Interface de communication Série
ACIA-6850 de Motorola

De nos jours, il existe un très grand nombre de protocoles de communication en série et d'interfaces matérielles disponibles dans l'industrie électronique. La plupart d'entre eux sont axés sur les largeurs de bande de transmission élevées, comme les normes, plus récentes, USB 2.0 et 3.0. Certaines de ces normes proviennent du passé, mais elles sont encore répandues, en particulier comme interfaces de communication entre des modules sur la même carte. L'une d'elles est l'USART (Universal Synchronous/Asynchronous Receiver/Transmitter).
En général, tous les microcontrôleurs fournissent au moins un périphérique UART. Presque tous les microcontrôleurs STM32, Atmel et PIC de microchip fournissent au moins deux interfaces UART/USART. Exemple, le microcontrôleur STM32F4VGT6 qui équipe la carte STM32F4DISCOVERY fournit jusqu'à 4 USARTs et 2 UARTs.

6.1 Introduction aux UARTs et USARTs

Une interface UART/USART est un dispositif qui traduit une séquence parallèle de bits (généralement regroupés en un octet) en un flux continu de signaux circulant sur un seul fil. Lorsque l'information circule entre deux dispositifs à l'intérieur d'un canal commun, les deux dispositifs (ici, on se référera à eux comme émetteur et récepteur) doivent être d'accord sur le timing, c-à-d, le temps qu'il faut pour transmettre chaque bit individuel d'information. Dans une transmission synchrone, l'émetteur et le récepteur partagent une horloge commune générée par l'un d'entre eux (habituellement le dispositif qui sert de maître). Sur la Figure 1, nous avons un chronogramme typique. montrant le dispositif A envoyant un octet (01101001) en série au dispositif B en utilisant une horloge de référence commune. L'horloge commune est également utilisée pour convenir de l'instant où commencer l'échantillonnage de la séquence de bits : lorsque le dispositif principal commence à envoyer l'horloge sur la ligne dédiée, cela signifie qu'il va envoyer une séquence de bits. Dans une transmission synchrone, la vitesse et la durée de transmission sont définies par l'horloge : sa fréquence détermine la rapidité avec laquelle on peut transmettre un seul octet sur le canal de communication. Mais si les deux dispositifs impliqués dans la transmission de données conviennent de la durée nécessaire pour transmettre un seul bit et quand commencer et finir d'échantillonner les bits transmis, alors nous pouvons éviter d'utiliser une ligne d'horloge dédiée. Dans ce cas, nous avons une transmission asynchrone. La Figure 2 montre le chronogramme d'une transmission asynchrone. L'état inactif (c'est-à-dire qu'aucune transmission ne se produit) est représenté par le signal à l'état haut. La transmission commence par un bit START, représenté par le niveau bas. Le front descendant est détecté par le récepteur et une période de 1,5 bit après cela (indiquée sur la figure par T1.5bit),

l'échantillonnage des bits commence. Huit bits de données sont échantillonnés. Le bit le moins significatif (LSB) est généralement transmis en premier. Un bit de parité optionnel est ensuite transmis (pour la vérification des erreurs des bits de données). Souvent, ce bit est omis si le canal de transmission est supposé être sans bruit ou s'il y a une vérification d'erreur plus haut dans les couches de protocole. La transmission est terminée par un bit STOP, qui dure 1,5 bits.

Une interface USART (Universal Synchronous Receiver/Transmitter) est un dispositif capable de transmettre un message de données en série à l'aide de deux E/S, l'une servant d'émetteur (TX) et l'autre de récepteur (RX), plus une E/S supplémentaire comme ligne d'horloge, tandis qu'une interface UART (Universal Asynchronous Receiver/Transmitter) utilise uniquement deux E/S TX/RX (voir la Figure 3).

La présence d'une ligne d'horloge dédiée ou d'un accord commun sur la fréquence de transmission ne garantit pas que le récepteur d'un flux d'octets puisse les traiter au même taux de transmission du maître. Pour cette raison, certaines normes de communication, comme RS232 et RS485, offrent la possibilité d'utiliser des lignes dédiées au contrôle de flux matériel (Hardware Flow Control). Par exemple, deux périphériques qui communiquent à l'aide de l'interface RS232 peuvent partager deux lignes supplémentaires, nommées Request To Send (RTS) et Clear To Send (CTS) : l'émetteur positionne son RTS, ce qui indique au destinataire de commencer à surveiller sa ligne d'entrée de données. Lorsqu'il est prêt pour les données, le récepteur positionnera sa ligne complémentaire, CTS, qui indique à l'émetteur de commencer à envoyer des données et pour que l'émetteur commence à surveiller la ligne de sortie de données de l'esclave

Interface de communication série

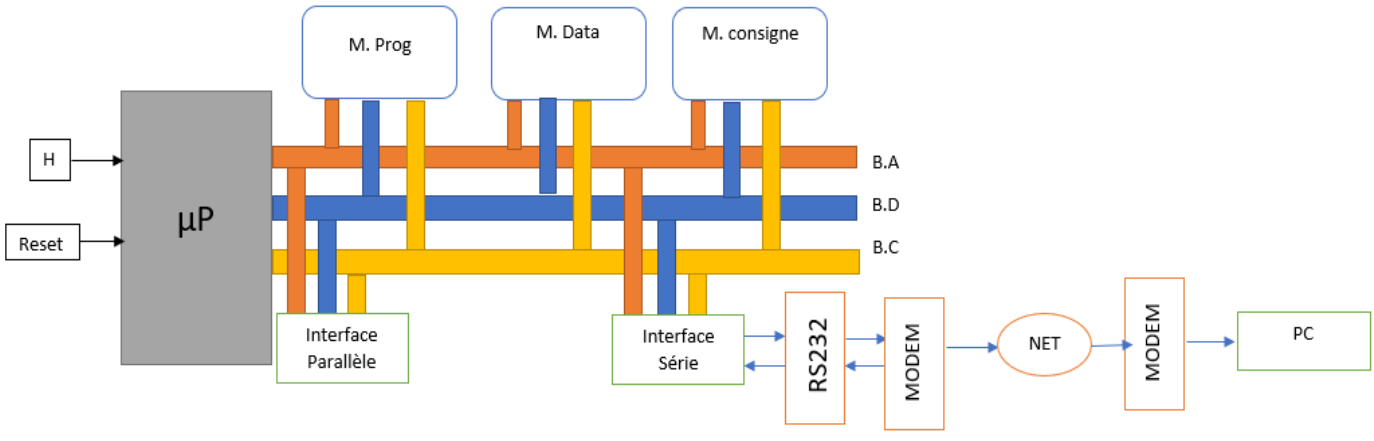


Figure : Interface de communication série

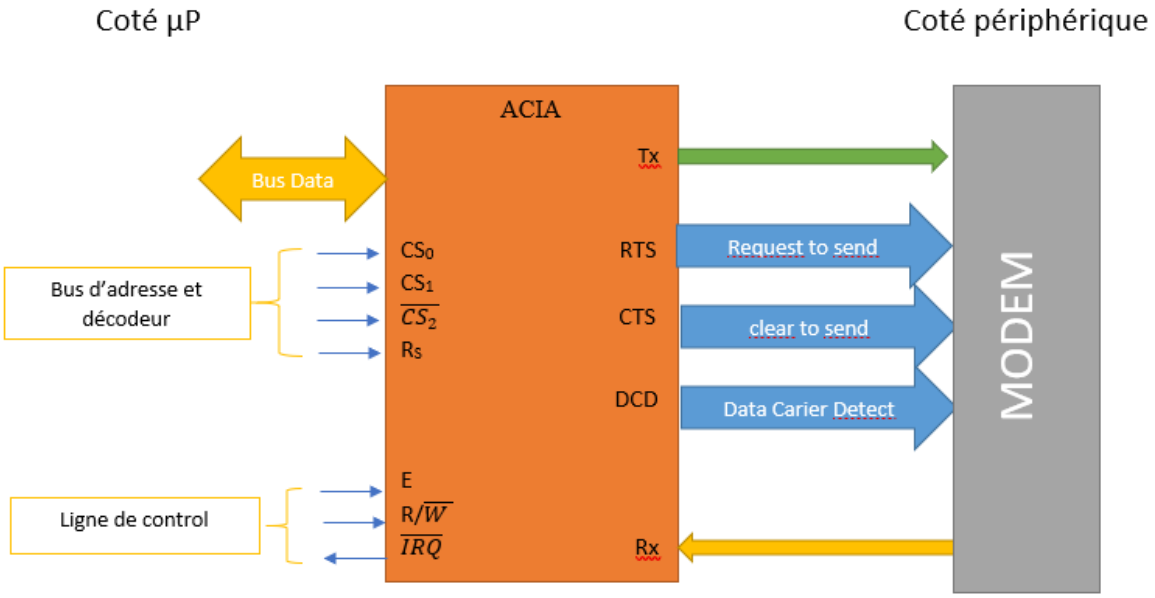
- L'interface de communication série nous permet de développer des applications comme :
  - Super vision industrielle.
  - SCADA (System Control and Data Acquisition).
  - Réseaux de capteurs sans fils (WSN).
  - Internet des objets (LOT).

- Le Tele control via wireless.
- Les capteursintelligent.

- L'ACIA 6850 de Motorola est une interface de communication série il permet de réaliser des liaisons séries entre le périphérique et le µP. Il reçoit du µP des données en parallèle et il les transmet au périphérique en série, comme il reçoit des données en série de ce périphérique et il les transmet en parallèle au µP.
- Donc l'objectif principal de ce circuit est d'assurer une communication parallèle-série et série-parallèle entre le µP et le périphérique. Cette communication est programmable de point de vue vitesse, sécurité, et format de donnée.

L'ACIA 6850 (Asynchrone Communication Interface Adaptor):

I. Structure Externe:



- L'ACIA se présente sous la forme d'un boitier (Dual in Line) DIL-24 broches.

a) Liaison avec µP :

- Bus de donnée D_0-D_7 : 8 lignes bidirectionnels à 3 états.
- Bus d'adresse : les lignes CS_0 $CS_1 et$ $\overline{CS_2}$ permettent de sélectionner le boitier.
- RS (Registre Select) : permet de sélectionner les registres internes.
- Bus de control :
  - E (Enable) : signal d'activation des échanges.
  - R/$\overline{W}$ (Read/ Write) : Lecture-écriture.
  - $\overline{IRQ}$ (Interupt Request) : cette sortie permet d'interrompre le µP.

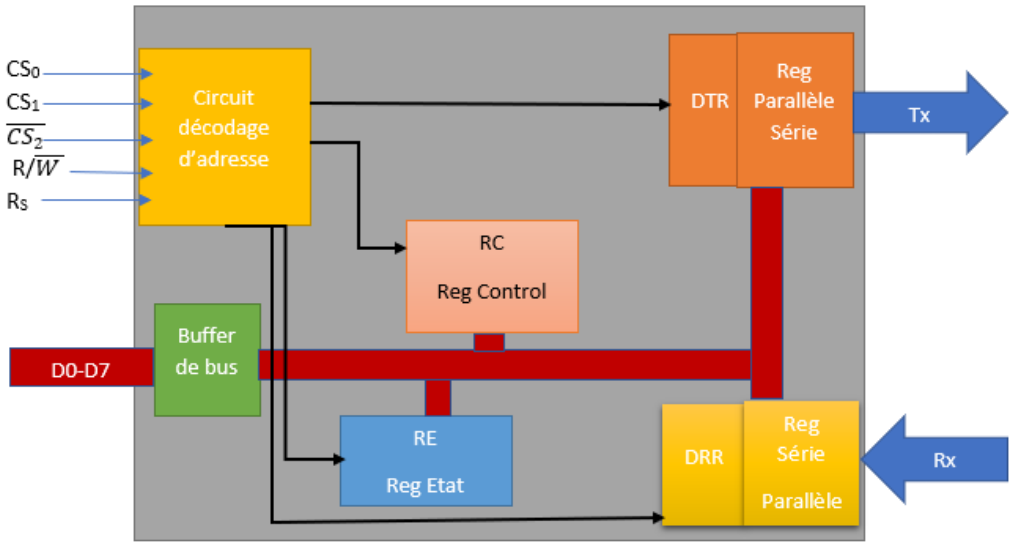
b) Coté périphérique :

- TxD (Data transmission Line) : ligne de transmission série, cette ligne assure la transmission des données en série de niveau TTL.
- RxD (Data ReceptionLine) : ligne de réception des données en série, cette ligne assure la réception des données en provenance du périphérique de niveau TTL.

- RTS (Request To Send) : sortie demande d'émission, elle permet de piloter un périphérique ou un MODEM (modulation/démodulation).
- $\overline{CTS}$ (clear to send) : entrée inhabitation de l'émetteur elle permet le control automatique de la fin de transmission par le modem non utilisé, elle doit être au niveau bas.
- $\overline{DCD}$ (Data Carier Detect) : entrée perte de la porteuse de donnée, elle permet le control de la réception non utilisée. Elle doit être au niveau bas.
- Txck (transmission clock) : horloge de transmission de données.
- Rx ck (receptionclock) : horloge de réception de données.

II. Structure Interne:

Synoptique :



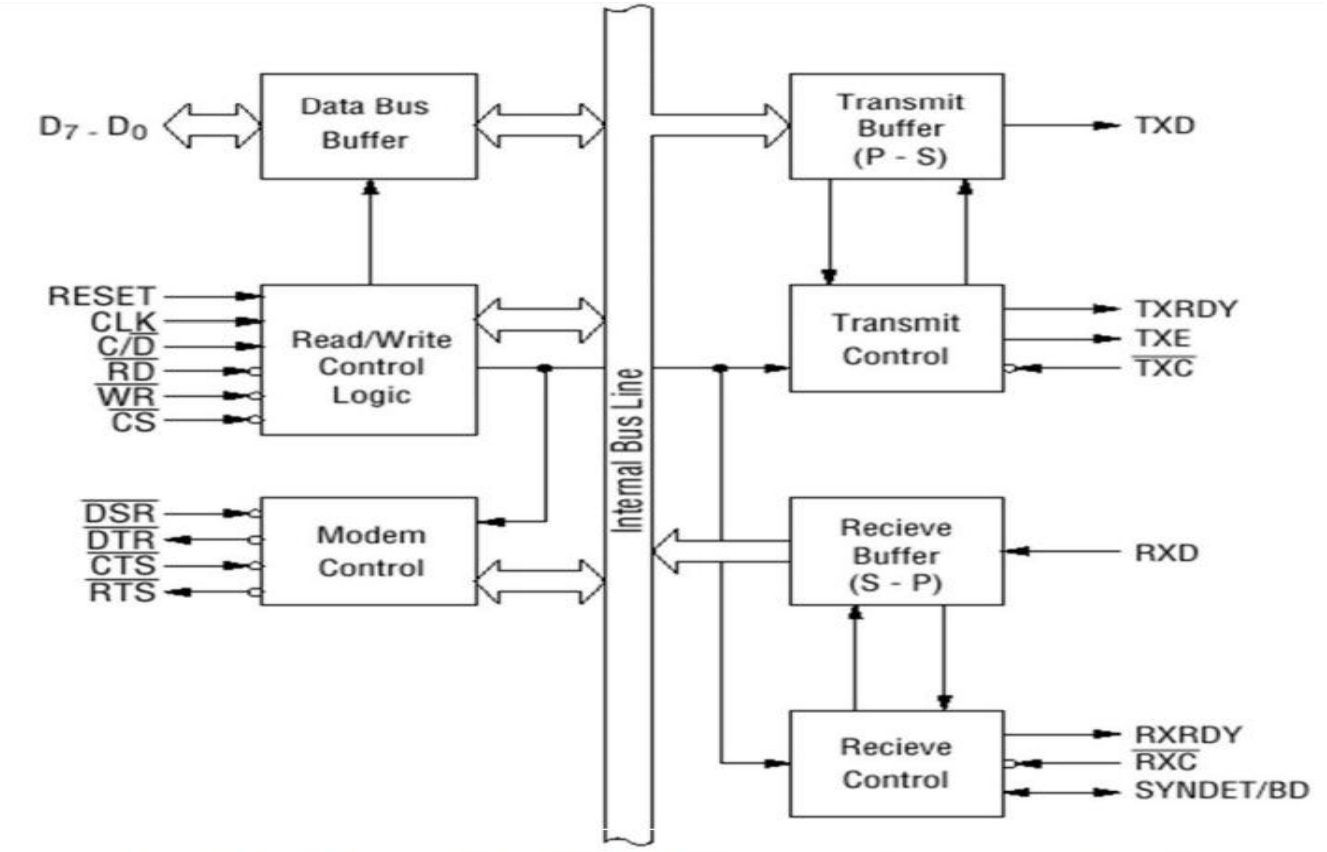


Fig 3.7 Block diagram of the 8251 USART (Universal Synchronous Asynchronous Receiver Transmitter)

Organisation interne (ACIA) :

- **RTD (Registre de transmission de données)** : c'est un registre de transmission à écriture seulement, il contient l'octet à émettre via un registre à décalage parallèle-série, les données seront présentes sur la ligne T_xD et synchronisé par une horloge de transmission.

- **RRD (Registre de réception de données) :**c'est un registre à accès en lecture seulement, il contient le mot reçu après un décalage série -parallèle, les données sont reçues via la ligne R_xDcadencéepar la fréquence de l'horloge de réception.

- **RC (Registre de control) :**c'est un registre à écriture seulement, il permet de programmer le protocole de communication série (vitesse, le format du mot, et le mode).

- **RE (Registre d'Etat) :**c'est un registre à accès en mode lecture seulement, il contient des informations sur l'état du mot reçue.

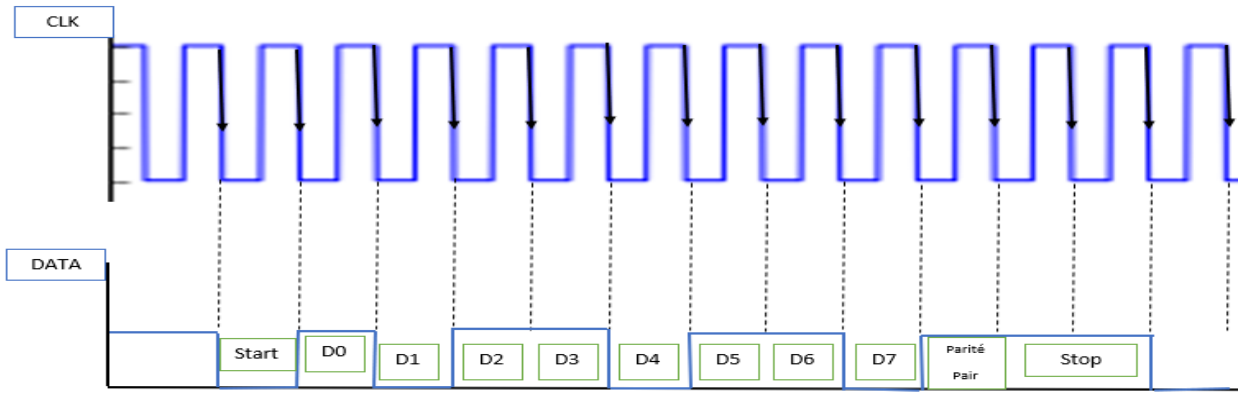
Adressage des registres interne de l'ACIA :

Bus	A_{15}		A_1	A_0	R/$\overline{W}$	Registre sélectionné
	CS_0	Décodage $CS_1\overline{CS_2}$		RS		
Adr	1	1	0	0	0	RS
Adr	1	1	0	0	1	RE
Adr+1	1	1	0	1	0	RTD
Adr+1	1	1	0	1	1	RRD

Exemple :

- Si Adresse de sélection d'ACIA est Adr : 70
  ⇒ RC et RE ⇒ Adresse = 70, même adresse seulement l'un à accès en mode lecture et l'autre en mode écriture.
  ⇒ RTD et RRD ⇒ Adresse = 71, même adresse seulement l'un à accès en mode lecture et l'autre en mode écriture.

III. Programmation de l'ACIA:



- La communication série gérée par l'ACIA utilise la procédure de START-STOP, chargée caractère transmis ou reçu à un format de 7 ou 8 bits avec ou sans contrôle de parité. Il est précédé par un bit START et suivi par 1 ou 2 bits STOP, chacun de ces bits est synchronisé sur une horloge mais la suite des caractères est asynchrone.

Programmation du registre de contrôle RC

RC_7	RC_6	RC_5	RC_4	RC_3	RC_2	RC_1	RC_0

RC_7	Control Rx
0	Masquer Int
1	Valider Int

RC_4	RC_3	RC_2	Longueur	Parité	Nbits Stop
0	0	0	7 bits	Pair	2
0	0	1	7 bits	Impair	2
0	1	0	7 bits	Pair	1
0	1	1	7 bits	Impair	1
1	0	0	8 bits	Sans	2
1	0	1	8 bits	Sans	1
1	1	0	8 bits	Pair	1
1	1	1	8 bits	Impair	1

RC_1	RC_0	Control Tx/Rx
0	0	1/1 Horloge
0	1	1/16 Horloge
1	0	1/64 Horloge
1	1	Reset ACIA

NOTE : 1 bauds = 1bit/sec.

- Les bits RC_1, RC_0 : permettent de sélectionner le rapport de division dans l'Horloge de Tx/Rx.

Dans le cas ou RC_1, RC_0 = 1 1 ⇒ une remise à zéro (Reset) de l'ACIA est confirmé pour l'initialisation de l'ACIA.

- Les bits RC_4, RC_3, RC_2 : permettent de sélectionner le format du mot reçu ou transmis.

- Les bits RC_6, RC_5 : actifs en transmission, ils permettent d'assurer une synchronisation avec le modem (Dans le cas pas de modem : 0 0)

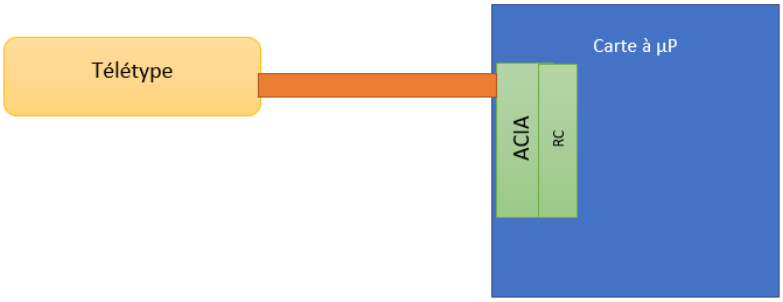
- Le bit RC_7 : actif en réception, ce bit lorsqu'il est à 1, il permet d'interrompre le µP quand le registre de réception est plein.

Exemple de programmation de registre de contrôle :

- On veut recevoir d'un télétype une donné et la stocker en mémoire à l'adresse : 8000.
- Le protocole de communication série est comme suite :

Fréquence Horloge ACIA : 1760 Hz, vitesse de réception : 110 bauds.

- Le Télétype transmet un mot de 7 bits avec parité Impaire et 2 bits Stop.
- Ecrire un programme d'initialisation de l'ACIA.



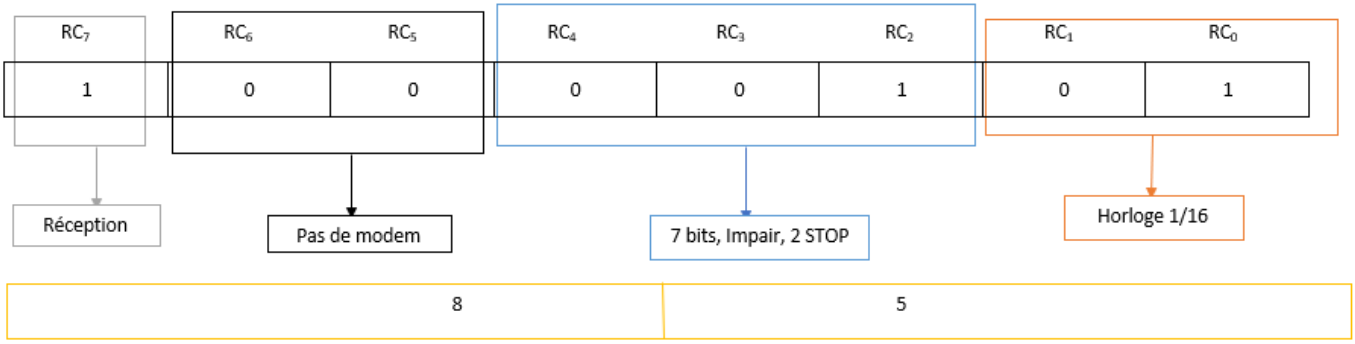
- On suppose que l'adresse de base de l'ACIA : 04

04 ⟶ Adresse : RC et RE

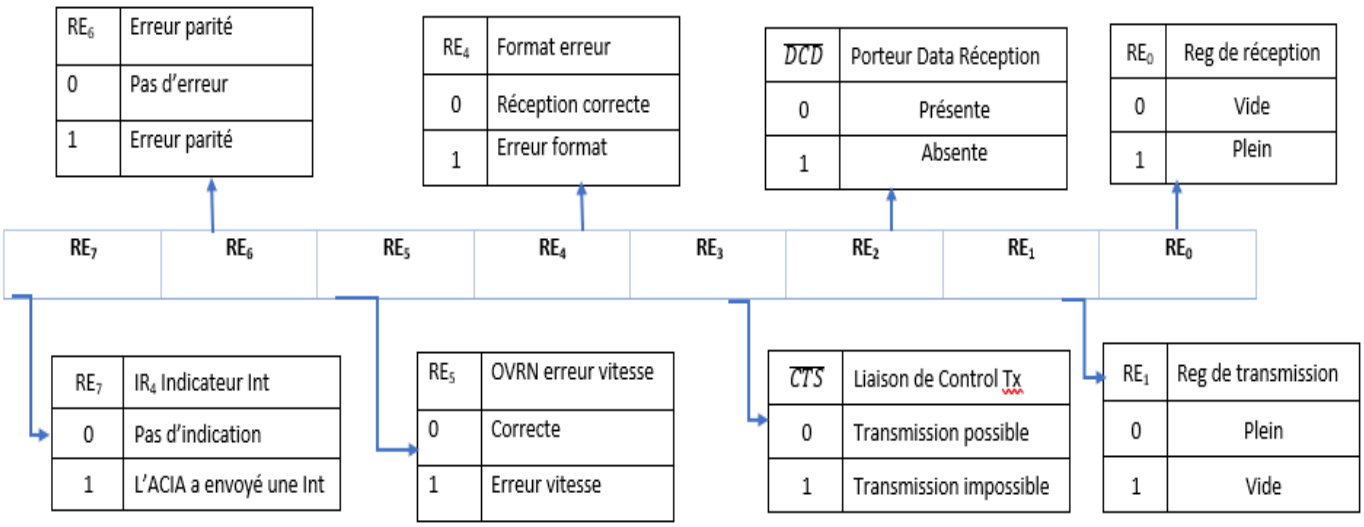
05 ⟶ Adresse : RTD et RRD

- Solution :

MVI A, 03
OUT 04 ⟶ Master Reset de l'ACIA RC0=1 et RC1=1
MVIA, 85 ⟶ Programmer le protocole, voir figure en bas.
OUT 04
IN 05 ⟶ Lire registre reception
STA 8000 ⟶ Stocker dans case mémoire 8000
état des bits du registre de contrôle.



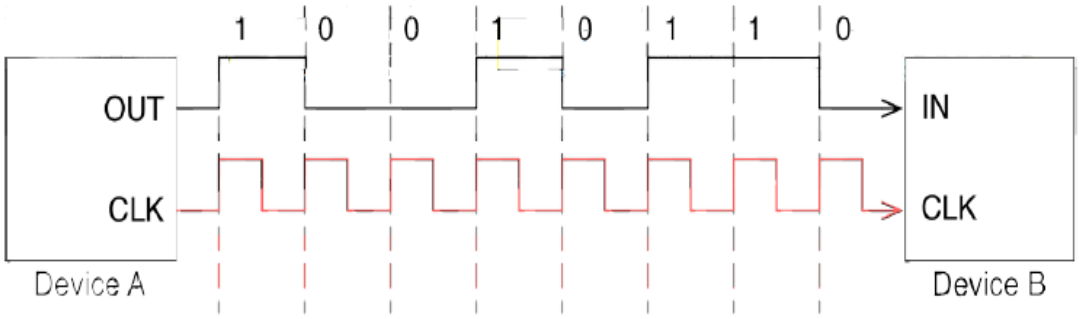
Rôle des bits (Indicateurs) du registre d'état : ce registre permet de synchroniser la communication entre deux ACIA ou interfaces série en général et de contrôler l'état de l'information reçue si elle respect le protocole de communication et s'il n y a pas d'erreurs. Comme il est visible les bits RE_0, RE_1 RE_2 et RE_3 sont utilisés pour assurer une bonne transmission des données alors que les bits RE_4-RE_7 sont utilises pour détecter les erreurs lors de la transmission.



L'UART de Intel :

Une interface UART/USART est un dispositif qui traduit une séquence parallèle de bits (généralement regroupés en un octet) en un flux continu de signaux circulant sur un seul fil.
Lorsque l'information circule entre deux dispositifs à l'intérieur d'un canal commun, les deux dispositifs (ici, on se référera à eux comme émetteur et récepteur) doivent être d'accord sur le timing, c-à-d, le temps qu'il faut pour transmettre chaque bit individuel d'information. Dans une transmission synchrone, l'émetteur et le récepteur partagent une horloge commune générée par l'un d'entre eux (habituellement le dispositif qui sert de maître).

Sur la Figure suivante, nous avons un chronogramme typique. montrant le dispositif A envoyant un octet (01101001) en série au dispositif B en utilisant une horloge de référence commune. L'horloge commune est également utilisée pour convenir de l'instant où commencer l'échantillonnage de la séquence de bits : lorsque le dispositif principal commence à envoyer l'horloge sur la ligne dédiée, cela signifie qu'il va envoyer une séquence de bits.

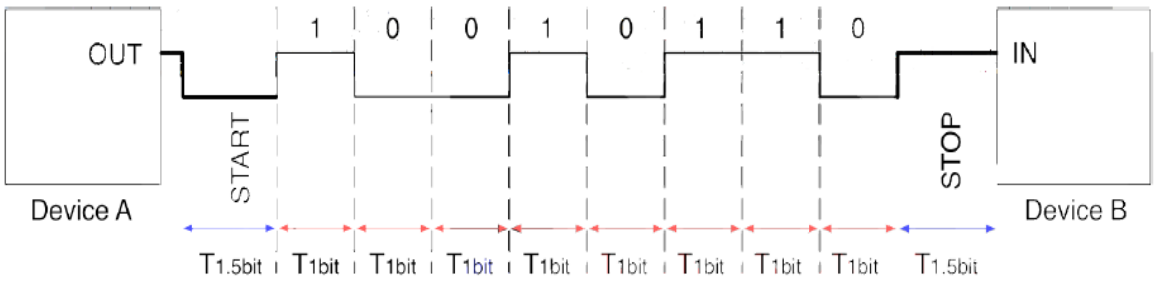


Communication série entre deux dispositifs utilisant une source d'horloge partagée.

Transmission Synchrone et Asynchrone :

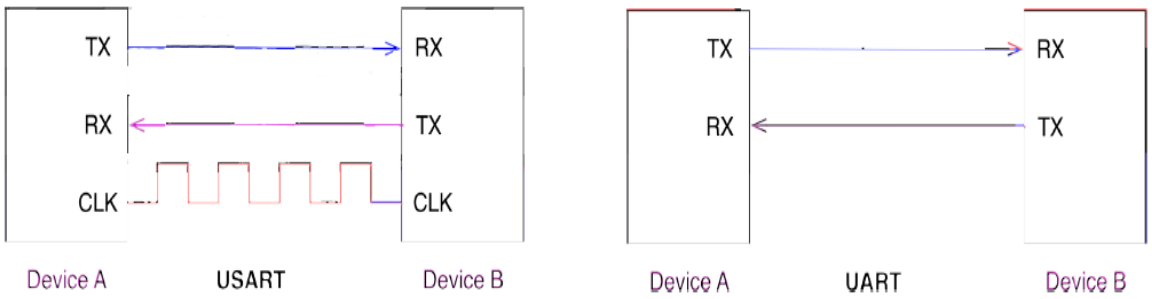
Dans une transmission synchrone, la vitesse et la durée de transmission sont définies par l'horloge : sa fréquence détermine la rapidité avec laquelle on peut transmettre un seul octet sur le canal de communication. Mais si les deux dispositifs impliqués dans la transmission de données conviennent de la durée nécessaire pour transmettre un seul bit et quand commencer et finir d'échantillonner les bits transmis, alors nous pouvons éviter d'utiliser une ligne d'horloge dédiée. Dans ce cas, nous avons une transmission asynchrone.

La figure suivante montre le chronogramme d'une transmission asynchrone. L'état inactif (c'est-à-dire qu'aucune transmission ne se produit) est représenté par le signal à l'état haut. La transmission commence par un bit START, représenté par le niveau bas. Le front descendant est détecté par le récepteur et une période de 1,5 bit après cela (indiquée sur la figure par T1.5bit), l'échantillonnage des bits commence. Huit bits de données sont échantillonnés. Le bit le moins significatif (LSB) est généralement transmis en premier. Un bit de parité optionnel est ensuite transmis (pour la vérification des erreurs des bits de données). Souvent, ce bit est omis si le canal de transmission est supposé être sans bruit ou s'il y a une vérification d'erreur plus haut dans les couches de protocole. La transmission est terminée par un bit STOP, qui dure 1,5 bits.



Chronogramme d'une communication série sans ligne d'horloge dédiée.

Une interface USART (Universal Synchronous Receiver/Transmitter) est un dispositif capable de transmettre un message de données en série à l'aide de deux E/S, l'une servant d'émetteur (TX) et l'autre de récepteur (RX), plus une E/S supplémentaire comme ligne d'horloge, tandis qu'une interface UART (Universal Asynchronous Receiver/Transmitter) utilise uniquement deux E/S TX/RX (voir la Figure 3).



Différence des signaux entre une USART et une UART.

La présence d'une ligne d'horloge dédiée ou d'un accord commun sur la fréquence de transmission ne garantit pas que le récepteur d'un flux d'octets puisse les traiter au même taux de transmission du maître. Pour cette raison, certaines normes de communication, comme RS232 etRS485, offrent la possibilité d'utiliser des lignes dédiées au contrôle de flux matériel (Hardware Flow Control).

Par exemple, deux périphériques qui communiquent à l'aide de l'interface RS232 peuvent partager deux lignes supplémentaires, nommées Request To Send (RTS) et Clear To Send (CTS) : l'émetteur positionne son RTS, ce qui indique au destinataire de commencer à surveiller sa ligne d'entrée de données. Lorsqu'il est prêt pour les données, le récepteur positionnera sa ligne complémentaire, CTS, qui indique à l'émetteur de commencer à envoyer des données et pour que l'émetteur commence à surveiller la ligne de sortie de données de l'esclave.