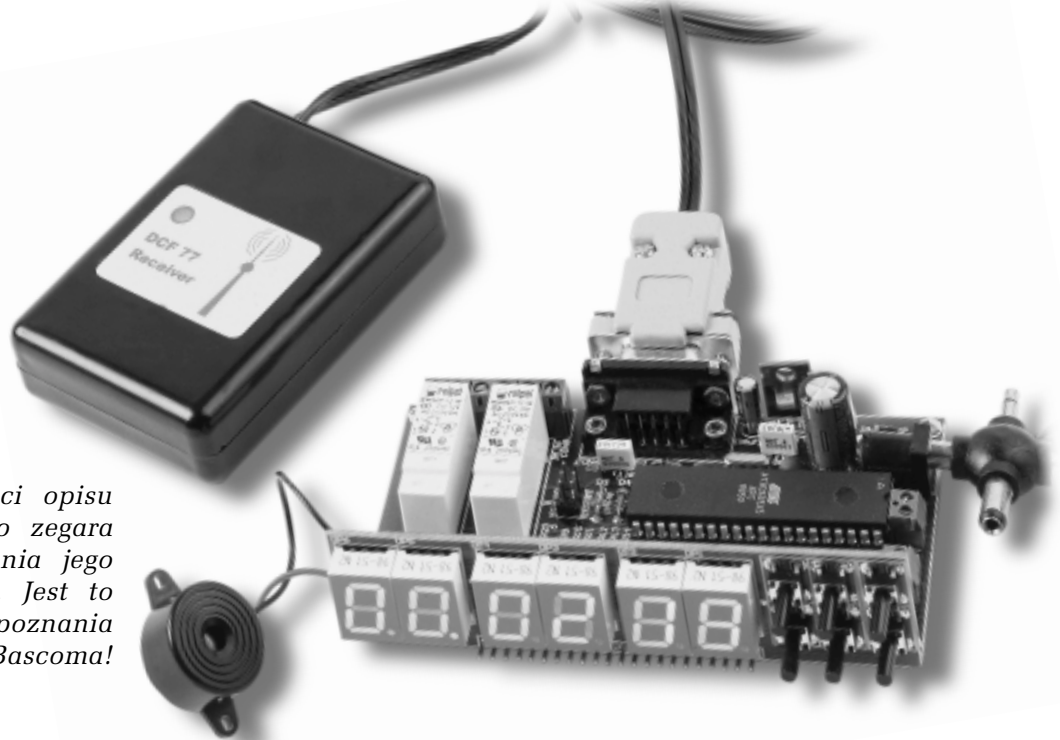


# Programowany zegar z DCF77, część 2

## AVT-5022

*W drugiej części opisu programowanego zegara z DCF77 autor odsłania jego programowe tajniki. Jest to doskonała okazja do poznania możliwości Bascoma!*



Chyba najciekawszym fragmentem programu sterującego zegarem jest podprogram analizujący odbierany sygnał DCF77 i korygujący aktualny czas oraz datę. Przedstawiono go na **list. 3**.

Uruchomiony został *TIMER1*, skonfigurowany do pracy z preskalerem o stosunku podziału 64. A zatem częstotliwość podawana na wejście rejestru tego timera wynosi dokładnie 125000Hz. Należy też zauważyć, że w przypadku wystąpienia przepełnienia timera1 nastąpi skok do podprogramu *DCF\_START* (dyrektywa *ON TIMER DCF\_START*).

Podczas konfiguracji programu udzielone zostało także zezwolenie na obsługę przerwania zewnętrznego *INT0*. To, czy przerwanie będzie inicjowane opadającym, czy wstępującym zboczem sygnału możemy określić za pomocą polecenia konfiguracyjnego *CONFIG INT0 = FALLING* lub *CONFIG INT0 = RISING*. Podczas analizy podprogramu dekodowania transmisji musimy pamiętać, że sygnał DCF został odwrócony w fazie (zanegowany) przez układ

z tranzystorem T7.

Omówiliśmy sposób dekodowania sygnału DCF, który okazał się niezbyt skomplikowany. Pozostała jednak otwarta jedna sprawa: skąd program wie, że jest to sygnał DCF i należy rozpocząć jego dekodowanie? Na szczęście jednoznaczne określenie startu transmisji sygnału DCF nie jest także sprawą trudną. Zauważmy, że transmisja kończy się na 58 impulsie, a impuls 59 w ogóle nie występuje. Czas trwania przerwy pomiędzy początkami wszystkich impulsów jest stały i wynosi dokładnie jedną sekundę. Jest tylko jeden wyjątek: przerwa pomiędzy ostatnim i pierwszym impulsem wynosi nie jedną, ale dwie sekundy! To właśnie zjawisko wykorzystamy do jednoznacznego określenia początku transmisji sygnału (kodu) DCF.

Podczas występowania impulsów DCF *TIMER1* był wykorzystywany do pomiaru ich czasu trwania. Podczas pomiaru czasu prawidłowego impulsu timer1 nie ulegał nigdy przepełnieniu, a maksymalna wartość jego rejes-

```

List. 3.
'Podprogram analizujący odbierany kod DCF77
Dcf:
  Pause_counter = 0 'wyzzerowanie zmiennej określającej czas upływający pomiędzy impulsami,
                    'patrz: listing podprogramu DCF_START
  Dcf_flag = Not Dcf_flag 'zmienna określająca poziom impulsu DCF zostaje zanegowana

  If Dcf_flag = 0 Then 'jeżeli na wejściu INT0 jest stan niski, co oznacza początek impulsu, to:
    Config Int0 = Rising 'przerwanie INT0 ma reagować na wstępujące zboczne sygnału
    Dcf_receiving_flag = 1 'ustaw zmienną sygnalizującą fakt odbierania impulsu
    Counter1 = 0 'wyzzeruj rejestr timer0
    Start Timer1 'w celu zmierzenia czasu trwania impulsu uruchom timer0
  Else 'jeżeli na wejściu INT0 jest stan wysoki, co oznacza koniec odbierania impulsu, to:
    Stop Timer1 'wstrzymaj prace timer0
    Config Int0 = Falling 'przerwanie INT0 ma reagować na opadające zboczne sygnału
    Dcf_receiving_flag = 0 'wyzzeruj zmienną sygnalizującą fakt odbierania impulsu

'W tym momencie program ma "trochę czasu" na dokonanie analizy czasu trwania odebranego
'uprzednio impulsu DCF i wyciągnięcie z niej odpowiednich wniosków. Dla impulsu o czasie trwania
'100 ms, czyli oznaczającego logiczne 0, stan timer0 powinien wynosić 12500. A zatem,
'uwzględniając konieczny margines błędu wynikający z cech transmisji AM:

  If Timer1 < 15000 And Timer1 > 10000 Then Dcf_bit = 0
'jeżeli stan rejestru timer0 zawiera się pomiędzy 10000 a 15000, to odebrany bit ma wartość 0

  If Timer1 > 20000 And Timer1 < 30000 Then Dcf_bit = 1
'jeżeli stan rejestru timer0 zawiera się pomiędzy 20000 a 30000, to odebrany bit ma wartość 1

'Czas trwania impulsu DCF drastycznie wykraczający poza zadane wartości może świadczyć tylko o
'błędzie w transmisji i musi powodować jej anulowanie. A zatem:
  If Timer1 < 10000 Then Start_dcf_flag = 0
'anuluj transmisję jeżeli zawartość rejestru timer0 jest mniejsza od 10000

  If Timer1 < 20000 And Timer1 > 15000 Then Start_dcf_flag = 0
'anuluj transmisję jeżeli zawartość rejestr timer0 jest mniejsza od 20000 i większa od 15000

  If Start_dcf_flag = 1 Then
'Jeżeli czas trwania odebranego impulsu mieścił się w zadanym przedziale i została określona jego
'wartość logiczna, to w zależności od numeru impulsu program musi wykonać następujące czynności.
  Select Case Dcf_counter
    Case 1: _sec = 0 'jeżeli odebrany został impuls o numerze 0, to wyzeruj
                    'zmienną sekund
    Case 21: Dcf_temp.0 = Dcf_bit'bit 0 zmiennej pomocniczej DCF_TEMP
            'przyjmuje wartość odebranego impulsu
    Case 22: Dcf_temp.1 = Dcf_bit'bit 1..
    Case 23: Dcf_temp.2 = Dcf_bit'bit 2..
    Case 24: Dcf_temp.3 = Dcf_bit'bit 3..
            Dcf_min = Dcf_temp 'zakończyła się transmisja informacji
            'o jednostkach aktualnej minuty i w związku z tym
            'zmienna pomocnicza DCF_MIN przyjmuje obliczoną wartość
            'zmiennej DCF_TEMP,
            'która następnie zostaje wyzerowana
    Case 25: Dcf_temp.0 = Dcf_bit'bit 0 zmiennej pomocniczej DCF_TEMP
            'przyjmuje wartość odebranego impulsu
    Case 26: Dcf_temp.1 = Dcf_bit'bit 1..
    Case 27: Dcf_temp.2 = Dcf_bit'bit 2..
            Dcf_temp = Dcf_temp * 10 'zakończyła się transmisja informacji
            'o dziesiątkach aktualnej minuty i w związku z tym
            'obliczona wartość zostaje pomnożona przez 10,
            'a następnie:
            Dcf_min = Dcf_temp + Dcf_min 'obliczona zostaje ostateczna wartość minut
            Dcf_temp = 0 'zmienna pomocnicza DCF_TEMP zostaje wyzerowana

'Omawianie dekodowania godziny, dnia miesiąca, miesiąca, roku i dnia tygodnia nie ma chyba
'większego sensu. Odbywa się ono na takiej samej zasadzie, co dekodowanie aktualnej minuty i jego
'analiza nie wnosiłaby niczego nowego w zrozumienie zasady działania programu. Pozwólmy zatem
'programowi możliwie dekodować następną informację i przenieśmy się na sam koniec transmisji,
'sygnalizowany odebraniem 58 impulsu

    Case 58:
      Start_dcf_flag = 0 'koniec transmisji, zerujemy jej flagę
      _min = Dcf_min 'przyporządkowanie zmiennej minut odebranej wartości
      _hour = Dcf_hour 'przyporządkowanie zmiennej godzin odebranej wartości
      _month = Dcf_month 'przyporządkowanie zmiennej miesiąca odebranej wartości
      _day = Dcf_day 'przyporządkowanie zmiennej dnia miesiąca odebranej wartości
      _year = Dcf_year 'przyporządkowanie zmiennej roku odebranej wartości
      _week_day = Dcf_week_day 'przyporządkowanie zmiennej dnia tygodnia
                        'odebranej wartości
  End Select 'koniec wyboru kolejnego impulsu

  Incr Dcf_counter 'zwiększ wartość licznika impulsów o 1
  Counter1 = 0 'wyzzeruj rejestr timer1
  Start Timer1 'uruchom timer1
End If
Return

```

tru wynosiła ok. 30000. Po dokonaniu pomiaru Timer1 zostaje ponownie uruchomiony w innym celu: zmierzenia czasu trwania przerwy pomiędzy impulsami. Ta przerwa podczas trwania transmisji wynosi 800 lub 900ms, co wiąże się z jednokrotnym przepełnieniem Timer1, natomiast po jej zakończeniu prawie 2 sekundy, co spowoduje dwukrotne wystąpienie przerwania i dwukrotny skok do podprogramu DCF\_START.

```

Dcf_start:
Incr Pause_counter
'zwiększ o 1 wartość zmiennej
'określającej czas przerwy
'pomiędzy impulsami
If Pause_counter = 2 Then
  Pause_counter = 0
  Dcf_counter = 0
  'licznik impulsów kodu DCF
  'zostaje wyzerowany
  Start_dcf_flag = 1
  'wskaźnik rozpoczęcia i trwania
  'transmisji DCF zostaje
  'ustawiony na 1

```

```

'TU WŁAŚNIE NASTĘPUJE
'POCZĄTEK TRANSMISJI DCF77

```

```

End If
Return

```

Mam nadzieję, że analiza przedstawionych listingów fragmentów programu sterującego naszym zegarem pozwoli Czytelnikom na pełne zrozumienie sposobu dekodowania sygnału DCF i zasady pracy programowego zegara czasu rzeczywistego (RTC). Nie wspomnieliśmy jeszcze o dodatkowych funkcjach zegara: o sposobie realizacji timera i budzików, a także o stoperze.

Zarówno kontrola zgodności bieżącego czasu z ustawionym czasem budzików, jak i odliczanie czasu przez timer i stoper odbywa się w podprogramie SEC\_TIC, do którego wykonywany jest skok po upływie każdej sekundy (tak, jak to zostało ustalone w konfiguracji programowego RTC). Dodatkowo, w podprogramie SEC\_TIC jest instrukcja warunkowa pozwalająca na detekcję upływu kolejnych minut:

```

Sectic
If Temp2 <> _min Then
' Czynności do wykonania po
' upływie kolejnej minuty, np.
' porównanie czasu budzika
End If
' Pozostałe czynności do
' wykonania
Temp2 = _sec
Return

```

Podprogram „obsługi“ timera i jednego z budzików zamieszczono na list. 4.

Program sterujący pracą zegara został z konieczności opisany bardzo fragmentarycznie, ponieważ pełny jego listing zajmuje około 5 stron formatu A4. Chciałem jedynie przedstawić Czytelnikom jego najważniejsze fragmenty, a w szczególności sposób dekodowania sygnału DCF77.

### Montaż i uruchomienie

Na rys. 3 przedstawiono rozmieszczenie elementów na dwóch płytkach obwodów drukowanych, wykonanych na laminacie dwustronnym z metalizacją. Montaż zegara rozpoczynamy od wlutowania w płytkę rezystorów i podstawek pod układy scalone, a kończymy

List. 4.

```

Sectic:
If Temp2 <> _min Then
  Readeeprom Alarm_hours, 3 'odczytaj z pamięci EEPROM dane o godzinie budzenia
  Readeeprom Alarm_minutes, 4 'odczytaj z pamięci EEPROM dane o minucie budzenia
  If Alarm_hours = _hour And Alarm_minutes = _min And Alarm_on_flag = 1 Then
    'jeżeli odczytane dane są zgodne z aktualnym czasem oraz
    'jeżeli budzik był aktywny, to:
    Alarm_counter = 30 'licznik trwania alarmu zostaje ustawiony na 30
    Alarm_counter_flag = 1 'flaga włączenie alarmu zostaje ustawiona na 1
    Set Portd.6 'zostaje włączony przełącznik
  End If
  ' dalsze czynności do wykonania po upływie minuty
  .....
End If

If Timer_on_flag = 1 Then
  'jeżeli timer jest włączony, to
  Decr Timer_seconds 'zmniejsz wartość sekund timera
  If Timer_seconds = 255 Then 'jeżeli wartość sekund wynosi 255, to:
    Decr Timer_minutes 'zmniejsz wartość minut
    Timer_seconds = 59
    If Timer_minutes = 255 Then 'jeżeli wartość minut wynosi 255, to:
      Decr Timer_hours
      Timer_minutes = 59 'wartość minut staje się równa 59
      If Timer_hours = 255 Then 'jeżeli wartość godzin wynosi 255, to:
        Timer_on_flag = 0 'koniec pracy timera
        Readeeprom Timer_minutes,6 'ponownie odczytaj z pamięci EEPROM zapisane
        'tam wartość minut timera
        Readeeprom Timer_seconds, 7 'ponownie odczytaj z pamięci EEPROM zapisane
        'tam wartość sekund timera
        Readeeprom Timer_hours, 5 'ponownie odczytaj z pamięci EEPROM zapisane
        'tam wartość godzin timera
        Reset Portd.6 'wyłącz przełącznik timera
        Beep 'wygeneruj sygnał akustyczny
      End If
    End If
  End If
End If

'w podprogramie SECTIC odbywa się także odliczanie czasu trwania alarmu budzika:
'jeżeli alarm był uaktywniony, to zmniejsz wartość jego licznika
If Alarm_counter_flag = 1 Then Decr Alarm_counter

'jeżeli licznik czasu trwania alarmu osiągnął 0, to zresetuj sygnalizację włączenia alarmu
If Alarm_counter = 0 Then Alarm_counter_flag = 0

'jeżeli alarm jest aktywny, to naprzemiennie włączaj sygnalizację akustyczną
If Alarm_counter_flag = 1 And Flash_flag = 1 Then Portd.4 = Not Portd.4

Return
    
```

na zamontowaniu kondensatorów elektrolitycznych i pozostałych elementów o dużych gabarytach.

Zmontowane płytki musimy połączyć ze sobą za pomocą kątowych goldpinów. Taki sposób montażu zapewni nie tylko solidne połączenie mechaniczne, ale także ustawienie płytek idealnie pod kątem prostym względem siebie.

Odbiornik DCF77 dołączamy do złącza DB9 umieszczonego na tylnej krawędzi płytki bazowej zegara. Odbiornik nie wymaga oddzielnego źródła zasilania, a o jego działaniu świadczy migotanie diody umieszczonej w jego obudowie oraz punktu dziesiątego na pierwszym wyświetlaczu zegara.

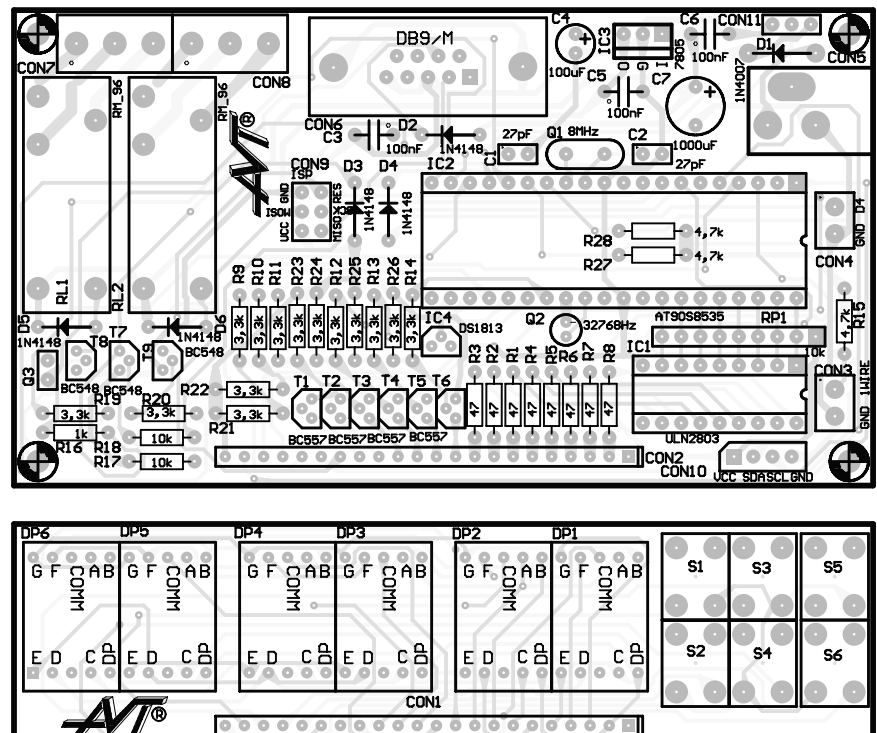
Sygnal DCF77 nadawany jest na bardzo niskiej częstotliwości - 77kHz. Dlatego fala radiowa rozchodzi się przy ziemi i jest podatna na zakłócenia bardzo zależne od warunków pogodowych i pory dnia. Zdarza się, że w niektórych rejonach o silnych zakłóceniach odbiór tego sygnału jest czasami niemożliwy. Dotyczy to zwłaszcza dużych aglomeracji miejskich. Zwiększenie zakłóceń następuje o wschodzie i zachodzie słońca oraz w obecności urządzeń

elektronicznych (monitory, komputery, telewizory, silniki itd.). Dlatego bardzo ważne jest, aby znaleźć dla odbiornika jak najlepsze miejsce. Istotne jest także jego zorientowanie względem nadajni-

ka oraz odległość odbiornika od urządzeń elektronicznych (zalecane jest minimum 2m). Dotyczy to w szczególności komputerów i komparatorowych silników elektrycznych, zarówno AC, jak i DC. Natomiast, wbrew wcześniejszym obawom i złym doświadczeniom sprzed paru lat, nie stwierdziłem poważniejszych zakłóceń pracy odbiornika wywoływanych przez procesor sterujący zegarem. Odbiornik pracował poprawnie nawet po umieszczeniu go w odległości kilku centymetrów od procesora.

Odbiornik powinien leżeć na płaskiej powierzchni (dioda LED do góry), nie może leżeć pod kątem, ani w pobliżu metalowych przedmiotów. Należy uzyskać jak najlepszy sygnał poprzez obracanie odbiornika wokół jego osi. Można to poznać po regularnym zapalaniu się diody LED. Powinna ona zapalać się co 1s na czas ok. 0,1 do 0,2 sekundy i gasnąć.

Układ zegara zmontowany ze sprawdzonych elementów nie wymaga uruchamiania i działa natychmiast po włożeniu w podstawkę zaprogramowanego procesora. Jednak to, że działa, nie oznacza wcale, że już umiemy go obsługiwać. Zajmijmy się więc nieco rozbudowanymi procedurami obsługi.



Rys. 3. Rozmieszczenie elementów na płytkach drukowanych.

**Tab. 1. Zestawienie funkcji pełnionych przez klawiaturę zegara.**

	S1	S2	S3	S4	S5	S6
Czas	Przejdźcie do kolejnej funkcji		+ S6 – przejdźcie do ustawiania czasu			+ S3 - przejdźcie do ustawiania czasu
Ustawianie czasu		Koniec ustawiania	Zmiana minut			Zmiana godzin
Data	Przejdźcie do kolejnej funkcji					
Ustawianie daty		Koniec ustawiania	Zmiana dnia miesiąca			Zmiana miesiąca
Budzik 1	Przejdźcie do kolejnej funkcji					
Ustawianie budzika 1		Koniec ustawiania	Zmiana minut			Zmiana godzin
Budzik 2	Przejdźcie do kolejnej funkcji	Budzik włączony/ /wyłączony				
Ustawianie budzika 2		Koniec ustawiania	Zmiana minut			Zmiana godzin
Timer	Przejdźcie do kolejnej funkcji	Start Timer		Stop Timer	Reset timer	
Ustawianie timera		Koniec ustawiania	Zmiana minut	Zmiana sekund		Zmiana godzin
Stoper	Przejdźcie do kolejnej funkcji	Start stoper		Stop stoper	Reset stoper	

Po pierwszym włączeniu zasilania układ przechodzi automatycznie w tryb wyświetlania aktualnego czasu, z tym że na wyświetlaczach ukazuje się początkowo godzina 00:00, a zegar rozpoczyna zliczanie czasu od tej wartości. Mamy teraz dwie możliwości do wyboru: albo poczekać na odebranie transmisji DCF77 i automatyczne skorygowanie wskazywanego czasu i dat, albo wykonać to ręcznie. Ponieważ jednak oczekiwanie na zdekodowanie transmisji może trwać do 2 minut (nawet przy dobrych warunkach propagacyjnych), dokonajmy ręcznej korekty czasu.

Podczas wyświetlania czasu, podobnie jak przy korzystaniu z innych funkcji zegara, możemy przejść w tryb ustawiania naciskając jednocześnie klawisze S3 i S6.

Zmiana trybu pracy sygnalizowana jest trzykrotnym sygnałem akustycznym, którego zadaniem jest ostrzeżenie operatora, że dalsze naciśnięcie klawiszy może wprowadzić istotne zmiany w pracy zegara.

Do ustawiania czasu, daty oraz innych wartości wykorzystujemy także klawisze S3 i S6. Naciskanie klawisza S3 powoduje cykliczną zmianę minut, a klawisza S6 - godzin. Ustawianie czasu kończymy naciskając klawisz S2. Zestawienie funkcji inicjowanych za pomocą poszczególnych klawiszy klawiatury zegara zamieszczono w **tab. 1**.

Na **rys. 4** przedstawiono wykorzystanie kropek dziesiętnych na wyświetlaczach. Sygnalizują one różne tryby pracy zegara z wyjątkiem budzików. Wejście w tryb ustawiania lub kontroli budzików sygnalizowane jest bowiem wyświetleniem „A1” lub „A2” na dwóch pierwszych wyświetlaczach.

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



Rys. 4. Znaczenie kropek dziesiętnych przy cyfrach wyświetlacza.

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/lipiec01.htm> oraz na płycie CD-EP07/2001B w katalogu PCB.