

# Biblioteki mikroprocesorowych procedur standardowych



## Procedura przerwania czasowego z kalendarzem

*Przedstawiamy kolejną procedurę dla procesorów rodziny MCS-51. Tym razem jest to zegar połączony z kalendarzem - temat bardzo popularny wśród konstruktorów urządzeń mikroprocesorowych.*

Poniżej przedstawimy krótką procedurę, która obsługuje przerwania od timera, umożliwiającą utworzenie zegara z kalendarzem do roku 10000. Założeniem projektu tak „żywoznego“ zegara nie była silna wiara w możliwość przetrwania architektury 8051 przez tak wiele lat, lecz chęć realizacji kalendarza gregoriańskiego.

W kalendarzu gregoriańskim, w odróżnieniu od kalendarza juliańskiego, lata przestępne występują co 4 lata z pewnymi wyjątkami. Generalna zasada jest taka: rok jest przestępny, jeśli liczba określająca go jest podzielna przez 4. Liczba roku podzielna przez 100, ale nie podzielna przez 400 oznacza rok zwykły, to jest ten wyjątek. Wynika z tego, że lata 1700, 1800, i 1900 były latami zwykłymi, zaś zbliżający się rok 2000 będzie rokiem przestępnym.

### Realizacja projektu

Procesor 8051 posiada dwa układy czasowe, oznaczone jako T0 i T1. Układy czasowe działają niezależnie od pracy jednostki centralnej i komunikują się z nią wystawiając sygnały przerwania. Każdy z timerów posiada parę rejestrów TH i TL. Są to rejestry licznikowe, zliczające zdarzenia albo cykle zegara. W zależności od wzajemnego połączenia rejestrów TH i TL, timery T0 i T1 pracują w jednym z czterech try-

bów pracy, oznaczanych liczbami 0..3. W tabeli 1 przedstawiono krótką charakterystykę tych trybów.

Oprócz tych rejestrów, istnieje szereg bitów sterujących zliczaniem. Każdy z timerów posiada osobny zestaw takich bitów. Oto one:

- GATE - bit bramkowania, jego ustawienie oznacza, że zliczane impulsy są bramkowane sygnałem wysokim na linii INTx ( $x=0,1$ ); wyzerowanie bitu oznacza, że bramkowania nie ma;
- C/T# - (counter/timer) bit wyboru źródła zliczanych impulsów, stan niski oznacza, że impulsy pochodzą z zegara procesora i są zliczane cykle procesora (timer), stan wysoki oznacza, że impulsy pochodzą z pinu Tx (licznik zdarzeń), przy czym zliczenie następuje po wykryciu zmiany stanu na pinie Tx z wysokiego na niski w dwóch sąsiednich cyklach procesora, czyli maksymalna częstotliwość zdarzeń jest 24-krotnie mniejsza od częstotliwości zegara procesora;
- TRx - (timer run) bit globalnego zezwolenia na zliczanie, stan wysoki włącza zliczanie, stan niski - wyłącza;
- TFX - (timer full) znacznik żądania przerwania, ustawiany po przepełnieniu rejestrów licznikowych, zerowany automatycznie po wejściu do procedury obsługi danego przerwania.

Lokacja powyższych bitów kontrolnych jest następująca:

- znaczniki GATE i C/T# razem z dwoma bitami M0 i M1, kodującymi tryb pracy, zajmują jedną połowę rejestru specjalnego TMOD (adres 89H). Młodsza połówka rejestru TMOD do-

**Tabela 1.**

Tryb	Charakterystyka
0	8-bitowy timer/licznik, THx jest 8-bitowym licznikiem, TLx jest 5-bitowym preskalerem
1	16-bitowy timer/licznik
2	8-bitowy licznik z automatycznym załadowaniem; THx przechowuje wartość, która jest zapisywana do TLx po jego przepełnieniu
3	timer T0: następuje rozdzielenie źródeł sygnału dla TH0 i TLO; TLO jest zależny od kontrolnych bitów T0, a TH0 jest zależny od kontrolnych bitów T1 timer T1: jest zatrzymany

tyczy T0, a starsza T1, co pokazano poniżej:

```
GATE C/T# M1 MO   GATE C/T# M1 MO
Timer T1          Timer T0
```

- znaczniki TR0, TFO (timera T0) i TR1, TF1 zajmują starszą półówkę rejestru specjalnego TCON (adres 88H), co widać niżej:

```
8FH 8EH 8DH 8CH 8BH 8AH 89H 88H
TFI TRI TFO TRO
```

Jest to rejestr adresowany bitowo, o czym świadczy młodsza cyfra jego adresu: jest to 0 albo 8.

Do naszych celów wykorzystamy tryb 2 timera T0. Jest to tryb 8-bitowego licznika z przeładowaniem. Wydawać by się mogło, że tryb ten jest mało efektywny. Rzeczywiście 8 bitów zapewnia nam zliczenie tylko 256 cykli, a więc stosunkowo często będzie potrzebna obsługa przerwania od T0. Z tego punktu widzenia o wiele lepszym rozwiązaniem jest praca w trybie 0, a jeszcze lepszym w trybie 1. Załadowanie rejestrów TH0 i TLO wartością początkową jest możliwe w procedurze obsługi, a jeśli odliczamy czas 1 sekundy, to do zliczenia mamy 1000000 cykli procesora przy zegarze 12MHz.

Dla eksperymentu ustalmy zatem tryb 1 dla timera T0 i niech przerwaniem będzie zgłaszane co 50000 cykli. Dostajemy przerwanie 20 razy na sekundę. W procedurę obsługi wkomponujemy rozkazy do załadowania pary rejestrów TH0, TLO liczbą 65536-50000=15536. Trzeba bowiem wiedzieć, że liczniki w tym procesorze zliczają w przód. Okazuje się jednak, że tak zbudowany zegar ma dokładność zegarka mechanicznego. Czemu tak się dzieje? W momencie przepełnienia licznika następuje ustawienie flagi TFO, co jest dla systemu przerwania mikroprocesora informacją o żądaniu obsługi przerwania. Jednak żądanie to nie musi być obsługane natychmiast.

Istnieje kilka powodów takiego stanu rzeczy. Po pierwsze, w momencie zgłoszenia obsługi przerwania może być obsługiwane przerwaniem o wyższym priorytecie. Po drugie, w momencie zgłoszenia obsługi przerwania jest wykonywana instrukcja wielobajtowa, a więc nie można jej zatrzymać w trakcie realizacji. Po trzecie, zostało jednocześnie zgłoszo-

ne do obsługi przerwania o tym samym priorytecie, ale o wyższej kolejności obsługi (takim przerwaniem dla T0 jest INT0). Po czwarte, jest wykonywana instrukcja RETI albo instrukcja zapisu danych do rejestru IE lub IP.

Każdy z wymienionych warunków wyklucza obsługę przerwania w danym cyklu procesora. Proces arbitrażu obsługi przerwania jest procesem bez pamięci, czyli w następnym cyklu maszynowym rozpoczyna się od początku i też nie musi skończyć się skokiem do procedury obsługi przerwania. Z dużą pewnością możemy więc powiedzieć, że zegarek nasz będzie pracował z dokładnością o rząd wielkości gorszą niż zapewnia rezonator kwarcowy.

Wady tej nie ma tryb 2 timera. Zapisanie zawartości TH0 do TLO zachodzi sprzętowo, niezależnie od momentu obsługi przerwania. Ważne jest jedynie to, aby obsługa przerwania została zakończona przed następnym ustawieniem flagi TFO.

Na list.1 przedstawiono procedurę obsługi przerwania od T0 wraz z wbudowanym kalendarzem. Zakładamy, że procesor jest taktowany zegarem 12MHz. Zapis liczby 6 do TH0 oznacza, że żądanie obsługi przerwania od T0 będzie pojawiać się co 250µs. Ustalamy więc zmienną (SEK\_250US), w której będziemy zliczać impulsy w odstępach 250µs. Zliczenie 40 takich odcinków daje 10ms. Powinna teraz pojawić się zmienna zliczająca 10-milisekundowe odcinki czasu - SEK\_10MS. Sto takich odcinków czasu daje już jedną sekundę. Następnie zliczane są minuty, potem godziny, dni, dni tygodnia, miesiące i lata. W przypadku lutego procedura musi rozważyć, czy dany rok jest przestępny, czy może jest rokiem zwykłym.

Ów podprogram można oczywiście modyfikować, dodając np. fragmenty kodu związane z wyświetlaniem sekwencyjnym na wskaźniku siedmiosegmentowym LED, obsługą klawiatury i inne, silnie związane z czasem procesy.

Wszystkie zmienne kodujące jednostki czasu (od sekundy do stulecia) są wyrażone w kodzie naturalnym binarnym.

**Mirosław Lach, AVT**

### Listing 1.

```
;FRAGMENT OPROGRAMOWANIA WIELU ZEGAROW TEGO
; SWIATA WYPOSAZONYCH W MIKROPROCESOR 80C51
; LUB JEMU PODOBNY ZEGAR 12MHZ, CO DAJE CYKL
; ROZKAZOWY LUS
SEK_250US EQU 30H
SEK_10MS EQU SEK_250US+1
SEKUNDY EQU SEK_10MS+1
MINUTY EQU SEKUNDY+1
GODZ EQU MINUTY+1
DNI EQU GODZ+1
DNI_TYG EQU DNI+1
MIESIACE EQU DNI_TYG+1
LATA EQU MIESIACE+1
STULECIA EQU LATA+1
EA EQU 0AFH
ETO EQU 0A9H
TRO EQU 8CH
ORG 0
LJMP RESTART
ORG 0BH
LJMP TO_SERVIS
RESTART:
MOV TMOD,#0000010B;TRYB 2|T0
TAKTOWANY ZEGAREM
SETB ETO
SETB EA
MOV TH0,#6
SETB TRO
MAIN:
SJMP MAIN
T0_S1A:
AJMP T0_S1
T0_SERVIS:
PUSH ACC
PUSH PSW
DJNZ SEK_250US,T0_S1A
MOV SEK_250US,#40
DJNZ SEK_10MS,T0_S1A
MOV SEK_10MS,#100
INC SEKUNDY
MOV A,SEKUNDY
CJNE A,#60,T0_S1
MOV SEKUNDY,#0
INC MINUTY
MOV A,MINUTY
CJNE A,#60,T0_S1
MOV MINUTY,#0
INC GODZ
MOV A,GODZ
CJNE A,#24,T0_S1
MOV GODZ,#0
INC DNI
INC DNI_TYG
MOV A,DNI_TYG
CJNE A,#8,T0_S2
MOV DNI_TYG,#1
T0_S2:
MOV A,DNI
CJNE A,#29,T0_S3
; MINELO 28 DNI MIESIACA
MOV A,MIESIACE
CJNE A,#2,T0_S1
LCALL PRZESTEPNY
JNC T0_S1
SJMP T0_S7
T0_S3:
CJNE A,#30,T0_S5
; MINELO 29 DNI MIESIACA
MOV A,MIESIACE
CJNE A,#2,T0_S1
SJMP T0_S7
T0_S5:
CJNE A,#31,T0_S4
; MINELO 30 DNI MIESIACA
MOV A,MIESIACE
CJNE A,#4,T0_S8
SJMP T0_S7
T0_S8:
CJNE A,#6,T0_S9
SJMP T0_S7
T0_S9:
CJNE A,#9,T0_S10
SJMP T0_S7
T0_S10:
CJNE A,#11,T0_S1
SJMP T0_S7
T0_S4:
CJNE A,#32,T0_S1
; MINELO 31 DNI MIESIACA
T0_S7:
MOV DNI,#1
INC MIESIACE
MOV A,MIESIACE
CJNE A,#13,T0_S1
MOV MIESIACE,#1
INC LATA
MOV A,LATA
CJNE A,#100,T0_S1
MOV LATA,#0
INC STULECIA
T0_S1:
POP PSW
POP ACC
RETI
;
; PODPROGRAM USTALANIA PRZESTEPNOSCI ROKU
; WEJSCIE: NIC
; WYJSCIE: CV=0 ROK PRZESTEPNY
; CY=1 ROK ZWYKLY
PRZESTEPNY:
MOV A,LATA
CLR C
RRC A
JNC PRZEST1
RET
PRZEST1:
RRC A
JNC PRZEST2
RET
PRZEST2:
JZ PRZEST3
RET
PRZEST3:
MOV A,STULECIA
RRC A
JNC PRZEST4
RET
PRZEST4:
RRC A
RET
```