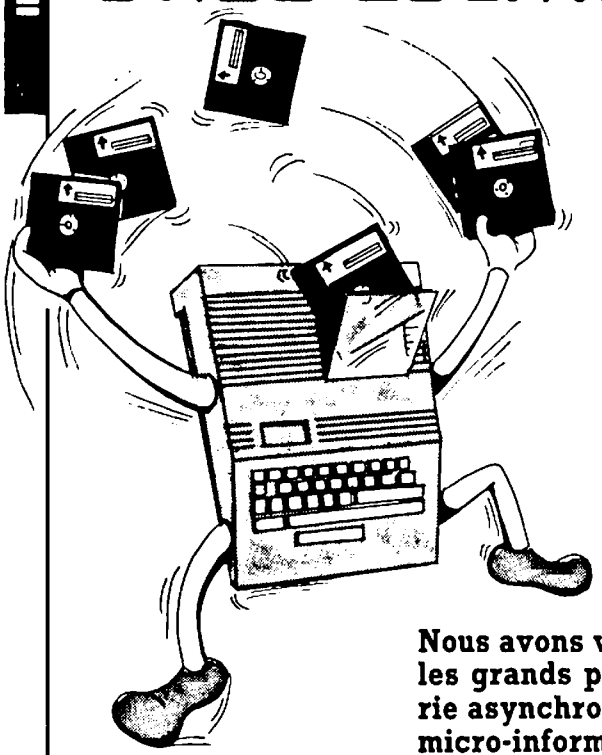
L'ABC DE LA MICRO-INFORMATIQUE



LES UART

Nous avons vu, le mois dernier, quels étaient les grands principes d'une transmission série asynchrone, transmission très utilisée en micro-informatique, tant amateur que professionnelle. Nous vous proposons aujourd'hui d'étudier la mise en œuvre du circuit typique d'interface série asynchrone : l'UART, ce qui signifie, rappelons-le, Universal Asynchronous Receiver Transmitter soit, en bon français, émetteur récepteur universel asynchrone.

L'étude d'un tel circuit dans le cadre d'une série comme la nôtre est possible sans devoir trop particulariser car, bien qu'étant produits par de très nombreux fabricants de circuits intégrés, il n'existe en fait que très peu d'UART différents. Le modèle que nous avons choisi est le plus répandu et existe chez au moins dix fabricants différents. Autant dire que c'est toujours lui, ou presque, que l'on rencontre.

PRESENTATION

Comme le montre la figure 1, et comme le laissait pressentir son appellation, un UART est en fait un circuit contenant deux parties distinctes : un émetteur asynchrone et un récepteur asynchrone.

Comme nous pouvons le constater, la partie émission est constituée principalement de deux registres : un registre de mémorisation des données à transmettre et un registre de transmission proprement dit. Un bloc baptisé logique de contrôle se charge de la commande de ces registres et du cadencement de la transmission à partir d'une horloge externe.

La partie réception est, elle aussi, constituée de deux registres analogues : un registre de réception des données et un registre de mémorisation des données. Une logique de contrôle se charge, là aussi, de la gestion des divers signaux de service.

Un cinquième registre, commun à l'émission et à la réception, est le registre dit de contrôle ou de commande. C'est grâce à lui que nous allons pouvoir programmer divers modes de fonctionnement de notre UART. Par ailleurs, et bien que cela ne se voie pas explicitement sur le synoptique de la figure 1, on peut considérer que certaines des informations disponibles en sortie du bloc logique de contrôle de la partie réception sortent en fait d'un registre d'état de l'UART, nous verrons pourquoi dans un instant.

Muni de tous ces éléments, que sait donc faire un tel circuit et quelles en sont les caractéristiques principales ? Nous allons vous en donner ci-après un bref aperçu :

– Les UART de ce type sont entièrement compatibles TTL sur toutes leurs pattes et sont alimentés sous une tension unique de 5 V. Des versions antérieures ont existé qui demandaient aussi une tension de – 12 V. On ne doit plus en trouver aujourd'hui sur le marché, même si certains revendeurs en ont encore en stock et cherchent à les écouler...

– Ils peuvent émettre et recevoir jusqu'à des vitesses de 64 000 bits par seconde, soit 6 400 caractères par seconde environ.

– La synchronisation des données reçues sur l'horloge de réception est évidemment interne et automatique.

– Ils savent générer automatiquement le bit de start, le ou les bits de stop et la parité paire, impaire ou inexistante.

– Ils peuvent émettre et recevoir des mots de 5, 6, 7 ou 8 bits.

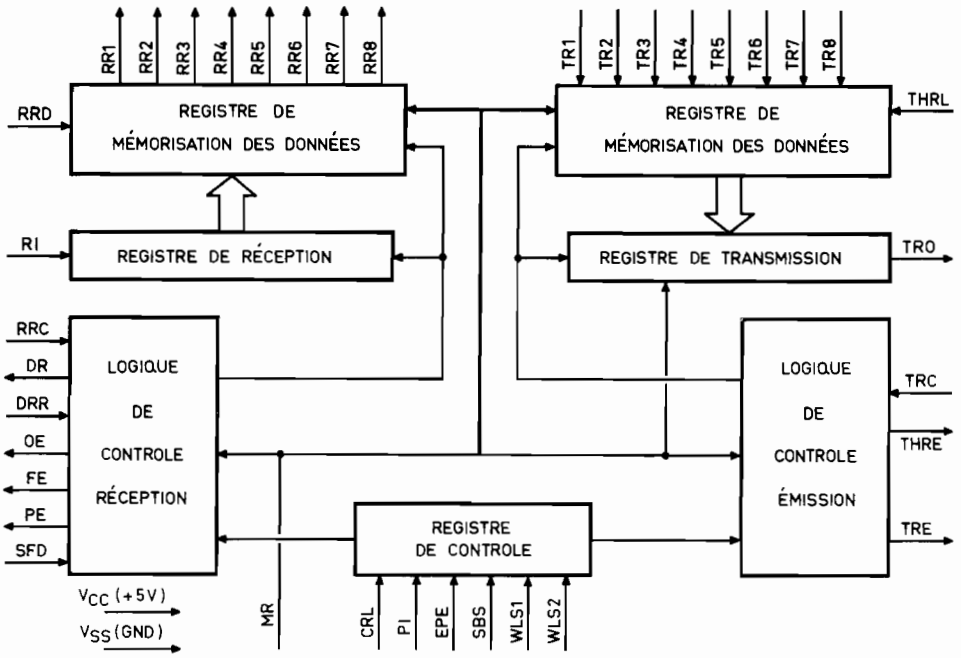


Fig. 1
Synoptique interne d'un UART.

– Ils peuvent fonctionner à des vitesses différentes en émission et en réception.
– Ils disposent d'une panoplie de signaux d'état très complète qui permettent à l'utilisateur de savoir comment se déroule la transmission en cours.
– Précisons pour en rester là avec ce survol des caractéristiques que ces circuits sont peu coûteux eu égard aux fonctions accomplies puisqu'on les trouve à moins de 80 F.

ANALYSE DETAILLEE

Maintenant que nous connaissons les grandes lignes du produit, nous pouvons nous intéresser de manière plus précise aux divers signaux disponibles, ce qui nous conduira tout naturellement à vous montrer comment il s'utilise. Pour ce faire, et bien que cela soit un peu fastidieux, nous allons passer en revue les noms et fonctions des diverses pattes du boîtier.

Commençons par les signaux de service communs aux parties émission et réception, car cela ira relativement vite. Nous avons :

– Une patte de masse appelée GND ou VSS selon les fabricants.
– Une patte d'alimentation positive à relier à toute tension comprise entre 4,75 V et 5,25 V (normes TTL) et baptisée généralement VCC. Le circuit, réalisé en technologie N.MOS dans la majorité des cas, consomme environ 30 mA.
– Une patte de RESET ou de remise à zéro baptisée généralement RESET ou MR (Master Reset). Le fait de mettre cette patte au niveau 1 remet à zéro les registres de réception et de transmission, met les signaux de contrôle à l'état inactif et remet à zéro les lignes d'état. Son utilisation n'a rien d'obligatoire. Cette ligne doit être reliée à la masse si elle n'est pas utilisée.

Tous les autres signaux présents sur le boîtier appartiennent en propre soit au sous-ensemble émission, soit au sous-ensemble réception. Commençons par ce dernier qui est le plus simple d'emploi.

Les lignes RR1 à RR8 (Receiver Register) sont les lignes de réception de données sur lesquelles on trouve le caractère reçu sous forme série. RR8 est le bit de poids fort et RR1 celui de poids faible. Ces lignes sont en logique trois états et sont sous le contrôle de RRD (Receiver Register Disconnect). Si RRD est au niveau bas, RR1 à RR8 fonctionnent normalement. Si RRD est au niveau haut, les lignes RR1 à RR8 sont placées en état haute impédance et peuvent donc être considérées comme isolées du circuit. Une telle utilisation est intéressante lorsque l'UART est connecté sur le bus d'un microprocesseur par exemple.

La ligne RI (Receiver Input) est l'entrée des données sous forme série.

La ligne RRC (Receiver Register Clock) est l'entrée d'horloge de réception. Cette ligne doit recevoir un signal carré de fréquence égale à 16 fois la fréquence de transmission exprimée en bits par seconde.

Ainsi, pour une transmission à 300 bauds ou 300 bits par seconde, une horloge à 16 x 300, soit 4 800 Hz, doit être utilisée. Ce facteur 16 est dû au processus de resynchronisation de l'horloge sur les données reçues qui a lieu dans l'UART.

La ligne DR (Data Received) indique, lorsqu'elle est au niveau haut, qu'un caractère complet a été reçu et est disponible dans le registre de mémorisation des données reçues, c'est-à-dire encore sur RR1 à RR8.

Pour remettre DR à zéro, la ligne DRR (DR Reset) est là. Il suffit de lui appliquer un niveau bas pour remettre DR à zéro. Cela se fait lorsque les données présentes sur RR1 à RR8 ont été lues par leur destinataire.

La partie réception de l'UART pourrait se suffire de ces signaux puisque l'on dispose de tout ce qu'il faut pour recevoir des caractères sous forme série, mais, comme nous vous l'avons dit ci-avant, un certain nombre d'informations relatives à la qualité de la liaison peuvent nous être fournies par l'UART. Libre à vous de les utiliser ou non, bien sûr.

Ces signaux sont au nombre de trois :

– PE (Parity Error) qui, lorsqu'il passe au niveau haut, indique une erreur de parité sur le caractère reçu. Signalons à ce propos que cette erreur ne signifie pas nécessairement une erreur de transmission ; en effet, il suffit que vous soyez en présence d'une transmission en parité paire alors que vous avez sélectionné par erreur sur votre UART une parité impaire pour qu'il vous indique erreur de parité alors que tout se passe bien. Précisons que quel que soit l'état de ce signal, le caractère reçu est quand même disponible sur RR1 à RR8. L'UART ne prend aucune décision à votre place et vous fournit, quoi qu'il advienne, ce qu'il a reçu.

– FE (Framing Error) qui, lorsqu'il passe au niveau haut, indique une erreur de format. Cette erreur peut avoir plusieurs causes : le nombre de bits reçus ne correspond pas à celui attendu, le nombre de bits de stop reçus ne correspond pas à celui attendu ou encore le caractère reçu n'est pas conforme du tout à ce que l'on rencontre normalement sur une liaison série asynchrone. Ici encore et malgré la présence de cette erreur, l'UART fournit « ce qu'il a compris » sur ses sorties RR1 à RR8. Dans les deux premiers cas vus ci-avant, la donnée reçue peut être correcte. Dans le dernier cas, c'est plus aléatoire bien évidemment.

– OE (Overrun Error) qui, lorsqu'il passe au niveau haut, indique une erreur de recouvrement ou de survitesse. Cette erreur se produit lorsque l'utilisateur de l'UART ne vient pas lire assez vite les données présentes sur RR1 à RR8 et ne remet donc pas à zéro en conséquence le signal DR par la ligne DRR. Dans ces conditions, les caractères successifs reçus s'écrasent les uns sur les autres dans le registre de réception et conduisent à des données qui n'ont plus aucun sens sur RR1 à RR8. Des trois erreurs décelables au niveau de l'ART, celle-ci est la plus grave car, lorsqu'elle se produit, on est certain que les données reçues sont réellement mauvaises.

Comme pour les lignes RR1 à RR8, il est possible de faire passer OE, FE et PE en haute impédance grâce à la ligne SFD (Status Flag Disconnect). Il suffit pour cela de mettre SFD au niveau haut.

La partie émission ne comprend pas plus de signaux et, comme vous allez le constater, ceux-ci sont, en quelque sorte, la réciproque de ceux que nous venons de voir pour la réception.

Les lignes TR1 à TR8 (Transmitter Register) sont les entrées de données parallèles sur lesquelles doit être appliqué le mot à émettre. TR8 est le bit de poids fort et TR1 le bit de poids faible. Le mot appliqué sur ces lignes ne peut rentrer dans le registre de mémorisation des données à transmettre que si la ligne THRL (Transmitter Holding Register Load) est maintenue au niveau bas. En outre, lorsque cette ligne passe au niveau haut, les données contenues dans le registre de mémorisation des données à émettre sont transférées dans le registre d'émission. Si le registre d'émission est déjà occupé car il est en train d'émettre un caractère, le transfert entre le registre de mémorisation et celui d'émission est mis en attente, ce qui ne génère pas d'erreur au sein du circuit. L'utilisateur doit, par contre, éviter de présenter trop vite une nouvelle donnée sur TR1 à TR8 et il lui faut pour cela tester l'état de signaux dont nous allons voir le rôle maintenant.

Lorsque le registre de mémorisation des données est vide et, donc, lorsqu'il peut accepter un nouveau caractère, l'UART le signale au moyen de la ligne THRE (Transmitter Holding Register Empty) qui passe au niveau haut. Lorsque c'est le registre d'émission par contre qui est vide, c'est à la ligne TRE (Transmitter Register Empty) de passer au niveau haut pour le signaler.

Deux autres lignes sont plus « classiques » en ce qui concerne cette partie émission : TRO (Transmitter Register Output) qui est la sortie des données sous forme série et TRC (Transmitter Register Clock) qui est l'entrée d'horloge d'émission. Comme pour la partie réception de l'UART, cette dernière doit être à une fréquence égale à 16 fois la fréquence de transmission exprimée en bauds ou bits par seconde. Rappelons que, comme l'UART sait émettre et recevoir à des vitesses différentes, TRC peut être à une fréquence totalement différente de RRC.

Les autres lignes disponibles côté émission servent à fixer le format de transmission choisi parmi les nombreuses possibilités offertes par les diverses normes relatives aux transmissions séries asynchrones. Nous y trouvons :

– PI (Parity Inhibit) qui, lorsqu'elle est mise au niveau haut, interdit la génération d'un bit de parité. Dans ces conditions, le ou les bits de stop suivent immédiatement le dernier bit du caractère transmis.

– EPE (Even Parity Enable) qui, lorsqu'elle est mise au niveau haut, fait générer une parité paire alors que cette dernière est impaire lorsque EPE est au niveau bas. Cette génération est évidemment soumise à l'autorisation de PI. Si PI interdit la parité, l'état de EPE est sans importance.

– SBS (Stop Bit Select) permet de choisir le nombre de bits de stop. Si SBS est au niveau haut, chaque caractère est suivi par 2 bits de stop alors qu'il n'y en a qu'un dans le cas contraire. En outre, sur certains UART de haut de gamme, le fait de sélectionner une transmission à 5 bits (voir ci-après) et de mettre SBS au niveau haut fait générer 1 bit de stop 1/2, ce qui est utile pour certains standards de transmission anciens mais que l'on rencontre encore quelquefois.

– WLS2 et WLS1 (Word Length Select 2 et 1) sélectionnent le nombre de bits transmis pour chaque caractère compte tenu des indications du tableau de la figure 2. Il est évident que, si vous désirez transmettre du code ASCII, les deux premières options vous sont interdites car il faut impérativement disposer de 7 bits pour ce dernier. Le standard 6 bits ne se rencontre d'ailleurs quasiment jamais alors que le 5 bits ne se trouve plus guère que sur les vieux téléimprimeurs travaillant en code Baudot.

WSL2	WSL1	Longueur du mot
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

Fig. 2. – Les divers formats de mots autorisés.

Toutes les informations appliquées sur ces lignes de contrôle ne sont prises en compte par le circuit que si la ligne CRL (Control Register Load) est mise au niveau haut, en permanence ou temporairement.

Précisons, en outre, que la partie réception de l'UART ex-

ploite aussi ces informations. En particulier elle considére que le format des caractères reçus doit être conforme à celui des caractères émis, donc à ce qui a été sélectionné au moyen de ces lignes. Ce n'est pas très contraignant car on utilise rarement deux formats de transmission différents sur une même liaison série.

QUELQUES COMMENTAIRES

Comme nous vous l'avons expliqué en début d'article, les explications que nous venons de vous donner sont valables pour un très grand nombre d'UART de fabricants différents. On peut citer, sans que la liste qui suit soit limitative : les TR1863 et TR1865 de Western Digital, les COM 8017 et COM 8502 de SMC, l'AY 3 1015 de General Instruments, etc.

Bien que ces circuits aient des fonctions et des brochages identiques au point d'être interchangeables sans nécessiter aucune modification de câblage, leurs fabricants ne donnent pas toujours à leurs pattes les mêmes noms que ceux que nous avons utilisés ci-avant. Afin de vous permettre de faire le rapprochement, vous trouverez en figure 3 le brochage de tous ces circuits sur lequel figurent les noms que nous avons utilisés dans cette étude (noms utilisés sur la fiche du TR 1883 de Western Digital).

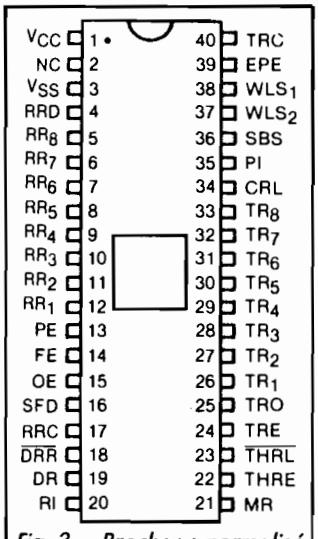


Fig. 3. – Brochage normalisé des UART étudiés dans cet article.

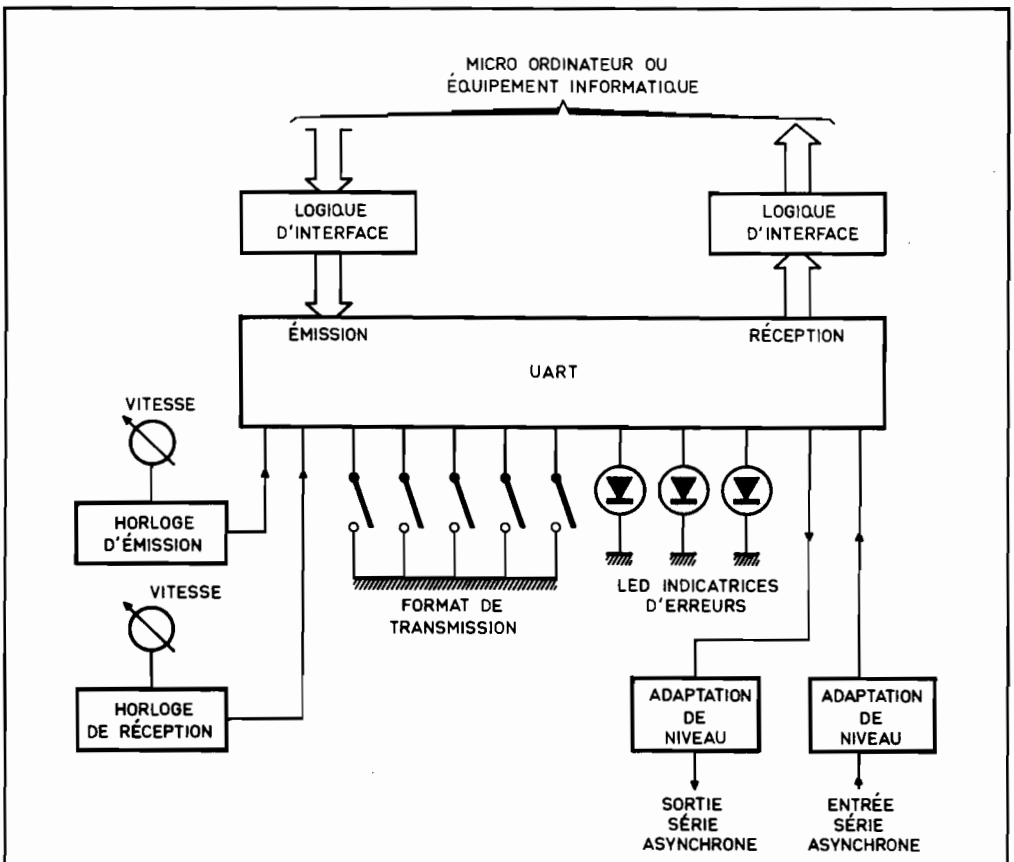


Fig. 4. – Synoptique de mise en œuvre d'un UART.

L'UTILISATION D'UN UART

Comme vous avez pu le constater à la lecture de ce qui précède, et bien que l'UART accomplisse l'essentiel des tâches nécessaires sur une liaison série asynchrone, un certain nombre de signaux logiques doivent être gérés par le montage auquel il est couplé.

La figure 4 est un exemple d'utilisation très schématique qui n'est là que pour rappeler l'essentiel. Nous y voyons, en particulier :

– Deux générateurs d'horloges, un pour l'émission et un pour la réception. En principe, ces générateurs peuvent produire diverses fréquences afin de permettre à l'UART d'accepter les diverses vitesses de transmission normalisées que l'on peut rencontrer sur des liaisons séries asynchrones.

– Un bloc d'interrupteurs qui permettent à l'utilisateur de sélectionner un format de transmission parmi ceux autorisés par l'UART.

– Un jeu de voyants ou diodes électroluminescentes qui signalent les éventuelles erreurs de réception.

– Une circuiterie logique, tant à l'émission qu'à la réception, qui permet d'interfacer l'UART avec le micro-ordinateur ou l'équipement informatique chargé de le piloter.

– Une circuiterie d'adaptation de niveau qui transforme les signaux TTL émis par l'UART en signaux à un niveau différent et, réciproquement, qui adapte les niveaux des signaux reçus en niveaux TTL compris par l'UART.

Ces deux derniers points nécessitent un développement assez long et, comme nous avons déjà utilisé pas mal de papier ce mois-ci, nous y consacrerons notre prochain article.

C. TAVERNIER