



# Mój pierwszy zegar mikroprocesorowy



Do czego to służy prezentowany układ?

Powyższe pytanie jest chyba nieco retyryczne, przecież każdy wie, do czego służy zegarek: do wskazywania aktualnego czasu. Jednak możliwości urządzenia, które za chwilę opiszemy wykraczają ponad możliwości przeciętnego zegarka, a w każdym razie ponad możliwości zegarków prezentowanych jak dotąd w EdW. Nasz nowy zegarek, zbudowany z wykorzystaniem zaledwie jednego układu scalonego - procesora typu AT89C4051 posiada następujące funkcje:

- Wskazywanie bieżącego czasu w formacie: godziny/minuty/sekundy
- Wskazywanie aktualnej daty w formacie: dzień miesiąca: miesiąc
- Budzik działający w cyklu dobowym.
- Ustawianie aktualnego czasu
- Ustawianie aktualnej daty
- Ustawianie czasu budzenia
- Włączanie i wyłączanie budzika
- Wprowadzanie korekty czasu wyrównującej niedokładności spowodowane zastosowaniem taniego, popularnego rezonatora kwarcowego
- Pomiar temperatury w zakresie od - 20 do +100 stopni Celsjusza dokonywany za pomocą cyfrowego termometru DS1820
- Możliwość sterowania wyświetlaczy siedmiosegmentowych LED o wysokości 57mm.

Nie są to może parametry zbyt imponujące, ale jak na tak małe i proste urządzenie całkowicie wystarczające. Zegary elektroniczne należą do grupy urządzeń najchętniej budowanych przez hobbyistów elektroników. Ja także bardzo lubię projektować i wykonywać układy takich zegarów i dlatego opisane urządzenie jest pierwszym, ale nie ostatnim zegarem zbudowanym w technice mikroprocesorowej, który zostanie opisany na łamach EdW.

Wracamy jednak do opisywanej konstrukcji, która jest układem przeznaczonym głównie dla początkujących konstruktorów,

możliwym do wykonania podczas wolnego popołudnia. Chciałby jeszcze tylko awansem odpowiedzieć na pewne głosy krytyki, z którymi być może spotka się opracowany przeze mnie zegarek.

Wyświetlacze alfanumeryczne LCD mają same zalety i tylko dwie wady: mały wymiar wyświetlanych znaków i ich słabą czytelność przy zmiennych warunkach oświetlenia. Ta wada może być powodem słusznej krytyki układu, którego wskazania będą widoczne tylko z niewielkiej odległości i przy dobrym oświetleniu. Na tę krytykę mogę odpowiedzieć dwoma stwierdzeniami:

- nasz zegarek jest urządzeniem przeznaczonym do pracy na biurku lub nocnym stoliku gdzie małe wymiary cyferek nie powinny być szczególnie uciążliwe. Można także zastosować wyświetlacz z podświetlaniem.

Bez jakichkolwiek przeróbek można do naszego zegara dołączyć... wyświetlacze LED o wysokości aż 57mm! Takie rozwiązanie pozwoli na zastosowanie układu w roli zegara tablicowego, widocznego nawet z odległości kilkudziesięciu metrów!

## Opis układu

Schemat elektryczny proponowanego układu zegarka został pokazany na **rysunku 1**. Komentowanie tak prostego hardware'u nie miałoby chyba większego sensu. Wystarczy stwierdzić, że do wyświetlania danych został zastosowany standardowy wyświetlacz alfanumeryczny LCD 16\*1, element dobrze znany Czytelnikom EdW, a w szczególności Uczestnikom i Sympatykom kursu BASCOM College. Do sterowania pracą zegara wykorzystane będą cztery przyciski S1 ... S4, a miniaturowy przekaźniczek RL1 może posłużyć do włączania i wyłączania urządzeń peryferyjnych służących jako sygnalizatory budzika. Ogromne znaczenia ma zainstalowanie w systemie dwóch magistrali transmisji danych: 1WIRE i I<sup>2</sup>C. O ich użyteczności przekonamy

się za chwilę, podczas analizy programu sterującego zegarem. Biermy się zatem do pisania programu dla naszego zegarka.

Co właściwie mamy wykonać? Czy coś zupełnie nowego, co pojawiło się na świecie z nastaniem ery elektroniki? Bynajmniej, nasz zegar będzie się składał, podobnie jak zegary mechaniczne, z wielu kółek zębatych, które zazębiając się o siebie i obracając z ustaloną prędkością, wskazują nam aktualny czas. Patrząc na program sterujący pracą naszego czasomierza, nie zobaczymy niczego innego, niż to, co widział Ludwik XVI, zaglądając w głąb swoich ukochanych zegarów. Zrealizujemy metodami elektronicznymi to samo, co mechanicy tworzyli już od setek lat metodami klasycznymi.

Co jest podstawą konstrukcji każdego zegara mechanicznego? Zwykle takie malutkie kółeczko, nazywało się chyba balans albo jałko podobnie. To ono decydowało o precyzji zegara, taktując ze stałą częstotliwością jego mechanizm. Właściwie to kółko było jakby okrągłym wahadłem, ale istotne jest tylko to, że posiadało małe ząbki, które napędzały większe kółeczko - sekundnik. Taki właśnie balans, wahadło decydujące o precyzji naszego zegara musimy teraz sobie zaprojektować, wykorzystując do tego celu nowo poznany składnik każdego systemu mikroprocesorowego - timer.

Ponieważ balans naszego zegarka ma pracować nieustannie, niezależnie od innych czynności wykonywanych przez procesor, najlepiej byłoby wykorzystać do jego budowy timer pracujący w trybie 2 - AUTORELOAD. Niestety, napotykamy tu na pewne ograniczenie wynikające z zasady pracy timera w tym trybie: maksymalną wartość liczby, którą możemy załadować do timera, równą 255. Taki balans kręciłby się z zawrotną prędkością, przerywania następowałyby jedno po drugim, co mogłoby niejednokrotnie spowodować inne funkcje wykonywane

przez procesor. Zrezygnujemy zatem z proponowanego trybu i zastosujemy tryb 1 pracy timera, z programowym ładowaniem i uruchamianiem timera.

A zatem Timer0 procesora musi być skonfigurowany w następujący sposób:

**Config Timer0 = Timer , Gate = Internal , Mode = 1  
On Timer0 Timer0\_int**

Wewnętrzny oscylator procesora pracuje z częstotliwością 11059200Hz, co wynika z zastosowania taniego i popularnego rezonatora kwarcowego o tej właśnie częstotliwości rezonansowej. Pamiętajmy jednak, że częstotliwość ta jest wewnętrznie dzielona przez 12 i dopiero potem używana jako zegar maszynowy procesora. A zatem realna częstotliwość taktowania procesora wynosi  $11059200/12 = 921600\text{Hz}$ .

Liczba ta znacznie przekracza pojemność zastosowanego timera, który wobec tego zostanie zmuszony do kilkukrotnego zliczania w ciągu sekundy. Następnym krokiem będzie zatem znalezienie jak największej liczby, która musi spełnić następujące warunki:  
- być mniejszą lub równą 65536  
- wynik dzielenia 921600 przez tę liczbę musi być liczbą całkowitą.

Liczbą tą jest 15:  $11059200/15 = 61440$ , czyli że wykorzystywać będziemy prawie całą pojemność timera. W trybie 1 maksymalna pojemność timera określona jest liczbą dwubajtową i wynosi 65536. Wynika z tego, że aby osiągnąć przepelnienie timera i wygenerowanie przerwania we właściwym czasie, timer musi za każdym razem rozpocząć zliczanie od wartości  $65536 - 61440 = 4096$ . Teore-

tycznie! Nie zapominajmy bowiem, że na utrzymanie, przeładowanie i ponowne uruchomienie timera procesor potrzebuje także trochę czasu. Z obliczeń i z doświadczeń przeprowadzonych w symulacji programowej wynika, że na te operacje procesor "zużyje" aż 54 takty zegarowe. A zatem liczba ładowana do rejestrów timera musi wynosić 4150.

Każde wystąpienie przerwania pochodzącego z Timer1 powoduje skok programu do procedury zawartej w podprogramie obsługi przerwania TIMER0\_INT, pokazanej na **listingu 1**. Jest to najważniejsza część programu obsługującego nasz zegar, w której realizowane są wszystkie funkcje związane z odliczaniem sekund, minut, godzin, dni i miesięcy. Z premedytacją zrezygnowałem ze zliczania lat: można przecież nie wiedzieć, która jest godzina, można zapomnieć jaki mamy dzień miesiąca czy nawet miesiąc, ale rok? Byłby to z pewnością szczyt roztargnienia, wart uwiecznienia w księdze Guinnessa! Rezygnacja ze zliczania lat pociągnęła za sobą jedną, drobną niedogodność: program nie jest w stanie rozpoznać lat przestępnych. Jednak nie sądzę, aby skorygowanie wartości dnia miesiąca raz na cztery lata było dla kogokolwiek zbyt facygujące!

Mam nadzieję, że podprogramy pokazane na **listingu 1** i **2**, realizujące najważniejsze funkcje naszego zegara są całkowicie zrozumiałe dla Czytelników i że wyjaśnienia może wymagać jedynie funkcja korekty czasu, dokonywanej automatycznie raz na dobę.

Do stabilizacji pracy wewnętrznego oscylatora procesora 89C4051 wykorzystany został popularny i tani rezonator kwarcowy 11.0659200MHz. Popularny i tani może niekiedy oznaczać: "nie najwyższej jakości" i tak właśnie jest niejednokrotnie w praktyce.

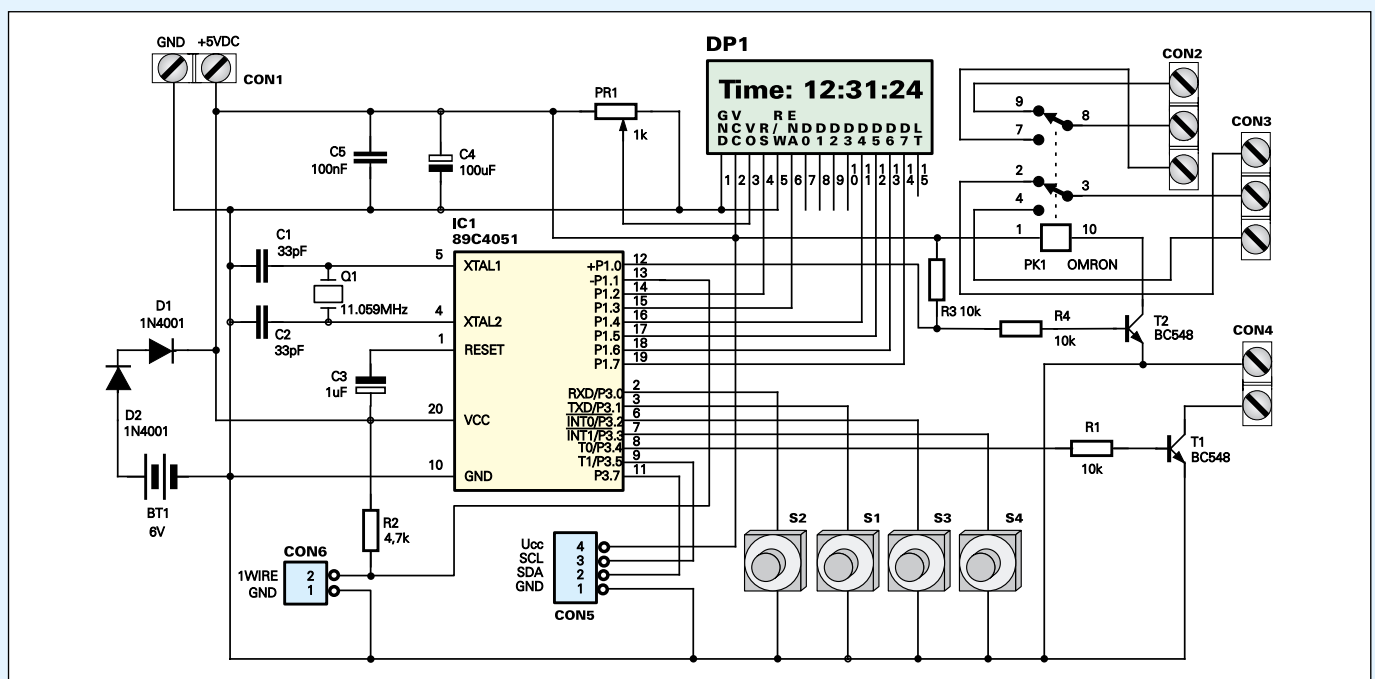
Rozrzut parametrów tych oscylatorów jest dość znaczny, co powoduje odchyłki wskazywanego czasu od czasu wzorcowego dochodzące niekiedy nawet do kilku - kilkunastu sekund na dobę. W roku 2001 trudno pogodzić się z taką "precyzją" i należy znaleźć jakieś rozwiązanie dające naszemu zegarowi cechy morskiego chronometru.

## 'Listing 1

```

Timer0_int:
Counter0 = 4150      'załaduj do rejestru timera1
                    'wartość 4150
Start Timer0      'uruchom timer1
Incr Int_temp     'zwiększ o 1 zmienną pomocniczą
                  'zliczającą kolejne przerwania
If Int_temp = 15 Then
    'jeżeli zmienna pomocnicza
    INT_TEMP przyjęła wartość 15 (zliczono 15 wystąpień
    'przerwania timera1), to:
    Seconds_flag = 1 'zmienna pomocnicza
    SECONDS_FLAG przyjmuje wartość 1 (patrz listing3)
    Incr Seconds     'zwiększ o 1 wartość sekund
    If Hours = 12 And Minutes = 0 And Seconds = 30
        And Correction_flag = 1 Then
        'jeżeli jest godzina 12:00:30,
        a także zmienna CORRECTION_FLAG jest równa 1, to:
        Seconds = Seconds + Seconds_correction
        'dokonaj korekty wskazywanego
        'czasu
        Correction_flag = 0
        'po dokonaniu korekty ustaw
        'ponownie zmienną CORRECTION_FLAG na 0,
        'co pozwoli na dokonanie kolejnej korekty za
        '24 godziny
End If
Int_temp = 0      'zmienna pomocnicza zliczania
                  'przerwań zostaje wyzerowana
If Seconds = 60 Then
    'jeżeli wartość sekund zwiększyła
    'się do 60, to:
    Seconds = 0   'wyzeruj wartość sekund, oraz
    Incr Minutes 'zwiększ wartość minut
    If Alarm_hour = Hours And Alarm_minute = Minutes
        And Alarm_flag = 1 Then
        'jeżeli aktualny czas jest zgodny
        'z czasem ustawionym dla budzika i jeżeli udzielone
        'zostało zezwolenie na włączenie sygnału budzenia
        '(zmienna ALARM_FLAG), to:
        Set Rel : SET T1 'włącz przełącznik alarmu
        'i tranzystor sterujący piezo (w konfiguracji
        'programu użyto REL ALIAS P1.0 i T1 ALIAS P3.4)
    
```

Rys. 1 Schemat ideowy



```

Else          'jeżeli powyższe warunki nie są
              spełnione, to:
Reset Rel    'wylącz przełącznik alarmu
End If       'koniec warunku
If Minutes = 60 Then
              'jeżeli wartość minut stała się
              równa 60, to:
Minutes = 0  'wartość minut staje się równa
              0, oraz:
Incr Hours   'zwiększa się wartość godzin
If Hours = 24 Then
              'jeżeli wartość godzin stała się
              równa 24, to:
Hours = 0    'wartość godzin staje się równa
              0, oraz:
Incr Day     'zwiększ wartość dni
Correction_flag = 1
              'udziel zezwolenia na korektę
              czasu w rozpoczynającym się dniu
Call Days    'wezwij podprogram obliczający
              dni w danym miesiącu (patrz listing 2)
End If       'koniec warunku
End If       'koniec warunku
End If       'koniec warunku
Return       'koniec obsługi przerwania TIMER1
    
```

## 'Listing 2

```

Sub Days
If Day = 29 And Month = 2 Then
              'jeżeli wartość dnia zmieniła się
              na 29 dzień lutego, to:
Incr Month   'zwiększ wartość miesiący
Day = 1      'dzień miesiąca staje się równy 1
End If       'koniec warunku
If Day = 31 Then
              'jeżeli wartość dnia miesiąca
              osiągnęła 31, to
Select Case Month 'w zależności od aktualnego
miesiąca wezwij podprogram zakończenia miesiąca:
Case 4 : Call Month_incr
Case 6 : Call Month_incr
Case 9 : Call Month_incr
Case 11 : Call Month_incr
End Select
End If       'koniec warunku
If Day = 32 Then
              'jeżeli wartość dnia miesiąca
              osiągnęła 32, to:
Select Case Month 'w zależności od aktualnego
miesiąca wezwij podprogram zakończenia miesiąca:
Case 1 : Call Month_incr
Case 3 : Call Month_incr
Case 5 : Call Month_incr
Case 7 : Call Month_incr
Case 8 : Call Month_incr
Case 10 : Call Month_incr
Case 12 : Call Month_incr
End Select
End If       'koniec warunku
If Month = 13 Then
              'jeżeli wartość miesiąca
              osiągnęła 13, to wartość miesiąca staje się równa 1
End Sub
Sub Month_incr
Incr Month   'zwiększ wartość miesiący
Day = 1      'dzień równy 1
End Sub
    
```

Takich rozwiązań może być wiele. Każdy z nas ma do dyspozycji, i to całkowicie za darmo, zegar wskazujący czas z dokładnością do 1 sekundy na 5 milionów lat. Mam tu na myśli atomowy wzorzec czasu DCF dostępny za pośrednictwem fal radiowych na terenie całej Europy. Jednak zegar wykorzystujący do korekty czasu sygnał DCF jest urządzeniem dość skomplikowanym lub raczej wymagającym bardziej skomplikowanego programu i nieco "lepszego" procesora niż nasz miniaturowy zegarek. A zatem synchronizacją wskazań zegara za pomocą sygnału DCF zajmijmy się później, a dla naszego prostego układziku poszukajmy innego, tańszego i prostszego rozwiązania. Zastosowanie kosztownego generatora kwarcowego, odpornego na zmiany temperatury i dającego dokładność rzędu

ułamka sekundy na dobę niepotrzebnie zwiększyłoby koszt wykonania zegara i dlatego zdecydowałem się na dość niekonwencjonalne rozwiązanie, i ile wiem nie spotykane dotąd w konstrukcjach zegarów. Wykorzystujemy tu fakt, że błąd wskazań zegara "domowego", pracującego w mniej więcej stałych warunkach termicznych jest raczej stały i nie zmienia się w funkcji czasu. Te parę sekund na dobę możemy sobie "odpuścić", byleby błąd wskazań nie kumulował się w miarę upływu kolejnych dni. Temu właśnie służy funkcja korekty czasu, wykonywana raz na dobę, punktualnie o godzinie 12:00:30. Do licznika sekund zostaje wtedy dodana lub odjęta ustalona uprzednio doświadczalnie wartość z przedziału -20 ... +20 sekund. Większa wartość korekty nie została przewidziana, ponieważ rezonator kwarcowy dający odchyłki większe niż 20 sekund na dobę należy po prostu wyrzucić lub zastosować w układzie, w którym precyzja generatora kwarcowego nie jest zbyt istotna.

Sposób wprowadzanie korekty czasu został pokazany na **listingu 5**.

Program obsługujący nasz zegar może pracować w kilku, zmienianych za pomocą klawiszy S1 i S4 pętach programowych. Wykonują one następujące funkcje:

- wyświetlanie aktualnego czasu
- wyświetlanie aktualnej daty
- wyświetlanie czasu budzenia
- ustawianie czasu i jego ewentualnej korekty
- ustawianie daty
- ustawianie czasu budzenia
- włączanie i wyłączanie budzika
- wyświetlanie aktualnej temperatury

Wszystkie te podprogramy są w istocie bardzo podobne do siebie i dlatego też dla przykładu omówimy tylko dwa z nich: wyświetlanie czasu i ustawianie czasu. Prezentowanie całego listingu programu w tym artykule nie miałyby większego sensu, tym bardziej że jego wersja źródłowa została umieszczona na stronie [www.edw.com.pl](http://www.edw.com.pl).

Na **listingu 3** został pokazany podprogram realizujący funkcję wyświetlania aktualnego czasu. Zasada jego działania została wyczerpująco opisana w komentarzach, a dodatkowego wyjaśnienia wymaga jedynie cel wysyłania informacji o aktualnym czasie na magistralę I<sup>2</sup>C.

Magistrala I<sup>2</sup>C, szczegółowo opisywana w ramach kursu BASCOM COLLEGE, jest jednym z najważniejszych "naczyni krwionośnych" wielu systemów mikroprocesorowych. Umożliwia ona dołączanie do systemu różnych układów peryferyjnych, pełniących niejednokrotnie dość złożone funkcje, a sterowanych za pośrednictwem tylko dwóch wyprowadzeń procesora. W przypadku naszego zegara magistrala I<sup>2</sup>C umożliwi dołączenie do niego dwóch lub trzech (jeżeli chcemy także wyświetlać sekundy) modułów wyświetlaczy LED o znacznych wymiarach. Są to moduły AVT-859, opisane przez niżej podpisanego w numerze 8/00 Elektroniki Praktycznej. Zastosowanie takich wy-

świetlaczy, niestety relatywnie kosztownych, o wysokości 57 mm, umożliwi obserwację wskazań zegara z odległości do kilkudziesięciu metrów i nawet w całkowitej ciemności.

## 'Listing 3

```

Sub Display_time
Cls          'wyczyść ekran wyświetlacza
Seconds_flag = 1
              'zmienna pomocnicza
              SECONDS_FLAG przyjmuje wartość 1, zezwalającą
              na wyświetlenie na ekranie
              aktualnego czasu
Waitms 255  'zaczekaj 255 ms
Do
If Seconds_flag = 1 Then
              'jeżeli jest zezwolenie na
              wyświetlenie czasu, to:
Seconds_flag = 0
              'anuluj zezwolenie aż do upływu
              kolejnej sekundy (patrz listing 1)
Home        'ustaw kursor na początkowej pozycji
Lcd "Time: " 'wyświetl napis informacyjny
If Hours < 10 Then Lcd "0";
              'jeżeli wartość godzin jest
              mniejsza od 10, to wyświetl zero wiodące
Lcd Hours ; ";" 'wyświetl wartość godzin
If Minutes < 10 Then Lcd "0";
              'jeżeli wartość minut jest mniejsza
              od 10, to wyświetl zero wiodące
Lcd Minutes ; ":";
              'wyświetl wartość minut
If Seconds < 10 Then Lcd "0";
              'jeżeli wartość sekund jest
              mniejsza od 10, to wyświetl zero wiodące
Lcd Seconds 'wyświetl wartość sekund
X = Makebcd(hours)
              'przedstaw wartość godzin w
              kodzie BCD
I2csend 112 , X
              'wyslij otrzymaną wartość na
              magistralę I2C pod adres 112
X = Makebcd(minutes)
              'przedstaw wartość minut w
              kodzie BCD
I2csend 114 , X
              'wyslij otrzymaną wartość na
              magistralę I2C pod adres 114
X = Makebcd(seconds)
              'przedstaw wartość sekund w
              kodzie BCD
I2csend 116 , X
              'wyslij otrzymaną wartość na
              magistralę I2C pod adres 116
End If      'koniec warunku i wyświetlania danych
Set S1     'spróbuj ustawić stan wysoki na
              przycisku S1
If S1 = 0 Then Call Display_date
              'jeżeli próba nieudana (przycisk
              naciśnięty) to przejdź do podprogramu
              wyświetlania daty
Set S4     'spróbuj ustawić stan wysoki na
              przycisku S4
If S4 = 0 Then Call Settime
              'jeżeli próba nieudana, to wezwij
              podprogram ustawiania czasu
Loop
End Sub
    
```

**Listing 4** pokazuje podprogram ustawiania bieżącego czasu. Podprogram ustawiania daty zbudowany jest prawie identycznie i nie ma w tej chwili sensu szczegółowo się nim zajmować.

## 'Listing 4

```

Sub Settime
Cls          'wyczyść ekran wyświetlacza
Setting_flag = 1
              'zmienna pomocnicza
              SETTING_FLAG przyjmuje wartość 1 zezwalającą na
              wyświetlenie danych
Temp1 = Hours 'zmienna pomocnicza TEMP1
              przyjmuje wartość godzin
Temp2 = Minutes
              'zmienna pomocnicza TEMP2
              przyjmuje wartość minut
    
```



```

Waitms 255 'zaczekaj 255 ms
Do
Do
If Setting_flag = 1 Then
    'jeżeli zmienna pomocnicza
    SETTING_FLAG ma wartość 1, to:
    Home 'ustaw kursor na pierwszej pozycji
    Lcd "Set time: " 'wyświetl komunikat
    If Temp1 < 10 Then Lcd "0";
    'jeżeli wartość zmiennej TEMP1
    jest mniejsza od 10, to wyświetl zero wiodące
    Lcd Temp1 ; ""; 'wyświetl wartość zmiennej
    TEMP1
If Temp2 < 10 Then Lcd "0";
    'jeżeli wartość zmiennej TEMP2
    jest mniejsza od 10, to wyświetl zero wiodące
    Lcd Temp2 ; 'wyświetl wartość zmiennej
    TEMP2
Setting_flag = 0 'wprowadź zakaz wyświetlania
    danych aż do wprowadzenia zmiany którejś
    z wartości
End If 'koniec warunku i koniec wyświetlania
Waitms 255 'zaczekaj 255 ms
Set S2 'spróbuj ustawić stan wysoki na
    przycisku S2
If S2 = 0 Then 'jeżeli próba nieudana (przycisk
    naciśnięty), to:
    Incr Temp1 'zwiększ wartość zmiennej
    TEMP1
Setting_flag = 1 'udziel zezwolenia na aktualizację
    wyświetlania danych
If Temp1 = 24 Then Temp1 = 0
    'jeżeli zmienna TEMP1
    przekroczyła wartość 23, to zmienna TEMP1
    przyjmuje wartość 0
End If 'koniec warunku
Set S3 'spróbuj ustawić stan wysoki na
    przycisku S3
If S3 = 0 Then 'jeżeli próba nieudana (przycisk
    naciśnięty), to:
    Incr Temp2 'zwiększ wartość zmiennej
    TEMP2
Setting_flag = 1 'udziel zezwolenia na aktualizację
    wyświetlania danych
If Temp2 = 60 Then Temp2 = 0
    'jeżeli zmienna TEMP2
    przekroczyła wartość 59, to zmienna TEMP1 przyjmuje
    wartość 0
End If 'koniec warunku
Set S4 'spróbuj ustawić stan wysoki na
    przycisku S4
If S4 = 0 Then 'jeżeli próba nieudana (przycisk
    naciśnięty), to:
    Hours = Temp1 'zmienna godzin przyjmuje
    wartość zmiennej TEMP1
    Minutes = Temp2 'zmienna minut przyjmuje wartość
    zmiennej TEMP2
    Seconds = 0 'licznik sekund zostaje
    wyzerowany
    Call Set_correction
    'wezwiń podprogram ustawiania
    korekty czasu
End If
Loop
End Sub
    
```

Warto jeszcze przyjrzeć się podprogramowi ustawiania korekty czasu przedstawionemu na **listingu 5**.

Nie mieliśmy jeszcze okazji wspomnieć o kolejnej funkcji naszego zegara, nie związanej tym razem z pomiarem upływu czasu. Jest nią wbudowany w program obsługi zegara moduł pomiaru temperatury w otoczeniu.

Do pomiaru temperatury wykorzystany został popularny układ termometru cyfrowego DS1820 produkcji firmy DALLAS. Procedury odczytu temperatury z tego układu i wyświetlania wyników pomiaru na wyświetlaczu LCD zostały pokazane na **listingu 6**. Procedura kalkulacji otrzymanych z układu danych w celu uzyskania rozdzielczości 0,1 stopnia jest wzorowana na programie opublikowanym przez firmę MCS Electronics na jej stronie [www.mcselec.com](http://www.mcselec.com).

Niestety, program ten zawierał pewne, dość istotne błędy, uniemożliwiające poprawne wyświetlanie temperatur ujemnych i w takiej postaci został niegdyś opublikowany przez mnie w *Elektronice Praktycznej*. Błędy algorytmu konwersji danych zostały zauważone i poprawione przez Kolegę **Artura Klimuszko** ([artx@poland.com](mailto:artx@poland.com)), a ja korzystam z okazji do złożenia Mu podziękowań.

## 'Listing 5

```

'UWAGA: zmienna SECONDS_CORRECTION musi
    być zadeklarowana jako SINGLE, aby umożliwić przy-
    mowanie wartości mniejszych od zera
Sub Set_correction
Cls 'wyczyść ekran wyświetlacza
Do
Home 'ustaw kursor na pierwszej pozycji
Lcd "Correction: "; Seconds_correction ; "
    'wyświetl komunikat i aktualną
    wartość korekty czasu
Waitms 255 'zaczekaj 255 ms
Set S2 'spróbuj ustawić stan wysoki na
    przycisku S2
If S2 = 0 And Seconds_correction < 20 Then Incr
    Seconds_correction
    'jeżeli przycisk naciśnięty i wartość
    współczynnika korekty jest mniejsza od 20, to zwiększ
    wartość zmiennej SECONDS_CORRECTION
Set S3 'spróbuj ustawić stan wysoki na
    przycisku S3
If S3 = 0 And Seconds_correction > -20 Then Decr
    Seconds_correction
    'jeżeli przycisk naciśnięty i wartość
    współczynnika korekty jest większa od -20, to zmniejsz
    wartość zmiennej SECONDS_CORRECTION
Set S4 'spróbuj ustawić stan wysoki na
    przycisku S4
If S4 = 0 Then Call Display_time
    'jeżeli przycisk naciśnięty, to powróć
    do podprogramu wyświetlania czasu
Loop
End Sub
    
```

## 'Listing 6

```

Sub Display_temperature
1wreset 'inicjalizacja magistrali 1WIRE
If Err = 1 Then 'jeżeli do magistrali nie jest
    dołączony termometr DS1820, to:
Cls 'wyczyść ekran wyświetlacza
Lcd " NO SENSOR"
    'wyświetl komunikat o braku
    termometru
Wait 1 'zaczekaj 1 sekundę
Call Display_time 'ponownie wezwij podprogram
    wyświetlania aktualnego czasu
End If
Do
1wwrite &HCC 'żądanie zgłoszenia się układu
    dołączonego do magistrali 1WIRE
1wwrite &H44 'rozpoczęcie konwersji analogowej
    wartości zmierzonej temperatury na postać cyfrową
Waitms 255 'zaczekaj 255 ms
Waitms 255 'zaczekaj 255 ms
Waitms 255 'zaczekaj 255 ms
Set S1 'spróbuj ustawić stan wysoki na
    przycisku S1
If S1 = 0 Then Call Display_time
    'jeżeli próba nieudana (przycisk
    naciśnięty), to wezwij podprogram wyświetlania czasu
    Call Read1820
    'wezwiń podprogram odczytu
    danych z termometru DS1820
Cls 'wyczyść ekran wyświetlacza
Locate 1, 3 'ustaw kursor na wskazanej pozycji
Lcd "T = " 'wyświetl literę "T"
Locate 1, 12 'ustaw kursor w 12 kolumnie
Lcd Chr(0); "C" 'wyświetl symbol stopnia
If T < 10 And T >= 0 Then
    'jeżeli temperatura mniejsza od 10
    a większa od zera, to:
    Locate 1, 7 na pozycji 7
    
```

```

Lcd "0"; T 'wyświetl zero wiodące
Else 'w przeciwnym przypadku
If T < 0 And T > -10 Then
    'jeżeli temperatura mniejsza od zera, to:
    Locate 1, 7
    T = T * -1
    'przelicz wartość temperatury
    Lcd "0"; T
    'dodaj zero wiodące ze znakiem "-"
    w przeciwnym przypadku:
Else
    Locate 1, 8
    'ustaw kursor na pozycji 8
    Lcd T 'wyświetl wartość temperatury
    T = T / 10
    'podziel wartość temperatury przez 10
    Locate 1, 7
    'cofnij kursor na pozycję 7
    Lcd T ; ""
    'wyświetl wartość temperatury
    + znak dziesiętny
End If 'koniec warunku
End If 'koniec warunku
Loop
End Sub
Sub Read1820
1wreset
1wwrite &HCC 'żądanie zgłoszenia się układu
    dołączonego do magistrali 1WIRE
1wwrite &HBE 'żądanie podanie wyniku pomiaru
    temperatury
Bd(1) = 1wread(9)
    'odczytanie wyniku
Tmp = Bd(1) And 1
    'ta i dalsze linie: konwersja odczytanego
    wyniku do postaci liczby dziesiętnej
If Tmp = 1 Then Decr Bd(1)
T = Makeint(bd(1), Bd(2))
T = T * 50
T = T - 25
T1 = Bd(8) - Bd(7)
T1 = T1 * 100
T1 = T1 / Bd(8)
T = T + T1
T = T / 10
End Sub
    
```

Na tym możemy zakończyć pobieżne omawianie programu sterującego pracą naszego zegarka. Chciałbym, aby Czytelnicy potraktowali zaprezentowane listingi jedynie jako wskazówki do samodzielnego napisania programu sterującego pracą zegara. Oczywiście, zaprogramowane procesory będą dołączane w kitach, a także dołączane do płyt obwodów drukowanych, ale samodzielne napisanie programu będzie zawsze największą satysfakcją dla Konstruktorów.

## Montaż i uruchomienie

Na **rysunku 2** została pokazana mozaika ścieżek płytki obwodu drukowanego wykonanego na laminacie jednostronnym oraz rozmieszczenie na niej elementów. Tak właściwie to mamy do dyspozycji aż dwie płytki: druga z nich może posłużyć jako w miarę estetyczna płyta czołowa, umożliwiająca szybkie wykonanie obudowy do zegarka.

Montaż układu wykonujemy typowo, z następującym wyjątkiem: **wyświetlacz alfanumeryczny i przyciski S1...S4 muszą zostać zamontowane od strony ścieżek, po uprzednim wlotowaniu wszystkich pozostałych elementów. Do zamocowania wyświetlacza wygodnie będzie użyć pojedynczego szeregu goldpinów. Przed ostatecznym zamontowaniem wyświetlacza musimy bardzo dokładnie sprawdzić jakość wykonania pozostałych połączeń,**

ponieważ po zamontowaniu tego elementu utracimy dostęp do położonych pod nim ścieżek i punktów lutowniczych.

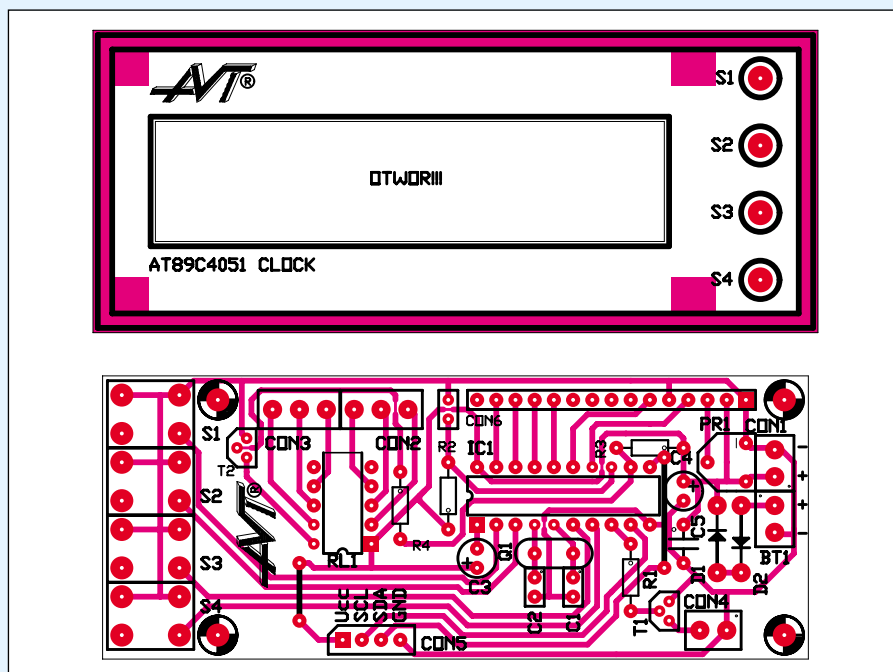
Po zmontowaniu płytki bazowej zegara pozostanie nam już tylko połączenie jej z płytą czołową, o ile oczywiście zechcemy wykorzystywać ten dodatkowy element. Połączenia możemy dokonać za pomocą czterech śrubek M3, których łebki przylutujemy do dużych punktów lutowniczych zlokalizowanych na spodniej stronie płyty czołowej. Podłużne punkty lutownicze umieszczone także na spodniej stronie płyty czołowej mogą posłużyć do przylutowania do nich paszków laminatu lub cienkiej blachy stanowiących boczne ścianki obudowy zegara.

Po wykonaniu tych wszystkich czynności możemy już włączyć zasilanie zegara, czyli dołączyć do złącza CON1 napięcie +5VDC, stabilizowane. Jako źródło zasilania najlepiej wykorzystać zasilacz tzw. "wtyczkowy" o dowolnej obciążalności prądowej.

Pobór prądu przez nasz zegar jest bardzo mały i z wyświetlaczem alfanumerycznym bez podświetlania nie przekracza 20mA. Na płytce umieszczone zostało dodatkowe złącze, oznaczone jako BT1. Do tego złącza możemy dołączyć baterię składającą się z czterech ogniw R6 lub AAA, pełniącą rolę zasilania awaryjnego, podtrzymującego wskazania zegara podczas przerw w zasilaniu z sieci.

### Funkcje klawiszy S1...S4

**Klawisz S1:** Kolejne naciśnięcia tego klawisza powodują przechodzenie układu do wyświetlania: czasu, daty, czasu ustawiania budzika, wyświetlania temperatury.

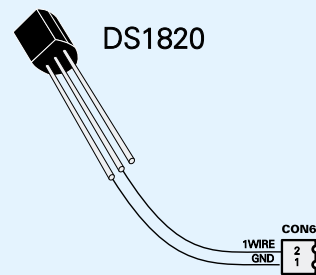


Rys. 2 Schemat montażowy

**Klawisz S4:** Naciśnięcie tego klawisza podczas wyświetlania czasu, daty lub czasu budzenia powoduje przejście układu w tryb ustawiania wymienionych wartości. Powtórne naciśnięcie S4 powoduje powrót do wyświetlania danych.

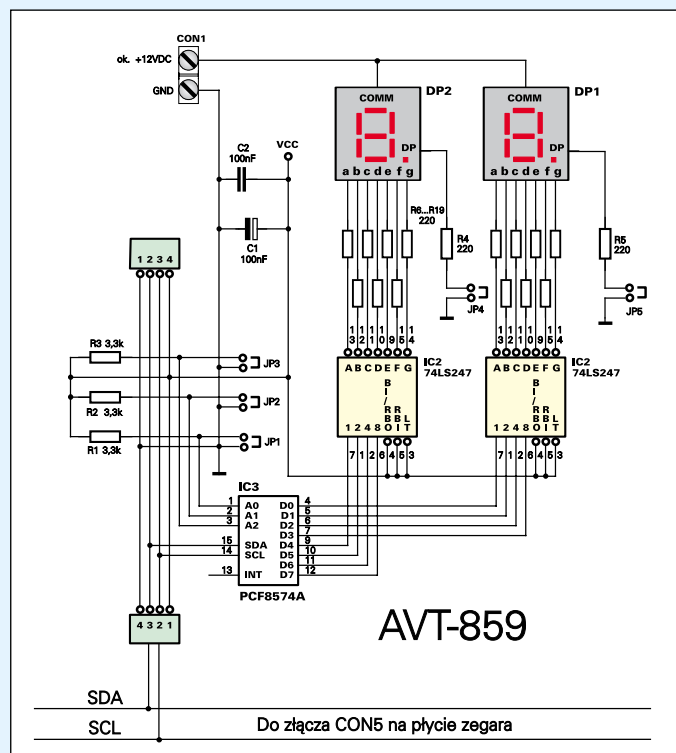
**Klawisz S3:** W trybie ustawiania cyklicznie zwiększa wartość godzin lub dni miesiąca. Podczas ustawiania korekty czasu zmniejsza jej wartość. W trybie wyświetlania danych włącza/wyłącza budzik.

**Klawisz S2:** W trybie ustawiania cyklicznie zwiększa wartość minut lub miesiąca. Podczas ustawiania korekty czasu zwiększa jej wartość.



Rys. 3

Rys. 4



Na rysunku 3 został pokazany sposób połączenia termometru DS1820 z układem zegara. Przewód łączący termometr z zegarkiem może być praktycznie dowolnej długości (do 100 m).

Na rysunku 4 przedstawiono schemat połączenia zegara z modułem wyświetlaczy siedmio-segmentowych LED AVT-859. Drugi moduł AVT-859 podłącza się do linii identycznie.

### Wykaz elementów

<b>Kondensatory</b>	
C1, C2	..... 33pF
C3	..... 1µF/16V
C4	..... 100µF/16V
C5	..... 100nF
<b>Rezystory</b>	
PR1	..... potencjometr montażowy miniaturowy 1kΩ
R1, R3, R4	..... 10kΩ
R2	..... 4,7kΩ
<b>Półprzewodniki</b>	
D2, D1	..... 1N4001
IC1	..... zaprogramowany procesor AT89C4051
T1, T2	..... BC548
Dodatkowo układ termometru DS1820	
<b>Pozostałe</b>	
DP1	..... wyświetlacz alfanumeryczny 16*1
CON1, CON4	..... ARK2 (3,5mm)
CON2, CON3	..... ARK3 (3,5mm)
CON5	..... 4x goldpin
PK1	..... OMRON 5V
Q1	..... rezonator kwarcowy 11.059MHz
S1, S2, S3, S4	..... przycisk microswitch 16mm
Rząd 14 goldpinów do zamocowania wyświetlacza	

Zbigniew Raabe  
e-mail: zbigniew.raabe@edw.com.pl

**Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2488**