

# Microprocesseur

## TP3 : Communications UART

Dans ce TP, nous allons mettre en œuvre un support de communication entre le kit microprocesseur et l'ordinateur. Pour cela nous utiliserons le module USART (Universal Synchronous/Asynchronous Receiver Transmitter) mis en œuvre sur un port RS232. Le protocole UART est un des moyen de communication possible grâce au module USART. La simplicité de ce protocole permettra de ne s'attarder que sur la gestion logicielle de la communication.

Pour effectuer la communication, nous allons transférer des caractères ASCII avec l'ordinateur au moyen d'un câble *null-modem* RS232. Le connecteur du kit est identifié RS232 sur la carte.

Côté logiciel, Nous utiliserons TeraTerm qui émule un terminal physique sur la liaison série. Il est disponible depuis le menu *Démarrer*. Lancez-le. Une première fenêtre vous demande quel type de communication vous souhaitez, sélectionnez 'série' sur le port COM1. Dans le menu *Configuration/Port Série*, définissez les paramètres de la communication :

- vitesse au choix (il est nécessaire de configurer la même vitesse sur la carte)
- 8 bits de données
- Pas de parité (none)
- 1 bit de stop.
- Pas de contrôle de flux (none).

Cette configuration de communication sera à utiliser également pour le module USART. Tout caractère frappé dans la zone de texte du terminal sera envoyé au kit par la liaison, mais ne sera pas affiché. Tout caractère émis par le kit sera affiché dans cette même zone de texte.

### 1) Transfert de données brutes

#### a) Configuration du PIO.

Dans un premier temps, téléchargez et ouvrez le projet TP3. Ce projet fait clignoter une LED pendant son fonctionnement, cela permet d'identifier si le programme fonctionne quand on n'observe rien sur la liaison.

La première étape consiste à configurer le PIO pour que le module USART contrôle les broches de communication. L'USART a besoin de deux broches pour fonctionner correctement. Elles sont situées sur le PIOA. La documentation du PIO est fournie au chapitre 27 à partir de la page 227. Des masques sont fournis pour gérer les broches plus rapidement, ils correspondent aux constantes `UART_RX_PIN` et `UART_TX_PIN` qui représentent respectivement les broches de réception (RX) et d'envoi (TX). Sur les deux modules USART que propose le micro-contrôleur, nous allons travailler avec le module USART0.

Comment désactiver le mode 'PIO' pour les broches RX et TX pour que le module y accède directement ? Pour ces broches, le module USART0 correspond au périphérique A (ch 10.4 / p33). Comment activer la connexion entre les broches et le périphérique ?

Du côté USART, il faut d'abord tester si le module reçoit bien les signaux avant de continuer une configuration plus évoluée. Pour cela, on utilise un mode décrit au paragraphe 30.6.8.3 (Test Modes) page 328 : le mode 'Remote loopback'. Dans ce mode, le signal RX est directement relié au signal TX sans être interprété. Par contre, il est quand même nécessaire d'activer les canaux d'émission et de réception (Control Register).

Quels sont les symboles qui permettent de s'adresser au module USART0 ? (adresse de base et Identifiant). C'est le registre 'UART mode register' qui permet d'activer le mode 'Remote loopback'. Comment ?

Réalisez l'ensemble de la configuration sur la carte. N'oubliez pas de connecter le câble série entre la carte et l'ordinateur. Le système fonctionne si le programme hyperterminal affiche les caractères que vous écrivez dans la zone de texte. Vérifiez bien l'utilité de la liaison en débranchant la carte temporairement : la liaison doit s'interrompre (il est préférable de déconnecter le câble au niveau de l'ordinateur pour des raisons de robustesse mécanique). Concrètement, chaque caractère frappé au clavier est envoyé à la carte, puis renvoyé de la carte vers l'ordinateur pour être affiché.

#### b) Configuration de l'USART et envoi basique.

Le module qui gère la liaison USART est documenté dans le chapitre 30. Pour effectuer une communication, il faut d'abord configurer correctement le mode. Le fonctionnement est le mode normal. Pour le reste, reprenez les mêmes paramètres que ceux que vous avez utilisés pour *TeraTerm*. Le débit se règle dans le registre BRGR (Baud Rate Generator Register). Exprimez d'abord la valeur à écrire sous forme théorique.

Le module devrait être configuré correctement. Si c'est le cas, *TeraTerm* affiche des caractères 'A' périodiquement, en phase avec l'animation des LEDs. Retrouvez la ligne qui provoque l'envoi d'un caractère.

#### c) Envoi de texte.

Nous allons maintenant nous intéresser à un envoi de mot complet. La liaison ne fonctionne que pour un caractère à la fois, il faut donc envoyer le mot caractère par caractère. L'abus de langage 'envoyer un mot' signifie qu'il n'y a pas d'attente entre les envois successifs des caractères du mot. L'envoi d'un caractère étant (très) lent par rapport au fonctionnement du processeur, il existe un moyen de savoir si le canal d'envoi est occupé. Quel est ce moyen ?

Envoyez un mot d'au moins cinq caractères au lieu de la lettre 'A'.

Comment gérez-vous l'attente entre les envois de deux caractères successifs ? (il ne doit pas y avoir de temps mort entre deux envois de lettres successifs)

#### d) Réception de texte.

Comment repérer qu'un caractère a été reçu par le module ?

Où lire la valeur du caractère reçu ?

Faites un programme qui renvoie les caractères reçus en leur ajoutant la valeur 1. Si ce programme fonctionne, vous obtiendrez un comportement proche du 1.a) mais la réponse à 'A' sera 'B', la réponse à 'B' sera 'C' et ainsi de suite...