

Nom i Cognoms: \_\_\_\_\_ Possible Solució \_\_\_\_\_

- 1) Volem implementar una rutina Sleep() pel PIC18F45K22 amb un oscil·lador de 8MHz que trigui N segons ( $0 < N < 1000$ ) en executar-se. Per això hem decidit fer servir:
- el Timer0 (mode 16 bits, entrada Fosc/4 i pre-scaler constant a 2),
  - la rutina d'interrupció de baixa prioritat, que permetrà mesurar un retard més petit, i
  - una variable global de sincronització.

Completa la rutina Sleep() perquè funcioni correctament. (2 PUNTS)

Donat que la freqüència de l'oscil·lador es de 8MHz i que el Timer està configurat per incrementar-se a un regim de  $\frac{1}{4}$  d'aquesta freqüència i amb un pre-scaler de 2 podem concloure que el registre comptador del timer s'incrementarà cada 8 ticks de rellotge, és a dir, cada microsegon. Com que a cada crida per desbordament programem el timer a un valor de 65535-1000 podem deduir que es desborda cada 10000 ticks ) i que a la variable Sincro per tant, se li assigna el valor de 1 cada centèsima de segon (1microsegon \* 10000).

Podem fer servir aquesta variable de sincronització per tal de comptar el numero de vegades que Sincro es posa a 1.

El numero de segons que volem que trigui la funció en executar-se es correspondrà a  $100 * N$  vegades que Sincro es posa a 1 on N es el número de segons.

Una possible solució a la rutina Sleep és:

```
int Sincro=0; // variable global per sincronitzar

void Sleep( int N)
{
    long cont = 0;
    While( cont < N * 100; cont ++ )
    {
        While(sincro == 0);
        Sincro = 0;
        cont++;
    }
}

void interrupt RSI_LOW (void)
{
    If (TMR0IF && TMR0IE)
    {
        TMR0=65535-10000;
        Sincro=1;
        TMR0IF=0;
    }
}
```

- 2) Volem comptar el nombre de polsos que produeix un senyal extern que hem connectat al pin T1CKI del PIC18F45K22 (corresponent al pin d'entrada del Timer 1 configurat en mode comptador, Prescaler del Timer1= 1, amb interrupció d'overflow activada i en mode 16 bits). Si en 1 segon hem tingut 2 interrupcions del TMR1IF, calcula el valor màxim i mínim possibles per la freqüència que ens arriba a T1CKI. (1,5 PUNTS)

De l'enunciat deduïm que les dues situacions extremes son:

- 1) que el comptador del timer hagi arribat al segon desbordament (interrupció) exactament quan finalitza el primer segon

$$F_{\text{senyal\_minima}} = 65536 * 2 = 131.072;$$

- 2) Que just quan acaba el segon el timer estigui a un sol tick de desbordar-se o de produir una tercera interrupció dins del primer segon. Per aquest cas

$$F_{\text{Senyal\_Max}} = 65536 * 3 - 1 = 196.607$$

Nom i Cognoms: \_\_\_\_\_ Possible Solució \_\_\_\_\_

- 3) Penses que es podria emular per software (actuant sobre un PIN per codi) la funcionalitat del PWM tenint exactament les mateixes prestacions que la que implementa la unitat CCP? **Justifica la resposta! (1,5 PUNTS)**

Si l'enunciat ens demanés implementar l'emulació amb timers parells la resposta hagués estat NO. Donat que PWM per hardware té una resolució per fixar el període de Duty Cycle de 10 bits que s'incrementa a la freqüència de l'oscil·lador. En cap cas amb un timer parell podem obtenir ni aquesta resolució ni aquesta cadència. Ara be, podríem fer servir Timers imparells que si que poden incrementar-se a la freqüència de l'oscil·lador i a més son de 16 bits. Una possible solució passaria per programar dos timers imparells programats a la freqüència que volem pel PWM i pel DutyCycle. L'única cosa que hauria de fer la rutina d'interrupció seria posar a '0' o a '1' un pin extern segons correspongui.

- 4) Quin és l'error màxim (en ticks de timer) que es pot cometre al llegir un timer de 16 bits si accedim de forma incorrecta a la part baixa i la part alta (sense control d'errors)?

**Justifica la resposta. (1 PUNT)**

Imaginem un timer amb un valor 0xFFFF

Si llegim la part alta obtindrem 0xFF. Després al següent cicle, quan llegim la part baixa, el timer haurà fet overflow (TIMER=0x0000), obtindrem 0x00 i direm que el valor del timer és 0xFF00. En conseqüència, l'error màxim és de 65280 tics

- 5) Volem generar un PWM de freqüència 2500Hz i *Duty Cycle* del 5%. Tenim un oscil·lador de 8MHz connectat al PIC de laboratori. Configura tot el necessari perquè aparegui aquest senyal al pin CCP1. **(2,5 PUNTS)**

Triarem el CCP1 funcionant amb el Timer 2 tot i que podria ser qualsevol CCP treballant amb el timer 2, 4 o 6.

Per generar una freqüència de 2500Hz trobem que:

$$1/2500 = [PR2+1] * PRE * 1/8000000.$$

Si triem el PRE més petit que permeti un  $PR2 < 2^8$ , trobem que  $PR2=199$  i  $PRE=4$

Per un DC del 5% trobem que:

$$5\% = CCPR1L:CCP1CON<5:4> / (4 * (PR2+1)) \Rightarrow CCPR1L:CCP1CON<5:4> = 40 = 0b101000$$

La configuració és:

TRISCbits.RC2=1 //disable TRIS

CCPTMRS0=XXXXXX00 //select CCP1 with timer 2

PRE2 =199;

CCPR1L=0b00001010 //Load DC MSbs

CCP1CON=XX0011XX // CCP1CON<5:4>=00 & CCP1 in PWM mode

T2CON=X0000101 // PRE=4 & TIMER2 ON

TRISCbits.RC2=0 //enable TRIS

- 6) Connectem un senyal que dona un pols positiu de 10us d'ample als pins CCP1 i CCP2 del microcontrolador.

Nom i Cognoms: \_\_\_\_\_ Possible Solució \_\_\_\_\_

Si tenim configurats els registres CCPTMRS0=0x00, CCPTMRS1=0x00, CCP1CON=0x05, CCP2CON=0x04, T1CON=0x11, TMR1GE=0 i després del flanc de baixada llegim els registres CCPR1 i CCPR2 amb valors 0x124 i 0x13D respectivament, quin és l'oscil·lador del micro?

**Justifica la resposta (1,5 PUNTS).**

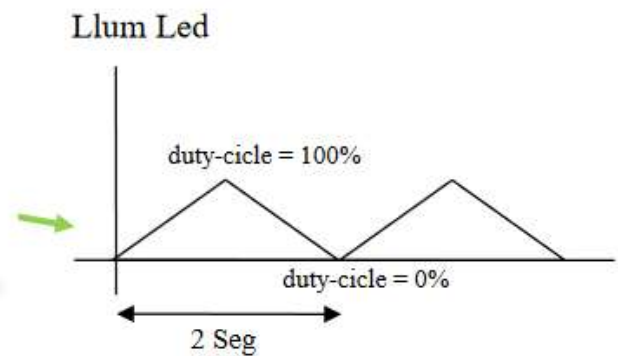
El CCP1 està configurat per capturar flancs de pujada i el CCP2 captura flancs de baixada. Tots dos fan servir el Timer1 que està configurat amb PRE=2 i CS=Fosc/4.

El 10us del flanc queden capturats entre els tics de CCP2-CCP1 =25 tics

Sabem que  $1\text{tic} = 4 \cdot \text{PRE} / \text{Fosc}$  i que  $1\text{tic} = 10\mu\text{s} / 25$ . Si resollem trobem que  $\text{Fosc} = 20\text{MHz}$

### Qüestions de laboratori NLAB2

A la pràctica de PWM vam generar un senyal modulats en amplitud de pols que feia oscil·lar el seu Duty Cycle de 0 a 100% amb un període de 2 segons. La sortida la vam veure associada al LED RC2 de la placa.



Respon breument les següents qüestions:

- Podem treure la sortida PWM per qualsevol pin del microcontrolador?

No, només per aquells que tenen associada una sortida de unitat CCP.

- Quin dels registres de la unitat CCP vam modificar per aconseguir el període de 2 segons.

Cap. Els registres de la unitat CCP es feien servir per generar la freqüència de PWM (1kHz) i el Duty Cycle variable. Perquè la variació del Duty tingués un cicle de 2 segons havíem de fer servir un Timer independent o una variable comptador.

- A Proteus, com vam veure que el senyal de PWM estava ben configurat i complia amb la freqüència i el duty-cycle?

Amb l'eina oscil·loscopi virtual de Proteus.

- Es pot fer servir el pin RC2 quan també utilitzem la GLCD de la placa?

Si, la GLCD utilitza el Port B i el Port D. El C queda lliure.