

Projet d'architecture des ordinateurs : UART RS-232

Universal Asynchronous Receiver Transmitter

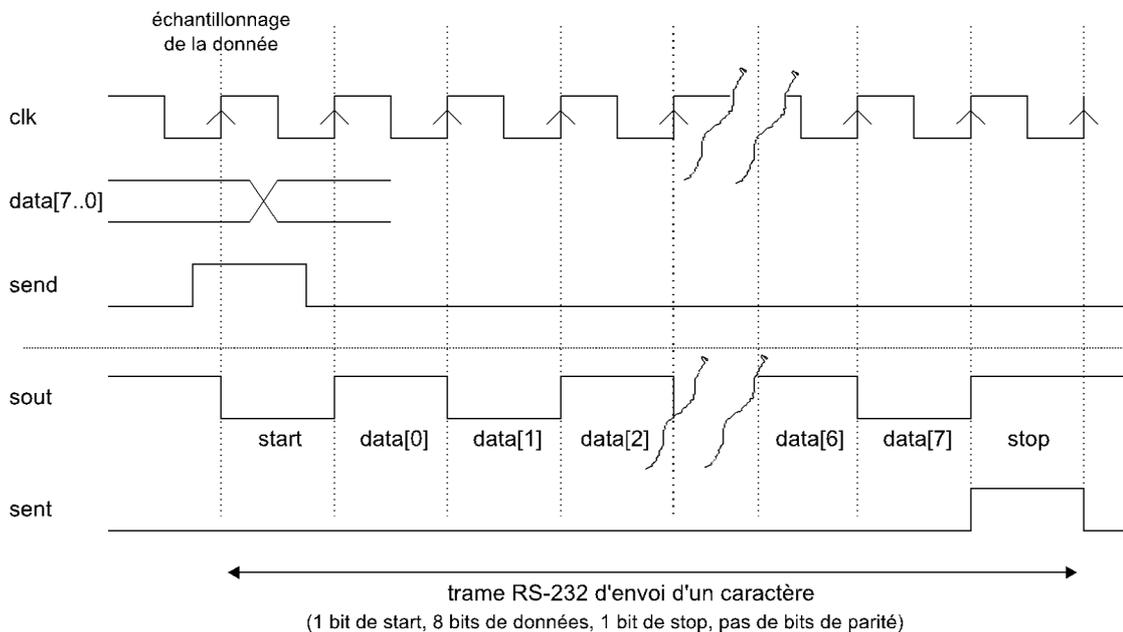
But du projet

- Construire un émetteur et un récepteur à la norme de transmission RS-232
- Utiliser ce module pour faire communiquer deux cartes entre elles

Norme de communication RS-232

La norme RS-232 est une norme de communication série simple et robuste, encore largement utilisée pour communiquer avec des équipements tels que des terminaux et des instruments de laboratoire.

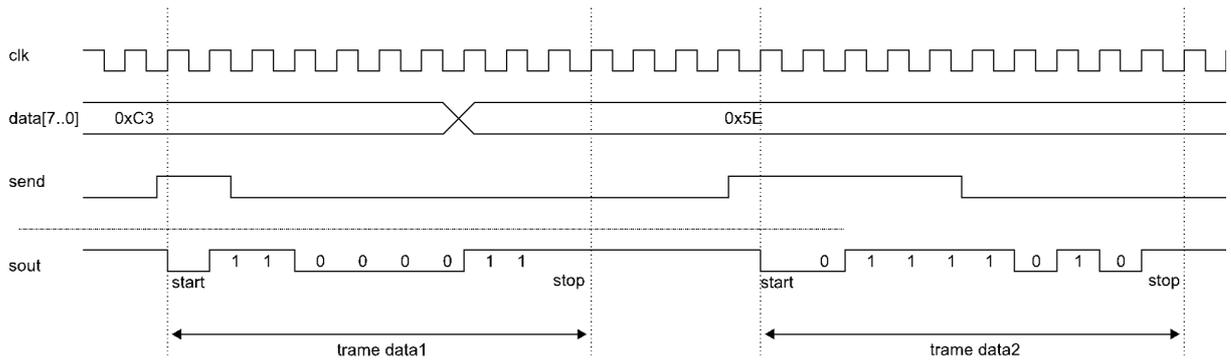
La chronologie précise de l'émission sur une ligne `out` d'un octet `data[7..0]` est donnée sur la figure suivante :



Remarques :

- La donnée à transmettre est échantillonnée au premier front montant de `clk` où `send=1`
- Un émetteur et un récepteur ne sont reliés entre eux que par le signal `out`; l'émetteur et le récepteur ont chacun leur propre horloge `clk`, de même fréquence approximative, mais pas nécessairement en phase. C'est pourquoi on parle de protocole asynchrone.

Le diagramme suivant montre la transmission de deux données successives data1 et data2 :



Le délai entre 2 trames peut être nul, le bit 'stop' d'une trame étant immédiatement suivi du bit 'start' de la suivante.

1. Émetteur

Le module émetteur à construire aura l'interface suivant :

```
module uart_em(rst, clk, data[7..0], send : sent, sout)
```

Il produit donc les signaux `sout` et `sent` avec la chronologie décrite sur les schémas précédents, en fonction de la donnée à transmettre `data[7..0]`.

2. Récepteur

Le module récepteur à construire aura l'interface suivant :

```
module uart_rec(rst, clk, sin : dout[7..0], rcvd)
```

Ce module doit reconstruire sur `dout[7..0]` la donnée entrante `sin` sérialisée par l'émetteur. L'horloge `clk` aura une fréquence quadruple de la fréquence de transmission, ce qui permettra de se recaler en phase par rapport à l'émission. Le signal `rcvd` passera à 1 durant 1 cycle d'horloge, lorsqu'une donnée aura été reçue. On ne gèrera pas les cas d'erreur.

3. Bataille navale

On reliera 2 cartes entre-elles pour leur faire échanger des octets :

- L'octet à envoyer sera placé sur les switches `sw[7..0]`, sa valeur sera affichée sur les 2 afficheurs 7-segments de gauche ; l'octet reçu sera affiché sur les 2 afficheurs 7-segments de droite
- L'horloge d'émission sera le bit 22 du module `clock28`, l'horloge de réception le bit 20 ; la transmission d'un octet durera environ 2s
- Le signal `send` d'envoi sera commandé par le bouton `btn[0]`
- La ligne d'émission sera la broche 1 du connecteur JA, notée `ja_out[0]`
- La ligne de réception sera la broche 1 du connecteur JC, notée `jc_in[0]`

On pourra relier une carte à elle-même ou relier 2 cartes entre-elles, comme sur les photos suivantes. Si on utilise 2 cartes, brancher les prises USB sur le même ordinateur pour qu'elles partagent les masses.

