

Obrotowy timer z alarmem akustycznym

kit

2780

AVT

Do czego to służy?

Opisany w artykule układ to timer z alarmem akustycznym, wzorowany na klasycznych minutnikach mechanicznych przeznaczonych do użytku w kuchni. Posiada jedno pokrętko, za pomocą którego nastawia się czas pozostały do zasygnalizowania alarmu (np. 3 minuty – po tym czasie jajko zostanie ugotowane na miękko). Można zaprogramować jeden z 23 czasów: od kilku sekund po kilka tysięcy godzin. Czas pozostały do uruchomienia alarmu pokazuje wyświetlacz składający się z 24 diod LED ułożonych wokół pokrętki. Dokładność czasu zadziałania alarmu jest bardzo dobra, gdyż częstotliwość sygnału zegarowego jest stabilizowana kwarcem.

Jak to działa?

Schemat ideowy układu widoczny jest na rysunku 1. Opiera się on na popularnym mikrokontrolerze AT89C2051 oraz dwóch rejestrach przesuwnych CMOS 4094. Pomysł tego układu powstał, gdy zastanawiałem się, co zrobić z impulsatorem – elementem stykowym wyposażonym w pokrętko i dwa wyjścia dające informacje, w którą stronę następuje jego obrót. Choć w literaturze elektronicz-

nej opisano już wiele różnego rodzaju timerów, nie spotkałem się jeszcze z podobnym układem, równie prostym w obsłudze. Intuicyjność obsługi urządzenia wynika w dużej mierze z użycia impulsatora zamiast przycisków, dzięki któremu obsługa wygląda podobnie jak w mechanicznych minutnikach (gdzie pokrętko, którym nastawia się czas, jest jednocześnie wskaźnikiem pozostałego czasu).

Impulsator ma dwa wyjścia (oznaczone jako A i B). Podczas pełnego obrotu osi każde z wyjść generuje ciąg impulsów (w modelu – 24 impulsy na obrót). Uzyskany przebieg prostokątny na jednym kanale różni się od przebiegu z drugiego kanału o pewne przesunięcie fazowe, i co istotne, różnica ta może być dodatnia lub ujemna. Analizując różnicę w fazie obu kanałów, otrzymujemy informację o kierunku obrotu osi impulsatora. W układzie zajmuje się tym oczywiście mikrokontroler.

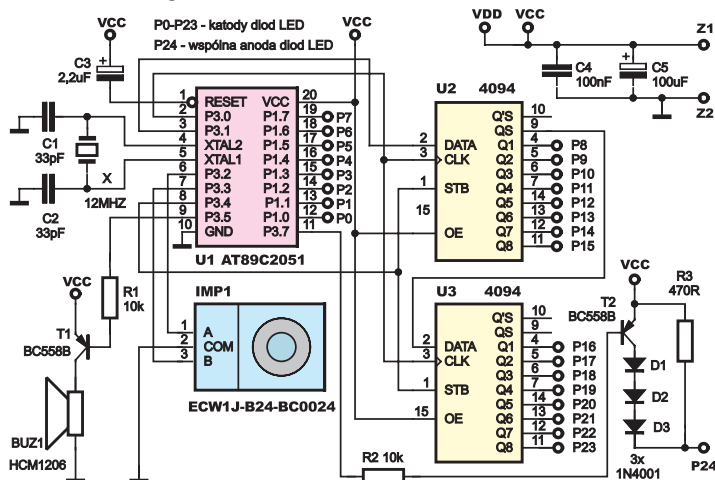
Oprócz tego mikrokontroler steruje diodami LED stanowiącymi wyświetlacz. I tu występuje wiadomy problem - procesor 89C2051 ma zbyt małą liczbę portów, aby możliwe było bezpośrednie sterowanie

wyświetlaczem wyposażonym w 24 diody LED (których nie widać na schemacie, widać natomiast miejsca podłączenia katod każdej diody: P0..P23 oraz wspólny punkt podłączenia anody każdej z nich: P24). Dlatego też wspomagają go dwa rejestry przesuwne 4094. Procesor steruje nimi za pośrednictwem 3 portów, które podają odpowiednie przebiegi na wejścia CLK (P3.0), DATA (P3.1), ale tylko do U2 – linia DATA U3 podłączona jest do wyjścia szeregowego U1 i STROBE (P3.4). Ostatecznie 8 diod w wyświetlaczu sterowanych jest bezpośrednio z portów mikrokontrolera, zaś 16 pozostałych z wyjść układów U2 i U3.

Wyświetlacz działa w dwóch trybach: pierwszy to tryb „linijkowy”, czyli wszystkie diody - począwszy od diody sygnalizującej czas 0s (dioda „zerowa”) do diody sygnalizującej pozostały czas do włączenia alarmu - są zaświecone, natomiast w drugim, „punktowym” trybie świecą tylko dwie diody - zerowa i wskazująca pozostały czas. Wyświetlacz pracuje jako linijkowy jedynie podczas ustalania czasu impulsatorem i po trzech sekundach przechodzi do trybu punktowego. Jest to spowodowane chęcią uzyskania minimalnego poboru prądu. Stąd układ z tranzystorem T2. Podczas pracy wyświetlacza w trybie linijkowym, T2 przewodzi i diody świecą jasno, natomiast podczas pracy w trybie punktowym prąd jest ograniczany przez rezystor R3 i diody świecą trochę słabiej. Diody prostownicze D1-D3 obniżają napięcia na diodach LED, co ogranicza prąd, ale nie powoduje efektu przygasania diod w miarę wzrostu liczby zaświeconych diod. Urządzenie ma własny buzzer przełączany z częstotliwością 10Hz podczas wystąpienia alarmu. Jego pracą steruje tranzystor T1 zasilany z portu P3.5 mikrokontrolera.

Program na mikrokontroler napisany został w BASCOM-ie dla procesorów rodziny '51. Można go ściągnąć z Elportalu (www.elportal.pl). Procesor pracuje z kwarcem 12MHz, a więc jeden cykl maszynowy

Rys. 1 Schemat ideowy



zajmuje dokładnie jedną mikrosekundę. Jest to wygodne podczas programowania poszczególnych czasów nastaw.

Co 10ms wykonywana jest procedura obsługi przerwania *Refresh*, której kod widoczny jest na **listingu 1**. Na początku tej procedury do rejestrów *timer* 0 *T10* i *Th0* wpisywana jest liczba równa 65535-10000 po to, żeby za dokładnie 10ms procedura została znów wykonana. Głównym licznikiem czasu w programie jest zmienna globalna *Time_count*. Korzysta z niej zarówno procedura sterująca wyświetlaczem, jak też procedura obsługująca użytkownika. Ponieważ zmienna *Time_count* jest typu *long*, może ona osiągnąć wartość 4294967295, czyli największy czas, jaki można zaprogramować, wynosi 42949672,95 sekund. Za każdym razem procedura sprawdza, czy licznik nie jest czasem równy którejś z 24 wartości umieszczonych pod etykietą *Time_tab* w programie. Są to dane umieszczone w pamięci programu i są równe poszczególnym czasom alarmu - patrz **listing 2** dostępny w Elportalu w materiałach dodatkowych do tego numeru EdW. Wartości te oczywiście można zmieniać i dostosować do swoich potrzeb, trzeba jednak uważać, żeby każda następna była większa od poprzedniej (inaczej program zadziała błędnie).

Jeśli okaże się, że wartość licznika *Time_count* jest równa którejś z tych wartości, wtedy uruchomiona zostaje procedura sterująca wyświetlaczem, czyli nastąpi odświeżenie wyświetlacza. Procedura sterująca wyświetlaczem widoczna jest na **listingu 3** (umieszczony w Elportalu, w materiałach dodatkowych do tego numeru). Jako argumenty tej procedury podawane są: tryb, w jakim pracuje wyświetlacz, oraz liczba z przedziału 0..23 odpowiadająca numerowi diody, która będzie wskazywała czas pozostały do alarmu. Warty zwrócenia uwagi jest fragment (pętla *for..next*) sterujący rejestrami przesuwymi.

Ujemne impulsy pochodzące od impulsatora powodują przerwanie *INT0* lub *INT1* (w zależności od kierunku obrotu) i w konsekwencji wywołanie procedury *Przekret*, której kod widoczny jest na **listingu 4** (w Elportalu).

Na początku tej procedury zostaje zdekodowany sygnał z obu kanałów impulsatora i w zależności od kierunku obrotu do zmiennej *Time_count* zostaje przypisana odpowiednia wartość (pierwsza z lewej lub prawej) zaczerpnięta z linii *DATA*.

Zmiany są pokazywane na wyświetla-

czu, ponieważ na końcu procedury zostaje uruchomiona procedura *Pokaz*.

Montaż i uruchomienie

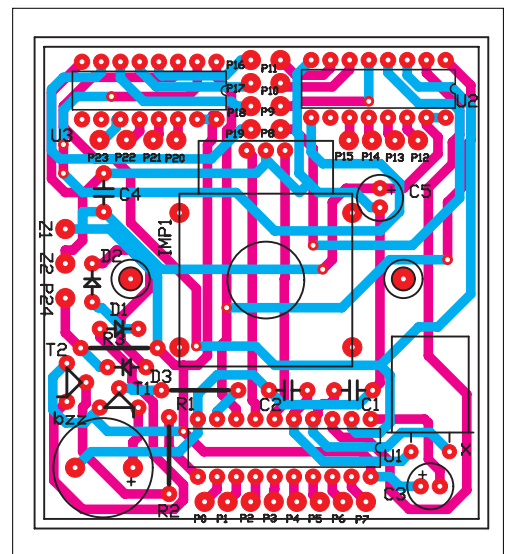
Układ modelowy został zmontowany na płycie uniwersalnej. Po jego uruchomieniu została zaprojektowana płytka drukowana pokazana na **rysunku 2** i na niej można zmontować układ. Całość montujemy w obudowie plastikowej, np. Z54S. Jak widać, na płycie nie ma miejsca na diody LED, dlatego też trzeba je umieścić w pokrywie obudowy, wokół otworu, przez który wychodzi oś impulsatora. Anody diod lutujemy ze sobą, natomiast katody łączymy za pomocą krótkich przewodów do punktów lutowniczych na płycie (P0..P24). Płytka drukowana ma dwa otwory mocujące. Mocujemy ją do obudowy najlepiej za pomocą śrubek M3 przykręcanych

Listing 1.

```
Refresh:           'procedura obsługująca przerwanie od T0
T10 = &B11101111 'co 10ms
Th0 = &B11011000 'Th0.T10=65535-10000
If M < 300 Then
  Incr M
  Tryb = 1
Else
  Tryb = 0
End If
Incr K
K = K Mod 5      'licznik modulo 5
If Alarm = 1 Then 'procedura generująca sygnał alarmu
  If K = 0 Then 'co 50ms następuje zmiana stanu buzzerka
    cpl P3.5
  End If
Else
  Set P3.5      'jesli alarm nie jest uruchomiony to wylacz buzzerk
End If

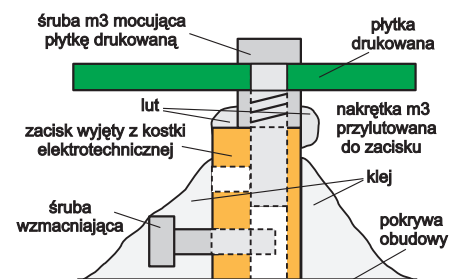
Restore Time_tab 'ustaw wskaznik na pierwszym elemencie tablicy
For V1 = 0 To 23
  Read T11
  If Time_count = T11 Then 'jesli licznik jest rowny elementowi tablicy
    'znajdz jego numer
    Call Pokaz(tryb , V1) 'pokaz na wyswietlaczu jego numer
  End If
Next V1

If Time_count > 0 Then 'tutaj następuje dekrementacja licznika za kazdym
  'wystapieniem przerwania T0 ale tylko wtedy jesli
  'ten licznik jest wiekszy od 0
  Decr Time_count
  'If Time_count = 0 Then
  Set Alarm 'ustaw znacznik alarmu
Else
  Reset Alarm 'zeruj znacznik alarmu
End If
End If
Return
```



Rys. 2 Schemat montażowy

Rys. 3



w pewien sposób do pokrywy obudowy (np. do wsporników z nakrętką przyklejonych od wewnętrznej strony pokrywy). Na **rysunku 3**

przedstawiony został przykładowy sposób zamocowania płytki drukowanej. W pokrywie wiercimy również otwór w miejscu,

gdzie umieszczony jest buzzer.

Układ zasilamy napięciem 5V, ale można użyć trzech baterii AA połączonych szeregowo, co daje w sumie 4,5 V. W obudowie oprócz koszyeczka na baterie dobrze jest umieścić jeszcze gniazdo zasilania. Ponieważ prąd pobierany przez układ wynosi kilka mA, warto wyposażyć go również w wyłącznik zasilania, szczególnie gdy planujemy używać baterii. W celu ułatwienia nastawiania timera można opisać każdą diodę np. cienkim flamastrem wodoodpornym albo wydrukować nalepkę z opisem.

Wykaz elementów

R1,R2.....	10k Ω	T1,T2.....	BC558B
R3.....	470 Ω	U1.....	AT89C2051
C1,C2.....	33pF	U2,U3.....	4094
C3.....	2,2 μ F	IMP1.....	ECW1J-B24-BC0024
C4.....	100nF	BUZ1.....	HCM1206
C5.....	100 μ F	X12MHz	
D1-D3.....	1N4001	LED.....	24 diody LED
		Obudowa Z54S	

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2780

Piotr Wójtowicz

piotr.wojtowicz@edw.com.pl