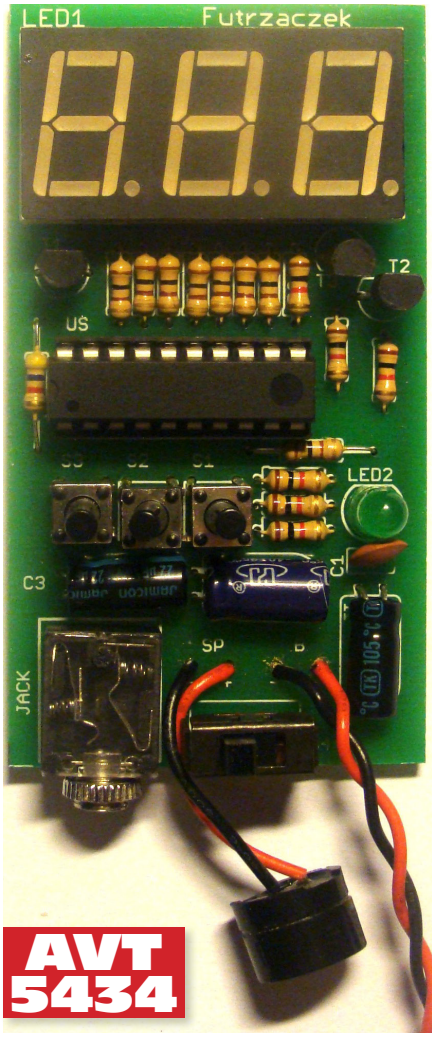
Metronom dla muzyków

Równomierne wybijanie rytmu jest dla początkującego muzyka zadaniem niełatwym, jednocześnie istotnym dla uzyskiwania postępów w nauce gry. Dostępne w handlu urządzenia, mające pomagać w tej czynności, są przeważnie przeładowane najróżniejszymi funkcjami, a przez to drogie. Niniejszy układ jest tani i prosty w montażu, zaś jego jedyną – i najistotniejszą zarazem – funkcją jest wystukiwanie równomiernego rytmu. ***Rekomendacje:*** *przyrząd może przydać się muzykom i jest alternatywą dla drogich rozwiązań fabrycznych.*



Metronom może być zasilany z baterii lub zasilacza sieciowego. Ma sygnalizację optyczną, wewnętrzny głośniczek oraz wyjście słuchawkowe. Za pośrednictwem trzycyfrowego wyświetlacza LED można ustawić częstotliwość powtórzeń w zakresie 30…250 BPM (ang. Beats Per Minute – uderzeń na minutę) i wybrać jeden z pięciu dźwięków. Parametry te zostają zapisane w nieulotnej pamięci EEPROM, by przy następnym uruchomieniu urządzenia, nie zachodziła konieczność ponownego ich ustawiania.

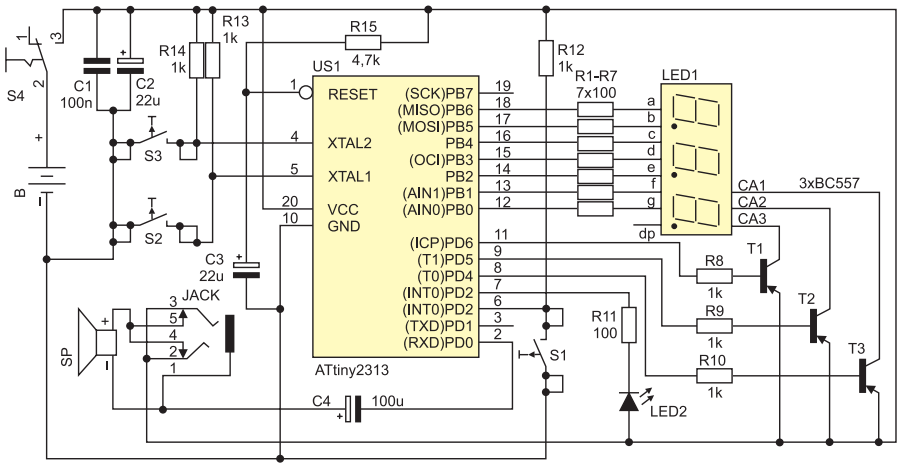
Zasada działania

Schemat ideowy metronomu pokazano na **rysunku 1**. „Sercem" urządzenia jest mikrokontroler US1 typu ATtiny2313. Do wyprowadzeń portu B, za pośrednictwem rezystorów R1…R7, dołączone są katody trzycyfrowego wyświetlacza LED1, który jest sterowany w trybie multipleksowanym: wspólne dla każdej cyfry anody zasilane są poprzez tranzystory T1…T3 z trzech wyprowadzeń portu D. Zastosowanie tranzystorów było konieczne, ponieważ wydajność prądowa poszczególnych wyprowadzeń mikrokontrolera nie przekracza 20 mA. Dioda LED2 służy do sygnalizacji optycznej wybijanego rytmu. Przyciski S1…S3 są dołączone pomiędzy masę a wejście mikrokontrolera. Rezystory R12…R14 utrzymują na nich poziom wysoki podczas oczekiwania na wciśnięcie któregokolwiek z przycisków. Obwód rezystor R15 – kondensator C3 odpowiada za restart mikrokontrolera po włączeniu zasilania. Kondensatory C1 i C2 filtrują zakłócenia, które mogłyby rozprzestrzeniać się po układzie.

Z pinu 2 (Portd0) jest uzyskiwany sygnał akustyczny. Ponieważ wyprowadzenia układów serii AVR – pracujące w charakterze wyjścia – mają mniejszą rezystancję wewnętrzną na poziomie niskim, więc głośnik jest włączony pomiędzy to wyjście, a dodatni biegun zasilania. Składową stałą oddziela kondensator C4. Gniazdo *Jack* ma wewnętrzny przełącznik, który odłącza głośnik SP po włożeniu do niego wtyczki. Obydwa kanały w wyjściu słuchawkowym są połączone równolegle. Dzięki takiemu rozwiązaniu, układy regulacji głośności, znajdujące się w słuchawkach, mogą działać poprawnie, niezależnie od ich konstrukcji. Istnieje również możliwość wyprowadzenia sygnału np. do wzmacniacza. Połączenie szeregowe mogłoby stwarzać w tej kwestii problemy.

Omówienie programu

Program został napisany w środowisku BASCOM-AVR. Jego kod zamieszczono na **listingu 1**. Program używa obu timerów



Rysunek 1. Schemat ideowy metronomu

W ofercie AVT*

AVT-5434 A AVT-5434 B
AVT-5434 UK

Podstawowe informacje:

- Płytka drukowana o wymiarach 43 mm×78 mm.
- Zasilanie: 2,7…5 V/maks. 90 mA.
- Rytm ustawiany w zakresie 30…250 BPM.
- Mikrokontroler ATtiny2313, oprogramowanie w języku Bascom AVR

Dodatkowe materiały na CD lub FTP:

ftp://ep.com.pl, user: 85414, pass: 2nev3854

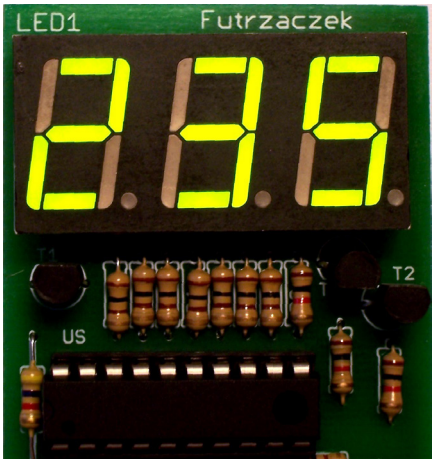
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

Projekty pokrewne na CD/FTP:

(wymienione artykuły są w całości dostępne na CD)

AVT-915 Elektroniczny metronom (EP 1/2006)
AVT-1109 Metronom cyfrowy (EP 12/1996)

* Uwaga:
Zestawy AVT mogą występować w następujących wersjach:
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.
AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlutowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf
AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)
Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). *http://sklep.avt.pl*



Fotografia 2. Wyświetlacz podczas zmiany wartości BPM

mikrokontrolera. Timer0, z preskalerem dzielącym częstotliwość zegara przez 8, steruje wyświetlaczami. Ma on pojemność 8 bitów i jest do niego ładowana wartość 156 (2^8-100=156), co oznacza, że podprogram obsługi przerwania wykonuje się ok. 800 razy na sekundę, zaś każda cyfra odświeżana jest z częstotliwością trzykrotnie mniejszą, wynoszącą ok. 266 Hz. Jest to wartość wystarczająca dla uzyskania wrażenia braku migotania wyświetlanych cyfr.

16-bitowy Timer1, z preskalerem dzielącym częstotliwość zegara przez 64, odpowiada za regularne emitowanie sygnałów dźwiękowych przez układ. Przerwanie wywoływane od jego przepełnienia jest wykonywane co 4 sekundy (1 MHz/64=15625 Hz oraz 2^{16}/15625 Hz=4 s. Ponieważ odstęp czasu między kolejnymi dźwiękami zawiera się w przedziale od 240 ms (60 s/250 PBM=0,24 s) do 2 s (60 s/30 BPM=2 s), wystarczy załadować do niego odpowiednią wartość, by uzyskać żądany interwał. Jest ona obliczana w podprogramie *Czas*. Najpierw oblicza się konieczną do załadowania wartość, zależną od BPM, która potem

Wykaz elementów

Rezystory: (0,25 W)
R1…R7, R11: 100 Ω
R8…R10, R12…R14: 1 kΩ
R15: 4,7 kΩ
Kondensatory:
C1: 100 nF/50 V (ceramiczny)
C2, C3: 22 µF/16 V (elektrolit.)
C4: 100 µF/16 V (elektrolit.)
Półprzewodniki:
LED1: wyświetlacz 7-segmentowy, 3 cyfry, wspólna anoda
LED2: dioda LED 5 mm, zielona
T1-T3 BC557
US1 ATtiny2313
Inne:
JACK gniazdo 3,5 mm stereo, do druku, z wyłącznikiem
S1-S3 monostabilne, styki SPST-NO, wym. (6×6×13) mm
S4 suwakowy, ON-ON, raster 4 mm
SP opis w tekście
podstawka DIL-20

Listing 1. Program sterujący pracą metronomu

```
 $regfile = „attiny2313.dat”  ,procesor ATTiny2313
$crystal = 1000000  ,oscylator wewnętrzny
Config Portb = Output  ,katody wyświetlaczy
Config Pind.6 = Output  ,setki
Config Pind.4 = Output  ,dziesiątki
Config Pind.5 = Output  ,jedności
Config Pind.3 = Output  ,dioda LED
Config Pind.0 = Output  ,głośniczek
Config Pind.1 = Output  ,nieużywane wyprowadzenie
Config Pina.1 = Input  ,switch S3 - zmiana tonu
Config Pina.0 = Input  ,switch S2 - BPM --
Config Pind.2 = Input  ,switch S1 - BPM++

Config Timer0 = Timer , Prescale = 8  ,timer do sterowania wyświetlaczami
Config Timer1 = Timer , Prescale = 64  ,timer do odliczania przerw
Declare Sub Cyfry  ,podprogram rozbijający liczbę na cyfry
Declare Sub Czas  ,podprogram obliczający wartość ładowaną do Timer1
Declare Sub Zakres  ,podprogram kontrolujący zakres BPM
On Timer0 Wyswietlaj  ,przerwanie od Timer0 powoduje skok do Tyswietlaj
On Timer1 Dzwiek  ,przerwanie od Timer1 powoduje skok do Dzwiek
Dim Setki As Byte  ,cyfra setek
Dim Dziesiatki As Byte  ,cyfra dziesiątek
Dim Jednosci As Byte  ,cyfra jedności
Dim Nr_wysw As Byte  ,numer uruchomionego wyświetlacza
Dim Bpm As Byte  ,BPM - ilość uderzeń na minutę
Dim Bpm_kopia As Byte  ,wartość BPM, kopiowana na czas obliczeń
Dim Ton As Byte  ,numer wybranego dźwięku
Dim Flaga_opoznienia As Bit  , podtrzymanie świecenia wyświetlacza
Dim Flaga_timer1 As Bit
Dim Wart_pocz As Word  ,wartość obliczona, ładowana do Timer1

Enable Interrupts  ,uruchamiany przerwania
Load Timer0 , 100  ,wpisujemy początkową wartość Timer0
Set Portd.0  ,włączamy rezystory podciągające
Set Portd.1
Set Portd.2
Set Portd.3
Set Portd.4
Set Portd.5
Set Portd.6
Set Porta.1
Set Porta.0

Set Flaga_opoznienia  ,wygasi wyświetlacz po pokazaniu BPM
Reset Flaga_timer1
Readeeprom Bpm , 5  ,wczytanie z EEPROM wartości BPM
Readeeprom Ton , 8  ,wczytanie z EEPROM numeru wybranego dźwięku
Zakres  ,jeżeli wartość będzie spoza zakresu (np. 0) - skoryguj ją
Czas  ,przeliczenie wczytanej wartości BPM na początkową wartość licznika
Cyfry ,rozbicie liczby BPM na trzy cyfry
Enable Timer0  ,włączenie Timer0 - pokazanie wczytanej z EEPROM wartości
Set Flaga_opoznienia  ,ustawienie flagi - wygasi wyświetlacz
Load Timer1 , Wart_pocz  ,załadowanie do Timer1 obliczonej wartości
Enable Timer1  ,uruchomienie Timer1 i rozpoczęnie „stukania”

Do
  If Pind.2 = 0 Then  ,po naciśnięciu S1
    Disable Timer1  ,wyłącz Timer1 - przerwania zakłócają wyświetlacz
    Set Flaga_opoznienia  ,ustaw flagę, która później wyłączy wyświetlacz
    Incr Bpm  ,zwiększaj wartość BPM
    Zakres  ,sprawdzaj, czy ustalona wartość mieści się w zakresie
    Cyfry  ,rozbicie liczby BPM na trzy cyfry
    Enable Timer0  ,włączenie Timer0 - uruchomienie wyświetlacza
    Waitms 100  ,odczekanie 100ms, by inkrementacja nie była zbyt szybka
  End If

  If Pina.0 = 0 Then  ,po naciśnięciu S2
    Disable Timer1  ,wyłącz Timer1-przerwania zakłócają wyświetlacz
    Set Flaga_opoznienia  ,ustaw flagę, która później wyłączy wyświetlacz
    Decr Bpm  ,dekrementacja wartości BPM
    Zakres  ,sprawdzaj, czy ustalona wartość mieści się w zakresie
    Cyfry  ,rozbicie liczby BPM na trzy cyfry
    Enable Timer0  ,włączenie Timer0 - uruchomienie wyświetlacza
    Waitms 100  ,odczekanie 100ms, by inkrementacja nie była zbyt szybka
  End If

  If Pina.1 = 0 Then  ,po naciśnięciu S3
    Set Flaga_timer1
    Disable Timer1  ,wyłącz Timer1-przerwania zakłócają wyświetlacz
    Incr Ton  ,inkrementuj wartość zmiennej Ton
    If Ton >= 5 Then  ,jeżeli wyszliśmy poza zakres...
    Ton = 0  ,...wróć do zera
  End If
  Setki = 33  ,wyłączenie cyfry setek
  Dziesiatki = Ton  ,numer dźwięku jest na środkowej cyfrze wyświetlacza
  Jednosci = 33  ,wyłączenie cyfry jedności
  Enable Timer0  ,uruchomienie wyświetlacza
  Wait 1  ,odczekanie sekundy, by inkrementacja była widoczna
End If

,po zwolnieniu wszystkich przycisków po zmienianej wartości BPM
If Flaga_opoznienia = 1 And Pina.0 = 1 And Pina.1 = 1 And Pind.2 = 1 Then
  Czas  ,dokonaj przeliczenia nowej wartości BPM
  Load Timer1 , Wart_pocz  ,załaduj ją do licznika Timer1
  Waitms 750  ,odczekaj 3/4 sekundy
  Disable Timer0  ,wyłącz wyświetlacz...
  Set Portd.4  ,...i wszystkie cyfry
  Set Portd.5
  Set Portd.6
  Writeeeprom Bpm , 5  ,zapisz nową wartość do EEPROM
  Reset Flaga_opoznienia  ,wygaś flagę - spełniła swoje zadanie
  Enable Timer1  ,włącz Timer1, czyli rozpocznij „pukanie”
End If
```

jest mnożona przez liczbę sekund w minucie. Część ułamkową podczas dzielenia traci się, gdyż wynik dzielenia jest zapisywany do zmiennej typu całkowitego. Nie trzeba się tym przejmować, ponieważ błąd (1/15625≈0,06 ‰) jest praktycznie niesłyszalny. Podprogram obsługujący przerwanie najpierw emituje dźwięk (jeden z pięciu), a potem załącza diodę LED2 na czas 400 µs – wystarczająco długo, by dostrzec jej mignięcie, z drugiej strony na tyle krótko, by między kolejnymi impulsami były wyraźne przerwy.

Wyboru liczby wystukiwanych impulsów na minutę dokonuje się przyciskiem S1 (zwiększanie wartości) lub S2 (zmniejszanie wartości). Lecz każdy z nich oprogramowany jest, poza wymienioną różnicą, tak samo: między kolejnymi inkrementacjami lub dekrementacjami zmiennej *Bpm* program wstrzymywany jest na ok. 100 ms; na tyle wolno, by można było dostrzec zmianę na wyświetlaczu, jednocześnie na tyle szybko, by regulacja odbywała się sprawnie. Po każdej iteracji pętli, podprogram *Cyfry* „rozbija”, przy pomocy operacji modulo, zawartość zmiennej *Bpm* na trzy zmienne, zawierające cyfry gotowe do wyświetlenia.

Wraz z naciśnięciem przycisku, zmienną *Flaga_opoznienia* ustawia się w stan wysoki. Po zwolnieniu zaś, odpowiada za obliczenie nowej wartości do załadowania do Timer1, podtrzymanie świecenia wyświetlacza przez 750 ms, zapisanie ustawionej wartości do pamięci EEPROM i na końcu – uruchomienie Timer1. Na czas pracy wyświetlacza jest on wyłączany (poleceniem *disable*), a to z tego względu, że przerwania od niego generowane, zakłócają proces multipleksowania. Widok wyświetlacza pokazano na **fotografii 2**.

Bardzo podobnie wygląda obsługa przycisku S3, którym jest dokonywany wybór sygnału dźwiękowego. Różnica polega na tym, że pracuje jedynie środkowa cyfra wyświetlacza (empirycznie ustalono, że liczba 33 wyłącza cyfrę), zaś inkrementowana zmienna *Ton* pracuje w pętli, podobnie jak *Bpm*, tyle, że znacznie węższej. Widok wyświetlacza w tej fazie pokazuje **fotografia 3**.

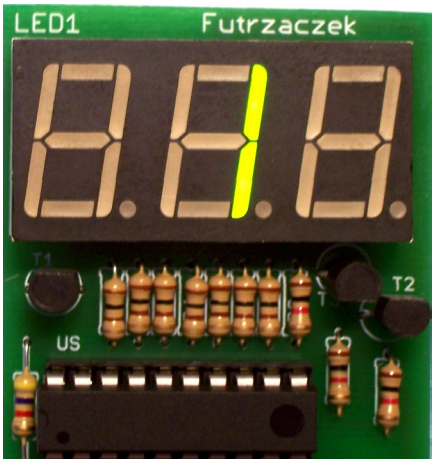
Do programu wpisano pięć różnych dźwięków. Z pewnym przybliżeniem (gdyż komenda *Sound* nie służy do generowania dokładnych przebiegów) można przyjąć, że jest to ton podstawowy 440 Hz i jego kolejne wielokrotności. Nic nie stoi na przeszkodzie, by je – według własnego gustu – zmodyfikować lub dopisać nowe, pamiętając jednocześnie o zwiększeniu górnej granicy zmiennej *Ton*.

Ostatnim omówionym tu elementem będzie sterujący wyświetlaniem podprogram *Wyswietlaj*. Jest on uruchamiany po przepełnieniu licznika Timer0. Najpierw jest ładowana nowa wartość, po czym są wyłączane wszystkie cyfry, poprzez ustawienie wyprowadzeń sterujących tranzystorami typu PNP. Następnie, odczytując wartość zmiennej *Nr_wysw*, dokonuje się wyboru jednego z nich i za pomocą polecenia *Lookup* wyprowadza przez PortB liczbę, zapisaną w postaci binarnej w bloku danych *Cyfry*, która powoduje włączenie odpowiednich segmentów.

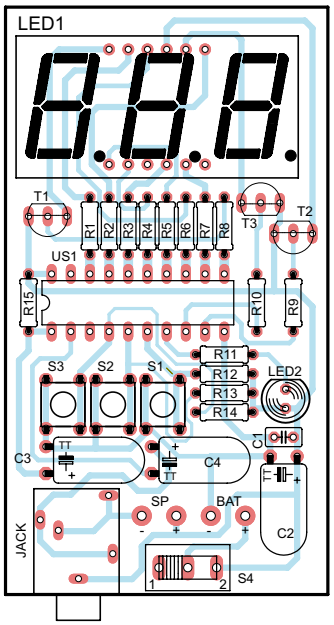
Program działa z domyślnymi, fabrycznymi ustawieniami fusebitów – wewnętrzny oscylator 8 MHz z podziałem częstotliwości zegara przez 8.

Montaż i uruchomienie

Układ zmontowano na jednostronnej płytce drukowanej o wymiarach 43 mm×78 mm, której schemat montażowy pokazano na **rysunku 4**. Montaż jest wykonywany typowo: od elementów najniższych po najwyższe. Pod układ US1 należy zastosować podstawkę. Kondensatory elektrolityczne powinny być wlutowane w pozycji leżącej – wówczas wyświetlacz, przyciski, przełącznik i diodę można bezpośrednio przystawić do ścianki. Gniazdo *Jack* jest tak usytuowane, że można je przykręcić nakrętką do obudowy. W układzie modelowym, jako głośnik SP, zastosowano głośniczek piezoelektryczny (bez



Fotografia 3. Wyświetlacz podczas wyboru rodzaju dźwięku



Rysunek 4. Schemat montażowy metronomu

generatora), o rezystancji cewki wynoszącej w przybliżeniu 16 Ω. Bardzo dobrze spisuje się przy napięciu zasilania przekraczającym 4 V, lecz poniżej tej wartości staje się za cichy, zwłaszcza do ćwiczeń z perkusją. Innych rozwiązaniem jest użycie zwykłego głośnika elektrodynamicznego o impedancji cewki 8 Ω lub 16 Ω. Zwiększona powierzchnia membrany zapewnia dobrą słyszalność, nawet przy zasilaniu napięciem 3 V.

Układ ATtiny2313 może być zasilany napięciem z przedziału 2,7…5,5 V. Pobór prądu przy włączonym wyświetlaczu wynosi, w zależności od liczby włączonych segmentów, 50…90 mA przy 5 V lub 30…60 mA przy 3 V. W trybie pracy, po wyłączeniu wyświetlacza, pobór prądu spada do kilku miliamperów. Oznacza to, że źródłem zasilania dla metronomu mogą być dwie lub trzy połączone szeregowo baterie o napięciu 1,5 V albo zasilacz sieciowy.

W układzie modelowym zastosowano wyświetlacz świecący w kolorze zielonym.

Listing 1. c.d.

```
,po zwolnieniu wszystkich przycisków po zmienianym tonie dźwięku
If Flaga_timer1 = 1 And Pina.0 = 1 And Pina.1 = 1 And Pind.2 = 1 Then
  Waitms 750  ,odczekaj 3/4 sekundy
  Disable Timer0  ,wyłączenie wyświetlacza
  Set Portd.4  ,wyłączenie anod wyświetlaczy
  Set Portd.5
  Set Portd.6
  Enable Timer1  ,uruchomienie Timer1 - rozpoczęnie „pukania”
  Writeeeprom Ton , 8  ,zapisanie numeru dźwięku do EEPROM
  Reset Flaga_timer1  ,wygaś flagę
End If

Loop
End

Sub Zakres:  ,podprogram sprawdzający zakres BPM
If Bpm >= 251 Then  ,jeżeli przekracza 250
  Bpm = 30  ,ustaw na 30
End If
  If Bpm <= 29 Then  ,jeżeli zaś jest mniejszy niż 30
    Bpm = 250  ,ustaw na 250
  End If
End Sub

Sub Cyfry  ,podprogram rozkładający liczbę BPM na cyfry
  Bpm_kopia = Bpm  ,skopiowanie wartości BPM
  Jednosci = Bpm_kopia Mod 10  ,wyłuskanie cyfry jedności
  Bpm_kopia = Bpm_kopia-Jednosci  ,odjęcie 1 - teraz ostatnie jest 0
  Bpm_kopia = Bpm_kopia / 10  ,podzielenie przez 10
  Dziesiatki = Bpm_kopia Mod 10  ,wyłuskanie cyfry dziesiątek
  Bpm_kopia = Bpm_kopia - Dziesiatki  ,jak wyżej
  Bpm_kopia = Bpm_kopia / 10
  Setki = Bpm_kopia  ,to, co zostało, jest cyfrą setek
End Sub

Sub Czas  ,podprogram obliczający wartość początkową licznika Timer1
  Wart_pocz = 15625 / Bpm  ,najpierw obliczenie ilości taktów na sekundę
  Wart_pocz = Wart_pocz * 60  ,a tu na minutę, poprzez mnożenie
End Sub

Dzwiek:  ,podprogram obsługujący przerwanie od Timer1
  Select Case Ton:  ,jeżeli zmienna Ton wynosi...
    Case 0:  ,...zero, to
      Sound Portd.0 , 5 , 1500  ,generuj dźwięk
    Case 1:  ,itd.
      Sound Portd.0 , 6 , 750
    Case 2:
      Sound Portd.0 , 7 , 375
    Case 3:
      Sound Portd.0 , 8 , 182
    Case 4:
      Sound Portd.0 , 10 , 61
  End Select
  Reset Portd.3  ,uruchom diodę LED
  Waitus 400  ,odczekaj krótki odcinek czasu
  Set Portd.3  ,wyłącz diodę LED
  Load Timer1 , Wart_pocz  ,ponownie załaduj do Timer1 wartość początkową
Return

Wyswietlaj:  ,wybranie wyświetlacza - za każdym razem następnego
  Load Timer0 , 100  ,załadowanie do Timer0 wartości początkowej
  Set Portd.6  ,wyłącz wszystkie wyświetlacze
  Set Portd.5
  Set Portd.4
  Select Case Nr_wysw  ,włączenie jednego wyświetlacza
  Case 0:  ,jeżeli Nr_wysw = 0, to
    Portb = Lookup(setki , Cyfra)  ,wystaw na PORTB cyfrę setek
    Reset Portd.6  ,uruchom cyfrę setek
  Case 1:  ,jeżeli Nr_wysw = 1, to
    Portb = Lookup(dziesiatki , Cyfra)  ,wystaw na PORTB cyfrę dziesiątek
    Reset Portd.5  ,uruchom cyfrę dziesiątek
  Case 2:  ,jeżeli Nr_wysw = 2, to
    Portb = Lookup(jednosci , Cyfra)  ,wystaw na PORTB cyfrę jedności
    Reset Portd.4  ,uruchom cyfrę jedności
  End Select
  Incr Nr_wysw  ,zwiększ wartość
  If Nr_wysw = 3 Then  ,jeżeli licznik wyświetlaczy się „przekręci”
    Nr_wysw = 0  ,należy go wyzerować
  End If
Return

Cyfra:    ,tablica przechowująca segmenty odpowiadające danej cyfrze
Data &B10000001 , &B11001111 , &B10010010 , &B10000110 , &B11001100 ,
&B10100100 , &B10100000 , &B10001111 , &B10000000 , &B10000100
```

Jest on czytelny nawet przy niskim napięciu zasilania, lecz lepszym rozwiązaniem może być zastosowanie wyświetlacza świecącego na czerwono, z racji mniejszej wartości napięcia przewodzenia zastosowanych w nim diod. W handlu dostępne są również wyświetlacze działające przy mniejszym, niż standardowe, prądzie świecenia każdego segmentu. Zastosowanie takiego przyczyni się do dalszego spadku poboru mocy, lecz wymagać będzie zwiększenia wartości rezystorów R1…R7. Dioda LED2 może być dowolna - warto jednak pamiętać, że zieleń jest kolorem, na który nasze oczy są wyczulone najbardziej. Diody białe lub niebieskie mogą nie funkcjonować przy zasilaniu ich napięciem 3 V.

Po włożeniu zaprogramowanego układu w podstawkę i włączeniu zasilania, układ rozpoczyna pracę bez jakichkolwiek czynności uruchomieniowych.

Michał Kurzela, EP