

Laboratorium Mikroprocesory

TIMERY MIKROKONTROLERA 8051



Timery – układy zliczające

2

- ▶ Mikrokontroler 8051 posiada dwa 16-bitowe układy zliczające (timery)
- ▶ Maksymalna wartość zliczania to $2^{16} = 65536$ impulsów
- ▶ Przejście ze stanu maksymalnej wartości do stanu zerowego sygnalizowane jest odpowiednią flagą.
- ▶ Działanie
 - ▶ Praca jako timer- zliczanie impulsów z rezonatora kwarcowego
 - ▶ Praca jako licznik – zliczanie impulsów na linii wejściowej mikrokontrolera



Timery – podłączenie

3

- ▶ Timer może liczyć impulsy zewnętrzne podłączone do linii procesora.
 - ▶ Timer 0 połączony jest z linią P3.4 zwaną też T0
 - ▶ Timer 1 połączony jest z linią P3.5 zwaną też T1
- ▶ Wszystkie rejestry związane z timerami znajdują się w obszarze rejestrów specjalnych. Należą do nich:
 - ▶ **TLO** - młodsza część Timera 0 (8 bitów)
 - ▶ **TH0** - starsza część Timera 0 (8 bitów)
 - ▶ **TL1** - młodsza część Timera 1 (8 bitów)
 - ▶ **TH1** - starsza część Timera 1 (8 bitów)
 - ▶ **TMOD** - ustawia tryb pracy timerów 0 i 1
 - ▶ **TCON** - 4 bity z tego rejestru sterują pracą timerów.



Timery – rejestr TMOD

4

- ▶ Ustawianie trybu pracy timerów 0 i 1
 - ▶ Cztery młodsze bity - Timer 0
 - ▶ Cztery starsze bity - Timer 1

TMOD							
TIMER 1				TIMER 0			
GATE	C/T	M1	M0	GATE	C/T	M1	M0

- ▶ Bit C/I decyduje o pracy jako timer lub licznik.
 - ▶ Stan 0 - zliczanie impulsów wewnętrznego zegara
 - ▶ Stan 1 – zliczanie impulsów zewnętrznych z wejścia Tx.
- ▶ Bit GATE decyduje o sposobie kontrolowania pracy timera.
 - ▶ Stan 0 - start/stop timera odbywa się przez ustawienie bitu TRx.
 - ▶ Stan 1 - dodatkowo o pracy timera decyduje stan linii INTx.
- ▶ Bity M1 i M0 ustawiają odpowiedni tryb pracy timera - 0...3.



Timery – rejestr TCON

5

- ▶ Zawiera po 2 bity sterujące dla każdego timera:
 - ▶ **TRO** (1 - start timera 0 , 0 - zatrzymanie timera 0).
 - ▶ **TR1** (1 - start timera 1 , 0 - zatrzymanie timera 1).
 - ▶ **TF0** (Flaga przepełnienia timera 0).
 - ▶ **TF1** (Flaga przepełnienia timera 1).
- ▶ Flaga przepełnienia - ustawiona automatycznie na 1 w momencie przekroczenia zakresu timera.



Timery – tryby pracy

6

- ▶ Tryb 0
 - ▶ praca jako rejestr 13 bitowy.
 - ▶ Maksymalna liczba zliczanych impulsów to 8192
 - ▶ Młodsza część timera jest traktowana jako rejestr 5 bitowy.
 - ▶ Przepełnienie młodszej części powoduje zwiększenie zawartości w starszej części timera
- ▶ Tryb 1
 - ▶ Praca jako rejestr 16 bitowy
 - ▶ Maksymalna liczba zliczanych impulsów to 65536



Timery – tryby pracy

7

▶ Tryb 2

- ▶ praca jako rejestr 8 bitowy.
- ▶ Maksymalna liczba zliczanych impulsów to 256
- ▶ Przepiętnienie młodszej części powoduje przeniesienie zawartości starszej części timera do młodszej
 - ▶ Pozwala na automatyczne ustawianie flagi przepiętnienia ze stałą częstotliwością

▶ Tryb 3

- ▶ Timer 1 jest zatrzymany
- ▶ Timer 0 działa jako 2 niezależne liczniki.
 - ▶ TL0 sterowane przez bity Timera 0
 - ▶ TH0 sterowane przez bity Timera 1



Timery – wyznaczanie CZASU

- ▶ Impuls zliczany przez timer pojawia się w odstępach:

$$T_i = \frac{12}{11,0592 [MHz]} \approx 1,085 \mu s$$

- ▶ Wartość 11,0592 MHz odpowiada częstotliwości rezonatora kwarcowego podłączonego do mikrokontrolera
- ▶ Liczbę impulsów (I) potrzebnych do odmierzenia zadanego czasu (T) można wyznaczyć w następujący sposób:

$$I = \frac{T}{T_i}$$

- ▶ Maksymalny czas odmierzony przez timer to około 71 ms



Timery – wyznaczanie CZASU

► Przykłady:

► $T = 50 \text{ ms}$

$$I = 50000 [\mu s] \cdot \frac{11,0592 [MHz]}{12} = 46080 = 180 \cdot 256$$

► $T = 10 \text{ ms}$

$$I = 10000 [\mu s] \cdot \frac{11,0592 [MHz]}{12} = 9216 = 36 \cdot 256$$

► $T = 20 \text{ ms}$

$$I = 20000 [\mu s] \cdot \frac{11,0592 [MHz]}{12} = 18432 = 72 \cdot 256$$

Timery – wyznaczanie CZASU

10

- ▶ Ustawienie Timera w trybie 1

- ▶ $T = 50\text{ ms}, I = 180 \cdot 256$

- ▶ $THx = 256 - 180$

- ▶ $TLx = 0$

- ▶ $T = 10\text{ ms}, I = 36 \cdot 256$

- ▶ $THx = 256 - 36$

- ▶ $TLx = 0$



Timery - przykład

11

```
LED      EQU      P1.7
;***** Ustawienie TIMERów *****
;TIMER 0
T0_G     EQU      0           ;GATE
T0_C     EQU      0           ;COUNTER/-TIMER
T0_M     EQU      1           ;MODE (0..3)
TIM0     EQU      T0_M+T0_C*4+T0_G*8
;TIMER 1
T1_G     EQU      0           ;GATE
T1_C     EQU      0           ;COUNTER/-TIMER
T1_M     EQU      0           ;MODE (0..3)
TIM1     EQU      T1_M+T1_C*4+T1_G*8

TMOD_SET EQU      TIM0+TIM1*16

;50[ms] = 50 000[ŠS]*(11.0592[MHz]/12) =
;      = 46 080 cykli = 180 * 256
TH0_SET  EQU      256-180
TL0_SET  EQU      0
;*****
;      LJMP      START
;      ORG       100H

START:
MOV      TMOD,#TMOD_SET           ;Timer 0 liczy czas
MOV      TH0,#TH0_SET             ;Timer 0 na 50ms
MOV      TL0,#TL0_SET
SETB     TR0                      ;start Timera

LOOP:
;Pętle mrugania diody TEST
CPL      LED
MOV      R7,#20                   ;odczekaj czas 20*50ms=1s

TIME_N50:
JNB      TF0,$                   ;czekaj, aż Timer 0
;odliczy 50ms
MOV      TH0,#TH0_SET             ;TH0 na 50ms
CLR      TF0                      ;zerowanie flagi timera 0
DJNZ     R7,TIME_N50              ;odczekanie N*50ms

SJMP     LOOP
```



- ▶ Napisz program, który będzie realizował funkcję zegara z alarmem
 - ▶ Format wyświetlania hh:mm:ss
 - ▶ Ustawianie godziny przy uruchomieniu programu
 - ▶ Wprowadzamy:
 - ▶ Godzinę (0-23)
 - ▶ Minuty (0-59)
 - ▶ Sekundy (0-59)
 - ▶ Ustawianie alarmu przy uruchomieniu programu
 - ▶ Wprowadzamy:
 - ▶ Godzinę (0-23)
 - ▶ Minuty (0-59)



Liczba dwucyfrowa

13

WPROWADZ:

LCALL WAIT_KEY ; Wczytaj liczbę dziesiątek

MOV B,#10 ; pomnóż

MUL AB ; przez 10

MOV R1,A ; zapisz liczbę w R1

LCALL WAIT_KEY ;wczytaj liczbę jedności

ADD A,R1 ; dodaj liczbę jedności do R1

RET ; wyjdź z podprogramu. Wynik w A.



- ▶ Ocena 4 – Realizacja zadania bez alarmu
- ▶ Ocena 4.5 – realizacja zadania z alarmem o nieokreślonym czasie trwania
- ▶ Ocena 5 - realizacja zadania z alarmem o określonym czasie trwania np. 2 s