

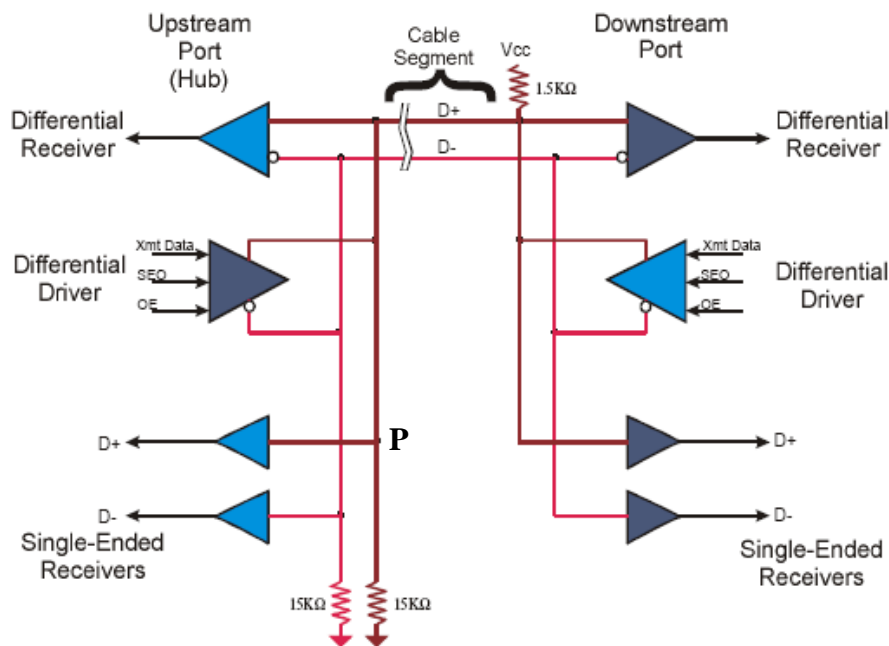
Nom i Cognoms: __Exemple de solució__

1) Aquest matí, els “niños de San Ildefonso” no saben si connectar els bombos a l'ordinador per I2C o per SPI. Com que van escasos de coure, volen triar la solució que gastí menys fils per connectar 2 bombos a un ordinador (que faria de master). Quants fils necessitaran en cada cas?

En I2C ens caldrà la línia de dades i adreces SDA i la línia de clock SCL; total 2 (a part del protocol podem considerar la referència a massa i l'alimentació).

En SPI ens caldran dos línies de selecció (una per esclau), el clock, les dades d'entrada i les dades de sortida; total 5 (a part del protocol podem considerar referència a massa i l'alimentació).

2) Mireu l'esquema de la figura.



2.1 Per a què serveix la resistència de 1.5KOhms connectada de D+ a Vcc a la part de dispositiu?

Serveix perquè el host USB pugui detectar la connexió d'un nou dispositiu (rebrà un flanc de pujada a D+) i alhora sabrà que és high-speed.

2.2 A la part del host, per a que serveixen els receptors Single-Ended de D+ i D- ?

Per detectar la connexió dels nous dispositius (flanc a un o altre) i per detectar l'estat final de paquet, on $D+ = D- = 0$, pel temps de dos bits.

bola extra: Quina tensió tindríem a l'entrada del connector Single-Ended (punt P) del host quan s'hi ha connectat un dispositiu com el de la figura?

A través de P, passarà una corrent de $I = V_{cc} / (15k + 1.5k)$, per tant la tensió serà $V_p = 15k / (15k + 1.5k) V_{cc}$. Si $V_{cc} = 5V$, $V_p = 4.54V$ o sigui un "1" lògic (hem suposat que l'impedància d'entrada al buffer de D+ és molt alta (cosa que és normal de suposar)).

3) Volem transmetre 10KB d'un equip A a un equip B, a través d'una línia sèrie configurada a 9600 bps, 8, N, 1. Com veieu, no fem servir bit de paritat i implementarem el següent protocol tipus *stop&wait* per verificar el correcte enviament de les dades:

- Les dades s'enviaran en paquets de 202 bytes, dels quals 200 seran les dades propiament dites i els dos darrers bytes seran la suma dels anteriors.
- El receptor comprovarà per cada paquet que la suma rebuda correspongui amb la de les dades rebudes i si és així enviarà un **ack**: 0xFF sinó, enviarà un **nack**: 0x00.
- Si ha rebut un **ack**, l'emissor enviarà el paquet següent i sinó repetirà l'últim.

3.1 Assumint que no hi haurà cap error en la comunicació, quant trigarem a enviar els 10KB d'informació?

Si tenim 10KB ens caldrà enviar $10 \cdot 1024 / 200 = 51,2$ paquets, és a dir 52.

Per cada paquet s'envien 202 bytes i es rebrà 1 byte de confirmació/error.

Cada byte que circuli necessitarà el temps de 10 bits: start+8dades+stop.

Per tant els bits totals que circularan seran:

$52 \text{ paquets} \cdot (202 + 1 \text{ bytes/paquet}) \cdot (10 \text{ bits/byte}) = 105560 \text{ bits}$

I el temps total sera:

$105560 \text{ bits} / 9600 \text{ bits/segon} = 10,995 \text{ segons}$

3.2 Si suposem que en promig hi ha un error de transmissió d'un bit per cada 20.000 bits enviats, quan trigarem a enviar els 10KB d'informació?

Si estadísticament falla 1 bit cada 20000 transmesos i cada paquet que enviem són 202 bytes $\cdot 10 \text{ bits/byte} = 2020 \text{ bits}$ podem assumir que fallarà un de cada 10 paquets, és a dir, un 10%. Els paquets totals seran ara 57 en comptes de 52, els bits totals seran 116116 i el temps anterior l'incrementem també en un 10% i ens dona 12,1 segons.