

Chapitre 6 : La Transmission des Données par Micro-ordinateurs

1. La Norme RS 232C

La norme RS-232C (ou jonction V24 normalisée par le CCITT) a été publiée en 1969 par l'électronique Industries Association. Établie initialement pour définir les connexions entre un terminal et un modem. Elle définit les caractéristiques électriques des circuits pour l'interface de deux appareils, elle donne des noms et des numéros aux différents fils nécessaires pour leur jonction.

Les prises de sorties ou les fiches de sorties sont de plusieurs sortes. Les prises 25 ou 9 broches de type D (appelées DB-25 et DB-9) sont les plus courants. Dans la pratique 9 broches seulement sont utilisées. Pour établir les différentes broches de la Jonction V24, nous allons procéder par étapes:

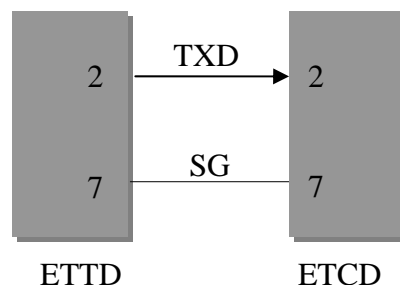
1.2. Transmission PC → Modem

Un seul appareil transmet (Connexion Unidirectionnelle)

La ligne 2: TXD (Transmitted Data ou transmission de données) ETTD → ETCD

La machine transmet par la Ligne 2, le modem reçoit sur la même ligne.

La ligne 7: SG (Signal Grand: Terme de protection); signal correspond au point de référence commun à partir duquel la polarité et la valeur de tension des autres lignes sont déterminées.



1.3. Contrôle de flux

Dans ce cas il est nécessaire que l'émetteur sache à quel moment le récepteur est prêt à recevoir. Pour cela il faut que le récepteur envoie un signal vers l'émetteur indiquant qu'il est prêt ou non, ce protocole s'appelle contrôle de flux ou validation de transfert (Handshaking) ou poignée de main en français. Il existe deux types contrôle de flux; câblé et programmé.

- *Le contrôle de flux programmé:* les signaux de contrôle sont des caractères envoyés sur les lignes de données et non sur les lignes de contrôles. Il s'agit des caractères de commandes XON (DC3) et XOFF (DC1) (Voir la table ASCII).

- *Le contrôle de flux Câblé:* Le récepteur envoie un signal (sur la ligne de contrôle) (de tension positive) tant qu'il est prêt à recevoir, lorsque l'émetteur reçoit une tension négative, il sait qu'il doit

stopper l'émission. Pour installer un contrôle de flux câblé, il faut ajouter au moins une ligne supplémentaire pour transporter le signal.

La ligne 6 : DSR (Data Set Ready : poste de données prêt)

Un récepteur se sert souvent d'une autre ligne de contrôle, pour mieux contrôler le transfert venant d'un émetteur, on ajoute donc une autre ligne.

La ligne 5: CTS (Clear To Send : prêt à émettre)

Dans ce cas, l'émetteur ne transmet que lorsque les deux lignes de contrôles sont dans l'état haut (tension positive). Il arrive souvent que les deux lignes soient reliées entre eux.

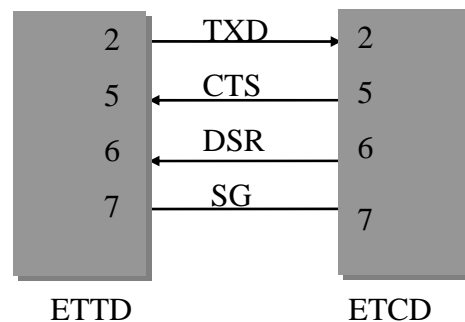


Figure 23. Transfert d'un ETTD vers un ETCD avec contrôle de flux

1.4. Transmission Modem → PC

Pour qu'on ETCD transmet avec un ETTD, on se sert de la ligne 3 de l'ETCD comme ligne de transfert de donnée,

La ligne 3: RXD (Receive data : Réception de donnée), cette ligne sert à la réception de données du côté ETTD.

Si un contrôle de flux est nécessaire, on utilise deux autres lignes de contrôle de l'ETTD.

La ligne 20: DTR (Data Terminal Ready : poste de données prêt) sert à envoyer le signal de contrôle du PC vers l'ETCD,

La ligne 4: RQS (Request to send : demande pour émettre), est la seconde ligne de contrôle du PC vers l'ETCD.

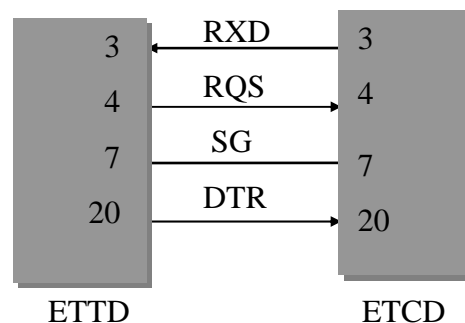


Figure 24. Transfert d'un ETCD vers un ETTD avec contrôle de flux

1.5. Communication Bidirectionnelle Modem - PC

Les deux appareils transmettent et reçoivent en même temps. Il faut au moins 3 lignes (2, 3, 7) pour la transmission de données dans chaque direction.

Si on ajoute les lignes de contrôle dans chaque direction, les lignes (6 et 20) et les lignes (4 et 5), on aura au total 7 lignes.

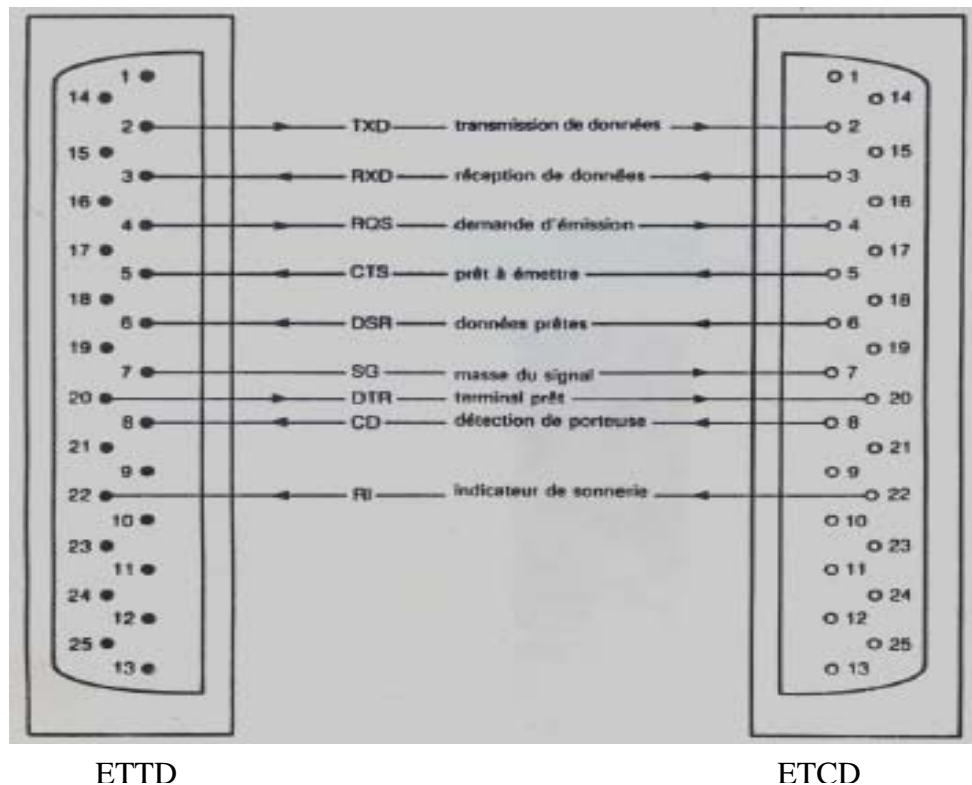


Figure 25. Branchement Standard d'un ETTD à un ETCD

On ajoute à ces lignes, deux lignes supplémentaires pour permettre à un modem de fournir d'avantage d'information à un ordinateur.

La ligne 8: CD (Carrier detect: détection du signal), indique la présence du signal de la porteuse.

La ligne 22: RI (Ring indicator: indicateur d'appel), avertit le modem qu'un appareil éloigné est entrain de l'appeler.

1.5. Communication Bidirectionnelle PC - PC

Pour transmettre des données en local entre deux micro-ordinateurs, le problème devient plus délicat. Le brochage doit pouvoir simuler la présence d'un modem. Cette forme de brochage pour une connexion directe ordinateur-ordinateur est appelée Null Modem ou câble croisé. Sa présentation est celle de la Figure 26.

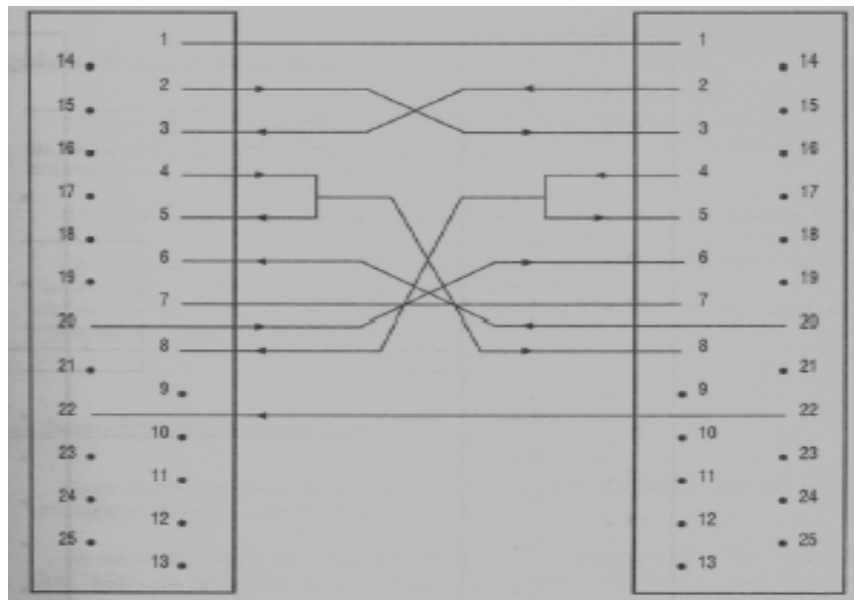


Figure 26. Connexion ETTD – ETTD (Null Modem)

Au niveau des broches 4 et 5, on peut tout simplement réaliser un strap. Normalement, elles servent à l'initialisation du circuit; la broche 5 (Clear To Send) renvoie à l'ordinateur ce signal lorsque le modem a établi la connexion avec l'autre modem en bout de ligne. Comme ici la connexion est directe, on rend implicite cette fonction grâce au strap. On peut aussi réaliser d'autres câblages plus simples.

2. Le Contrôleur de Communication Asynchrone

Le Contrôleur de Communication ou Émetteur Récepteur Asynchrone Universel (ERAU ou UART en anglais) est un circuit intégré chargé de la gestion de communication asynchrone.

2.1. Les Fonctions du Contrôleur de Communication

L'ERAU exécute principalement 4 tâches :

- Convertit des signaux parallèles, issue de a l'U.C. en signaux séries pour permettre la communication avec l'extérieur,
- Ajoute les bits de départ, d'arrêt et de parité en émission, et se débarrasse des même bits en réception et transmis le caractère à l'U.C.,
- S'assure que les bits soit émis avec la vitesse adéquate, calcul le bit de parité sur les caractères émis et reçus et rend compte à l'U.C. de toute erreur détectée,
- Gère le processus de contrôle de flux et rend compte à l'U.C. de l'état des lignes de contrôles entrantes.

2.2. Les Registres Principaux de l'ERAU

L'ERAU possède plusieurs registres, localisés à des emplacements mémoires internes, ils contiennent :

- Le dernier caractère reçu,
- Le prochain caractère à émettre,
- L'état courant des signaux de contrôle de flux,
- L'information indiquant si l'ERAU est prêt ou non à recevoir un caractère pour l'émettre, etc.

Ces registres sont de 3 sortes :

- Les registres de contrôles, qui reçoivent les commandes de l'U.C.
- Les registres d'état, qui informe l'U.C. de ce qui arrive vers L'ERAU.
- Les registres tampons, qui recueillent les caractères en instance d'émission ou de traitement.

La façon dont les registres sont manipulés dépend de l'architecture de l'ordinateur auquel l'ERAU est associé.

2.2.1. Les Registres de Contrôle

L'ERAU possède 4 registres de contrôle.

- **Le registre de contrôle de la ligne 3FB** (pour Com1) et **2FB** (pour Com2)

(Line Control Register), ce registre est utilisé pour initialiser les paramètres de communication.

Bits	Signification
0 et 1	Longueur des mots
2	bit d'arrêt (bit stop)
3	Valide la parité
4	Sélectionne la parité
5	bit de parité à un
6	Signal Break (dépassement du time out)
7	bits d'accès à la bascule de division (DLAB) (Division Latch access bit) ou à d'autres registres.

Les bit 0 et 1 (Longueur du mot)

Bit 0	Bit 1	Longueur du mot
0	0	5
0	1	6
1	0	7
1	1	8

Le bit 2 (Bit d'arrêt)

- 0 1 seul bit d'arrêt
- 1 2 bits d'arrêt

Le bit 3 (validation de la parité)

- 0 aucun bit de parité
- 1 1 bit de parité existe

Le bit 4 (sélectionne la parité)

- 1 parité paire
- 0 parité impaire (le bit 4 est ignoré tant que le bit 3 = 0)

Le bit 5 (Valeur de la parité)

bit3	bit4	bit5	
1	1	1	Valeur de parité = 0 parité paire
1	0	0	Valeur de parité = 1 parité impaire

Le bit 6 (Signal Break)

- 0 Force la société série
- 1 pas de sortie série

Le bit 7 bits d'accès à la bascule (deux registres) de division DLAB (Division Latch access Bit)

- 1 Une opération de L/E provoque un accès au diviseur de vitesse de transfert,
- 0 Une opération L/E provoque l'accès aux tampon de réception ou d'émission ou bien aux registres de validation d'interruption.

- **Les Bascules de Division de la Vitesse de Transfert (générateur de vitesse)**

Correspondent à deux registres qui sont:

- Bascule de division, octet de poids faible d'adresse **3F8** (pour Com1) et **2F8** (pour Com2)

(**DLL**: Division Latch less Significant Bit)

- Bascule de division, octet de poids fort d'adresse **3F9** (pour Com1) et **2F9** (pour Com2)

(**DLM**: Division Latch Most Significant Bit)

On peut accéder à ces deux registres si le DLAB = 1

Remarque:

Si le DLAB = 0, on peut accéder, soit au buffer de Transmission, soit au buffer de Réception 3F8 (pour Com 1) et 2F8 (pour Com 2)

La valeur de vitesse de transfert est établie en introduisant dans les deux registres le nombre par lequel la fréquence de l'horloge d'entrée (1,8432 MKZ) doit être divisée, la fréquence résultante vaut 16 fois la vitesse de transfert.

Vitesse	Décimal	Hex	DLM	DLL
300	384	180	1	80
1200	96	60	0	60
2400	48	30	0	30
4800	24	18	0	18
9600	12	0C	0	0C

Figure 27. Diviseurs de la vitesse de transfert pour différentes vitesses courantes

- **Registre de contrôle du modem 3FC** (pour Com1) et **3FC** (pour Com2)

(Modem Control Register), ce registre commande les signaux de contrôle émis par l'ERAU, il contrôle l'interface avec un modem ou un autre périphérique (ex: imprimante)

bit	Abréviation	Nom (si le bit est mis à 1)
0	DTR	Poste de donnée prêt
1	RQS	Demande d'émission
2	out 1	Sortie utilisateur 1
3	out 2	Sortie utilisateur 2
4	Loop	Boucle du mode test
5 - 7	à zéro	

Si le **bit 0** = 1, la ligne DTR est positionnée, l'appareil éloigné est autorisé à émettre,

Si le **bit 0** = 0, la ligne DIR n'est pas positionnée, l'ERAU demande à l'appareil éloigné de ne pas émettre.

Le bit 1 fonctionne de la même façon que **le bit 0**,

Si il est mis à 1 la ligne RQS est positionnée, Si il est mis égal à 0 la ligne RQS n'est pas positionnée.

Les bit 2 et 3 contrôlent les sorties auxiliaires dégénérées par l'utilisateur OUT 1 et OUT 2.

Les bits 4 valide le mode test par diagnostic

Les 5 à 7 sont toujours à zéro.

2.2.2. Les Registres d'Etat

Il y a 3 registres d'état qui rendent compte à l'U.C. de ce qui se passe dans différentes parties de l'ERAU.

- **Registre d'état de la ligne 3FD** (pour Com1) et **2FD** (pour Com2)

Ce registre sert à fournir des informations sur l'émission et la réception des données.

bits	Abréviation	Nom	Signification (si le bit est mis à 1)
0	DR(data Ready)	donnée prête	Un caractère a été reçu et placé dans le registre tampon de réception (il est toujours à 1 tant que le caractère n'a pas été lu.
1	OE(overrun error)	erreur de cadence	Un caractère a été lu avant que le précédent le soit, écrasement d'un caractère par un autre.
2	PE (Erreur de parité)	erreur de parité	le bit de parité reçu est incorrect.
3	FE (Frame Error)	erreur de trame	Un caractère reçu ne contient pas de bit d'arrêt.
4	BI (Signal de Break)		un signal d'interruption a été reçu
5	Registre de saisie de l'émetteur vide THRE (Transmitter Holding Register Empty)		L'ERAU est à prêt à accepter un nouveau caractère.
6	TSRE(Transmitter shift register Empty)		Registre de décalage de l'émetteur vide, il attend un caractère en provenance du registre de saisie de l'émetteur Ce bit ne peut être que lu. le registre de décalage sert à la procédure de conversion parallèle /série
7	toujours à Zéro		

- **Registre d'état du Modem 3FE** (pour Com1) et **2FE** (pour Com2)

(Modem Status register), ce registre donne des informations sur l'état des lignes de contrôle de flux.

bits	Nom	Signification (si le bit est mis à 1)
0	Delta CTS	La ligne prêt à mettre a changé
1	Delta DSR	La ligne poste de données prêtes à changer
2	TERI(Trailing Edge Ring Indicator)	Front arrière de l'indicateur d'appel: l'indicateur d'appel a changé du haut vers le bas
3	Delta RLSD (Receive line signal Detect)	La détection du signal de la ligne de réception a changé
4	CTS	L'entrée prêt à émettre est à l'état Haut (OK)
5	DSR	L'entrée poste de données prêt est à l'état haut (OK)
6	RI	L'indicateur d'appel est à l'état haut
7	RSLD	La détection du signal de la ligne de réception est à l'état haut

Les bit 0, 1 et 3 sont des bits delta, ils indiquent qu'un changement c'est produit depuis la dernière lecture du registre, ces bits sont mis à 1 lorsque la tension des lignes individuelles change, ils indiquent si elles sont à l'état haut ou l'état bas.

Les bits 4 à 7 donne l'état réel des lignes correspondantes, ils indiquent si ils sont à l'état haut ou à l'état bas

Si le bit correspondant est mis à 1, cela signifie que la ligne est dans l'état haut; la communication est alors autorisée

Si le bit correspondant est mis à 0, cela signifie que la ligne est dans l'état bas, ce qui interdit la communication

2.2.3. Les Registres tampons

Il y a deux sortes: tampon d'émission et tampon de réception

- **Le Registre tampon d'émission 3F8** (pour Com1) et **2F8** (pour Com2), c'est le registre de saisie de l'émetteur ou buffer d'émission (Transmitter Holding Register), recueille le nouveau caractère à émettre, le registre d'état de la ligne indique l'instant où le caractère est émis.

- **Le Registre tampon de réception 3F8** (pour Com1) et **2F8** (pour Com2), c'est le registre de réception ou buffer de réception (Receive Buffer Register), il extrait le dernier caractère reçu. Une fois lu, le registre d'état de la ligne indique que le tampon de réception est vide.

2.3. La Programmation de l'ERAU

2.3.1. Les Routines DOS et BIOS

Se sont des fonctions (des micro programmes) livrés avec le système d'exploitation de la machine. Il y a ceux qui sont dans la ROM, ce sont ceux du BIOS et ceux qui sont chargés en mémoire lors de l'initialisation du système, ce sont ceux du DOS.

En communication série, on peut avoir recours à deux fonctions DOS et à quatre fonctions BIOS. Elles sont appelées par l'intermédiaire d'interruption. Une interruption est déclenchée par l'instruction **int86**

2.3.2. Les 2 Fonctions DOS

Ces deux fonctions sont appelées en exécutant l'interruption numéro 21H.

La 1^{ère} fonction DOS de communication sert aux entrées série elle est réalisée en mettant le registre AH à 3 (numéro de la fonction) et en exécute l'interruption numéro 21H:

- AH = 3
- appel de l'interruption 21H

Le Dos attend de recevoir un caractère du port Com1 ou Com2 et le retourne dans AL (contient le caractère reçue)

La 2^{ème} fonction concerne les sorties séries.

On met dans AH 4 (numéro de la fonction), le caractère à émettre dans DL et en exécute l'interruption numéro 21H

- AH = 4
- DL = Caractère à émettre
- appel de l'interruption 21H

Remarque : Pas de moyens d'initialiser les paramètres de communication.

2.3.3. Les Fonctions Bios

Les quatre fonctions Bios sont appelées par l'intermédiaire de l'interruption numéro 14 H. Le registre AH contient toujours le numéro de la fonction (0 à 3), le registre DX contient le numéro du port, zéro pour Com1 et 1 pour Com 2

- DX= n° du port (0 ou 1)
- AH = n° de la fonction

La fonction n°: 0

Sert à initialiser les paramètres de communication. On y accède on mettant:

- DX = n° du port
- AH = 0
- AL = l'octet représentant les paramètres de communication
- appel de l'interruption 14 H

* Les bits 0 et 1 de l'octet définissent la longueur du caractère

- 1 1.....8 bit de données (longueur du caractère)
- 1 0.....7

Le bit 0 est toujours à 1

* Le bit 2 indique le nombre de bit d'arrêt

- 0 1 bit d'arrêt
- 1 2 bit d'arrêt

* Les bits 3 et 4 indique la parité

- 0 0 Aucune
- 1 0 impair
- 0 1 aucune (parité pas pris en compte)
- 1 1 pair

* Les bits 5 à 7 indiquent la vitesse de transmission

bit 7	bit 6	bit 5	Vitesse
0	0	0	110
0	0	1	150
0	1	0	300
0	1	1	600
1	0	0	1200
1	0	1	2400
1	1	0	4800
1	1	1	9600

Exemple:

On veut initialiser les paramètres de communication avec les valeurs suivantes :

Vitesse de transmission égale à 1200 bits/s, 8 de données, 1 bit d'arrêt et pas de parité

7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	1

$$2^7 = 2^6 \times 2 =$$

$$64 \times 2 = 128$$

$$128 + 3 = 131 \text{ décimal ou } 83h$$

Lorsque les paramètres sont initialisées, la fonction retourne l'état courant du port dans le registre AX = 131

La Fonction n°: 1

La fonction n°: 1 sert à l'émission des caractères, On l'appelle en mettant:

- AH = 1
- DX = numéro du port (0 pour Com1 et 1 pour Com 2)
- AL = Caractère à envoyer
- appel de l'interruption 14 H

Remarque:

La mise en oeuvre consiste à exécuter la fonction saisie de l'état du port (**n° 3**) puis la fonction d'émission, une fois que l'on sait que les conditions sont favorables. (les lignes de contrôle de flux DSR et DTR sont positionnées).

En retour le registre AH rend compte de toute condition d'erreur, si le bit 7= 0, l'émission a réussi; s'il est à 1, les bits restants indique le type d'erreur

bit 7 de AH

- 0 l'émission est bonne
- 1 erreur

La Fonction n°:2

Fonction de réception de caractères. Elle est appelée en mettant :

- AH = 2 (n° de la fonction)
- DX = n° du Port
- puis en exécute l'interruption 14 H

Le Bios attend un caractère en provenance du Port série soit reçu ou bien qu'un dépassement de temps soit atteint, lorsqu'un caractère est reçu, il est placé dans le registre AL.

Le registre AH = contient toutes les conditions d'erreurs.

Si AH = 0 (tout les bits sont mis à zéro), il n'y a pas d'erreur

Si AH ≠ de zéro alors les bits 0 à 7 bits indiquent le type d'erreur.

La Fonction n°: 3

Elle sert à la saisie de l'état du port (Get port status)

On y accède en mettant :

- AH = 3
- DX = n° du port
- appel de l'interruption 14 H

AX contient l'état du port.

Bits du registre AH	Signification si le bit est mis à 1
7	Erreur de dépassement de temps
6	Registre à décollage de l'émetteur vide
5	Registre de saisie de l'émetteur vide
4	Break détecte
3	Erreur de trame
2	Erreur de parité
1	Erreur de cadence
0	Données prêtes

Bit du registre AL	Signification si le bit est mis à 1
7	Détection signal de la ligne de réception
6	Indicateur d'appel positionné
5	Poste de données prêt
4	Prêt à émettre
3	Signal de la ligne de réception détecté
2	Front arrière de l'indicateur d'appel
1	Poste de données prêt delta
0	Prêt à émettre delta

Note : Delta signifie que le signal concerné a changé depuis la dernière fois que l'état du port a été lu.

Par exemple, le bit 1 du registre AL indique si l'état du signal DSR (poste de donnée prêt) a changé ou non depuis la dernière lecture du port.

2.3.4. Les avantages des routines DOS et BIOS

Il y a trois avantages essentiels:

- économiser l'effort de programmation et l'espace dans le fichier programme
- assurer une meilleure compatibilité avec d'autre machine utilisant le DOS et le BIOS

- dans le futur, si il y a des machines PC. avec une nouvelle architecture, les routines DOS et BIOS seront les mêmes.

Remarque :

Dans le cas de la communication série, la quantité de fonctions réalisables par les routines DOS et BIOS est limitée.

2.4. La Manipulation au niveau bit

En communication série, en utilisant les routines DOS ou BIOS ou en programmation directe de l'ERAU. On doit fréquemment accéder à certains bits des différents registres de l'adaptateur de communication, soit pour lire, soit pour écrire.

Pour cela, il faut se familiariser avec le tableau suivant :

Bits d'un octet	Valeurs
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

2.4.1. Test d'un bit

L'opérateur **&** du C, sert à réaliser l'opérateur **ET logique** avec cet opérateur (&) on peut déterminer la valeur de chaque bit d'un octet. Pour déterminer si un bit est mis à 1, on emploie l'opérateur & avec la valeur du bit.

Exemple :

Soit une variable notée **etat** (octet AH par exemple) qui représente l'état du port de communication, on veut savoir si le bit 4 de cette variable est mis à 1

If (etat & 16) ou If ((etat & 0x 10)) == 0 {
}

Si etat = 40

etat	00101000
configuration	<u>00010000</u>
ET logique	00000000

On obtient zéro comme résultat, ce qui indique que le bit 4 n'est pas mis à un

Si $\text{etat} = 52$

etat	00110100
configuration	<u>00010000</u>
ET logique	00010000

Le résultat est non nul, donc le bit 4 est mis à 1.

Comment on peut tester deux bits ?

2.4.2. Mise à un d'un bit

L'opérateur $|$ (barre Verticale) du C effectue le **OU logique**, pour mettre un bit à 1, on utilise l'opérateur $|$ (barre verticale) avec la valeur de ce bit.

On écrit: $\text{parcom} | = 8$

Cela mis le bit 3 à 1 et affecte la nouvelle valeur à la variable parcom.

Exemple : Si parcom vaut 37, le calcul donne

parcom	00100101
valeur du bit 3	<u>00001000</u>
OU logique	00101101

Comment mettre deux bits à 1.

2.4.3. Mise à Zéro d'un bit

L'opérateur le tilde (\sim) en C correspond à l'opérateur complément à un. Il met à zéro tous les bits à un, et met à 1 tous les bits à zéro.

Pour mettre un bit à zéro, on applique le **ET logique** entre l'octet de départ et le **complément à un** de la valeur des bits qu'il faut mettre à zéro.

Exemple:

Pour mettre à zéro le bit 5 de l'octet param, on écrit:

$\text{param} \& = \sim 32$

Si param vaut 106 en décimal, on aura:

32	000100000	(valeur du bit 5)
~ 32	111011111	(complément à un de 32)

Ensuite, on applique ET logique au complément à un de 32 et param

param	001101010
~ 32	<u>111011111</u>
ET	001001010

Le bit 5 est maintenant mis à zéro

2.5. Lecture/Ecriture dans un port de communication de l'ERAU

Pour lire à partir d'un port de communication, on peut écrire :

Carparam = int inp(port); Lit un octet à partir du port de communication et le met dans la variable Carparam

int outp (portcom, valeur); Envoie la valeur au port de communication

Exemple:

- Le registre d'état de la ligne pour Com1 se trouve à l'adresse 3FDH

Pour lire ce registre d'état de la ligne, on écrit :

Unsignad Adport = 0x 3FD

int état;

état = inp(adport); récupère les paramètres de com1 dans la variable état

- Le registre de contrôle du modem Com2 se trouve à l'adresse 2 FCH

Pour lui envoyer l'octet octmod

Unsigned adport = 0x2FCH

outp (adport, octmod);

Exercice:

Ecrire un programme d'émission et de réception de caractères

```

/*****
* Fonction : Initialisation de la liaison série (RS232)
* Arguments : void
*****/

/*-----
vitesse de communication: 9600 baud
parité: none (pas de parité)
nombre de stop bit: 1
longueur du mot de données: 8 bits
-----*/

void initpar()
{
    union REGS inregs,outregs;
    inregs.x.dx = 0;

```



```
inregs.h.ah = 0;  
inregs.h.al = 0xE3;  
int86(0x14,&inregs,&outregs);  
return;  
}
```