

Nom i Cognoms: _____ Possible Solució _____

- 1) Si tenim un microcontrolador amb un rellotge molt imprecís (per exemple fent servir un circuit RC per generar el *clock* en comptes d'un cristall de quars), i ens hem de comunicar amb un altre dispositiu, quin d'aquests dos protocols tindrà més problemes: SPI o UART?

Raona la resposta. (1,5 PUNTS)

La resposta correcta és que la UART tindrà molts més problemes de comunicació donat que es tracta d'un port de comunicació asíncron, és a dir, on emissor i receptor han de mantenir el seu propi rellotge de transmissió de dades. Si el rellotge del que disposem no és gaire precís probablement es doni la circumstància de que els rellotges en els dos sistemes es dessincronitzin i es provoquin errors de transmissió. En un SPI no es dona el cas donat que hi ha el senyal de rellotge que indica quan una dada ha estat enviada i pot ser llegida. Aquesta senyal de rellotge la genera el màster, es comparteix i es transporta d'un sistema a un altre fent servir una línia específica, i per tant no es produirien errors degut a aquesta limitació.

- 2) Amb una configuració sèrie de 9600 bauds, 8 bits/Byte, Paritat Senar, 1 bit d'Stop i sense tenir en compte el software, quant trigaries a enviar 128bytes? Justifica-ho. (1 PUNT)

Observem que per cada byte de dades que volem enviar en realitat estem enviat 11.

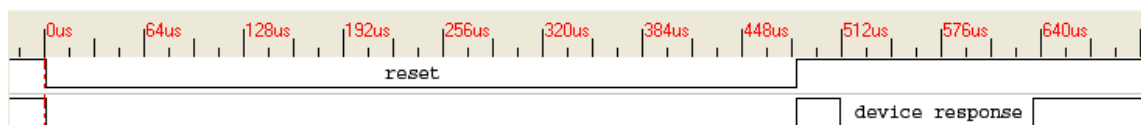
1 StartBit + 8 Dades + 1 Paritat + 1 Stop bit = 11 bits.

Es a dir, per transferir 128 bytes de dades hem d'enviar $128 * 11 \text{ bits} = 1408 \text{ bits}$.

Com la velocitat de transmissió es de 9600 bauds (bits/segon) tenim que el temps en que triguem en anviar 1408 bits serà $1408/9600 = 0,146 \text{ segons}$.

- 3) Fes una funció amb el codi necessari per detectar si a un pin del microcontrolador (suposem A0) s'hi ha connectat un dispositiu One-Wire (sols volem saber si s'ha connectat o no). A sota teniu el cronograma de descobriment de dispositius one-wire.

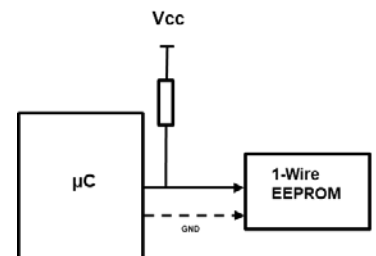
Explica bé la resposta. (2,5 PUNTS)



```
int One_Wire_Present ( void )
```

```
{
    TRISA0 = 0           // mode escritura (agafem línia)
    PORTA.RA0 = 0;       // fixem un 0
    Delay(480)           // delay 480 micros (reset a un possible slave)
    TRISA0 = 1           // alliberem la linea (torna a estat idle = 1)
    Delay(32)            // esperem 32 micros
    if (!PORTA.RA0)      // només tornem CERT si algu (l'slave) l'ha fixat a 0
        return(TRUE);
    return(FALSE);       // si la línia segueix en idle vol dir que l'slave no esta present.
}
```

```
// return 0 or 1
```

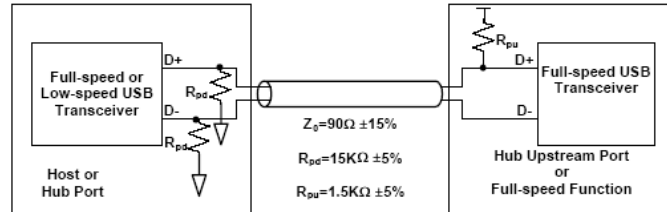


Nom i Cognoms: _____ Possible Solució _____

- 4) Per què serveix la resistència de Pull-Up del dispositiu (esclau) que es connecta al bus USB?
Justifica la resposta! (1 PUNT)

Té un doble propòsit:

Per una banda, al ser més petita que la RPD
ens assegura un nivell suficient per a que el
host detecti que s'ha connectat un dispositiu



slave. Per l'altra banda, al tenir la resistència a la línia D+, el host pot saber que el dispositiu connectat és de tipus Full-speed. Si tinguéssim la resistència a la línia D- el dispositiu seria low-speed.

- 5) Si amb un convertidor AD de 10 bits hem fet correctament una lectura i ADRES=0x034 quan a l'entrada hi havia 0.1 Volts i sabem que VREF- està connectat a Vss (0 Volts), quina tensió hi ha al pin VREF+? **Justifica la resposta. (2 PUNTS)**

Si el convertidor és de 10 bits sabem que el resultat de la conversió estarà entre 0 i $2^{10} - 1 = 1023$

Llavors sabem que $ADRES = 1023 * (0,1 - VREF-) / (VREF+ - VREF-)$

Si resolem trobem que $VREF+ = 1,97V$

- 6) Volem configurar una comunicació sèrie asíncrona a 115200 bauds, 8 bits de dades, sense paritat i un bit d'stop amb un PIC18F45k22 funcionant amb un clock de 4MHz.
- Quins valors de SYNC, BRGH, BRG16 i SPBRG triaríeu?
 - Quin error cometeu amb la configuració triada? **(2 PUNTS)**

Volem configurar l'UART de forma asíncrona, llavors **SYNC = 0**.

La configuració amb l'error més petit és **BRGH=1; BRG16=1;**

$n = (Fosc / (115200 * 4)) - 1 = 7,68 \rightarrow$ **SPBRG=8;**

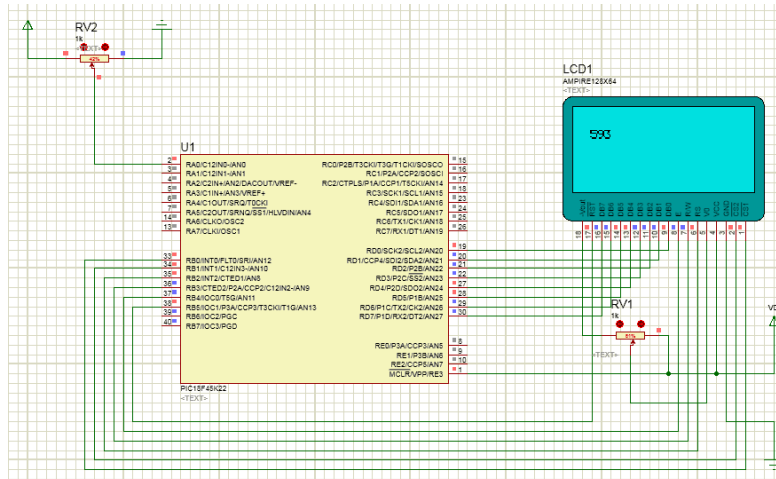
La velocitat real serà $Fosc / 4 * (8 + 1) = 111111$ bauds

L'error comès és $(111111 - 115200) / 115200 = -0,035 = -3,5\%$

Nom i Cognoms: _____ Possible Solució _____

Qüestions de laboratori NLAB3

A la pràctica del convertidor AD vas connectar un potenciòmetre al pin A0 i veure el seu valor a la pantalla GLCD. Respon breument les següents qüestions:



- Hauríeu pogut connectar el potenciòmetre a qualsevol pin del microcontrolador?
No, sols als que tenen la funcionalitat d'entrada analògica. En el manual se'ls identifica com a ANx a més del seu nom estàndard, per exemple RA0/AN0, RA1/AN1, etc.
- Quin registre vas tocar per configurar el PIN com a analògic?
El registre ANSELx fan la funció d'Analog Select pels pins. En el cas del pin 0 del port A hauríem de posar a 1 el bit 0 del registre ANSELA; Faríem: $ANSELA = 0x01$
- Quins eren els valors màxims i mínims (enter) que es pintaven a la GLCD?
El resultat de la conversió queda al registre ADRES, que és de 10 bits sense signe, per tant el valor anava de 0 a 1023.
- De quin ordre de magnitud era el TAD per bit que vas triar?
El TAD, que és el clock base per la màquina d'estats de l'AD ha de ser de l'ordre de us (microsegons), el manual ens diu que pot estar en 1 i 25 microsegons en condicions normals.
- En el vostre codi, cada quant temps llegíeu el valor del potenciòmetre per pintar-lo a la GLCD?
Depèn de l'estratègia seguida, amb fer-ho entre 1 i 10 vegades per segon (dins d'un bucle amb delay o amb la interrupció d'un Timer) és suficient i no satura d'escriptures la GLCD. Si volguéssim anar al límit (cada conversió d'AD, amb el TAD i TACK mínim) podríem arribar a saturar la GLCD i que no es veiés el resultat.