

chronogramme d'une liaison série asynchrone : chapitre 16 sur les UART

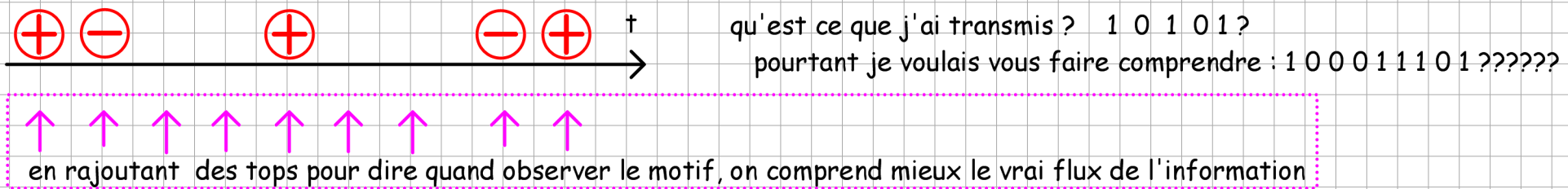
transmettre de l'information bit par bit : concept d'envoi d'un octet , les bits les uns après les autres, à l'aide d'un registre à décalage :

chargement parallèle et envoi sériel bit par bit au rythme d'une horloge qu'on appelle BAUDRATE : nombre de bit transmis par seconde

Transmettre de l'information nécessite :

-- Distinguer les motifs élémentaires (bit 0 et bit 1) ou plusieurs bits d'un coup (penser à QAM) \oplus veut dire 1 , \ominus veut dire 0

-- Savoir à quel instant l'information est disponible : MARQUEUR de synchro de lecture du motif



-- Savoir comment découper le flot de bits et les regrouper pour reconstituer des octets : concept d'encapsulation ... MARQUEUR de synchro

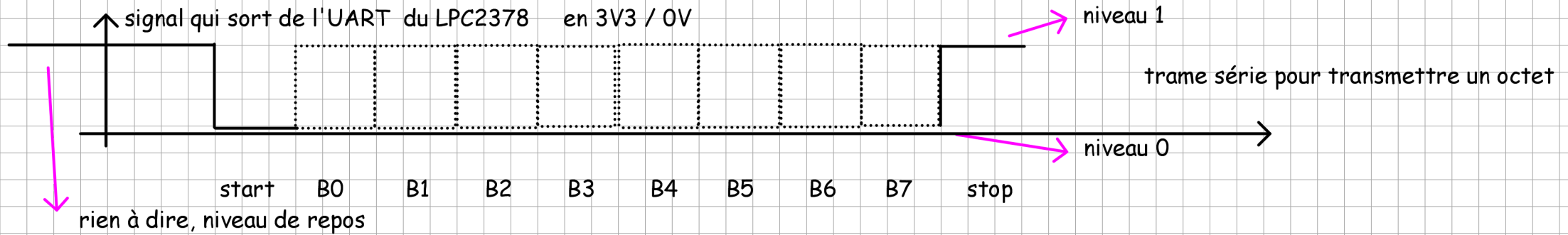
les lettres sont des motifs , regroupés en mots : MARQUEUR de mot = ESPACE

Les mots sont regroupés en phrases : MARQUEUR de phrase = MAJUSCULE et POINT

Les phrases sont regroupées en paragraphes : MARQUEUR de paragraphe = Tabulation et RETOUR à la ligne

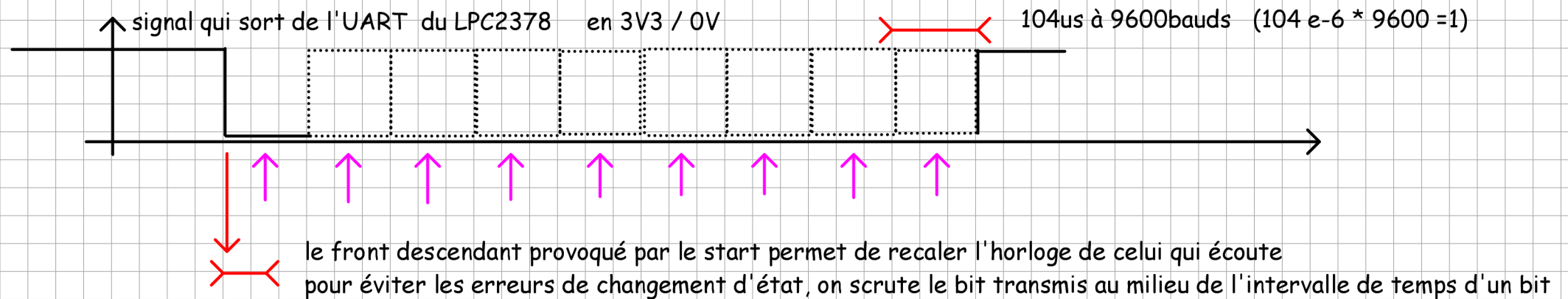
Les paragraphes sont regroupés en chapitre : MARQUEUR de chapitre = Saut de page

LA LIAISON série asynchrone permet de transmettre des octets à un certain débit appelé baudrate :

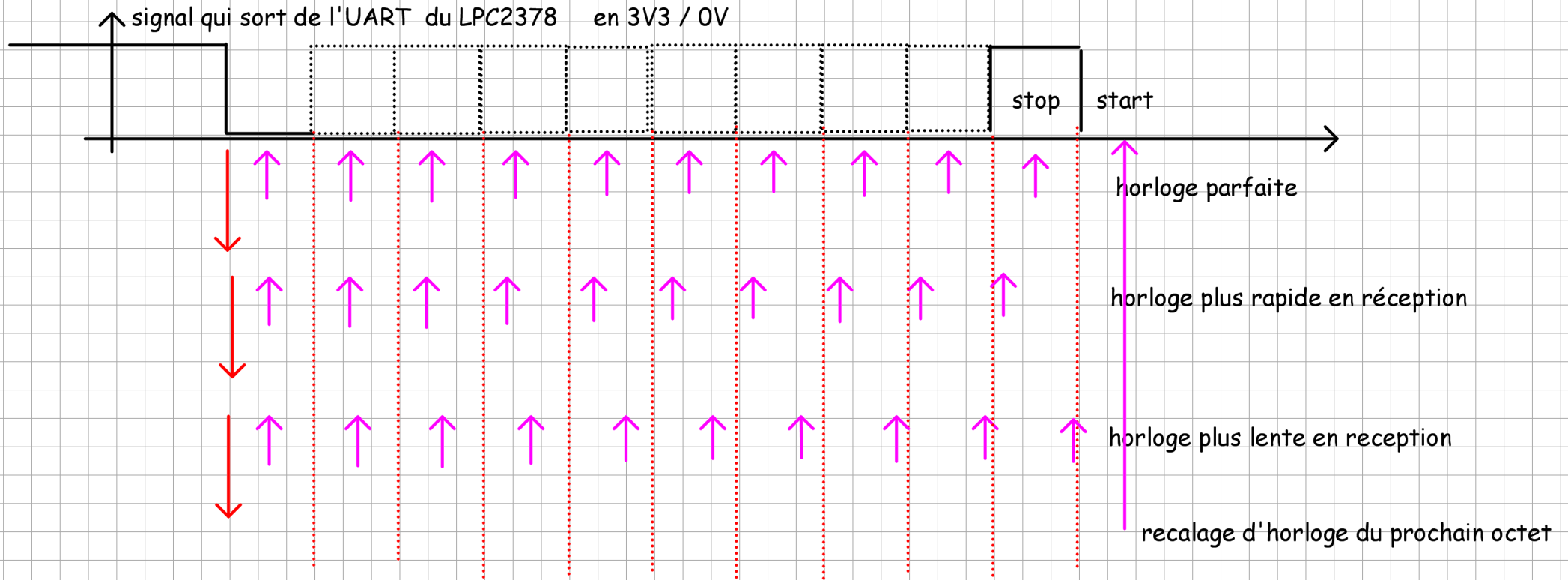


1er soucis , celui qui écoute n'a pas l'horloge pour savoir quand attraper les bit, et il ne sait pas s'il doit écouter...

Le concept start / stop permet de dire : Je commence à parler, tu peux resynchroniser ton horloge (on postule qu'on a la même à peu près)

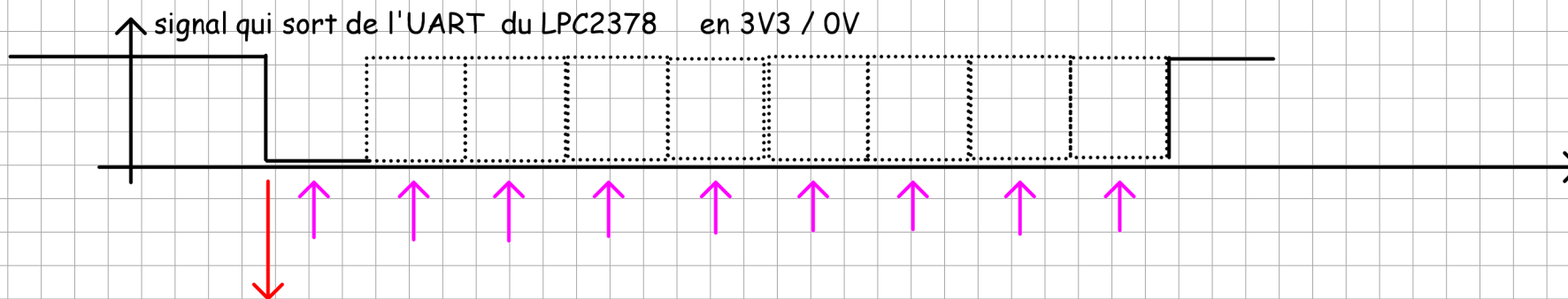


il y a une certaine tolérance d'horloge pour celui qui écoute, on peut être plus rapide ou plus lent



quelle erreur tolérable? On est au milieu du bit la première fois (50% de la durée d'un bit)
On en transmet 10, on a le droit de cumuler jusqu'à $10 * 5\%$ d'erreur

il suffit que l'horloge de l'émetteur et du récepteur soient calées à 5% près pour réussir à recevoir l'octet

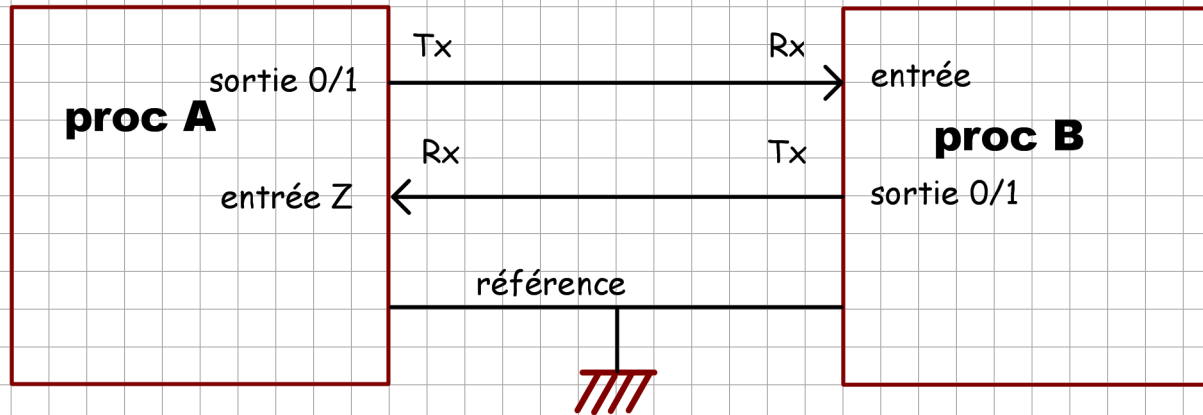


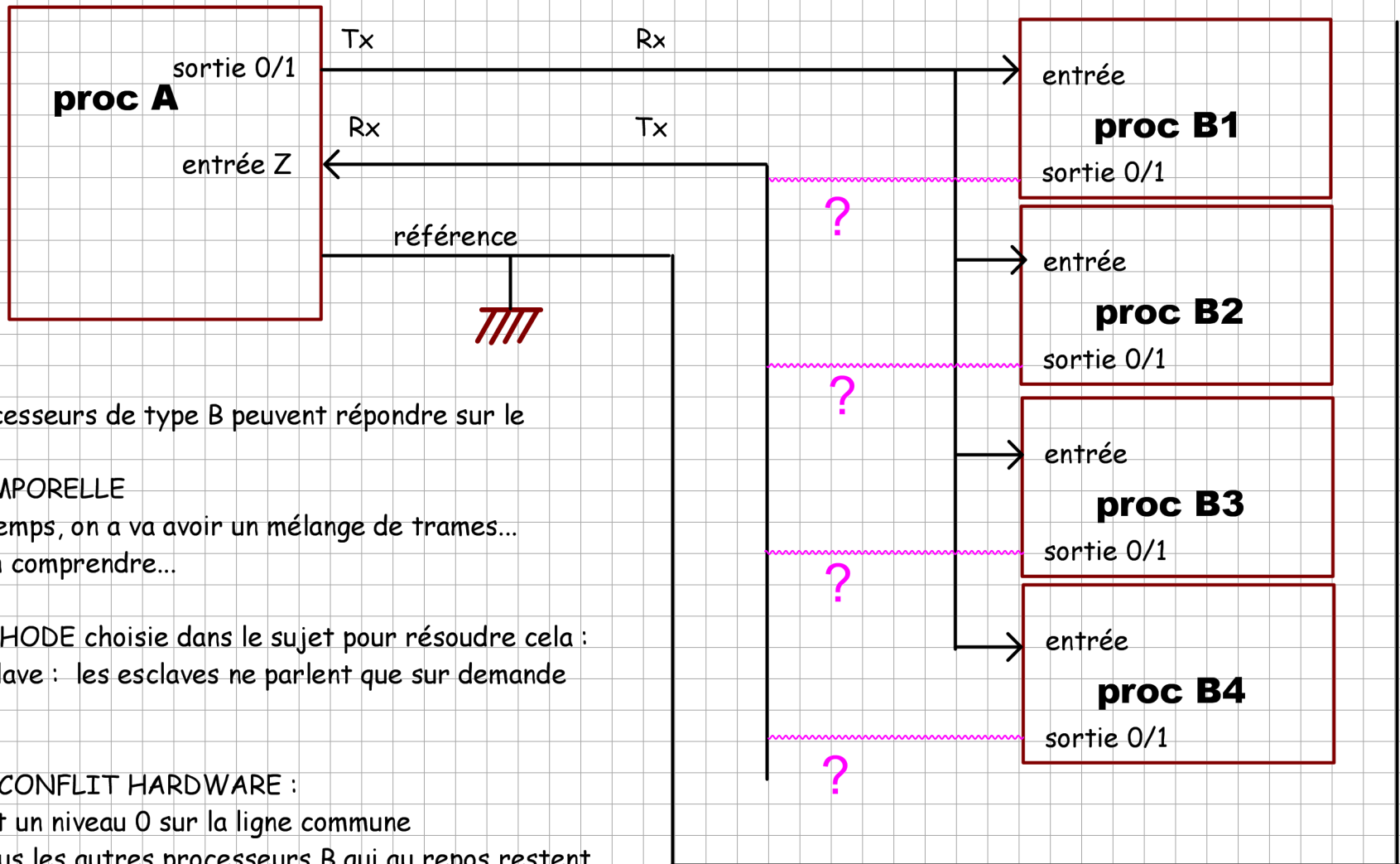
Du point de vue du programmeur : on transmet des octets

il existe un flag d'évènement qui dit : on peut envoyer un octet !

il existe un flag d'évènement qui dit : on a reçu un octet !

Du point de vue hardware pour communiquer





est ce que tous les processeurs de type B peuvent répondre sur le même fil ?

PROBLEMATIQUE TEMPORELLE

s'ils parlent en même temps, on a va avoir un mélange de trames...

Le processeur A va rien comprendre...

IL EXISTE UNE METHODE choisie dans le sujet pour résoudre cela :
un concept maître /esclave : les esclaves ne parlent que sur demande
du maître

PROBLEMATIQUE DE CONFLIT HARDWARE :

si le processeur Bx veut un niveau 0 sur la ligne commune
il est en conflit avec tous les autres processeurs B qui au repos restent
à 1

si les processeurs parlent chacun à leur tour, il faut que 0 l'emporte sur les 1 de repos