

TD ASR2

Transmission asynchrone ~ Contrôle de flux

On s'intéresse à la transmission asynchrone entre deux ordinateurs munis chacun d'un port série. Les caractéristiques de la liaison seront les suivantes :

- données codées sur 8 bits,
- parité paire,
- 1 bit START, 2 bits STOP,
- 9600 bps.

Question 0

On désire utiliser un câble avec 3 fils. Donner le câblage de la jonction entre les 2 ordinateurs.

Question 1

Quel est le nombre de bits transférés pour un caractère codé sur 8 bits ?

Question 2

Quelle est la durée qui sépare l'émission de deux caractères ?

Question 3

Un module RS232 contient les primitives suivantes :

Action InitialiserRS232 (Entrée *Config* : Octet
NuméroPort : Octet);

Cette action permet d'initialiser le port série *NuméroPort* (0,1,...) avec la configuration *Config*. Les bits de l'octet *Config* ont la signification suivante :

<u>Bits</u>	<u>Description</u>	<u>Valeur</u>
0..1	Nombre de bits de données	10 : 7 bits 11 : 8 bits
2	Nombre de bits STOP	0 : 1 bit 1 : 2 bits
3..4	Parité	00 : pas de parité 01 : impaire 10 : pas de parité 11 : paire
5..7	Vitesse	000 : 111 bps 001 : 150 bps 010 : 300 bps 011 : 600 bps 100 : 1200 bps 101 : 2400 bps 110 : 4800 bps 111 : 9600 bps

Action EmissionCaractère (Entrée : *Caract* : Octet
NuméroPort : Octet);

Cette action envoie le caractère *Caract* sur le port série *NuméroPort*.

Action RéceptionCaractère (Entrée : *NuméroPort* : Octet
Sortie : *Caract* : Octet);

Cette action permet de recevoir un caractère sur le port série *Numéroport*.

a) Action *Emetteur*

Ecrire une action *Emetteur* qui configure le port série 1 et qui envoie, sur ce même port, une suite de *P* caractères. (*P* constante prédéfinie)

b) Action *Récepteur*

Ecrire une action *Récepteur* qui configure le port série 1 et qui lit, sur ce même port, *P* caractères. Les caractères seront affichés sur l'écran, au fur et à mesure de leur arrivée.

Pour a et b, on effectue seulement la transmission de données, sans se préoccuper des phases d'ouverture et de fermeture.

Question 4

Donner une solution permettant le transfert d'un fichier. Quel est le temps minimal nécessaire pour le transfert d'un fichier de 120K ?

Question 5

Expliquez pourquoi les situations critiques suivantes peuvent avoir lieu lorsqu'on effectue un transfert en utilisant les actions *Emetteur* et *Récepteur* :

- a) perte des caractères lors de la réception,
- b) perte des caractères lors de l'émission.

Question 6

On se propose d'éviter les situations critiques de "type a" en utilisant une file circulaire de type FIFO (First In First Out) de la façon suivante : sur le site récepteur, l'arrivée d'un caractère sur le port série va déclencher la routine d'interruption *InterruptSérie* qui déposera le caractère dans une file circulaire. On dispose alors d'un module *RS232_Bufferisée* avec les primitives suivantes :

Action InitialiserRS232_Bufferisée (Entrée *Config* : Octet
NuméroPort : Octet);

(Idem à *InitialiserRS232*)

Action EmissionCaractère_Bufferisée (Entrée : *Caract* : Octet
NuméroPort : Octet);

(Idem à *EmissionCaractère*)

Fonction NombreCaractères (*NuméroPort* : Octet) : entier;

Cette fonction retourne le nombre de caractères contenus dans la file circulaire associée au port *NuméroPort*.

Action RéceptionCaractère_Bufferisée (Entrée : *NuméroPort* : Octet
Sortie : *Caract* : Octet);

Cette action fournit dans *Caract* le caractère en tête de la file circulaire associée au port *Numéroport* (et la file circulaire est mise à jour). A utiliser lorsque la file est non vide.

Donner les nouvelles versions des actions *Emetteur* et *Récepteur*. On ne demande pas d'écrire :

- la routine *InterruptSérie* de traitement de l'interruption série,
- les primitives du module *RS232_Bufferisée*.

Question 7

Cette solution n'est pas suffisante pour éviter toutes les situations de "types a". Expliquez pourquoi ? Donner brièvement les idées d'une solution correcte.