

Nom i Cognoms: _____

1) (2 punts)

Justifica si són certes aquestes afirmacions sobre el USB 2.0:

És síncron:

És half-duplex:

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 $f_{SCLK} = 10\text{Khz} \pm 10\%$.

Si enviem trames de 512 bits, quantifiqueu quin percentatge de bits rebrà incorrectament el slave per causa d'aquesta indeterminació.

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 ($0 \leq \text{fila} \leq 63$, $0 \leq \text{col} \leq 127$). 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. ($0 \leq x \leq 255$, $0 \leq y \leq 255$). 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 dels punts premuts (figura 1).

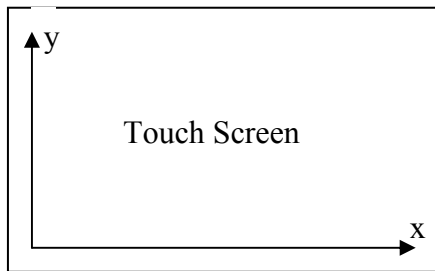


Figura 1

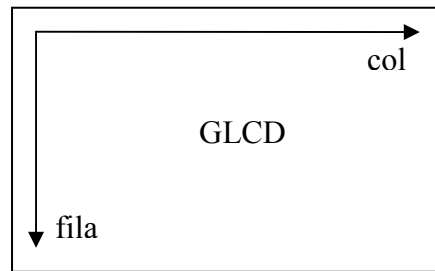


Figura 2

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);  
....  
    .... // Aquí cal calcular el valor de pinta_fila i pinta_col  
....  
    pinta_pixel_GLCD ( pinta_fila, pinta_col);
```

`pinta_fila =`

`pinta_col =`

Nom i Cognoms: _____

4) (2 punts)

Calcula quan trigariem 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.

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'*overflow* ? Raoneu la resposta.

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?

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

$SPBRG = (F_{osc} / (64 \times \text{Baud rate})) - 1$, 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