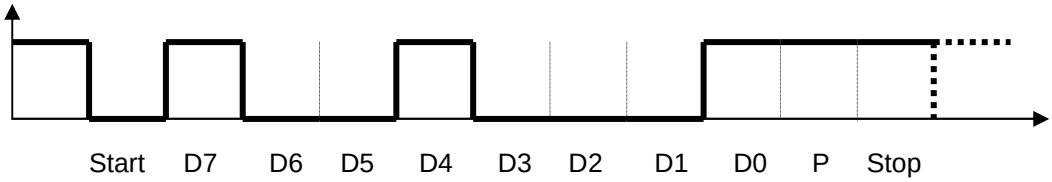
Communications séries asynchrones sur P18Fxx20

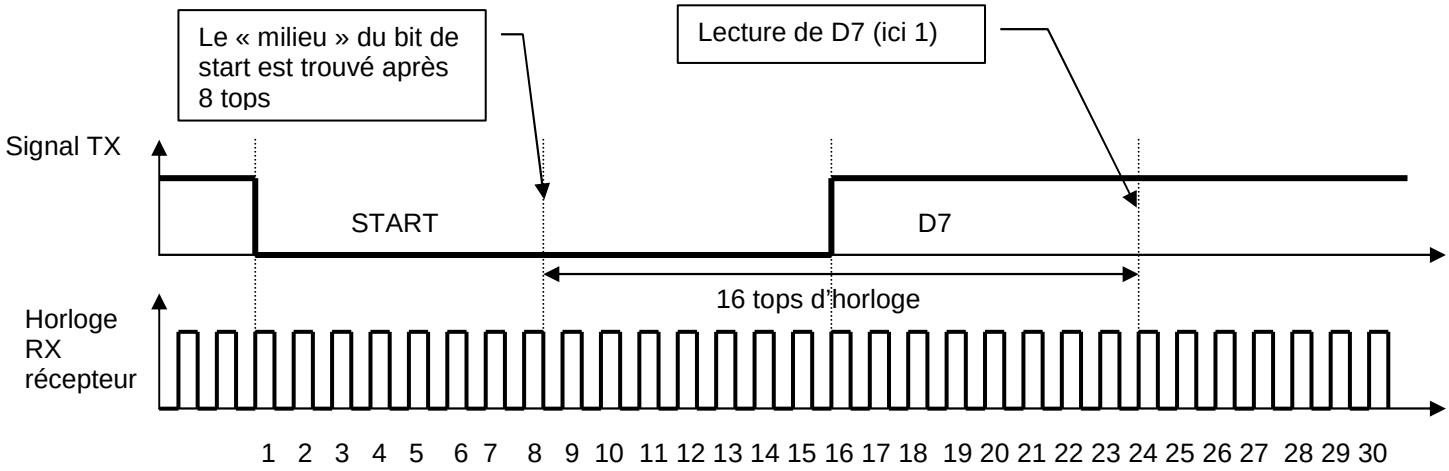
Les communications séries asynchrones suivent le format NZR (No Return to Zero). Ils communiquent avec le microprocesseur par l'intermédiaire du bus de données et en série avec l'extérieur par une liaison série asynchrone (sans horloge). Ils peuvent par ailleurs piloter un modem.
Exemple de trame : asynchrone 1start, 8 bits, parité paire, 1 stop : nombre 10010001



Parité paire : le bit de parité est positionné pour que l'ensemble des bits de donnée et le bit de parité représente un nombre de bits à 1 pair
Parité impaire : le bit de parité est positionné pour que l'ensemble des bits de donnée et le bit de parité représente un nombre de bits à 1 impair

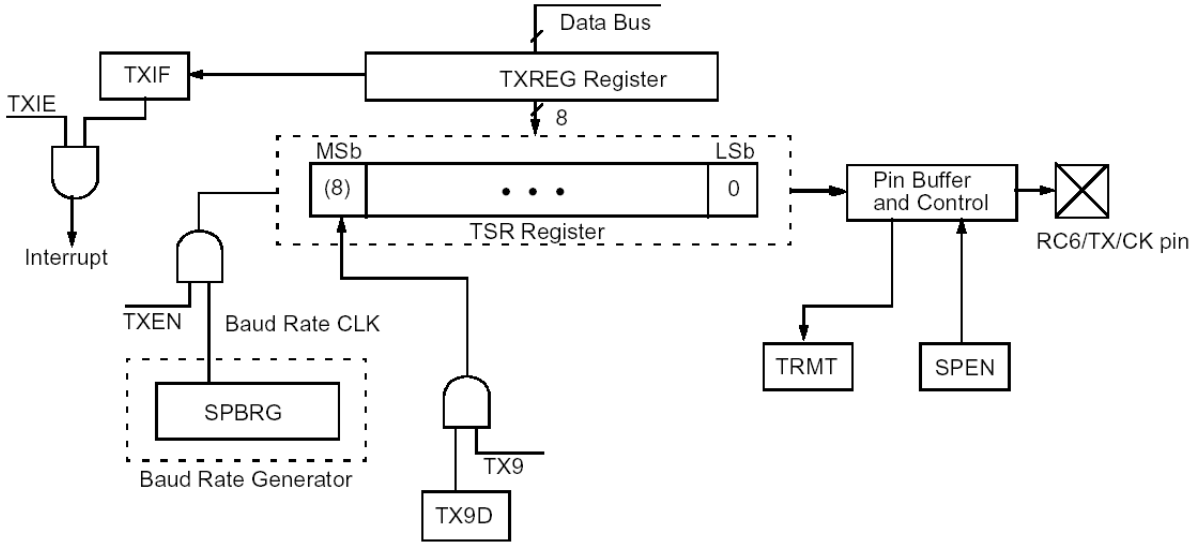
Dans ce type de transmission l'horloge de transmission est comprise dans le signal, le bit de start est utilisé par le récepteur pour se synchroniser avec l'émetteur. Cependant les deux horloges de transmission et de réception sont au départ très proche

L'horloge de réception possède une fréquence multiple de celle de transmission (en général x16 ou x64)
Dans le cas d'une division par 16 :
Lors de la réception du front descendant du bit de start, l'USART attend 8 tops d'horloge, le circuit reçoit alors théoriquement le milieu du bit de start, l'USART attend ensuite 16 tops d'horloge, le circuit reçoit alors le milieu de D7 et lit ce bit, l'USART attend ensuite 16 tops etc. L'horloge du récepteur est donc resynchronisée lors de la réception de chaque caractère.



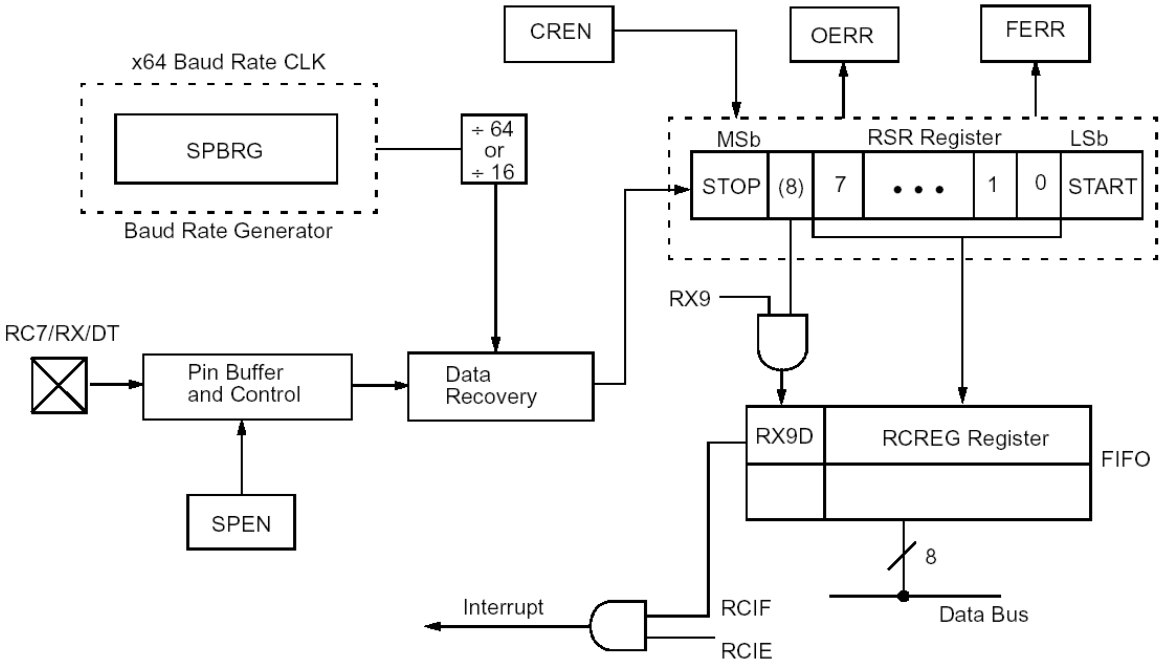
L'horloge RX (réception) doit donc toujours être supérieure à celle de TX (transmission). En réalité les deux horloges sont identiques et TX est divisé dans l'USART pour produire la vitesse de transmission souhaité.

USART : Transmission



TXREG : registre de transmission (tampon)
SPBRG : définit la vitesse de transmission (BAUD)
TXEN : valide l'horloge
TX9 : valide le 9ième bit
TX9D :9ième bit (donnée, adresse ou parité)
TXIE : autorise l'interruption
TXIF : drapeau d'interruption, indique que TXREG est vide
SPEN : configure TX/RX pin pour USART
TMRT : indique si TSR est vide

USART : Réception



CREN : active le récepteur asynchrone
SPBRG : définit la vitesse de transmission (BAUD)
SPEN : configure TX/RX pin pour USART
RCIE : autorise l'interruption en réception
RCIF : drapeau d'interruption de réception d'une donnée
RX9 : valide la prise en compte de D8 (adresse, donnée ou parité, traitement par logiciel)
OERR, FERR : indicateurs d'erreurs de réception

Registres généraux

TXSTA : Registre d'état et de contrôle des émissions

7	6	5	4	3	2	1	0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D

CSRC: Clock Source Select bit
Non utilisé en mode asynchrone
1 = mode maitre (horloge générée en interne depuis BRG)
0 = mode esclave (horloge externe)
TX9:
1 = transmission sur 9 bits
0 = transmission sur 8 bits
TXEN:
1 = transmissions activées
0 = transmissions désactivées
SYNC: USART Mode Select bit
1 = mode synchrone
0 = mode asynchrone
SENDB : Send Break Character bit
Mode asynchrone :
1 = Emet le caractère Break lors de la prochaine transmission (RAZ automatique)
0 = Fin de transmission du caractère Break
Inutilisé en mode synchrone
BRGH: High Baud Rate Select bit
Mode asynchrone :
1 = grande vitesse
0 = petite vitesse
Inutilisé en mode synchrone
TRMT: Transmit Shift Register Status bit
1 = registre de transmission TSR vide
0 = registre de transmission TSR plein
TX9D: 9th bit of Transmit Data
Peut être une adresse, une donnée ou un bit de parité

RCSTA: RECEIVE Registre d'état et de contrôle des réceptions

7	6	5	4	3	2	1	0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

SPEN: Serial Port Enable bit
1 = Active le port série (configure RX/DT et TX/CK comme des broches de port série)
0 = Désactive le port série
RX9: 9-bit Receive Enable bit
1 = Réception sur 9 bits, 0 = Réception sur 9 bits
SREN: Single Receive Enable bit
Inutilisé en modes asynchrones et en mode synchrone esclave
En mde synchrone et maitre
1 = Autorise une réception unique (effacé après la réception)
0 = interdit la réception
CREN: Continuous Receive Enable bit
En mode asnchrone
1 = Active le récepteur
0 = Désactive le récepteur
En mode synchrone
1 = Active la réception (CREN est prioritaire sur SREN)
0 = Désactive la réception
ADDEN: Address Detect Enable bit
En mode asynchrone sur 9bits (RX9 = 1):
1 = Active la détection d'adresse, autorise l'interruption et ne charge pas la donnée dans le buffer de réception quand RSR<8> =1
0 = Désactive la détection d'adresse, tous les bits sont envoyés dans le buffer de réception et le 9^{ième} bit peut être utilisé comme bit de parité
FERR: Framing Error bit (erreur de trame, généralement le bit de start ou stop n'a pas été détecté correctement)
1 = Framing error (mis à jour par une lecture de de RCREG et la réception du prochain octet)
0 = No framing error
OERR: Overrun Error bit - Indique qu'un caractère a été perdu
1 = Overrun error (effacé en effacant CREN), 0 = No overrun error
RX9D: 9th bit of Received Data
Peut être un bit d'adresse ou de donnée ou de parité et doit être géré par logiciel

Exemple en C18 :
Emetteur activé, transmission sur 8bits mode asynchrone, pas de Break.
Active le port série, récepteur activé sur 8 bits,

```
TXSTA = 0b00100000;
RCSTA = 0b10010000;
PIE1bits.TXIE=0;    // IT en emission désactivée
PIE1bits.RCIE=1;    // IT en reception activée (si nécessaire)
PIR1bits.TXIF=0;    // efface drapeau transmission
PIR1bits.RCIF=0;    // efface drapeau reception
```

Générateur de BAUD.

Le générateur de BAUD (BRG) repose sur un comptage de l'horloge Fosc (à ne pas confondre avec Fcycle=Fosc/4)
BRG peut être un compteur 8 bits ou 16 bits
`BAUDCONbits.BRG16=1 ; // compteur 16 bits`
Le bit BRGH (TXSTA) permet d'activer ou non le pre-divideur sur BRG
`TXSTAbits.BRGH=0; // BRG lent`

La vitesse de transmission dépend de l'oscillateur de BRG16 de BRGH et de la valeur dans SPBRGH:SPBRG. Un choix judicieux de BRGH et BRG16 permettra de réduire l'erreur sur la vitesse de transmission en fonction de la fréquence de l'oscillateur et la vitesse en BAUD souhaitée

BRG16	BRGH	Mode BRG/EUSART	Formule de calcul de la vitesse en Baud (n = ([SPBRGH:SPBRG])
0	0	8-bit/Asynchronous	Fosc/[64 (n + 1)]
0	1	8-bit/Asynchronous	Fosc/[16 (n + 1)]
1	0	16- bit/Asynchronous	
1	1	16-bit/Asynchronous	Fosc/[4 (n + 1)]

Exemple : Pour un PIC avec FOSC=32Mhz et une vitesse de transmission souhaitée de 9600 Bauds, mode 8 bits (BRG16=0) , avec prediviseur (BRGH=0)

Vitesse recherchée en BAUD= FOSC/(64 ([SPBRGH:SPBRG] + 1))
On recherche X= SPBRGH:SPBRG:
X = ((FOSC/Desired Baud Rate)/64) – 1= ((32000000/9600)/64) – 1= [51.08] = 51 (arrondi)
La vitesse réelle en BAUD sera = 32000000/(64 (51 + 1)) = 9615.38
L'erreur est donc : (BAUDcalculé – BAUDdésiré)/ BAUDdésiré
= (9615.38 – 9600)/9600 = 0.16%

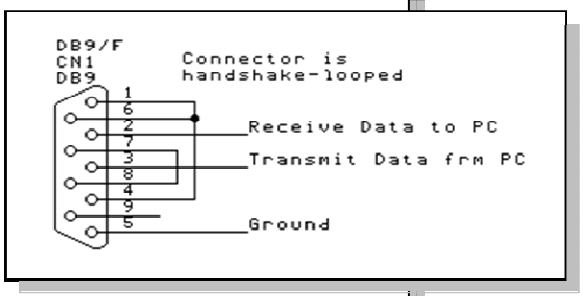
La configuration du générateur de BAUD (BRG) s'écrira :

```
BAUDCONbits.BRG16=0;
TXSTAbits.BRGH=0;
SPBRGH= 0x00;     // ligne inutile ici puisque BRG est sur 8 bits
SPBRG = 51;
```

BAUDCON: BAUD RATE CONTROL REGISTER

7	6	5	4	3	2	1	0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN

ABDOVF: Auto-Baud Acquisition Rollover Status bit
1 = BRG a été mis à jour durant le mode Auto-Baud Rate Detect (RAZ par logiciel)
0 = BRG n'a pas été mis à jour
RCIDL: Receive Operation Idle Status bit
1 = Une opération de réception est en attente
0 = Une opération de réception est active
RXDTP: Received Data Polarity Select bit (Asyncnronous mode only)
Asynchronous mode:
1 = la donnée en RX est inversée (0-1)
0 = la donnée en RX n'est pas inversée (1-0)
TXCKP: Clock and Data Polarity Select bit
Asynchronous mode:
1 = L'état de repos pour une transmission est 0
0 = L'état de repos pour une transmission est 1
Synchronous mode:
1 = L'état de repos pour l'horloge est 1
0 = L'état de repos pour l'horloge est 0
BRG16: 16-Bit Baud Rate Register Enable bit
1 = 16-bit Baud Rate Generator – SPBRGH et SPBRG forment SPBRG sur 16 BITS
0 = 8-bit Baud Rate Generator – SPBRG seulement (Compatible avec les anciens PIC), SPBRGH est ignoré
bit 2 **Unimplemented:** Read as '0'
WUE: Wake-up Enable bit
Asynchronous mode:
1 = EUSART echantillonne la broche RX en continu, une interruption est générée sur front descendnat, ce bit est effacé automatiquement lors du prochain front montant
0 = RX n'est pas surveillé
Synchronous mode:
Unused in this mode.
ABDEN: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Active la mesure automatique de la vitesse de réception au prochain caractère. (Nécessite la réception du caractère 55h) . Effacé automatiquement après la mesure
0 = La mesure de la vitesse de reception est desactivé.

Exemple : programme echo (PIC↔ PC) : Fosc=32MHz , comm : 9600,n,8,1

```
#include <p18f4620.h>
rom char mess[]="\nLes communications sont ouvertes\nTapez une touche ...\n\n";
// indique qu'un caractère est dans RCREG de l'USART
char data_recue(void)                  // reception d'une interruption
{       if (PIR1bits.RCIF)             /* char recu en reception*/
        {       PIR1bits.RCIF=0;       // efface drapeau
                return (1);            // indique qu'un nouveau caractère est dans RCREG
        }
        else return (0);               // pas de nouveau caractère reçu
}
// envoie un caractère sur USART
void putch(unsigned char c)            //putch est défini sur le port série
{   while(!TXSTAbits.TRMT);            // pas de transmission en cours ?
    TXREG=c;                           /* envoie un caractère */
    while(!PIR1bits.TXIF);
}
// envoie une chaine en ROM
void putchaine(rom char* chaine)
{       while (*chaine) putch(*chaine++);
}
void main(void)
{
        TXSTA = 0b00100000;
        RCSTA = 0b10010000;
        PIE1bits.TXIE=0;       // IT en emission désactivée
        PIE1bits.RCIE=0;       // IT en reception déactivée
        BAUDCONbits.BRG16=0;
        TXSTAbits.BRGH=0;
        SPBRG = 51;
        putchaine(mess);       // intro
        while(1)               // echo +1 ex: si 'a' est transmis 'b' est envoyé en echo
            { if (data_recue()) putch(RCREG+1); }
}
```