Nom i Cognoms: _____Possible Solució_____

Computer Interfacing. Tercer parcial, 16/1/2018

Responeu les preguntes a l'espai reservat. ***Justifiqueu*** *les respostes.*
Poseu el nom a tots els fulls.

1) Volem capturar un senyal de vídeo analògic provinent d'una càmera que té un sensor CCD de tal manera que a partir de la digitalització obtinguem un vídeo amb una resolució de **640x480 píxels per imatge i 30 imatges per segon**. Els píxels venen seqüencialment per una única línia i un cop convertit cada píxel estarà representat per un únic valor de 10 bits (la imatge no és en color).
   Penseu que seria possible realitzar aquesta digitalització amb el conversor AD del PIC18F4550 funcionant amb un oscil·lador a 20MHz? Justifica la resposta amb els càlculs pertinents *(2 punts)*.

Hi ha vàries formes de demostrar que no és possible fer la digitalització del senyal que demana l'enunciat.

La més simple és calcular la velocitat a la que hauríem de mostrejar cada píxel i veure que és impossible assolir-la.

De l'enunciat deduïm que hauríem de mostrejar 640x480x30 píxels en un segon, es a dir 9.216.000 píxels. Això vol dir que hauríem de mostrejar un píxel cada 0,1 microsegons. Sabem que el temps mínim TEÒRIC de mostreig que podríem aconseguir (independentment de la velocitat de l'oscil·lador del micro) és TACQ + 12TAD. Donades les restriccions del fabricant (TACQ min > 2,45 microsegons i TAD >= 0,8 microsegons) directament es pot veure que el temps mínim teòric de mostreig que podem assolir amb aquest micro és molt superior a les exigències del problema.

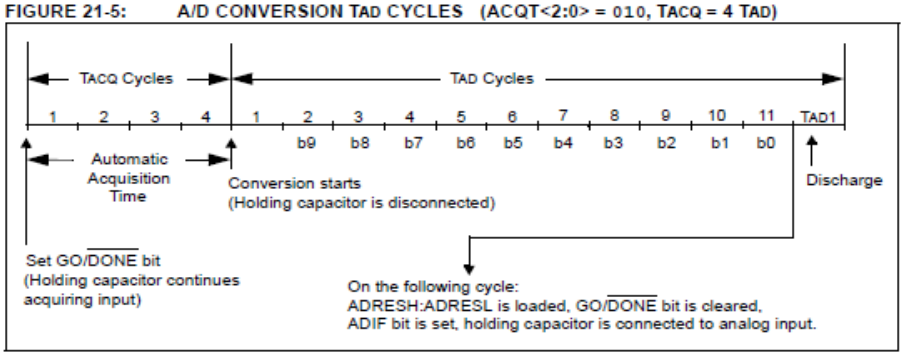
(2,45 microsegons + 12 * 0,8 microsegons) >> 0,1 microsegon.

2) Descriu breument els sistemes de detecció d'errors de comunicació del bus SPI *(1 punt)*.

El bus SPI NO disposa de cap sistema de detecció d'errors. És un dels seus inconvenients. Seria possible implementar-los però sempre amb una implementació feta i controlada per l'usuari i amb els sobrecostos en el temps d'enviament.

Nom i Cognoms: _____Possible Solució_____

FIGURE 21-5: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



Requirements:

Tad>0,8us
Tacq>2,45us

PIC18F family

	7	6	5	4	3	2	1	0
	ADFM	--	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
value after reset	0	0	0	0	0	0	0	0

ADFM: A/D result format select bit
0 = left justified
1 = right justified
ACQT2:ACQT0: A/D acquisition time select bits
000 = 0 TAD(1)
001 = 2 TAD
010 = 4 TAD
011 = 6 TAD
100 = 8 TAD
101 = 12 TAD
110 = 16 TAD
111 = 20 TAD
ADCS2:ADCS0: A/D conversion clock select bits
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
011 = FRC (clock derived from A/D RC oscillator)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64
111 = FRC (clock derived from A/D RC oscillator)

Note 1: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

Figure 12.10a ADCON2 register (PIC18F8X8X/8X2X/6X2X/2X20/4x20/1220/1320) (redraw with permission of Microchip)

Nom i Cognoms: _____Possible Solució_____

3) Es vol configurar la USART del microcontrolador PIC18F4550 per realitzar unes transmissions molt precises a 9600, 19200, 38400, 57600 i 115200 bps.
Quina seria la freqüència mínima més adequada de l'oscil·lador del sistema (F_{OSC}) per obtenir amb precisió les velocitats de transmissió referides anteriorment? *(2 punts)*.

TABLE 20-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	Fosc/[64 (n + 1)]
0	0	1	8-bit/Asynchronous	Fosc/[16 (n + 1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	Fosc/[4 (n + 1)]
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGH:SPBRG register pair

Per poder generar les freqüències de transmissió de 9600, 19200, 38400, 57600 i 115200 bps de forma exacte utilitzant la UART del pic18f, cal que la freqüència del sistema compleixi que

(Eq 1) Fosc / [4(n+1)] = 115200, 57600, 38400,19200,9600 o bé
(Eq 2) Fosc / [16(n+1)] = 115200, 57600, 38400,19200,9600 o bé
(Eq 3) Fosc / [64(n+1)] = 115200, 57600, 38400,19200,9600

També s'observa que si aconseguim configurar la UART amb la major freqüència de transmissió (115200 Hz) també s'obtenen fàcilment les altres freqüències doncs totes elles són exactament la meitat de l'anterior.

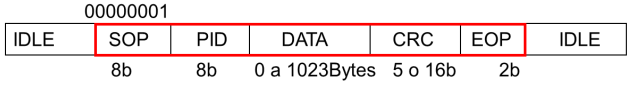
De la primera eqüació (Eq1) obtenim que Fosc / [4(n+1)] = 115200, si prenem el valor més baix de n = 0 obtenim que Fosc = 4 * 115.200 bps = 460.800 Hz. Amb la freqüència de 460.800 es poden generar exactament les altres freqüències que es soliciten utilitzant els valors enters de n=1, n=2, n=5, n=4. n = 11.

Si provem amb la Eq 2, tal que Fosc / [16(n+1)] = 115200, amb n = 0 ens dóna que la freqüència del sistema seria de 1.843.200 Hz = 1,8432 MHz (*), que és superior a la trobada anteriorment. El meteix passa amb l'Eq 3.

(*) Per curiositat, la freqüència de 1,8432 MHz és la freqüència que s'utilitzaba en la UART dels primers equips IBM PC (veure xip UART 8250).

4) Calculeu el temps que es trigaria en fer la transmissió per USB a FS (12 Mbps) d'un paquet que conté **64 Bytes** de dades com el representat a sota si TOTS ells tenen valor 255 (és a dir 64*8 uns). I si fossin 64 bytes tots ells a zero? *(1 punt)*.

Format dels paquets



La codificació USB imposa que cada sis uns consecutius s'insereixi un zero (anomenat stuffed bit). Llavors el temps que es triga en transmetre depèn del nombre de uns consecutius de les dades; si tots el bits del camp data són igual a 1, tenim que hi han 64x8 uns. El nombre de zeros afegits és igual a (64x8)/6 = 85 (la part decimal s'ignora). Llavors tenim que

Temps a transmetre el camp DATA (tots 1) = (512+85 stuffed bits) / 12 MHz = 49,75 useg
Temps a transmetre el camp DATA (tots 0) = (512 bits) / 12 MHz = 42,66 useg

Nom i Cognoms: _____Possible Solució_____

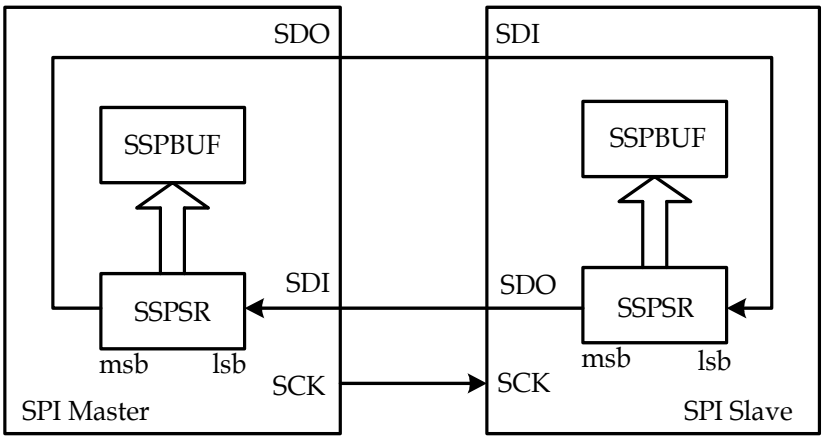
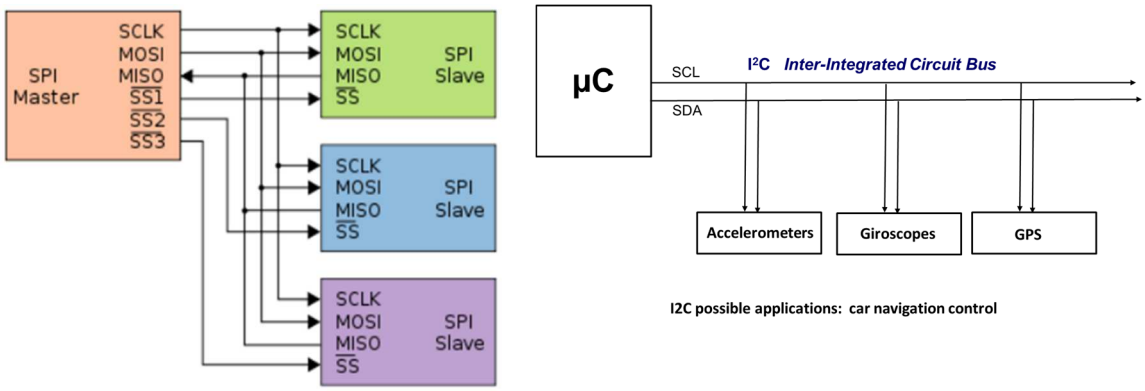
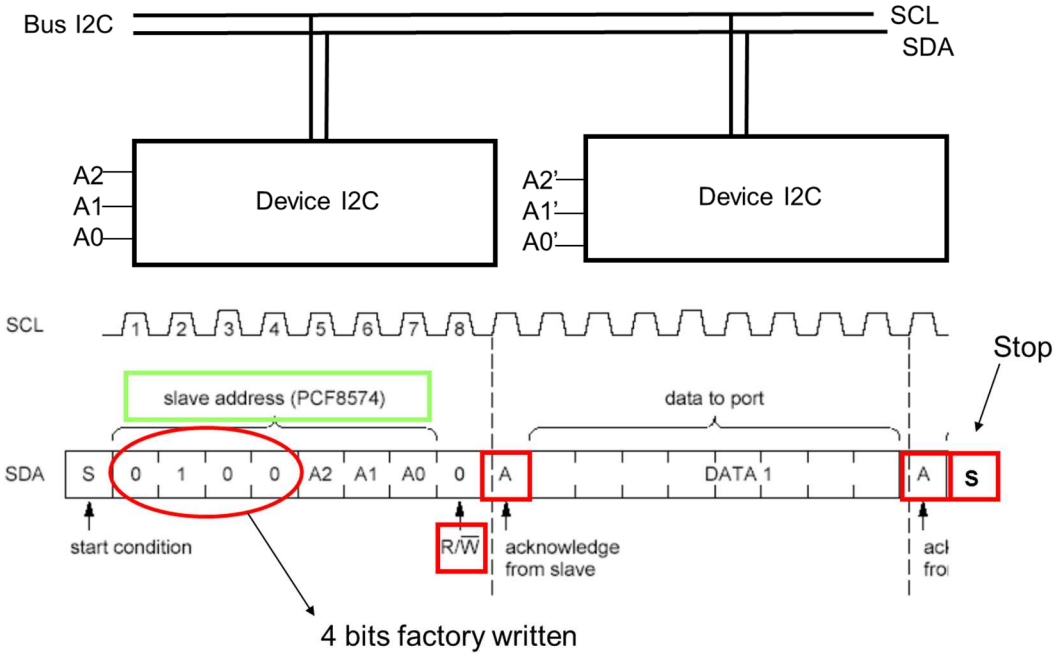


Figure 10.3 Connection between an SPI master and an SPI slave



SPI vs I2C SETUP

I²C *Inter-Integrated Circuit Bus*



Nom i Cognoms: _____Possible Solució_____

5) Tenim el conversor AD del PIC configurat amb una Vref- = 1V i Vref+ = 4V. Fosc és 8MHz, i el bit ADFM està a 1 (justificat a la dreta). Li posem un senyal a l'entrada (constant), el convertim (respectant les restriccions de Tad i Tacq) i obtenim als registres ADRESH:ADRESL el valor,

ADRESH : ADRESL
- - - - - - 01 00101111

Quína tensió analògica (en Volts) tenim a l'entrada del conversor AD? *(1,5 punts)*.

El valor trobat a ADRES és com sempre de 10 bits i com es veu, la part alta és a l'esquerra, per tant correspon al valor digital 303.
Tenim referències de +1 i +4 per tant el rang és 3V. Per tant, 3V correspon a 1024 valors digitals i cada unitat és 3V/1024=0,29mV

Així doncs, la tensió a l'entrada és: Vin= 1V (la referència-) + 303x0,29 = 1,887V

6) Volem enviar 240 Bytes per una línia sèrie configurada a 19200 bps, 8P1 (8 bits per byte, paritat parell i 1 bit de stop). Calcula, sense tenir en compte el software associat, ni el manegament o existència d'errors, el temps necessari per la seva transmissió *(1 punt)*.

Donat que tenim transmissió amb bit de paritat, per cada dada de 8 bits hem d'enviar-ne 11: 1 start + 8 data + 1 paritat + 1 stop.
Així doncs, el temps total serà de

240 Bytes x (11 bits / 1 Byte) x (1 segon/119200 bits) = 137,5 ms.

7) Indica si les afirmacions següents sobre el bus I2C són certes (**C**) o falses (**F**) *(1,5 punts, 0.25 per encert, -0,25 per error)*:

El bus I2C és diferencial. Per això té els cables D+ i D-.

Fals, la codificació és directa en tensió. Diferencial és per exemple USB.

El bus I2C està pensat per treballar dins d'un circuit imprès on hi comunica xips.

Cert, tal com el seu nom indica, InterIntegratedCircuit

El bus I2C és més ràpid que el bus SPI a l'hora de fer una transmissió de 8 bits.

Fals. I2C necessita indicar l'adreça de l'esclau a cada transmissió(+ start, ack..)

El bus I2C és asíncron. No té una línia de *clock* per marcar el temps de cada bit.

Fals. És síncron. La lína de clock és SCL.

Si en un bus I2C no funciona un dispositiu esclau, tots els altres fallaran també.

Fals, són independents entre ells, tenen adreça pròpia.

Un bus I2C pot tenir un màxim total de 7 dispositius connectats.

Fals, hi ha 7 bits per indicar l'adreça. Pot haver 8 màxim del mateix tipus.