

Cognoms, Nom \_\_\_\_\_ DNI \_\_\_\_\_

**Tota resposta sense justificar es considerarà nul·la !**

**P1. (1,5 punts)** Per controlar la velocitat de rotació d'un motor incorporem un sensor que produirà un flanc de pujada a cada volta del motor. Connectarem aquest flanc a un CCP configurat en mode capture cada flanc ascendent. Sabem que  $F_{osc}=8\text{MHz}$  i  $T1CON=0x03$  i que tot el necessari està ben configurat. Quina serà la velocitat mínima en RPM (Revolucions per Minut) que podem capturar si no controlem els overflows de Timer1?

Segons la configuració el timer està funcionant sense PRE i amb un  $CS=F_{osc}/4$ .

Com que no controlem els overflows de timer, sabem que com a màxim es pot calcular la velocitat del motor quan el temps d'una volta és més petit que el temps en el que el timer (de 16 bits) fa l'overflow:

$$T_{max} = 2^{16} \cdot \frac{4}{F_{osc}} \cdot PRE = 32768 \mu s$$

Si la volta més lenta que podem controlar triga  $32768 \mu s$ , això es correspon a una velocitat de:

$$\frac{1}{32768 \cdot 10^{-6} / 60} = 1831 \text{ RPM}$$

**P2. (1.5 punts)** Quin és el període màxim i mínim de PWM que podem generar amb el mòdul CCP del nostre PIC18 si  $F_{osc}=4\text{MHz}$ ? Quants duty cycles diferents podem crear quan configurem el mòdul amb el període màxim? I amb el mínim?

El període màxim s'obté amb el PR màxim i el PRE més gran possible:

$$T_{max} = (255 + 1) \cdot \frac{4}{4M} \cdot 16 = 4096 \mu s$$

$$Resolution = \frac{\log[4(255 + 1)]}{\log(2)} = 10 \text{ bits}$$

Amb un PR de 255 tenim una resolució de 10 bits per fer duty cycles i es poden fer fins a 1024 dutys diferents.

El mínim s'obté amb el PR mínim i sense PRE

$$T_{min} = (0 + 1) \cdot \frac{4}{4M} \cdot 1 = 1 \mu s$$

$$Resolution = \frac{\log[4(0 + 1)]}{\log(2)} = 2 \text{ bits}$$

Amb un PR=0 teinm dos bits de resolució i es poden fer fins a 4 dutys diferents.

**P3. (1 punt)** Comprensió dels protocols de comunicacions: UART, SPI, I2C.

Indica amb una X quina de les afirmacions és certa per cada plantejament (sols hi ha una correcta). Encert suma 1/4. Error resta 1/12 . Blanc no afecta. En aquesta pregunta, no cal justificar les respostes.

3.1 El bus 1-wire és

	sèrie asíncron full-duplex
X	sèrie asíncron half-duplex
	sèrie síncron full-duplex
	sèrie síncron half-duplex

### 3.2 El bus i2c és

	sèrie asíncron full-duplex
	sèrie asíncron half-duplex
	sèrie síncron full-duplex
X	sèrie síncron half-duplex

### 3.3 El bus spi és

	sèrie asíncron full-duplex
	sèrie asíncron half-duplex
X	sèrie síncron full-duplex
	sèrie síncron half-duplex

### 3.4 La tècnica de bit-banging consisteix en

	En la transmissió d'informació entre dispositius utilitzant perifèrics específics
X	En la transmissió d'informació per programari enlloc d'utilitzar un maquinari dedicat
	Generar un clock fent servir algun dels timers present al microcontrolador
	Demanar a una IA que ens doni el codi per programar el nostre microcontrolador

**P4. (1 punts)** En una transmissió de dades USB 1.1 calcula quant triga a enviar-se un data packet si volem enviar 100 Bytes de dades i el bus és Low Speed (1.5Mb/s)

**DATA (Data to be sent/received)**

SOP	DATA0/1	DATA	CRC	EOP
8b	8b	0 a 1203B	16b	2b

Podem enviar els 100 Bytes de dades encapsulats en un únic Data Packet. Llavors, el packet que enviem conté els següents bits:

8bits SOP + 8bits Data0/1 + 100Bytes dades (800bits) + 16bits CRC + 2bits EOP = 834bits

En enviar-ho es triga:

$$834bits \cdot \frac{1s}{1,5Mbits} = 556\mu s$$

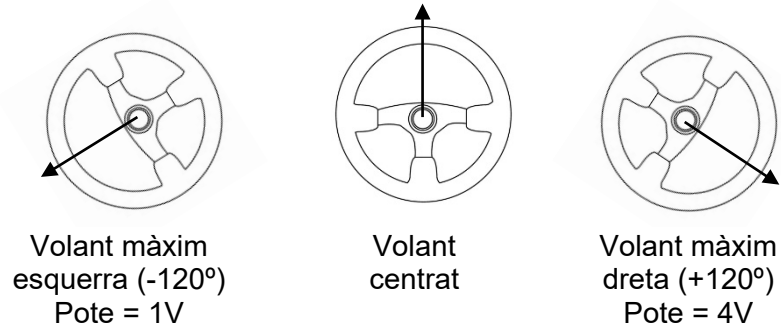
Cognoms, Nom \_\_\_\_\_ DNI \_\_\_\_\_

**Tota resposta sense justificar es considerarà nul·la !**

**P5. (1 punts)** Una marca de cotxes vol llegir el gir del volant usant un potenciòmetre. Amb el volant girat al màxim a l'esquerra ( $-120^\circ$ ) el pote donarà 1V. Girant al màxim a la dreta ( $+120^\circ$ ) donarà 4V.

Volem llegir els graus de gir del volant amb una resolució de  $0.2^\circ$ .

Quants bits haurà de tenir el nostre conversor A/D com a mínim per complir amb els requeriments? Les tensions de referència del ADC seran  $V_{REF-}=0V$  i  $V_{REF+}=5V$ .



Entre els dos extrems de gir del volant, tenim un rang de  $120^\circ - (-120^\circ) = 240^\circ$ . Com que es vol una resolució de  $0.2^\circ/\text{step}$ , llavors el número de steps que caldran en el rang de voltatge [1V...4V] serà de:

$$\text{Num steps}_{[1..4]} = \frac{240^\circ}{0.2^\circ/\text{step}} = 1200 \text{ steps}.$$

Com que el ADC treballa amb tensions de referència en el rang [0V... 5V], necessitem calcular quants steps d'ADC calen en el rang complet de tensions de referència.

$$\text{Num steps}_{[0..5]} = \frac{1200}{3} \cdot 5 = 2000 \text{ steps}$$

Per tenir 2000 steps, calen 2001 codis. Calculem quants bits necessitem per tenir 2001 codis:

$$nBits = \log_2(2001) = 10.97bits$$

Per tant, necessitem com a mínim un conversor ADC de 11 bits.

**P6. (1 punts)** El registre ADCON2 del ADC del micro està configurat al valor hexadecimal 0xAA. El nostre PIC treballa amb una  $F_{osc}=8 \text{ MHz}$ .

P6.1. Quin és el temps total de mostreig d'una senyal d'entrada analògica?

El valor 0xAA en binari és: 10101010. Si mirem la definició del ADCON2 al formulari, veurem que el camp ACQT = 0b101. I el camp ADCS = 0b010.

Aquests valors indiquen que tindrem un temps d'adquisició de 12 Tads. I un clock de conversió =  $F_{osc}/32$ .

$$T_{ad} = \frac{32}{F_{osc}} = \frac{32}{8MHz} = 4\mu s$$

El temps total de mostreig està format pel temps d'adquisició, més 11 Tad de la conversió, més 1 Tad extra per a la descàrrega del condensador de hold.

$$T_{mostreig} = T_{acq} + 12T_{ad} = 12 \cdot 4\mu s + 12 \cdot 4\mu s = 96\mu s$$

P6.2. Amb aquest temps de mostreig, quina serà la freqüència màxima que podrem convertir amb l'ADC sense tenir problemes d'aliasing?

Segons criteri de Nyquist, per evitar problemes d'aliasing cal que:  $F_{mostreig} > 2 \cdot F_{senyal}$ .

Sabem que  $F_{mostreig} = 1/96\mu s = 10416.67 \text{ Hz}$ .

$$\text{Per tant, } F_{senyal} < \frac{F_{mostreig}}{2} \approx 5208Hz$$

**P7. (2 punts)** Necessitem usar el perifèric UART del PIC18F45K22 per tal de transmetre dades sèrie en mode asíncron, sense paritat, amb 8 bits de dades, 1 bit d'stop i amb Baudrate=57600. La Fosc és de 4MHz.

Configura els bits necessaris dels registres TXSTA, RCSTA, BAUDCON, SPBRGH i SPBRG per a que funcioni la transmissió correctament.

Especifica clarament els càlculs que justifiquen les teves decisions i el % d'error que cometem en el Baudrate amb la configuració triada.

Primer de tot provem les 3 fórmules per saber quina s'ajusta millor al Baudrate desitjat.

Cas divisor 64:

$$n = \frac{Fosc}{64 \cdot BR} - 1 = \frac{4MHz}{64 \cdot 57600} - 1 = 0.085 \Rightarrow \text{arrodonim a 0. } BR_{real} = \frac{4MHz}{64 \cdot (0 + 1)} = 62500$$

$$\% Error = \frac{BR_{real} - BR_{ideal}}{BR_{ideal}} \cdot 100 = \frac{62500 - 57600}{57600} \cdot 100 = 8.5\% \Rightarrow \text{error excessiu (supera 5\%)}$$

Cas divisor 16:

$$n = \frac{Fosc}{16 \cdot BR} - 1 = \frac{4MHz}{16 \cdot 57600} - 1 = 3.34 \Rightarrow \text{arrodonim a 3. } BR_{real} = \frac{4MHz}{16 \cdot (3 + 1)} = 62500$$

$$\% Error = 8.5\% \text{ (mateix cas d'abans)} \Rightarrow \text{error excessiu (supera 5\%)}$$

Cas divisor 4:

$$n = \frac{Fosc}{4 \cdot BR} - 1 = \frac{4MHz}{4 \cdot 57600} - 1 = 16.36 \Rightarrow \text{arrodonim a 16. } BR_{real} = \frac{4MHz}{4 \cdot (16 + 1)} = 58823.53$$

$$\% Error = \frac{58823.53 - 57600}{57600} \cdot 100 = 2.12\% \Rightarrow \text{error acceptable! (no supera 5\%)}$$

Per tant, ens quedem amb divisor 4, que implica BRGH=1 i BRG16=1.

Bits a configurar al TXSTA: TX9=0, TXEN=1, SYNC=0, BRGH=1

Bits a configurar al RCSTA: SPEN=1

Bits a configurar al BAUDCON: BRG16=1

SPBRGH=0 i SPBRG=16

**P8. (1 punts)** Contesta breument les següents preguntes sobre la línia sèrie UART.

1) Disposa la línia sèrie d'algun sistema de detecció d'errors en els bits de dades? Si existeix aquest sistema, ¿està implementat en el perifèric del nostre PIC18F45K22?

Sí, el bit de paritat permet detectar errors en la transmissió de dades. No està implementat per hardware en el nostre PIC, tot i així per software el podríem calcular i enviar-lo usant el 9è bit de dades.

2) Volem connectar dos PIC parlant entre ells per línia sèrie (es transmetran dades en els dos sentits). Especifica les connexions que haurem de fer entre els dos micros.

Haurem de connectar el pin Tx del primer micro amb el pin Rx del segon. I viceversa (el Tx del segon amb el Rx del primer). Així com connectar els pins GND (Vss) per a que tots dos micros comparteixin els 0V.

3) Què significa que la línia sèrie sigui un protocol de comunicacions Full-Duplex?

Que es poden fer transmissions i recepcions de dades simultàniament.

4) Quan es transmet la informació per línia sèrie, quin bit de les dades comença a enviar-se primer: el LSbit o el MSbit?

Es comença enviant el Least Significant bit.

5) Quants bits en total s'envien en una trama de línia sèrie, necessaris per enviar 1 byte de dades?

Cal enviar 1 bit d'start + 8 bits de dades + 1 bit d'stop. Total: 10 bits