

**Laborator – Aplicații utilizând limbajul C, pentru microcontrolere**

În acest Laborator s-a pus accent pe cunoașterea și folosirea Timerelor T0, T1 și lucrul în Întrerupere. În cele ce urmează se dau cei mai importanți Regiștri utilizați în acest Laborator.

1. Registrul de control mod Timer / Numărător (TMOD)

MSB				LSB			
GATE	C/ $\bar{T}$	M1	M0	GATE	C/ $\bar{T}$	M1	M0
← TIMER 1 →				← TIMER 0 →			

**GATE** Control poartă când este setat. Timerul / numărător "x" este activat numai când pinul  $\overline{INTx}$  este HIGH și bitul de control "TRx" este setat. Când este șters, timerul "x" este activat dacă se setează bitul de control "TRx".

**C/ $\bar{T}$**  Selector timer sau numărător șters pentru funcționare ca timer (intrare de la sistemul de clock intern). Setat pentru funcționare ca numărător (intrare pe pinul "Tx").

M1	M0	FUNȚIONARE
0	0	Timerul 8048 "TLx" servește pentru prescalare pe 5 biți
0	1	Timerul / numărător de 16 biți "THx" și "TLx" sunt în cascadă; nu există prescalare
1	0	Timerul / numărător "THx" de 8 biți cu autoreîncărcare păstrează o valoare ce trebuie reîncărcată în "TLx" ori de câte ori el depășește
1	1	(Timer 0) TL0 este un timer / numărător de 8 biți controlat de către biții de control ai timerului 0 standard. TH0 este un timer de 8 biți controlat numai de biții de control ai timerului 1.
1	1	(Timer 1) timerul / numărător 1 este oprit

2. Registrul de control timer / numărător (TCON)

MSB				LSB			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT	SIMBOL	FUNȚIE
TCON.7	TF1	Flag depășire Timer 1. Setat hardware pe depășire Timer / Numărător. <b>Șters hardware</b> când procesorul vectorizează spre rutina de întrerupere sau prin ștergerea bitului prin software
TCON.6	TR1	Bit de control funcționare timer 1. <b>Setat / Șters prin software</b> pentru a comuta timerul / numărător on / off.
TCON.5	TF0	Flag depășire timer 0. Setat hardware pe depășire timer / numărător. <b>Șters hardware</b> când procesorul vectorizează spre rutina de întrerupere sau prin ștergerea bitului prin software
TCON.4	TR0	Bit de control funcționare timer 0. <b>Setat / Șters prin software</b> pentru a comuta timerul / numărător on / off.
TCON.3	IE1	Flag front întrerupere tip 1. <b>Setat hardware</b> când se detectează frontul unei întreruperi externe. Șters când s-a procesat întreruperea.
TCON.2	IT1	Bit control întrerupere tip 1. <b>Setat / șters software</b> pentru a preciza front căzător / nivel LOW pentru delanșarea întreruperilor externe.
TCON.1	IE0	Flag front întrerupere tip 0. <b>Setat hardware</b> când se detectează frontul unei întreruperi externe. Șters când s-a procesat întreruperea.
TCON.0	IT0	Bit control întrerupere tip 0. <b>Setat / șters software</b> pentru a preciza front căzător / nivel LOW pentru delanșarea întreruperilor externe.

### 3. Registrul de activare a întreruperilor (IE)

MSB			LSB				
EA (EAL)	X	X	ES	ET1	EX1	ET0	EX0

BIT	SIMBOL	FUNCȚIE
IE.7	EA sau (EAL)	Dezactivează toate întreruperile. Dacă EA = 0 nu va fi confirmată nici o întrerupere. Dacă EA = 1, fiecare sursă de întrerupere este activată în mod individual prin setarea sau ștergerea bitului său de activare
IE.6	–	Rezervat
IE.5	–	Rezervat
IE.4	ES	Activează sau dezactivează întreruperea pentru portul serial. Dacă ES = 0, întreruperea pentru portul serial este dezactivată.
IE.3	ET1	Activează sau dezactivează întreruperea la depășirea timerului 1. Dacă ET1 = 0 întreruperea pentru timerul 1 este dezactivată
IE.2	EX1	Activează sau dezactivează întreruperea externă 1. Dacă EX1 = 0, întreruperea externă 1 este dezactivată.
IE.1	ET0	Activează sau dezactivează întreruperea la depășirea timerului 0. Dacă ET0 = 0 întreruperea pentru timerul 0 este dezactivată
IE.0	EX0	Activează sau dezactivează întreruperea externă 0. Dacă EX0 = 0, întreruperea externă 0 este dezactivată.

### Programul 1

```

/*****
/*                               Program 1 ASC                               */
*****/

#include <reg513.h>                // foloseste Registrii cu Functii Speciale ale microcontrolerului C513 - INFINEON
unsigned int j;                   // declar variabila globala fara semn pe 2 octeti
void init (void);                 // declar functia init - folosita la initializarea Timerului 0

void main (void)                  // bucla principala de program
{
    init();                       // apeleaza functia init
    while (1)                     // bucla infinita
    {
        for (j=0x01; j< 0x80; j<<=1) // baleere de port
        {
            P3 = j;                // copie valoarea din j in Portul P3
            while (~TF0);          // cat timp TF0=0 (adica Timerul are o valoare cuprinsa intre 0 si 65535) - ramii aici!
            TF0=0;                 // sterge flagul TF0 (care automat s-a setat - atunci cand Timerul s-a autodepasit)
        }

        for (j=0x80; j> 0x01; j>>=1)
        {
            P3 = j;
            while (~TF0);
            TF0=0;
        }
    }
}

void init (void)
{
    TMOD = 0x01; // configurez Timerul T0 sa lucreze pe 16 biti (cu incrementare la 1us) - fiind pe 16 biti, Timerul se autodepaseste la 2^16 = 65535us
    TF0 = 0;     // flag din Registrul TCON - flag ce anunta autodepasirea Timerului
    TR0 = 1;     // pornesc Timerul T0 !
}
    
```

## Programul 2

```
/*
*****
Program 2 ASC
*****
*/

#include <reg513.h> // foloseste Registrii cu Functii Speciale ale microcontrolerului C513 - INFINEON
unsigned int j; // declar variabila globala fara semn pe 2 octeti
char q = 10; // declar variabila globala cu semn = 10
void init (void); // declar functia init - folosita la initializarea Timerului 0 al acestui microcontroler

void main (void) // bucla principala de program
{
    unsigned char y; // declar variabila "locala"
    init(); // apeleaza functia init
    while (1) // bucla infinita
    {
        for (j=0x01; j< 0x80; j<=<=1) // baleere de port
        {
            P3 = j; // copie valoarea din j in Portul P3
            for (y=0; y<q; y++) //
            { // bucla "for" utilizata pentru a se obtine o intarziere
                while (~TF0); // de valoare = 10 x 65535us = 0,65s
                TF0=0; //
            } //
        }

        for (j=0x80; j> 0x01; j>=>=1)
        {
            P3 = j;
            for (y=0; y<q; y++)
            {
                while (~TF0);
                TF0=0;
            }
        }
    }
}

void init (void)
{
    TMOD = 0x01; // configurez Timerul T0 sa lucreze pe 16 biti (cu incrementare la 1us) -
    TF0 = 0; // fiind pe 16 biti, Timerul se autodepaseste la 2^16 = 65535us
    TR0 = 1; // flag din Registrul TCON - flag ce anunta autodepasirea Timerului
}
}
```

### Programul 3

```

/*****
/*
/* Program 3 ASC
/* In acest program s-a realizat cu ajutorul Timerului T0
/* o intarziere de 50000us=50ms, in modul urmator:
/* Timrul T0, daca ar porni de la valoarea 0000, pana la
/* valoarea FFFF, i-ar trebui 65535us
/* dar, daca as incarca initial octetii corespunzatori
/* Timerului T0 ( TH0=octet superior, TL0=octet inferior )
/* cu valoarea 15535=0x3CAF (heza) (65535-50000=15535)
/* atunci pana la autodepasire (FFFF), i-ar trebui Timerului
/* exact 50000us - cat mi-am propus!
*****/

#include <reg513.h> // foloseste Registrii cu Functii Speciale ale microcontrolerului C513 - INFINEON
unsigned int j; // declar variabila globala fara semn pe 2 octeti
char q = 20, qq = 5; // declar variabile globale cu semn
void init (void); // declar functia init - folosita la initializarea Timerului 0 al acestui microcontroler

void main (void) // bucla principala de program
{
    unsigned char y; // declar variabila "locala"
    init(); // apeleaza functia init
    while (1) // bucla infinita
    {
        for (j=0x01; j< 0x80; j<=<=1) // baleere de port
        {
            P3 = j; // copie valoarea din j in Portul P3
            for (y=0; y<q; y++) //
            {
                while (~TF0); // bucla "for" utilizata pentru a se obtine o intarziere
                TR0 = 0; // opresc Timerul T0
                TF0 = 0; // sterg flagul ce s-a setat la depasirea Timerului T0
                TH0 = 0x3C; // 65535-50000=15535us=3CAF (in hexa) [incarc octetul superior]
                TL0 = 0xAF; // [incart octetul inferior al timerului T0]
                TR0 = 1; // pornesc Timerul T0
            }
        }

        for (j=0x80; j> 0x01; j>=>=1)
        {
            P3 = j;
            for (y=0; y<qq; y++) //
            {
                while (~TF0); // bucla "for" utilizata pentru a se obtine o intarziere
                TR0 = 0; // opresc Timerul T0
                TF0 = 0; // sterg flagul ce s-a setat la depasirea Timerului T0
                TH0 = 0x3C; // 65535-50000=15535us=3CAF (in hexa) [incarc octetul superior]
                TL0 = 0xAF; // [incart octetul inferior al timerului T0]
                TR0 = 1; // pornesc Timerul T0
            }
        }
    }
}

void init (void)
{
    TMOD = 0x01; // configurez Timerul T0 sa lucreze pe 16 biti (cu incrementare la 1us) -
    // fiind pe 16 biti, Timerul se autodepaseste la 2^16 = 65535us
    TF0 = 0; // flag din Registrul TCON - flag ce anunta autodepasirea Timerului
    TR0 = 1; // pornesc Timerul T0 !
}

```

## Programul 4

```
/******  
/*                               Program 4 ASC                               */  
/*Timerul T0 este utilizat pentru a se obtine o intarziere                    */  
/*de 50ms                                                                    */  
/*Timerul T1 este utilizat pentru a se obtine o intarziere                    */  
/*de 10ms                                                                      */  
/******  
  
#include <reg513.h> // foloseste Registrii cu Functii Speciale ale microcontrolerului C513 - INFINEON  
unsigned int j; // declar variabila globala fara semn pe 2 octeti  
unsigned char q = 10, qq = 50; // declar variabile globale cu semn  
void init (void); // declar functia init - folosita la initializarea Timerului 0 al acestui microcontroler  
  
void main (void) // bucla principala de program  
{  
    unsigned char y; // declar variabila "locala"  
    init(); // apeleaza functia init  
    while (1) // bucla infinita  
    {  
        for (j=0x01; j< 0x80; j<<=1) // baleere de port  
        {  
            P3 = j; // copie valoarea din j in Portul P3  
            for (y=0; y<q; y++) //  
            {  
                while (~TF0); // bucla "for" utilizata pentru a se obtine o intarziere  
                TR0 = 0; // opresc Timerul T0  
                TF0 = 0; // sterg flagul ce s-a setat la depasirea Timerului T0  
                TH0 = 0x3C; // 65535-50000=15535us=3CAF (in hexa) [incarc octetul superior]  
                TL0 = 0xAF; // [incart octetul inferior al timerului T0]  
                TR0 = 1; // pornesc Timerul T0  
            }  
        }  
        for (j=0x80; j> 0x01; j>>=1)  
        {  
            P3 = j;  
            for (y=0; y<qq; y++) //  
            {  
                while (~TF1); // bucla "for" utilizata pentru a se obtine o intarziere  
                TR1 = 0; // opresc Timerul T1  
                TF1 = 0; // sterg flagul ce s-a setat la depasirea Timerului T1  
                TH1 = 0xD8; // 65535-10000=55535us=D8EF (in hexa) [incarc octetul superior]  
                TL1 = 0xEF; // [incart octetul inferior al timerului T0]  
                TR1 = 1; // pornesc Timerul T0  
            }  
        }  
    }  
}  
  
void init (void)  
{  
    TMOD = 0x11; // configurez Timerul T0 si T1 sa lucreze pe 16 biti (cu incrementare la 1us) -  
    // fiind pe 16 biti  
    TF0 = 0; // flag din Registrul TCON - flag ce anunta autodepasirea Timerului  
    TF1 = 0; // flag din Registrul TCON - flag ce anunta autodepasirea Timerului  
    TR0 = 1; // pornesc Timerul T0 !  
    TR1 = 1; // pornesc Timerul T1 !  
}
```

## Programul 5

```

/*****
/*          Program 5 ASC          */
*****/

#include <reg513.h>                // foloseste Registrii cu Functii Speciale ale microcontrolerului C513 - INFINEON
unsigned char q = 70, qq = 20, y, j; // declar variabile globale cu semn
void init (void);                // declar functia init - folosita la initializarea Timerului 0 al acestui microcontroler
void delay (unsigned char r);     // declar functia de intarziere delay

void main (void)                  // bucla principala de program
{
    init();                       // apeleaza functia init
    while(1)                       // bucla infinita
    {
        while (1)                  // bucla infinita
        {
            for (j=0x01; j< 0x80; j<=<=1) // baleere de port
            {
                P3 = j;            // copie valoarea din j in Portul P3
                delay(qq);         // apeleaza functia de intarziere delay
            }

            for (j=0x80; j> 0x01; j>=>=1)
            {
                P3 = j;
                delay(q);
            }
        }
    }

void init (void)
{
    TMOD = 0x01;                  // configurez Timerul T0 sa lucreze pe 16 biti (cu incrementare la 1us) -
    TF0 = 0;                      // fiind pe 16 biti
    TR0 = 1;                      // flag din Registrul TCON - flag ce anunta autodepasirea Timerului
    // pornesc Timerul T0 !

void delay (unsigned char r)
{
    for (y=0; y<r; y++)           // bucla "for" utilizata pentru a se obtine intarzierea dorita
    {
        while (~TF0);            // stau aici 10ms
        TR0 = 0;                  // opresc Timerul T0
        TF0 = 0;                  // sterg flagul ce s-a setat la depasirea Timerului T0
        TH0 = 0xD8;               // 65535-10000=55535us=D8EF (in hexa) [incarc octetul superior]
        TL0 = 0xEF;               // [incarc octetul inferior al timerului T0]
        TR0 = 1;                  // pornesc Timerul T0
    }
}
}

```

