

Etude de la liaison série RS232

Objectif

Interfacer un PC avec la carte Altera via la liaison série asynchrone RS232

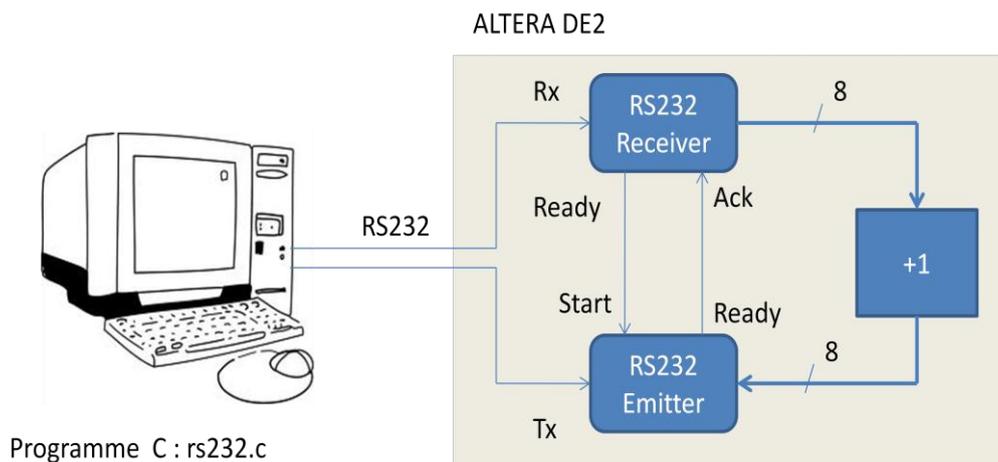
Le protocole RS232

Il existe sur internet une documentation fournie expliquant le fonctionnement de cette liaison série. Voir par exemple : <http://www.tavernier-c.com/serie.htm>

Mise en œuvre

Télécharger le fichier arcsys.lavenier.net/RS232/RS232.zip dans un répertoire, décompressez-le, puis ouvrez le projet TestRS232.bdf. Compilez également le programme C `rs232.c`.

Ce design comprend un exemple de mise en œuvre de l'interface RS232 à travers deux modules (*RS232Receiver* et *RS232Emitter*). Les octets reçus par le récepteur sont modifiés (incrémenté d'une unité) et ré-envoyés immédiatement vers l'émetteur. Un programme qui s'exécute sur le PC émet vers la liaison série les caractères frappés au clavier et récupère les données en provenance de cette même liaison avant de les afficher à l'écran.



Le module *RS232Receiver* détecte une transaction sur la ligne Rx. Le signal Ready est positionné à 0 pendant toute la transaction et passe à 1 dès qu'elle est terminée. Le module

RS232Receiver présente sur sa sortie l'octet reçu. Ce dernier est modifié (+1) puis envoyé au module RS232Emitter. L'émission débute en positionnant la commande Start à 1. Dans le cas présent, il correspond au signal Ready du module RS232Receiver. A partir de cet instant, le signal Ready du module RS232Emitter passe à 0 et y reste tant que la transaction est active. A la fin de la transmission, ce signal repasse à 1 et le signal à l'autre module qui se repositionne dans un état de réception.

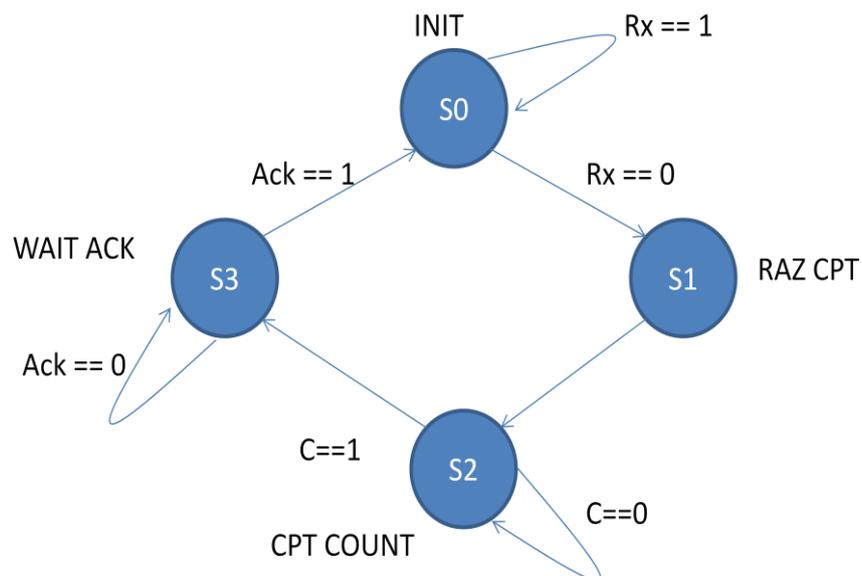
Le protocole RS232 implémenté possède les caractéristiques suivantes :

- Vitesse : 9600 bauds
- Données : 8 bits
- Parité : non

Une transaction correspond donc à l'émission d'un bit start (0) suivie de 8 bits de données (poids faible en tête).

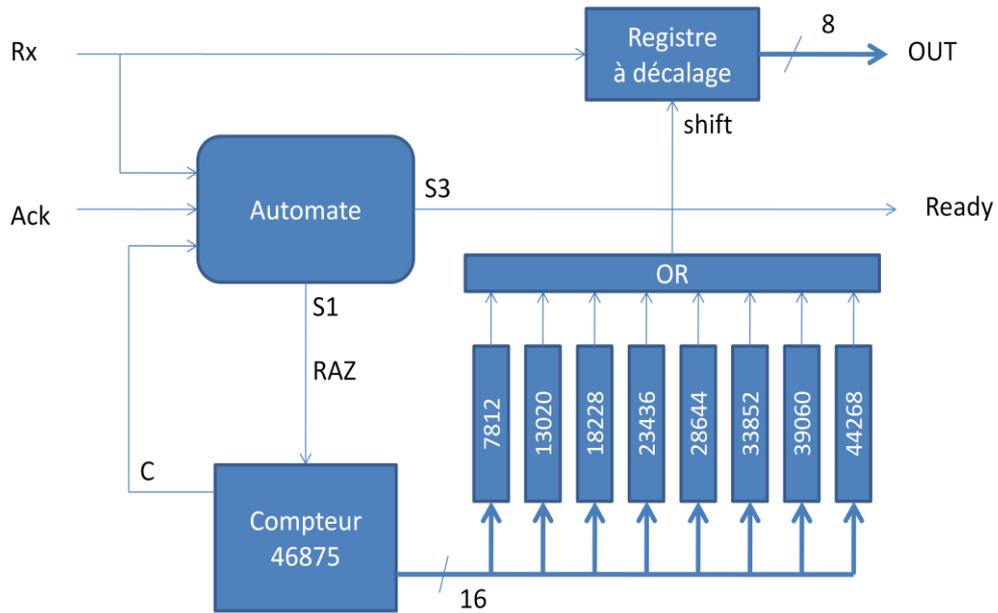
Module RS232Receiver

Le comportement de ce module peut être spécifié par l'automate suivant :

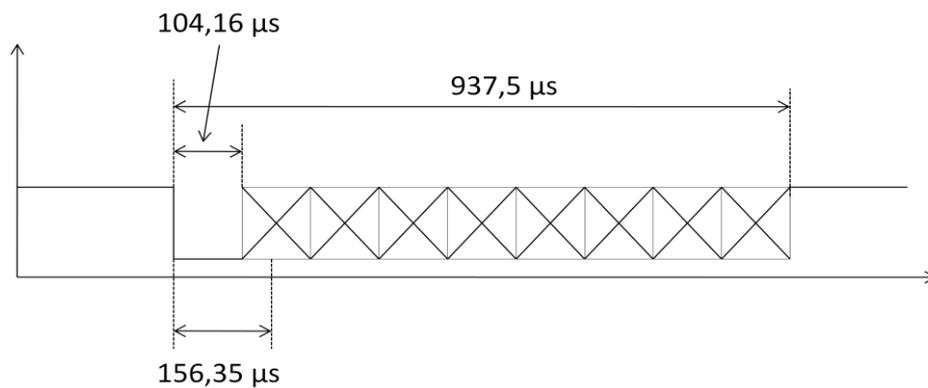


L'état S0 est l'état de repos : pas de transaction sur la ligne (le signal Rx est au niveau 1). Dès que Rx passe à 0, cela signifie qu'une donnée va être transmise. L'automate se positionne dans l'état S1 où il initialise un compteur à 0, puis gagne l'état S2 où le compteur est incrémenté au rythme de l'horloge. Ce compteur est un compteur modulo 46875, ce qui correspond à une période de 937 μ s avec une horloge à 50 MHz, où encore au temps d'une transaction de 9 bits (bit start + donnée sur 8 bits) sur la liaison RS232 (le signal C passe à un tous les 46875 cycles). L'automate reste dans l'état S2 tant que la transaction n'est pas terminée puis rejoint l'état S3 où il reste tant qu'un acquittement de la transaction n'a pas été réalisé (signal Ack).

Cet automate est couplé au dispositif suivant :

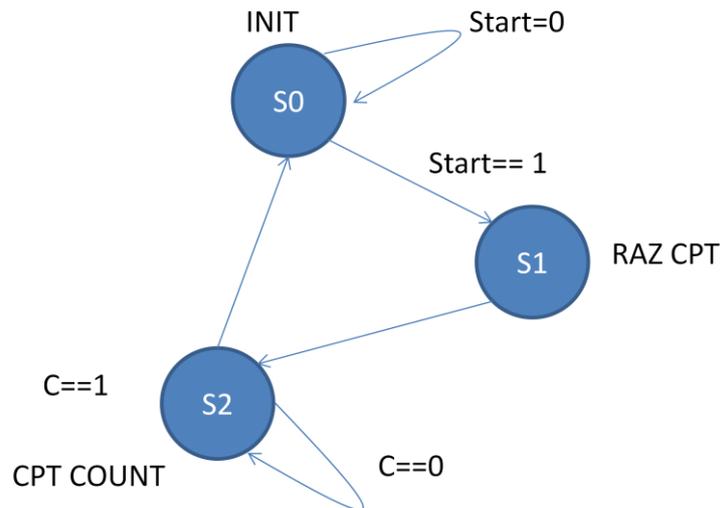


Rx est le signal en provenance de la liaison série. Il est connecté à l'automate (cf paragraphe précédent) et à un registre à décalage sur 8 bits. Ce registre à décalage est activé à des instants précis où l'information sur la liaison série est significative. Ici, l'échantillonnage est fait au milieu des bits 1, 2, 3, 4, 5, 6, 7, 8. Le bit 0 (bit start) n'est pas échantillonné. Ainsi, le premier bit de donnée est échantillonné au bout de 7812 cycles, soit après $156 \mu\text{s}$, ce qui correspond à un instant stable. Ces instants sont détectés avec des comparateurs.



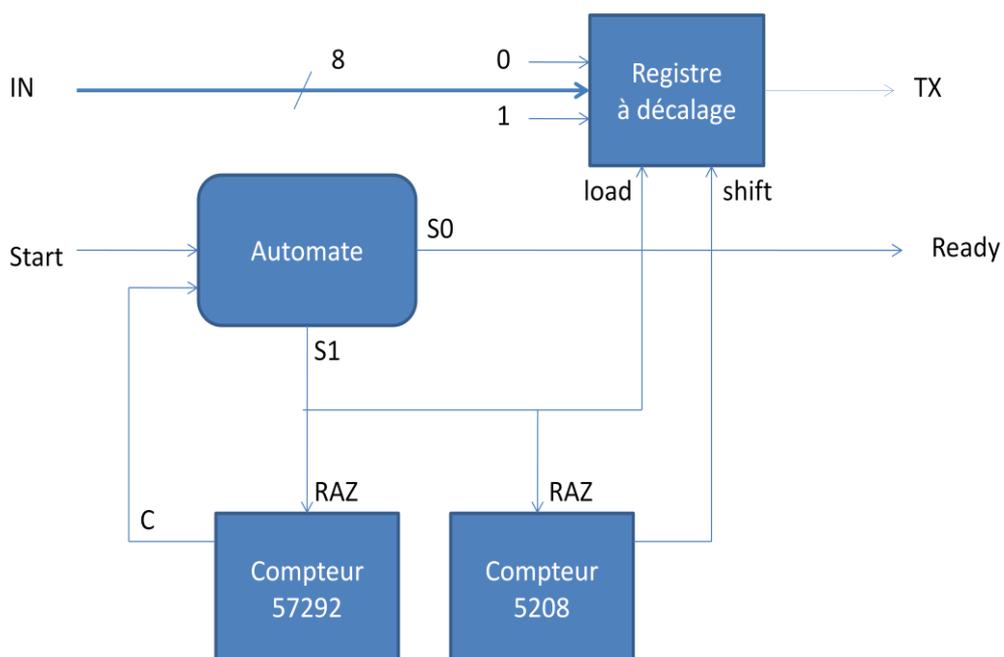
Module RS232Emitter

Le comportement de ce module peut être spécifié par l'automate suivant :



Le fonctionnement de cet automate est encore plus simple que celui du module RS232Receiver. L'état S0 est l'état de repos : pas de transaction sur la ligne. Le passage du signal Start de 0 à 1 provoque l'émission d'une donnée. L'automate se positionne alors dans l'état S1 où il initialise un compteur à 0, puis gagne immédiatement l'état S2 où le compteur est incrémenté au rythme de l'horloge. Ce compteur est un compteur modulo 57292, ce qui correspond au temps d'une transaction de 10 bits (bit start + donnée sur 8 bits + bit stop) sur la liaison RS232. L'automate reste dans l'état S2 tant que la transaction n'est pas terminée puis rejoint l'état S0.

L'automate est associé au dispositif suivant :



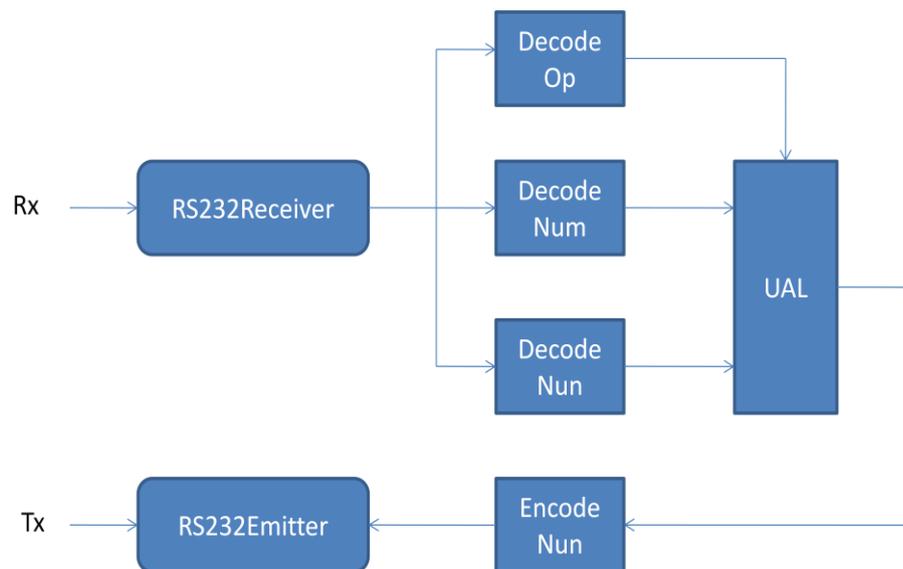
Le module reçoit en entrée un octet qui est chargé en parallèle dans un registre à décalage de 10 bits. Le bit 0 est positionné à 0 (bit Start), les bits 1 à 8 correspondent à l'information utile (octet à transmettre) et le bit 9 est positionné à 1. Le registre à décalage est activé (commande shift) toutes les 104 µs par un signal correspondant à la retenue sortante d'un compteur modulo 5208.

TP : Mini Calculette

Le but du TP est d'envoyer une chaîne de caractères représentant une opération arithmétique vers la carte et de récupérer une chaîne de caractères indiquant le résultat. Par exemple, l'envoi de la chaîne « 5 + 7 » doit retourner la chaîne « 12 ».

Pour simplifier, on considèrera seulement les opérations +, - et *. Les opérandes seront inférieurs à 1000 (3 chiffres maximum). On autorise aussi quelques modifications sur la chaîne de caractères à transmettre. En reprenant l'exemple précédent on pourra, par exemple, transmettre la chaîne « 005 007 + » qui a l'avantage d'avoir une taille constante et induire un décodage hardware plus simple.

Le schéma de principe de l'architecture à mettre en place est le suivant :



Les octets reçus du module RS232Receiver doivent être interprétés/décodés pour fournir à une unité arithmétique et logique (UAL) des opérandes sur N bits. De manière inverse, le résultat produit par l'UAL doit être encodé pour générer une chaîne d'octets. L'ensemble est, bien sûr, synchroniser par un automate (non représenté sur la figure).