

Nom i Cognoms: _____ **Una possible solució** _____

1) **Justifica** si són certes aquestes afirmacions sobre el bus SPI:

És síncron:

Cert, màster i esclaus comparteixen el senyal de clock SCK

És Half-duplex:

Fals, és Full-duplex. Podem transmetre i rebre alhora amb els senyals SDI i SDO.

2) Un enginyer de telecomunicacions pretén implementar una comunicació sèrie asíncrona per bit-banging utilitzant la placa EASYPIC6 del laboratori. Presenta la següent funció que implementa la recepció d'un caràcter de 1000 bits amb un bit de start, un de stop, sense paritat i a 19200 bps.

El senyal arriba per un pin d'entrada anomenat *PINX*, i la informació rebuda es guarda en una taula de bits, anomenada *buffer*, i declarada com a global.

```
void rebre_ls (void)
{
    int i;
    float T;

    inicialitza_port(); // configura PINX com a entrada i fa tot el que calgui
    T = 1.0/19200.0; // temps de bit
    while (PINX); // esperem bit de start (estat idle = '1')
    espera (1.5*T);
    for ( i=0; i<1000; i++)
    {
        Buffer[i] = PINX; // recepció bit a bit
        Espera (T);
    }
    espera (0.5*T); // bit de stop
}
```

Indiqueu si la recepció es faria de forma correcta, justificant els motius.

NO. Intentar transmetre tants bits de forma asíncrona d'aquesta manera és una bestiesa. La freqüència de transmissió mai serà exactament igual a la de recepció, i això provocarà que bit a bit, emissor i receptor es vagin dessincronitzant.

Hi ha diferents motius pels quals les freqüències de transmissió i receptor no siguin exactament iguals. P ex: en el codi no es té en compte el retard de les instruccions, l'error alhora de generar la base de temps (veure prob.3), derives dels oscil.ladors dels micros...

3) Calcular el valor que cal escriure en el registre SPBRG i BRGH per generar un *baudrate* de 12.200 bps si l'oscil·lador del sistema és de 16 MHz. Per fer-ho caldrà utilitzar les formules indicades a continuació:

a) $SPBRG = (F_{osc} / (16 \times \text{Baud rate})) - 1$, BRGH = 1 High Speed

b) $SPBRG = (F_{osc} / (64 \times \text{Baud rate})) - 1$, BRGH = 0 Low Speed

Utilitzant la formula a) dóna $SPBRG = [80,96] = 81$ i utilitzant la segona formula b) dóna $SPBRG = [19,49] = 19$.

Per poder decidir quina és la millor opció cal comparar l'error que es cometria en generar el *baudrate* en cada cas.

En el cas a) el *baudrate* seria de 12195 bps i en el segon cas b) 12500

L'error és menor en el primer cas. Llavors $SPBRG = 81$ i $BRGH = 1$

4) Per quins motius l'ample de banda màxim i efectiu de l'USB augmenta en passar de paquets de dades de 1 a 64 bytes de longitud? Per quin motiu aquest ample de banda disminueix en passar de paquets de dades de 256 a 1023 Bytes?

Table 6-2: Full-Speed Isochronous Bandwidth

Data Payload (Bytes)	Percentage of Frame Bandwidth/ Transfer	Max Xfers/ Frame	Maximum Bandwidth
1	1%	150	150KB/s
2	1%	136	272KB/s
4	1%	115	460KB/s
8	1%	88	704KB/s
16	2%	60	960KB/s
32	3%	36	1.152MB/s
64	5%	20	1.280MB/s
128	9%	10	1.280MB/s
256	18%	5	1.280MB/s
512	36%	2	1.024MB/s
1023	69%	1	1.023MB/s

En el primer cas, la velocitat efectiva augmenta doncs es treu un millor rendiment en transmetre més informació utilitzant el mateix *overhead* per paquet (CRC, permisos IN/OUT, etc.)

En el segon cas, la velocitat efectiva disminueix degut a la fragmentació del temps que imposen els *frames*; és a dir, un paquet no s'envia si queda partit entre dos *frames* (un *frame* comença cada 1 mseg). Llavors, en el cas que els paquets siguin molt grans no es podrà transmetre en el *frame* actual i s'haurà d'esperar al següent *frame*, perdent eficiència.

Nom i Cognoms: _____ **Una possible solució** _____

- 5) Es cert o fals que les transmissions de tipus *isochrones* en USB garanteixen el control d'errors i retransmissions però no l'ample de banda? I en les transmissions de tipus *bulk*?

Fals, en les tx. isòcrones l'ample de banda està assegurat. A l'assegurar l'ample de banda, no es poden fer retransmissions degudes a possibles errors. De fet en aquest tipus de transmissions no s'envia confirmació de recepció (ACK). S'utilitza per enviar informació que caduca amb la següent transmissió, p.ex. imatges de videoconferència. Cert, en les transferències *bulk* la prioritat és la integritat de les dades, no la velocitat. A cada paquet s'envia la confirmació, en cas d'errors el paquet es tornarà a transmetre.