

Cognoms i Nom: _____ Doc. Identitat: _____

Totes les respostes han d'estar degudament justificades

1) Explica en quins casos podem utilitzar la instrucció RETFIE FAST i en quins casos haurem d'utilitzar la instrucció RETFIE per retornar d'una rutina d'atenció a la interrupció? (1,5 pts.)

2) Es vol utilitzar el TIMER3 per a generar una base de temps. (1,5 pts.)

a) A quina freqüència s'activa el TMR3IF si T3CON= 0x91, i Fosc= 32MHz.

b) Escriu el codi C per a la inicialització del TIMER3 i del mecanisme d'interrupcions, i la rutina de servei a la interrupció, per a generar una interrupció de baixa prioritat cada 10 ms amb el TIMER3. Fosc= 32MHz.

3) Per una aplicació, s’han de comptabilitzar el nombre de canvis (flancs de pujada i flancs de baixada) que es produeixen en el senyal binari **S**, generat per un sensor, utilitzant un PIC18F4550 –Fosc= 8MHz-. Per les dues opcions proposades:

- 1) Connexió del senyal **S** a l’entrada RA4, i detecció dels canvis per enquesta d’aquesta entrada
- 2) Connexió del senyal **S** a l’entrada RB0, i detecció dels canvis per interrupcions INT0. (3 pts.)

<p>CODE 1 – Pooling</p> <pre>#include <xc.h> void init_pic1 () { // INIT PIC to read Sensor // by pooling } void main(void) { int comptador=0; byte lastS, S; init_pic1 (); S= PORTAbits.RA4; lastS= S; while (1) { S=PORTAbits.RA4; if (S!=lastS) comptador++; lastS= S; // APPLICATION CODE // EXECUTION TIME 350 mseg. } }</pre>	<p>CODE 2 - Interruption</p> <pre>#include <xc.h> int comptador=0; void interrupt ISR() { // Complete the // Interrupt Service Routine } void init_pic2() { // INIT PIC to read Sensor // using interruptions } void main(void){ init_pic2(); while (1) { // APPLICATION CODE // EXECUTION TIME 350 mseg. } }</pre>
---	--

- a) Completa la rutina **init_pic1** del codi 1 amb la finalitat de configurar adequadament el PIC segons l’opció 1.
- b) Completa les rutines **init_pic2** i **ISR** del codi 2 amb la finalitat de configurar i programar adequadament el PIC segons l’opció 2.

--	--

- c) Quina és la latència (temps entre el canvi del senyal **S** i l’actualització de la variable de comptatge) en el pitjor cas, en cada una de les opcions.

d) Per a cada opció, determinar el temps necessari, com a mínim, entre canvis consecutius de **S** per a que puguin ser detectats.

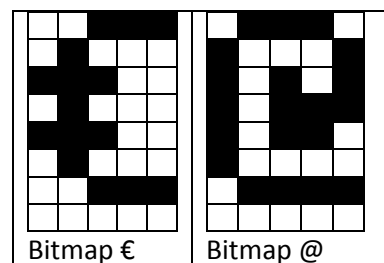
e) Quina de les dues opcions considereu millor. Justifiqueu la resposta.

4) A més dels caràcters alfanumèrics, es vol mostrar al GLCD diferents símbols (p.ex. €, @, \$, *, →, ...). Amb aquesta finalitat s'ha d'ampliar la llibreria "rutines_GLCD" que utilitzeu al laboratori, amb el mapa de caràcters, els bitmaps dels símbols (relació entre el codi, el símbol, i els valors dels píxels associats a cada símbol, els símbols es representa en un bitmap de 8x5 píxels), i la rutina writeSym per mostrar el símbol al GLCD. (2 pts.)

```
byte symbols [5*N]={ .... }; // Bitmap for N symbols, N<=100
void writeSym (byte page, byte y, byte codi); // Write symbol at GLCD
```

a) Completeu els valors del vector **symbols** per a representar € i @ segons el següents bitmaps. Contempleu únicament els símbols € (codi 0) i @ (codi 1) (N=2).

b) Escriure el codi de la rutina **writeSym** per mostrar un símbol, a la pàgina i columna indicats.



Nota: A les llibreries rutines_GLCD disposeu de les rutines/funcions void writeByte (byte page, byte y, byte data), byte readByte (byte page, byte y), void SetDot(byte x, byte y), void ClearDot (byte x, byte y), entre d'altres.

Cognoms i Nom: _____

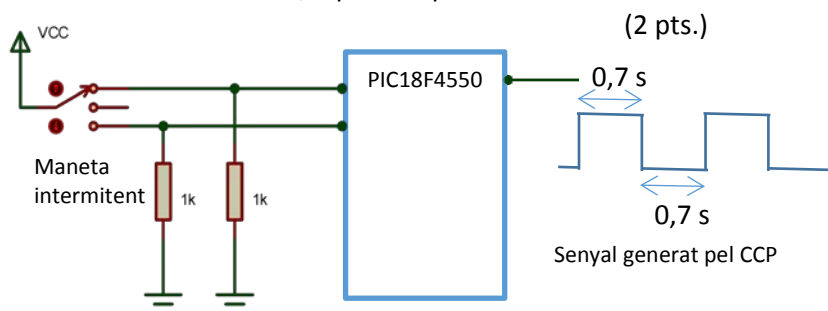
Doc. Identitat: _____

Totes les respostes han d'estar degudament justificades

5) Un fabricant de vehicles ens demana dissenyar diversos circuits pel seu nou model. Per raons econòmiques volen gestionar diferents dispositius del cotxe amb un únic microcontrolador PIC18F4550 que funciona a una freqüència de rellotge de 2MHz. Un dels dispositius que volen gestionar són els intermitents. Ens demanen llegir des del PIC dos senyals provinents de la maneta de l'intermitent del panel de control del conductor, i que indiquem si s'ha d'encendre l'intermitent esquerra o dret.

a) Com implementaríeu amb el PIC la lectura de l'estat de la maneta dels intermitents.

Quina/es unitat/s del PIC faries servir i com la/es configuraria/es?



b) El fabricant també vol generar des del PIC un senyal per encendre els llums dels intermitents. Durant l'activació dels intermitents, el temps en que els intermitents estan encesos (0,7 segons) ha de ser igual al que estan apagats. Volem implementar aquesta solució fent servir el mòdul CCP2 del PIC amb el TIMER3 com a timer associat.

Quins valors assignaries als registres T3CON, CCP2CON, CCPR2H i CCPR2L per tal d'implementar aquesta solució?

Indica només els bits que has de modificar i el seu valor. Justifica la resposta i fes els càlculs pertinents.

REGISTER 14-1: T3CON: TIMER3 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS
bit 7						bit 0

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
-1' = Bit is set
-0' = Bit is cleared
-x = Bit is unknown

- bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations
- bit 6, 3 **T3CCP2:T3CCP1:** Timer3 and Timer1 to CCPx Enable bits
1x = Timer3 is the capture/compare clock source for both CCP modules
01 = Timer3 is the capture/compare clock source for CCP2;
Timer1 is the capture/compare clock source for CCP1
00 = Timer1 is the capture/compare clock source for both CCP modules
- bit 5-4 **T3CKPS1:T3CKPS0:** Timer3 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value
- bit 2 **T3SYNC:** Timer3 External Clock Input Synchronization Control bit
(Not usable if the device clock comes from Timer1/Timer3.)
When TMR3CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.
- bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)
0 = Internal clock (FOSC/4)
- bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

REGISTER 15-1: CCPxCON: STANDARD CCPx CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
— ⁽¹⁾	— ⁽¹⁾	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

Unimplemented: Read as '0'⁽¹⁾

DCxB1:DCxB0: PWM Duty Cycle Bit 1 and Bit 0 for CCPx Module

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the two LSBs (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight MSBs of the duty cycle are found in CCPR1L.

CCPxM3:CCPxM0: CCPx Module Mode Select bits
0000 = Capture/Compare/PWM disabled (resets CCPx module)
0001 = Reserved
0010 = Compare mode: toggle output on match (CCPxIF bit is set)
0011 = Reserved
0100 = Capture mode: every falling edge
0101 = Capture mode: every rising edge
0110 = Capture mode: every 4th rising edge
0111 = Capture mode: every 16th rising edge
1000 = Compare mode: initialize CCPx pin low, on compare match, force CCPx pin high (CCPxIF bit is set)
1001 = Compare mode: initialize CCPx pin high; on compare match, force CCPx pin low (CCPxIF bit is set)
1010 = Compare mode: generate software interrupt on compare match (CCPxIF bit is set, CCPx pin reflects I/O state)
1011 = Compare mode: trigger special event, reset timer, start A/D conversion on CCPx match (CCPxIF bit is set)
11xx = PWM mode

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF ⁽¹⁾

REGISTER 9-10: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 ⁽¹⁾	U-0	R/W-1	R-1	R-1	R/W-0 ⁽²⁾	R/W-0
IPEN	SBOREN	—	—	—	—	POR	BOR

REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPUR	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—