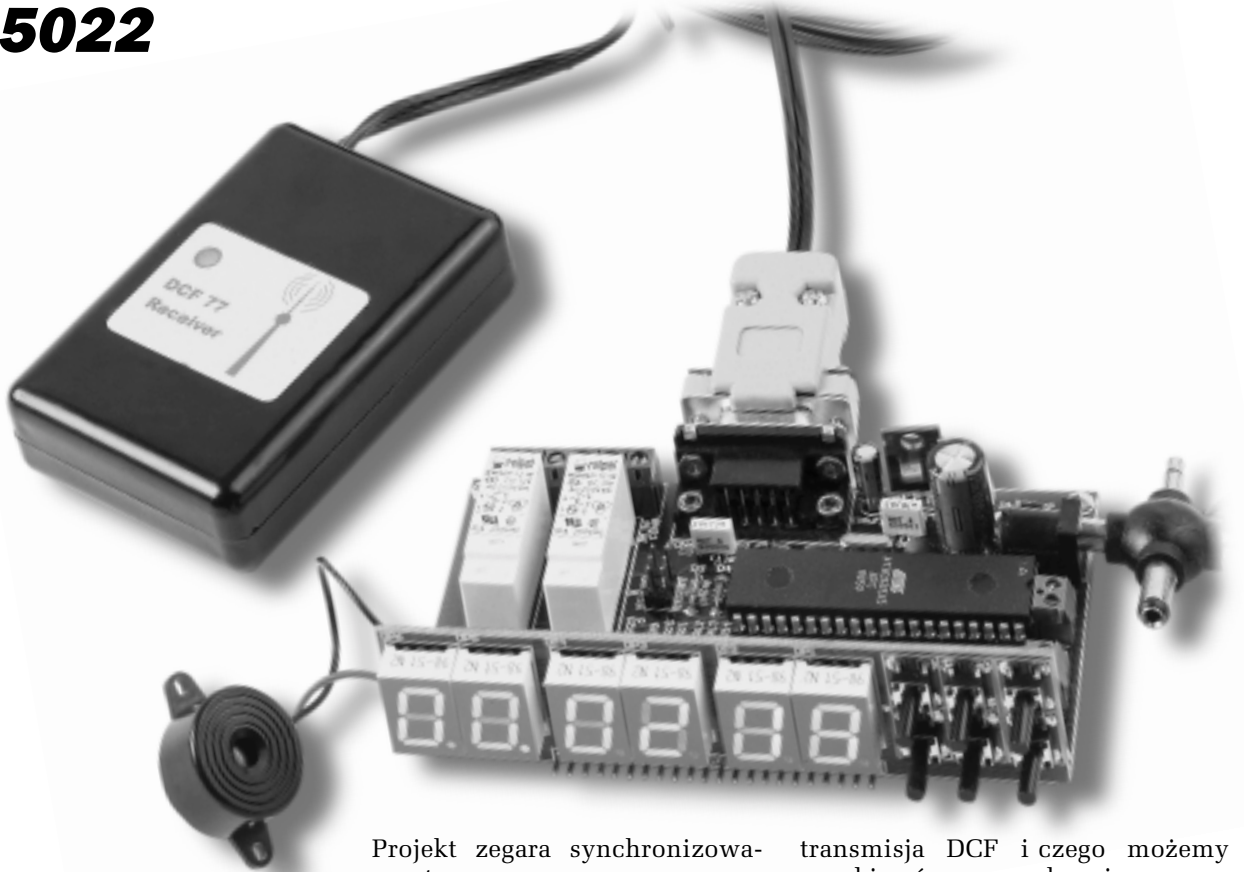


Programowany zegar z DCF77, część 1

AVT-5022



Proponowany układ jest „przeróbką“ układu opisanego w Elektronice Praktycznej kilka lat temu. Wydaje się jednak, że wskutek postępu w elektronice rok rozwoju tej dziedziny techniki to całe stulecie w innych dziedzinach.

Projekt zegara synchronizowanego atomowym wzorcem czasu DCF77 zaprezentowaliśmy w numerze EP7/94 i, jak na tamte czasy, był to układ bardzo nowoczesny. Trudno jednak nazwać go takim w roku 2001 i dlatego postanowiłem przedstawić Czytelnikom własne opracowanie takiego zegara, w którym wykorzystano nowoczesne, nie istniejące w połowie lat 90. elementy. Właściwie jedynym elementem zasługującym na uwagę jest naprawdę doskonały procesor, ośmielam się twierdzić, że najlepszy w swojej klasie: AT90S8535.

Zanim jednak przedstawimy szczegółowy opis układu, zapoznajmy się z podstawowymi parametrami nowego zegara DCF77.

Temat synchronizacji zegarów za pomocą sygnału DCF był poruszany na łamach Elektroniki Praktycznej dość dawno i dlatego chciałbym w największym skrócie przypomnieć, czym właściwie jest

transmisja DCF i czego możemy oczekiwać po synchronizowanym zegarze.

Zapewnienie synchronizacji czasu na terenie całego państwa czy nawet kontynentu było od dawna bardzo istotnym problemem. Nawet najdokładniejsze chronometry stosowane w nawigacji morskiej nie zapewniały dostatecznej precyzji, a pomiary astronomiczne były czynnością bardzo kłopotliwą. Do rozwiązania pozostawały więc dwa problemy: stworzenie wzorca czasu niezależnego od zjawisk astronomicznych i odpowiednio niezawodnego środka transmisji, zapewniającego błyskawiczne przekazywanie informacji na teren przynajmniej jednego państwa. Takim wzorcem czasu, praktycznie idealnie dokładnym, okazały się przemiany zachodzące w izotopach promieniotwórczych, a odpowiednim medium stały się fale radiowe. W wielu krajach (m. in. USA, Wielka Brytania)



Rys. 1. Przybliżony zasięg nadajnika DCF.

stworzono nadajniki czasu wzorcowego.

W Niemczech poradzano sobie z tym problemem już dość dawno, nadając sygnał z bardzo dokładnego wzorca czasu. Jest nim atomowy (cezowy) zegar czasu znajdujący się na Uniwersytecie w Braunschweig, którego błąd szacowany jest na mniejszy niż 1 sekunda na 5 milionów lat. W Mainflingen [50° 0,1' N, 09° 00' E] koło Frankfurtu nad Menem znajduje się nadajnik radiowy nadający na częstotliwości 77,5kHz (z mocą 50kW) dokładną informację czasową, będącą urzędowym wzorcem czasu w Niemczech.

Informacja o czasie jest kodowana w 59-bitowych słowach przesyłanych co sekundę i zawiera dane o czasie (godziny, minuty) i dacie (rok, miesiąc, dzień, dzień tygodnia). Dodatkowo zawarte są też zapowiedzi zmiany czasu i dodatkowej sekundy oraz informacja o tym, czy obowiązuje czas letni, czy zimowy. Aby zapobiec przekłamanom, dodano także bity parzystości pomagające wykryć błędy w transmisji.

Według oficjalnych danych, moc nadajnika zapewnia poprawny odbiór w zasięgu 2500km (rys. 1), a więc na terenie prawie całej kontynentalnej Europy (z wyłączeniem Islandii i części Finlandii). Praktyka jest jednak nieco inna, ponieważ ostatnio zapoznałem się z informacjami o poprawnym odbiorze transmisji DCF77 nawet na odległość do 5000 km.

Oczywiście, odbiór nie był stały, ale przy dobrych warunkach propagacyjnych możliwe było odebranie kilkudziesięciu poprawnych transmisji dziennie.

Do odbioru informacji DCF przeznaczone są specjalne odbiorniki DCF Receiver dostępne w sklepach AVT. Odbiornik taki jest niezbędny do wykorzystania wszystkich możliwości zegara.

Zrealizowanie układu zegara w technice mikroprocesorowej, zresztą jedynej możliwej do zastosowania w tak skomplikowanym systemie, radykalnie uprościło jego budowę. Nie polecałbym jej może zupełnie „zielonym” elektronikom, ale już średnio zaawansowani elektronicy, mający za sobą wykonanie kilku prostych układów, mogą zdecydować się na montaż zegara DCF77.

Opis działania

Schemat elektryczny zegara DCF77 pokazano na rys. 2. „Sercem” układu jest zaprogramowany procesor typu AT90S8535, otoczony niewielką liczbą elementów dyskretnych. Schemat zegara możemy podzielić na dwie części, odpowiadające fizycznemu rozmieszczeniu elementów na dwóch płytkach obwodów drukowanych. W górnej części schematu jest blok sześciu wyświetlaczy siedmiosegmento-

wych DP1..DP6 i klawiatura zbudowana z sześciu przycisków S1..S6. Ta część układu połączona jest z głównym blokiem zegara za pomocą złącza CON1 + CON2.

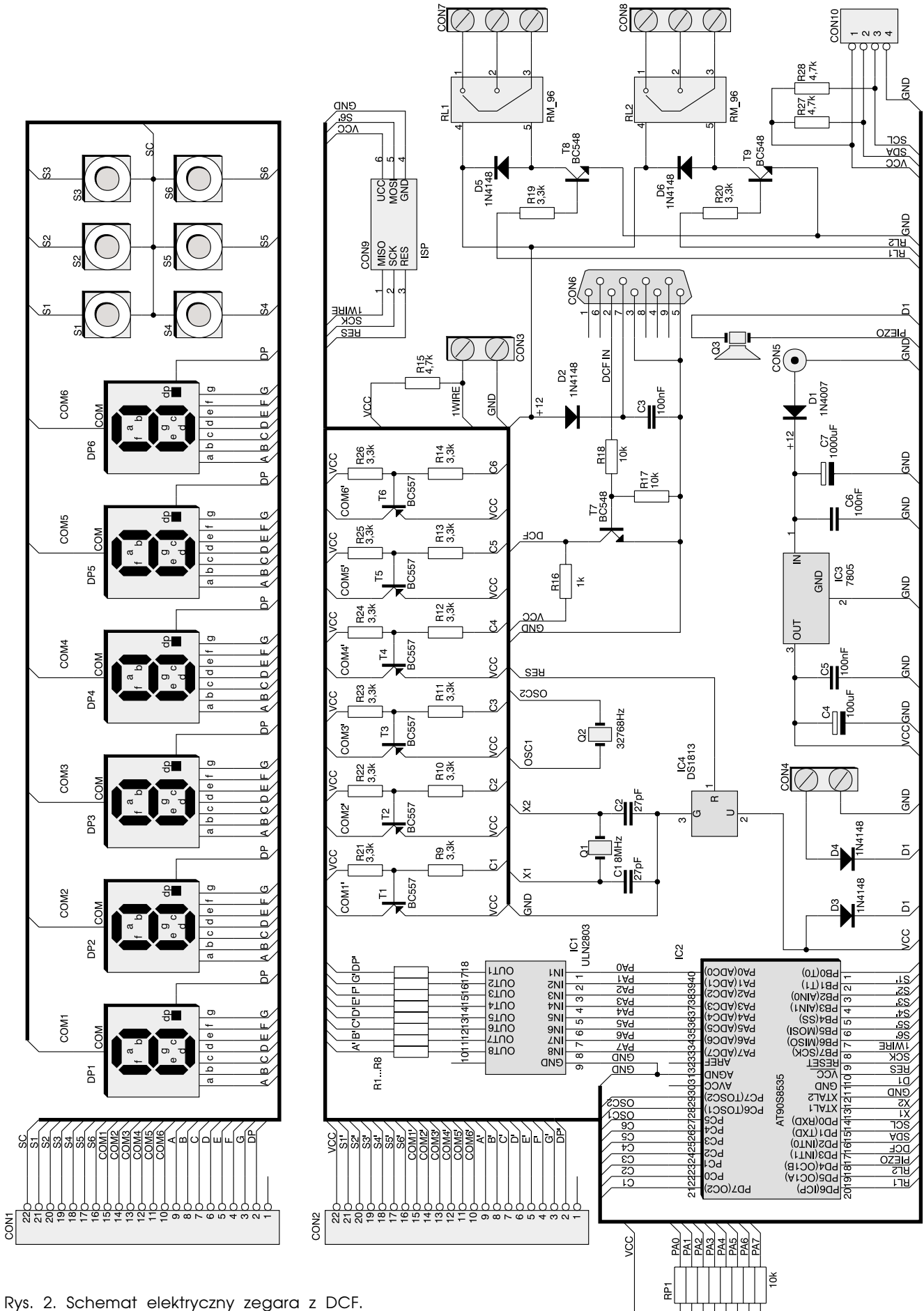
Zastosowanie wyświetlania multipleksowanego spowodowało dodanie do układu odpowiednich elementów sterujących segmentami i wspólnymi anodami wyświetlaczy. Segmenty oraz punkty dziesiętne wyświetlaczy sterowane są za pomocą driverów mocy zawartych w układzie IC1 - ULN2803, natomiast do wspólnych anod wyświetlaczy prąd jest doprowadzany za pomocą tranzystorów T1..T6.

Sygnał DCF doprowadzany jest z odbiornika do złącza DB9 - CON6, a następnie podlega inwersji i dostosowaniu do poziomu TTL w układzie z tranzystorem T7. Następnie kierowany jest na wejście przerwania INTO procesora.

W układzie zastosowano dwa rezonatory kwarcowe Q1 i Q2 współpracujące z procesorem. Rezonator Q1, o częstotliwości podstawowej 8MHz, jest „zwykłym” kwarcem współpracującym z oscylatorem systemowym procesora. Natomiast „zegarkowy” kwarc Q2 stabilizuje częstotliwość dodatkowego oscylatora RTC wbudowanego w strukturę procesora.

Podstawowe dane techniczne zegara DCF77:

- ✓ Wyświetlanie czasu w formacie: godzina, minuta, sekunda.
- ✓ Wyświetlanie daty w formacie: dzień, miesiąc, rok. Opcjonalnie data może być wyświetlana z pominięciem roku, na którego miejscu prezentowana może być informacja o aktualnym dniu tygodnia.
- ✓ Budzik 1 pracujący w trybie codziennym, czyli włączający sygnał budzenia niezależnie od dnia tygodnia.
- ✓ Budzik 2 automatycznie ignorujący sobotę i niedzielę jako dni wolne od pracy i nauki w szkole. Niezależnie od generacji sygnału akustycznego obydwa budziki sterują przekaźnikiem, do którego można podłączyć urządzenia o znacznym poborze mocy.
- ✓ Timer pracujący w zakresie od 1 sekundy do 59 sekund, 59 minut i 99 godzin. Timer steruje drugim przekaźnikiem o znacznej obciążalności styków.
- ✓ Stoper o zakresie liczenia identycznym jak timer.
- ✓ Informacje prezentowane są na typowych wyświetlaczach siedmiosegmentowych LED o standardowych wymiarach. Umożliwia to odczytywanie czasu z odległości kilku, a dla osób obdarzonych sokołim wzrokiem nawet kilkunastu metrów. Ostatnio zapoznałem się z listem od Czytelnika, który napotkał na ogromne problemy podczas prób dołączenia do zbudowanego układu wyświetlaczy o większych wymiarach, niż przewidziane w projekcie. Problem ten nie będzie nawet w najmniejszym stopniu dotyczył naszego zegara DCF77, ponieważ bez jakichkolwiek przeróbek możemy go wyposażyć w dodatkowe wyświetlacze o wysokości cyfr równej 57mm, co umożliwi odczyt z odległości nawet kilkudziesięciu metrów!
- ✓ Dane wprowadzane są do zegara za pomocą sześcioprzyciskowej klawiatury. W każdej chwili możemy dokonać ręcznej zmiany czasu, co nie było możliwe w poprzednim „wcieleniu” zegara DCF.
- ✓ Pomimo, że nasz zegar przeznaczony jest do stałej współpracy z odbiornikiem sygnału DCF77, może on także pracować jako zwykły zegar synchronizowany kwarcem „zegarkowym” 32768Hz. Jednak dokładność wskazań znacznie wtedy spada i jest taka, jaka może być dokładność seryjnie produkowanych i nie selekcyjowanych popularnych rezonatorów kwarcowych.
- ✓ Zegar może być wyposażony w awaryjne źródło zasilania, które przy zaniku napięcia w sieci energetycznej dostarcza prądu do zasilania samego tylko procesora.



Rys. 2. Schemat elektryczny zegara z DCF.

List. 1.

```

Sub Display_time
'Przed wejściem w pętlę programową, w której prowadzona będzie kontrola czasu bieżącego,
'a także jego ewentualna ręczna korekta, program określa rodzaj wyświetlania
'oraz zawiadamia o rozpoczęciu pracy za pomocą krótkiego sygnału akustycznego:

Display_type = 1
Short_beep
Do
'W programie zadeklarowane zostały trzy zmienne pomocnicze, których wartość bezpośrednio
'decyduje o tym, co aktualnie będzie ukazywało się na wyświetlaczach. Ponieważ w tym momencie
'mamy zamiar wyświetlać informacje o czasie, zmienne te przyjmują następujące wartości:

Disp1 = _sec
Disp2 = _min
Disp3 = _hour

'Z podprogramu wyświetlania i korekty daty możemy przejść do wyświetlania aktualnej daty,
'a dalej do innych funkcji. Ich zmiana dokonywana jest za pomocą naciśnięcia przycisku S1:

Reset Portb.0
If Pinb.0 = 1 Then Display_date

'Do funkcji ręcznej korekty aktualnego czasu możemy przejść po jednoczesnym naciśnięciu
'przycisków S3 i S6:

Reset Portb.2: Reset Portb.5

If Pinb.2 = 1 And Pinb.5 = 1 And Time_set_flag = 0 Then
'Wejście w tryb ręcznego ustawiania nowego czasu sygnalizowane jest trzema długimi
'(ok. 1 s) sygnałami akustycznymi. Następuje też zmiana wartości zmiennej pomocniczej
'TIME_SET_FLAG określającej tryb pracy:
For R = 1 To 3
Beep
Next R
Time_set_flag = 1
End If

'Naciśnięcie przycisku dołączonego do pinu 2 portu B powoduje zwiększanie wartości minut.
'Każda taka operacja sygnalizowana jest krótkim sygnałem akustycznym. Oczywiście, taka
'akcja jest możliwa tylko po wejściu programu w tryb ustawiania czasu.

Reset Portb.2
If Pinb.2 = 1 And Time_set_flag = 1 Then
Incr _min
If _min = 60 Then _min = 0
Short_beep
End If

'Naciśnięcie przycisku dołączonego do pinu 5 portu B powoduje zwiększanie wartości
'godzin. Każda taka operacja sygnalizowana jest krótkim sygnałem akustycznym.

Reset Portb.5
If Pinb.5 = 1 And Time_set_flag = 1 Then
Incr _hour
If _hour = 24 Then _hour = 0
Short_beep
End If

'Naciśnięcie przycisku dołączonego do pinu 1 portu B spowoduje wyjście układu z trybu
'ustawiania czasu, co zostanie zasygnalizowane długim sygnałem akustycznym.

Reset Portb.1
If Pinb.1 = 1 And Time_set_flag = 1 Then
_sec = 0
Time_set_flag = 0
Beep
End If

'A teraz jedna z najważniejszych funkcji programu: wysyłanie danych na magistralę I2C,
'co umożliwi dołączenie do naszego zegara dodatkowych wyświetlaczy o dużych wymiarach.
'Dane konwertowane są najpierw na kod BCD, a następnie wysyłane na I2C pod kolejne adresy
'układów PCF8574:
X = Makebcd(_hour)
I2Csmd 112, X
X = Makebcd(_min)
I2Csmd 114, X
X = Makebcd(_sec)
I2Csmd 116, X

Loop
End Sub
    
```

Podstawowym elementem wykonawczym zegara jest przetwornik piezoelektryczny z generatorem (Q3), sterowany bezpośrednio z wyjścia 4 portu D procesora. Dwa przekaźniki RL1 i RL2, których cewki zasilane są za pośrednictwem tranzystorów T8 i T9, można zastosować do sterowania urządzeniami o znacznym poborze mocy, zasilanych z sieci energetycznej. Ważnym elementem układu jest złącze magistrali I²C - CON10. Umożliwia ono dołączenie do zegara dodatkowych wyświetlaczy, np. modułów AVT-859 o wysokości 57mm.

Zegar powinien być zasilany napięciem stałym o wartości ok. 12VDC, doprowadzonym do złącza CON5. Stabilizator scalony zrealizowany na układzie IC3 dostarcza napięcia +5VDC, niezbędnego do zasilania części cyfrowej układu zegara.

Należy zwrócić uwagę na nietypowe zasilanie procesora, który jest dołączony do szyny zasilającej VCC o napięciu +5VDC za pośrednictwem diody D3. W związku z tym napięcie zasilania procesora jest zmniejszone o ok. 0,6V i wynosi tylko ok. 4,4VDC. Co spowodowało zastoso-

wanie tak nietypowego rozwiązania? To cała historia. Podczas uruchamiania kilku układów z procesorem AT90S8535, wykonywanych wbudowany w jego strukturę oscylator i generator przerwań RTC, napotkałem na nieoczekiwane i dziwaczne trudności. W niektórych układach oscylator nie działał w ogóle, a w innych pracował w niekontrolowany sposób, włączając się i wyłączając w nieoczekiwanych momentach. Ani sprawdzanie części hardware'owej układu, ani kodu programu nie dawało rezultatu, podobnie jak wertowanie karty katalogowej procesora. Na rozwiązanie problemu natknąłem się dopiero podczas lektury erraty do karty

Tab. 1. Sposób kodowania informacji DCF

Numer impulsu (numer sekundy)	Znaczenie impulsu
0	Początek transmisji. Zawsze = 0.
1-14	Przerwa, bez znaczenia - wszystkie zera.
15	0- antena normalna; 1- antena pomocnicza.
16	0-normalnie; 1- zapowiedz zmiany czasu (przez godzinę przed zmianą).
17-18	(w kolejności bity 18,17) 10-czas zimowy; 01-czas letni.
19	0-normalnie; 1-zapowiedz dodatkowej sekundy.
20	Start informacji czasowej. Zawsze = 1.
21-24	(w kolejności bity 24,23,22,21) jednostki minut w BCD.
25-27	(w kolejności bity 27,26,25) dziesiątki minut w BCD.
28	bit parzystości dla bitów 21-27.
29-32	(w kolejności bity 32,31,30,29) jednostki godzin w BCD.
33-34	(w kolejności bity 34,33) dziesiątki godzin w BCD.
35	bit parzystości dla bitów 29-34.
36-39	(w kolejności bity 39,38,37,36) jednostki dni miesiąca w BCD.
40-41	(w kolejności bity 41,40) dziesiątki dni miesiąca w BCD.
42-44	(w kolejności bity 44,43,42) dni tygodnia w BCD - 1=Pn; 7=Nd.
45-48	(w kolejności bity 48,47,46,45) jednostki miesiąca w BCD.
49	dziesiątki miesiąca w BCD.
50-53	(w kolejności bity 53,52,51,50) jednostki lat w BCD.
54-57	(w kolejności bity 57,56,55,54) dziesiątki lat w BCD.
58	bit parzystości dla bitów 36-57.
59	brak impulsu.

List. 2.

```

Multiplexing:
'Po zgłoszeniu przerwania timera0 wykonane zostaną następujące czynności:
Porta = 0      'wstępne ustawienie portu A w stan niski
Portc = 255    'wstępne ustawienie portu C w stan wysoki
Incr Digit_number 'zwiększenie numeru kolejnego wyświetlacza (kolejnej pozycji dziesiętnej)
If Digit_number = 7 Then Digit_number = 1 'zamknięcie cyklu zliczania wyświetlaczy do modułu 6

Select Case Digit_number 'w zależności od numeru wyświetlacza

Case 1:
'dziesiątki sekund
Temp = Displ / 10 'obliczenie wartości cyfry, która ma zostać ukazana na pierwszym
'wyświetlaczu
Porta = Lookup(temp, 7segment) 'przekodowanie otrzymanej wartości na kod
'wyświetlacza siedmiosegmentowego
Reset Portc.1 'włączenie zasilania anod segmentów pierwszego wyświetlacza

Case 2:
'sekundy
Temp = Displ / 10 'obliczenie wartości cyfry, która ma zostać ukazana na pierwszym
'wyświetlaczu
Temp = Temp * 10
Temp = Displ - Temp
Porta = Lookup(temp, 7segment) 'przekodowanie otrzymanej wartości na kod
'wyświetlacza siedmiosegmentowego
Reset Portc.0 'włączenie zasilania anod segmentów pierwszego wyświetlacza

Case 3:
'dziesiątki minut
'konstrukcja programu analogiczna do dziesiątek sekund

Case 4:
'minuty
'konstrukcja programu analogiczna do sekund

Case 5:
'dziesiątki godzin
Temp = Disp3 / 10
Porta = Lookup(temp, 7segment)
If Dcf_receiving_flag = 1 Then Porta = Porta + 1
'jedno z wielu dodatkowych uwarunkowań umieszczonych w podprogramie obsługi
'wyświetlania multipleksowanego, omówione dla przykładu. Dodanie 1 do wartości
'wysyłanej do portu A powoduje migotanie kropki dziesiętnej, synchronicznie
'z odbieranymi impulsami kodu DCF.
Reset Portc.5

Case 6:
'godziny
'konstrukcja programu analogiczna do sekund

End Select

Return
'Przekodowywanie wartości na kod wyświetlacza siedmiosegmentowego odbywa się na podstawie
'danych zawartych w poniższej tabeli:
7segment:
Data 252, 96, 218, 242, 102, 182, 190, 224, 254, 246
    
```

katalogowej, gdzie firma ATMEL umieściła rozbijającą uwagę:

„(..) When using an external 32 kHz crystal as asynchronous clock source for Timer2, the timer may count incorrectly at voltages above 4.0V. Keep the supply voltage below 4.0V when clocking Timer2 from an external crystal. (..)“, z której wynika, że przy kwarcu 32kHz napięcie zasilania powinno być mniejsze niż 4V.

Nie wnिकam w to, dlaczego generator kwarcowy 32768Hz przy zasilaniu procesora napięciem większym od 4V działa nieprawidłowo. Ważne jest tylko to, że obniżenie napięcia o 0,6V spowodowało natychmiastowe usunięcie problemów ze sprzętowym RTC. Jeżeli już procesor jest zasilany poprzez diodę separującą go od reszty układu, to nic prostszego jak zapewnić mu zasilanie awaryjne, które można dołączyć do złącza CON4.

Aby poznać działanie zegara, należy prześledzić sterujący nim program.

Zacznijmy od najprostszej funkcji, jaką jest zwykle wskazywanie aktualnego czasu, na razie bez omawiania sposobu synchronizowania go z sygnałem DCF77. Aby zbudować metodami programistycznymi zegar czasu rzeczywistego, zwykle musimy się trochę pomęczyć. Musimy obliczyć wartość, którą będzie przeładowywany timer odpowiedzialny za odmierzenie czasu, przygotować procedurę obsługi przerwania, w której zliczane będą sekundy i minuty określające upływający czas. Oprogramowanie zegara zostało napisane w języku MCS BASIC, w którym taki sposób tworzenia programowego RTC byłby zbyt nieudolny. A zatem wystarczy tylko napisać:

```

Config Clock = Soft,
Gosub = Sestic
    
```

aby poinstruować kompilator o konieczności utworzenia w programie zegara RTC, którego działanie oparte jest na przerwaniach otrzy-

mywanych z timera2. Od tego momentu mamy do dyspozycji, podobnie jak w starym QBASIC, dwie zmienne główne: TIME\$ i DATE\$ oraz zmienne pomocnicze: _sec, _min, _hour, _day, _month i _year. Wszystkie czynności związane z odliczaniem czasu, wyznaczeniem liczby dni miesiący czy też pilnowaniem kolejnych lat przestępnych są wykonywane odład automatycznie, a podprogram RTC zajmuje w pamięci procesora ok. 300B. Programowy RTC z BASCOM-a ma tylko jedną wadę, o której lojalnie informuje nas Autor programu: zawiera w sobie pluskwę, która da znać o sobie w roku 2101, gdyż zostanie on zidentyfikowany jako rok 2001. No cóż, chyba niewiele nas to obchodzi.

WYKAZ ELEMENTÓW

Rezystory

- RP1: R-PACK SIL10kΩ
- R1..R8: 47Ω
- R9..R14, R19..R26: 3,3kΩ
- R15, R27, R28: 4,7kΩ
- R16: 1kΩ
- R17, R18: 10kΩ

Kondensatory

- C1, C2: 27pF
- C3, C5, C6: 100nF
- C4: 100μF/10V
- C7: 1000μF/16V

Półprzewodniki

- D1: 1N4007
- D2..D6: 1N4148
- DP1..DP6: wyświetlacz siedmiosegmentowy LED wsp. anoda
- IC1: ULN2803
- IC2: AT90S8535
- IC3: 7805
- IC4: DS1813
- T1..T6: BC557
- T7..T9: BC548

Różne

- Q1: rezonator kwarcowy 8MHz
- Q2: rezonator kwarcowy 32768Hz
- Q3: przetwornik piezoelektryczny z generatorem
- CON1, CON2: goldpin kątowy 22pin
- CON3, CON4: ARK2 (3,5mm)
- CON5: gniazdko zasilania
- CON6: DB9/M lutowane w płytce
- CON7, CON8: ARK3
- CON10: 4 x goldpin
- RL1, RL2: RM96/12V
- S1..S6: microswitch

To, że wykorzystujemy timer2 do obsługi programowego RTC nie oznacza, że nie możemy wykorzystywać tak „miłej“ gratki, jak przerwania generowane dokładnie co jedną sekundę, także do innych celów. W poleceniu konfigurującym zegar czasu rzeczywistego możemy dodać (i dodaliśmy) dyrektywę, wskazującą do jakiego podprogramu ma nastąpić skok po upływie każdej kolejnej sekundy. W tym przypadku jest to podprogram *SECTIC*.

Na początek zajmijmy się najprostszym fragmentem programu, który odpowiedzialny jest za wyświetlanie i dokonywanie ręcznych korekt bieżącego czasu. Wspomnijmy tylko jeszcze, że pozostałe dwa timery zawarte w strukturze procesora: *TIMER0* i *TIMER1* zostały także uruchomione i skonfigurowane w następujący sposób:

```
Config Timer1 = Timer
Prescale = 64
Config Timer0 = Timer
Prescale = 64
On Timer1 Dcf_start
On Timer0 Multiplexing
```

Przerwania generowane przez timer0 wykorzystywane będą do sterowania multipleksowanym wyświetlaniem danych, a timer1 obsługiwać będzie procedury pomiaru czasu trwania impulsów kodu DCF. Przykładową procedurę pokazano na **list. 1**.

Podprogramy realizujące wyświetlanie i korektę daty, wyświetlania i ustawianie budzików i timerów są skonstruowane bardzo podobnie do programu wyświetlania czasu. Nie będziemy zatem ich analizować i przejdziemy do zapoznania się z podprogramem obsługi wyświetlaczy siedmiosegmentowych (**list. 2**). Skok do tego podprogramu następuje przy każdym wystąpieniu

przerwania timer0, wówczas zmiana wyświetlanych cyfr odbywa się z częstotliwością ok. 488Hz (pamiętamy o włączeniu preskalera timera o stopniu podziału 64). Podprogram obsługi wyświetlania multipleksowanego prezentuje w postaci bardzo uproszczonej, ale wystarczającej do zapoznania się z jego działaniem. Omawianie wszystkich dodatkowych uwarunkowań umieszczonych w tym podprogramie, a służących zróżnicowaniu wyświetlania w zależności od aktualnej funkcji zegara tylko gmatwałoby opis programu, nie wnosząc wiele w zrozumienie jego działania.

Zbigniew Raabe, AVT
zbigniew.raabe@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/czerwiec01.htm> oraz na płycie CD-EP06/2001B w katalogu PCB.