

# Elektroniczny metronom AVT-915

Każdy z nas ma jakąś pasję, której poświęca swój wolny czas. Niejednokrotnie bywa tak, że czas ten poświęcamy nie jednemu, ale kilku zainteresowaniom, niekoniecznie ściśle związanym ze sobą. Ja postanowiłem powiązać swoje zainteresowania tak, by za pomocą jednego rozwijać i udoskonalać drugie. W ten sposób powstał metronom, który jest efektem mojego zamiłowania do elektroniki i fascynacji muzyką.

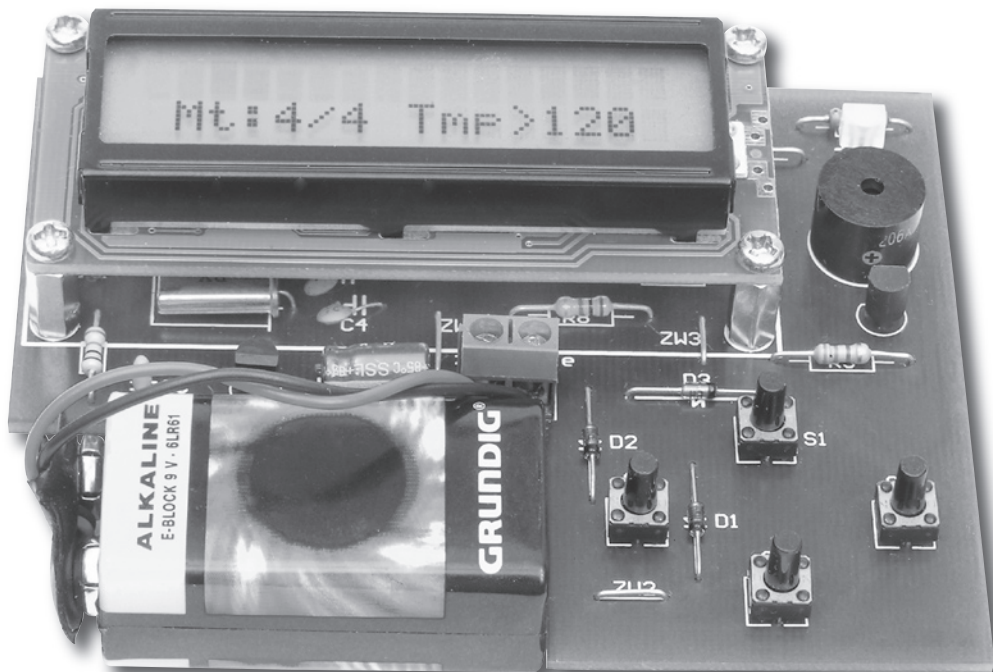
## Rekomendacje:

metronom to przyrząd, który z pewnością przyda się każdemu praktykującemu muzykowi. Jeśli ów muzyk oprócz umiejętności naciskania na klawisze potrafi również posługiwać się lutownicą, to satysfakcja z wykonania prezentowanego urządzenia będzie podwójna.



### PODSTAWOWE PARAMETRY

- Płytko o wymiarach: 113 x 81 mm
- Sygnalizacja ustawień na alfanumerycznym wyświetlaczu LCD
- Zakres wskazywania tempa: 40...250 uderzeń/min
- Zakres ustawiania metrum: 1/4...7/4
- Aakustyczna i świetlna sygnalizacja tempa
- Zasilanie: 9 V (np. z baterii 6F22R)



Rytm i tempo są to dwa nieodzowne elementy muzyki. Bez nich niemożliwe byłoby powstanie jakiegokolwiek piosenki czy melodii. Zatem umiejętność „równego” grania jest dla muzyka równie ważna jak artykulacja (sposoby wydobywania dźwięku), czy np. kształcenie słuchu i bywa, że jest doskonała latami. A ponieważ, jak mawiają sami muzycy „człowiek to nie automat”, więc jest to idealna okazja żeby połączyć muzykę z elektroniką.

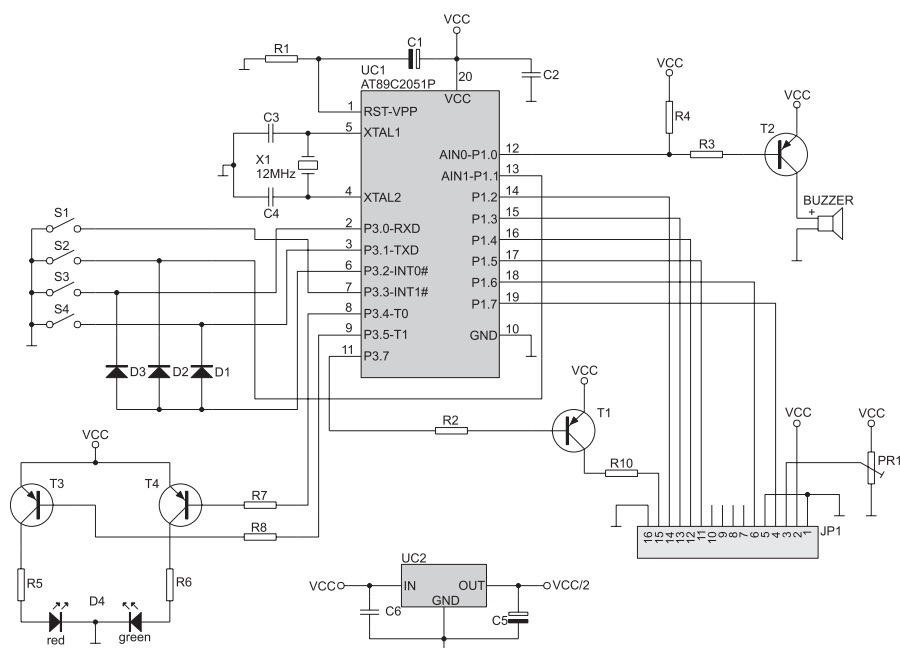
Sercem metronomu jest doskonale wszystkim znana kostka Atmela 89C2051. Jest to obecnie chyba najpopularniejszy mikrokontroler z rodziny procesorów '51. To jego niezaprzeczalna zaleta, a praktycznie objawia się to w dostępności dużej gamy programatorów oraz narzędzi programistycznych, min. środowiska MCS Bascom'51. Kostka ma 2 kB pamięci programu Flash, co w zupełności wystarczy do umieszczenia w niej prostego programu obsługującego podstawowe funkcje metronomu. Jest to jak sądzę dobrą wiadomością dla posiadaczy wersji Demo środowiska Basic, które umożliwia kompilowanie programów właśnie do 2 kB.

## Zasada działania

Tym co odróżnia proponowany przeze mnie metronom, od reszty układów tego typu dostępnych

na rynku, jest użycie wyświetlacza tekstowego LCD 2x16. Wyświetlane są na nim informacje na temat aktualnych ustawień, oraz punkt świetlny symulujący ruch wahadła klasycznego metronomu. Dostępne funkcje oczywiście zależą od oprogramowania układu. Udostępniony przeze mnie program umożliwia regulację tempa w zakresie od 40 do 250 BPM (czyli uderzeń na minutę), oraz wybór metrum z przedziału 1/4 do 7/4. „4” oznacza ćwierćnuty, użycie znaku „/” nie jest w tym przypadku poprawne z punktu widzenia muzycznego (ponieważ metrum to nie ułamek), jednak stwierdziłem że to jedyny rozsądny sposób na wyświetlenie tej informacji w jednej linii na wyświetlaczu tekstowym. Cyfry od 1 do 7 oznaczają, co którą ćwierćnutę w taktie akcentujemy. Skrajne położenie „wskazówki” metronomu jest dodatkowo sygnalizowane dźwiękiem buzzerka, oraz mrugnięciem diody. Gdy jest to nuta aktualnie akcentowana (początek taktu), dźwięk buzzerka jest dłuższy i dioda zapala się na czerwono. Pozostałe części taktu sygnalizowane są krótszym dźwiękiem i zapaleniem diody zielonej.

Metronom obsługujemy za pomocą czterech przycisków: UP, DOWN, Mt/Tmp, PAUSE.



Rys. 1. Schemat elektryczny mikroprocesorowego metronomu z wyświetlaczem

1. Mt/Tmp – przycisk ten umożliwia przełączanie pomiędzy wyborem opcji: Metrum (Mt), a Tempem (Tmp). Aktualnie wybrana opcja jest sygnalizowana znakiem „>” znajdującym się pomiędzy nazwą opcji, a jej wartością. Opcja nieaktywna w danym momencie oznaczona jest znaczkiem „:”. Jako opcja aktywna po włączeniu zasilania ustawiona jest funkcja Tmp. Przyciśnięcie tego przycisku, powoduje jednorazową zmianę aktualnej funkcji. Aby powtórnie zmienić funkcję należy przycisk puścić i przycisnąć ponownie. Po włączeniu zasilania wartość Tmp wynosi 120, a Mt 4/4.
- 2/3. Up/Down – przyciski te odpowiadają odpowiednio za zwiększenie bądź zmniejszenie wartości aktualnie wybranej funkcji, czyli zmianę tempa (przy aktywnej opcji Tmp), lub metrum (aktywna opcja Mt). Jednorazowe przyciśnięcie tych przycisków powoduje cykliczną zmianę tych parametrów, aż do puszczenia przycisku.
4. Pause – jak sama nazwa wskazuje przycisk ten powoduje chwilowe zatrzymanie działania metronomu. Na wyświetlaczu pojawia się napis „PAUSE”, po ok. 1 sekundzie wygaszone zostaje podświetlenie wyświetlacza LCD, a sam procesor wprowadzony zostaje w stan uśpienia. Ponowne naciśnięcie tego przy-

cisku powoduje „wybudzenie się” układu i powrót do stanu jak przed „zapauzowaniem”.

W stanie uśpienia (IDLE) procesor przechodzi w tryb zmniejszonego poboru mocy, co jest bardzo ważne z punktu widzenia układów zasilanych bateryjnie. Wprowadzenie funkcji PAUSE wynika głównie z faktu przeoczenia przez mnie kilku ważnych czynników na etapie projektowania metronomu. Otóż mikrokontroler 89C2051 mimo wielu swoich zalet nie ma wewnętrznej pamięci EPROM, w której mogłyby być przechowywane ustawienia układu po odłączeniu go od zasilania. Zatem każdorazowy zanik zasilania powoduje start układu, ze z góry zaprogramowanymi wartościami. „Zapauzowanie” układu nie powoduje utraty danych, i po przywróceniu układu do pracy wartości Tmp i Mt są identyczne jak przed jej wstrzymaniem. W stanie uśpienia (PAUSE) układ pobiera ok. 8 mA. Z punktu widzenia oszczędności energii korzystniejsze byłoby wprowadzenie układu w stan zamrożenia (POWER DOWN), jednak 2051 jest jedną z niewielu odmian '51 w których niemożliwe jest wybudzenie układu z zamrożenia za pomocą zewnętrznego przerwania.

### Układ elektryczny

Jak każdy układ mikroprocesorowy, tak i ten został zaopatrzony w sprzętowy reset. Dzięki temu początkowe warunki pracy mikro-

kontrolera po jego wyzerowaniu są zawsze jednakowe. W przypadku mikrokontrolerów rodziny '51 realizacja automatycznego zerowania jest niezwykle prosta, i sprowadza się do podłączenia zewnętrznego kondensatora (C1) i rezystora (R1).

Ze względu na oszczędność wprowadzeń wyświetlacz LCD został podłączony w trybie 6-bitowym, tj. 4 bity danych i 2 bity sterujące. Ponieważ Bascom słynie z wszelkich udogodnień, także i tym razem zwalnia nas z programowej obsługi wyświetlacza, która (jak wiedzą asemblerowcy) jest nieco bardziej skomplikowana dla sterowania 4-bitową magistralą danych niż 8-bitową. Nasza rola sprowadza się jedynie do określenia w programie sposobu podłączenia LCD. Ponieważ transmisja danych pomiędzy mikrokontrolerem a wyświetlaczem odbywa się tylko w jedną stronę, więc linia sterująca wyświetlacza R/W została na stałe podłączona do masy. Podświetlenie wyświetlacza sterowane jest z portu P3.7 poprzez tranzystor T1. Rezystor R10 został dobrany doświadczalnie, można go pominąć ponieważ podświetlenie wyświetlacza jest przystosowane do zasilania napięciem o wartości 5 V; jednak ze względu na dość duży pobór prądu (zasilanie bateryjne), warto zastosować ten rezystor w celu zredukowania poboru mocy. Rezystorem montażowym PR1 regulujemy kontrast wyświetlacza.

Porty P3.4 i P3.5 poprzez tranzystory T4 i T3 sterują pracą dwukolorowej diody LED. Obecność tranzystorów jest niezbędna ponieważ dioda ma wspólną katodę i nie da się jej bezpośrednio wysterować z portów mikrokontrolera. Zamiast niej można zastosować dwie osobne diody LED w odpowiednich kolorach. Wówczas pomijamy elementy T3, T4, R7, R8. Taka zmiana nie wymusza na nas nawet ingerencji w program, ponieważ i tak stanem aktywnym (zapalającym diody) jest zero.

Aby uprościć program i nie wprowadzać konieczności sprawdzania na każdym kroku stanu klawiszy sterujących, zostały one podpięte do linii obsługującej zewnętrzne przerwania. Jak wiadomo do dyspozycji mamy tylko dwa wyprowadzenia tego typu. Aby podłączyć do jednego portu

większą liczbę klawiszy musimy użyć 3 diod połączonych tak, jak na schemacie (D1, D2, D3). Każdy z przycisków (Up, Down, M/T) połączony został poprzez diodę do portu INT0 mikrokontrolera, oraz dodatkowo kolejno do portów P3.0, P1.1, P3.1. W ten sposób przyciśnięcie któregoś z nich wywołuje odpowiedni podprogram obsługi przerwania, w którym następnie sprawdzane są stany na ww. portach; stąd też wiemy, który przycisk został wciśnięty. Gwarantuje nam to błyskawiczną reakcję na naciśnięcie klawisza. Przycisk PAUSE został bezpośrednio połączony do końcówki INT1 co wiąże się z jego funkcją – wprowadzania mikrokontrolera w stan uśpienia i jego wybudzanie.

Bardzo ważną rzeczą jest podłączenie rezystorów podciągających (R4 i R9) do wyprowadzeń P1.1 i P1.0. Są to tzw. linie z alternatywnymi funkcjami wejścia/wyjścia. W tym przypadku mamy konkretnie do czynienia z wejściami komparatorów analogowych. Linie te mają nieco inną budowę niż np. linie portów P1 czy też P3, i nie zostały wyposażone w wewnętrzne rezystory podciągające. Jeżeli chcemy wykorzystywać te linie jako zwykłe wejścia lub wyjścia cyfrowe, konieczne jest użycie zewnętrznych rezystorów podciągających.

#### WYKAZ ELEMENTÓW:

##### Rezystory

R1, R2, R4, R7, R8, R9: 10 kΩ

R3: 3,3 kΩ

R5, R6: 270 Ω

R10: 100 Ω

PR1: 10 kΩ

##### Kondensatory

C1: 10 μF/10 V

C2, C6: 100 nF

C3, C4: 33 pF

C5: 47 μF/10 V

##### Półprzewodniki

Us1: 89c2051

Us2: 78L05

D1, D2, D3: 1N4148

D4: LED dwukolorowa (czerwono - zielona)

T1, T2, T3, T4: BC557

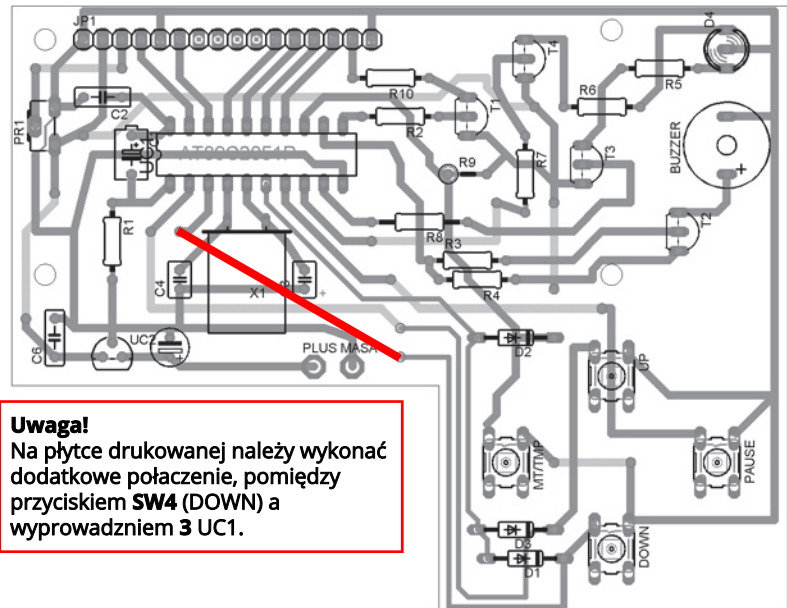
##### Inne

LCD: LCD 2x16 z podświetleniem

X1: rezonator kwarcowy 12 MHz

Bz: buzzer

Sw1, Sw2, Sw3, Sw4: mikro switche



**Uwaga!**  
Na płytce drukowanej należy wykonać dodatkowe połączenie, pomiędzy przyciskiem SW4 (DOWN) a wyprowadzeniem 3 UC1.

Rys. 2. Schemat montażowy układu

Do zasilania metronomu użyłem baterijki 9 V. Z powodu zbyt dużego napięcia baterijki, oraz z uwagi na to, że układy mikroprocesorowe są raczej wrażliwe na wahania napięcia, na wejściu układu znajduje się stabilizator scalony 78L05 utrzymujący napięcie na stałym poziomie, oraz odpowiednie kondensatory blokujące napięcie.

#### Program

W zasadzie jedynym zadaniem metronomu jest odmierzanie czasu. Toteż program napisany jest właśnie w taki sposób, aby sprawą nadrzędną było cykliczne włączanie pewnych funkcji. Do odmierzania czasu służą Timer0 i Timer1. Oba pracują w trybie drugim (mode 2) i chociaż w tym trybie ich „długość” wynosi maksymalnie tylko 8 bitów, co jak na nasze potrzeby w zupełności wystarcza. Dodatkową zaletą tego rozwiązania jest to, że w tym trybie liczniki przeładowują się automatycznie. Przeładowania obydwu Timerów, wywołują podprogramy obsługi przerwania, a te z kolei powodują inkrementację (zwiększenie o 1) rejestrów, odpowiednio Count0 i Count1. W ten sposób otrzymujemy czasy potrzebne do realizacji przypisanych funkcji. Zadaniem Timer0 jest mierzenie odcinków czasu między „uderzeniami” wahadełka. Timer1 natomiast, odmierza czas mrugnięcia diody oraz włączenia buzzerka. W przypadku obu timerów odblokowana została obsługa przerwania.

Czas przeładowania Timera0 został dobrany w taki sposób, aby regulacja tempa w zakresie od 40 do 250 uderzeń na minutę była jak najbardziej precyzyjna. Zatem dla 40 uderzeń na minutę wartość rejestru Count0 wynosi 255, a dla 250 równa jest 40. Dodatkowo czasy te zostały podwojone przez zastosowanie pętli FOR. Podfunkcja „czekaj” porównuje zawartość zmiennej Count0 ze zmienną „liczczas”. Zmienna ta jest obliczana za każdym razem kiedy zmienimy wartość tempa. A oto zależność z której korzystamy do obliczania wartości tej zmiennej:

$$\frac{60}{Tempo} = X[s]$$

X – czas pomiędzy uderzeniami wahadełka ([s] oznacza sekundy; ponieważ tempo podajemy w ćwierćnutach na minutę, więc aby otrzymać czas w sekundach korzystamy z powyższego wzoru).

$$\frac{X}{13} = Y[s]$$

Y – czas pomiędzy zmianą pozycji wahadełka taktowego (ponieważ wahadełko składa się z 3 kursorów, więc dla wyświetlacza 16 kolumnowego jeden pełny cykl będzie wynosił 13 pozycji wahadełka).

$$\frac{Y}{2 \cdot 226[\mu s]} = \frac{Y}{2 \cdot 226 \cdot 10^{-6}[s]} = licz-$$

„liczczas” – zmienna określająca liczbę przeładowań Timera0 pomię-

dzy przejściem wahadełka pomiędzy poszczególnymi pozycjami.

Po podstawieniu otrzymujemy:

$$\text{liczczas} = \frac{60}{\frac{2 \cdot 13 \cdot 226 \cdot 10^{-6} \cdot \text{Tempo}}{60 \cdot 10^6}} = \frac{10211}{26 \cdot 226 \cdot \text{Tempo}}$$

Wartość wpisywana do Timera1 jest dobrana doświadczalnie, w taki sposób aby przerwania przychodzące od obu timerów, oraz krotności tych przerwań nie następowały w tym samym czasie. Nachodzenie na siebie obu przerwań powodowało błędne działanie układu i metronom „zaczynał się”, mimo iż przerwanie od Timera0 ma wyższy priorytet niż przerwanie pochodzące od Timera1. Dioda oraz buzzerki sterowane są za pomocą Timera1. Jest on startowany w skrajnych położeniach wahadełka. Dla części taktu akcentowanych buzzerki zostaje włączony na ok. 35 ms, a dla części nieakcentowanych na ok. 12 ms. W ten sposób w miejscu akcentowanym uzyskujemy efekt gło-

śniejszego „stuknięcia”. Dioda zapalana jest na 35 ms, niezależnie od części taktu, natomiast akcent sygnalizowany jest czerwonym rozbłyskiem, podczas gdy pozostałe części taktu zielonym. Warto jeszcze dodać, że przed włączeniem buzzerki i diody, zostaje zablokowana obsługa przerw zewnętrznych. Wciśnięcie któregokolwiek z klawiszy funkcyjnych powoduje przerwanie wykonywania programu głównego i przejście do podprogramu obsługi przerwania aktualnie zgłoszonego. Wciśnięcie któregokolwiek z przycisków w momencie kiedy buzzerki jest aktywny, spowodowałoby jego działanie do momentu puszczenia przycisku. Jest to dosyć nieprzyjemny efekt przypominający odgłosy wydawane przez zawieszony komputer. Obsługa przerw zewnętrznych zostaje odblokowana zaraz po wygaszeniu diody i buzzerki. W pętli głównej programu został umieszczony listing realizujący rysowanie „wahadełka”. Podprogramy przerwań zewnętrznych realizują odpowiednio

funkcje przycisków podłączonych do wejść INTO i INT1.

### Montaż i uruchomienie

Poprawnie zmontowany i zaprogramowany układ działa od razu po podłączeniu zasilania. Nie ma konieczności jakichkolwiek regulacji. Środowisko Bascom mimo swoich licznych zalet ma jedną zasadniczą wadę – jego procedury są strasznie „pamięciożerne”. 2 kB pamięci pozwoliły jedynie na umieszczenie programu realizującego podstawowe funkcje metronomu. Jeżeli ktoś z Was czuje się na siłach, oraz posiada odpowiednią wersję Bascom, to na podstawie **list. 1** może rozbudować metronom o dodatkowe funkcje, jak np. granie beatów ósemkowych, triolowych, szesnastkowych. W tym celu kostkę 2051 należy wymienić na 4051, która ma identyczną architekturę wyprowadzeń, ale daje nam do dyspozycji pamięć o pojemności 4 kB.

**Marcin Sawicz**