Nom i Cognoms:_____Una possible solució_____

1) (2 punts)
Justifica si són certes aquestes afirmacions sobre el USB 2.0:

És síncron: FALS
Emissor i receptor no comparteixen senyal de rellotge. Cadascun té el seu.

És half-duplex: CERT
Les dades poden anar en qualsevol sentit, però no al mateix instant.

2) (1 punt)
Un micro amb un oscil.lador molt poc precís fa de master en una comunicació SPI. Degut a aquesta imprecisió, la freqüència del senyal de rellotge SCLK té una certa indeterminació. Concretament f_{SCLK} = 10Khz ±10%.
Si enviem trames de 512 bits, quantifiqueu quin percentatge de bits rebrà incorrectament el slave per causa d'aquesta indeterminació.

0%
Al tractar-se d'una comunicació síncrona, el slave rep el senyal SCLK del master. Per tant llegirà les dades de forma correcta encara que no sigui a una freqüència constant.

3) (2 punts)

pinta_pixel_GLCD (fila, col);

Encén el píxel que es troba en les coordenades esmentades en un LCD gràfic monocrom de 64x128 (0<=fila<=63, 0<=col<=127). La figura 2 ens mostra els eixos i origen de coordenades del GLCD.

posicio_touch_screen (&x, &y);

Obté la posició premuda en una pantalla tàctil ubicada sobre la GLCD. (0<=x<=255, 0<=y<=255). La figura 1 ens mostra els eixos i origen de coordenades de la pantalla tàctil.

calibra_touch_screen (&xmin, &ymin, &xmax, &ymax);

Demana a l'usuari que premi la pantalla tàctil en els seus extrems i retorna les coordenades del punts premuts (figura 1).

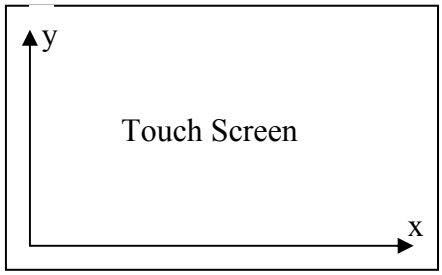

Figura 1

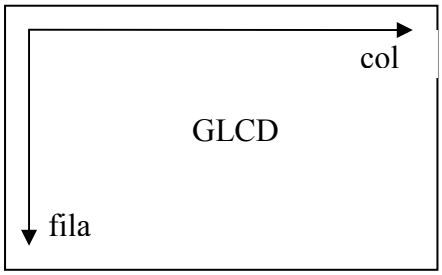

Figura 2

Es demana la expressió correcta de les variables pinta_ fila i pinta_col, per a que s'encengui el píxel just allà on ha premut l'usuari.

```
....
 calibra_touch_screen ( &xmin, &ymin, &xmax, &ymax);
....
posicio_touch_screen ( &x, &y);
....
....    // Aqui cal calcular el valor de pinta_fila i pinta_col
....
pinta_pixel_GLCD ( pinta_fila, pinta_col);
```

pinta_fila = 63 – 63 * (y – ymin) / (ymax – ymin)

pinta_col = 127 * (x –xmin) / (xmax – xmin)

Nom i Cognoms:_____Una possible solució_____

4) (2 punts)

Calcula quan trigaríem a enviar 12KB de dades (12x1024B), des del PIC18F4550 a un PC, si ho fem per una línia sèrie que està configurada a 19200bps, amb paraules de 8 bits, paritat parell i un bit de stop.

Cada transmissió serà: Start, 8 bits de dades, Paritat, Stop, 11 bits.

Enviarem: 12 KByte x 11 bits / Byte = 12x1024x11= 135168 bits.

Temps = 135168 bits / 19200 bits/segon = 7,04 segons.

5) (2 punts)

Mireu la següent rutina d'interrupció de recepció de línia sèrie del PIC 18F4550:

```
char buffer[];
int index;

#pragma code high_vector = 0x08
void high_interrupt (void)
{
        _asm
        goto  high_ISR;
        _endasm
}
#pragma code
#pragma interrupt high_ISR
void high_ISR (void)
{
        if (PIR1bits.RCIF){
                buffer[index++]=RCREG;
        }
}
```

La línia sèrie està configurada a 115200bps, 8 bits de dades, no paritat, 1 bit de stop, i NO hi ha cap control de fluxe.
Quin ha de ser el temps màxim entre dos serveis d'interrupció consecutius per evitar que hi hagi error d'*overrun* ? Raoneu la resposta.

L'overrun es produirà si arriba una nova dada abans de ser llegida l'anterior.

Cada dada transmessa (1 byte) implicarà l'enviament de 1 bit de start + 8 bits de dades + 1 bit de stop = 10 bits.
Per tant, arribarà una dada cada: 10 bits / 115200 bits/segon = 86,8 us.

Si no podem atendre una rutina d'interrupció de recepció de línia sèrie cada 86,8 us ens arribarà una dada sense haver llegit l'anterior. Encara que tinguéssim un buffer hardware o una FIFO d'algunes posicions, s'acabaria omplint també.

6) (1 punt)
Si volem configurar la línia sèrie a 2400bps, amb un PIC que té l'oscil.lador de 4MHz, hi ha alguna diferència entre configurar la línia sèrie amb BRGH en Low Speed o en High Speed?

SPBRG = (Fosc / (16 x Baud rate)) - 1, BRGH = 1 High Speed
SPBRG = (Fosc / (64 x Baud rate)) - 1, BRGH = 0 Low Speed

BRGH = 0

BAUD RATE (K)	Fosc = 4 MHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207
1.2	1.202	0.17	51
2.4	2.404	0.17	25
9.6	8.929	6.99	6
19.2	20.833	8.51	2
28.8	31.250	8.51	1
33.6	-	-	-
57.6	62.500	8.51	0
HIGH	0.244	-	255
LOW	62.500	-	0

BRGH = 1

BAUD RATE (K)	Fosc = 4 MHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-
1.2	1.202	0.17	207
2.4	2.404	0.17	103
9.6	9.615	0.16	25
19.2	19.231	0.16	12
28.8	27.798	3.55	8
33.6	35.714	6.29	6
57.6	62.500	8.51	3
HIGH	0.977	-	255
LOW	250.000	-	0

A les taules que ens dóna el fabricant (ja està tot calculat, no cal recalcular-ho per arribar a la mateixa conclusió o equivocar-se pel camí), veiem que en els dos casos:

- La velocitat real d'enviament serà de 2404 bauds.
- L'error de precisió comès serà del 0,17%

Per tant no hi haurà cap diferència entre fer servir el mode BRGH=0 amb el valor 25 al registre SPBRG o el mode BRGH=1 amb el valor 103 a SPBRG.