

4 – Transmission série

Objectifs : gestion de périphériques (UART, PLL), mise en œuvre d'une transmission série.

PREPARATION : LE CODE DE LA FONCTION `uart0_init` DE LA PARTIE 1.2 DEVRA ETRE PREPARE ET SERA VERIFIE AU DEBUT DE LA SEANCE DE TP.

REALISATION : LES PARTIES 1.2, 1.3 et 1.4 DOIVENT ETRE VALIDEES PAR L'ENCADRANT AU COURS DE LA SEANCE.

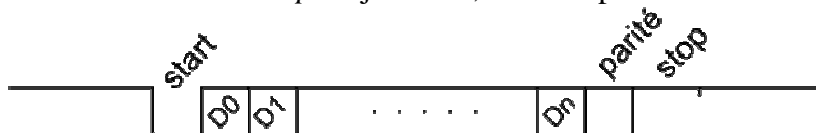
1.1 Description

L'objectif de cette séance est de mettre en œuvre une transmission série. Nous utiliserons pour cela l'UART0 de la carte de développement pour transmettre des messages vers le PC hôte (par l'intermédiaire du programme Hyper Terminal de Windows).

L'UART (Universal Asynchronous Receiver Transmitter) est un émetteur-récepteur asynchrone universel. Ce composant est utilisé pour faire la liaison entre un processeur et le port série. Le processeur manipule les données en parallèle (8-bit, 16-bit, etc), il faut donc transformer ces données pour les faire passer à travers une liaison série, qui utilise un seul fil pour faire passer tous les bits de données.

Une trame UART est composée de la façon suivante :

- Un bit de *start* toujours à 0, utilisé pour la synchronisation du récepteur.
- Les données, la taille peut varier généralement entre 5 et 9 bits.
- Eventuellement un bit de parité, pair ou impair.
- Et un bit de *stop* toujours à 1, la durée peut varier entre 1 et 2 temps bit.



Afin de faciliter l'interopérabilité entre périphériques, des vitesses de transmission sont normalisées, l'unité baud correspondant à un temps bit (110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps).

1.2 Initialisation

Dans notre application, nous allons opérer à une vitesse de transmission de 9600 bauds. Pour que la transmission puisse fonctionner, il faut impérativement veiller à ce que les vitesses de transmissions soient les mêmes entre les 2 unités communicantes. En particulier, il va falloir configurer la fréquence d'horloge du processeur `clk` pour qu'elle soit compatible avec la vitesse de transmission choisie (9600 bauds). La configuration de la fréquence de fonctionnement du processeur est possible à l'aide de la PLL (Phase Locked Loop).

Configurer le projet de la façon suivante :

- Créer un fichier `main_uart.c`, dans lequel vous écrirez une fonction `main()` qui pour l'instant appellera une fonction `uart0_init()`.
- Créer un fichier `uart0.c`, dans lequel vous écrirez la fonction `uart0_init()`.

La première tâche de la fonction `uart0_init()` consiste à modifier la valeur de la fréquence d'horloge du processeur initialement à 10 MHz, car cette valeur n'est pas un

multiple entier de 9600. Nous fixerons `cclk` à la valeur de 60MHz qui est elle compatible avec une vitesse de transmission de 9600 bauds.

Reportez-vous au manuel utilisateur pour la description exhaustive du fonctionnement de la PLL. Ajouter le code nécessaire pour configurer la fréquence d'horloge cpu (`cclk`) à 60MHz sachant que la fréquence de l'oscillateur `Focs` est à 10 MHz.

Les étapes à suivre sont :

- Configurer la PLL (`PLLCFG`) et rendre effective la modification (`PLLFEED`).
- Activer la PLL (`PLLCON`) et rendre effective l'activation (`PLLFEED`).
- Attendre que la PLL se stabilise sur la fréquence (`PLLSTAT`).
- Connecter la PLL comme source d'horloge (`PLLCON`, `PLLFEED`).
- Mettre l'horloge périphérique `pclk` à la valeur de l'horloge processeur `cclk` (`VPBDIV`).

Tester votre programme et vérifier la fréquence d'horloge en utilisant le PWM.

1.3 Configuration de l'UART

La deuxième tâche de la fonction `uart0_init()` consiste à configurer l'UART0.

Reportez-vous au manuel utilisateur pour la description exhaustive du fonctionnement de l'UART0. Vous devez successivement :

- Configurer les broches de la carte pour activer l'UART0 en transmission/réception (`PINSEL0`).
- Activation et reset des FIFOs (`UART0_FCR`).
- Configurer l'UART en format 8 bit. Ici il faudra aussi spécifier que l'on va diviser la valeur de l'horloge périphérique puisqu'elle fonctionne à 60 MHz et que l'on transmet à 9600 bauds (`UART0_LCR`).
- Spécifier la valeur de cette division (`UART0_DLL`, `UART0_DLM`). Quelle est cette valeur ?
- Désactiver l'accès la division de la valeur de l'horloge périphérique (`UART0_LCR`).

Compléter fonction `uart0_init()`.

1.4 Transmission de données

Pour tester vos programmes, il vous faudra d'abord connecter un câble série entre la sortie de l'UART0 et le port COM1. Ensuite, on utilisera le programme Hyper Terminal sous Windows qui permettra d'afficher les données envoyées par la carte LPC2106 vers le PC. Ouvrir un Hyper Terminal (programme → accessoires → communication) et configurer une connexion avec les paramètres suivants : port COM1, bps 9600, bits de données 8, parité aucun, bits d'arrêt 1, contrôle de flux aucun.

Vous écrirez successivement dans le fichier `uart.c` les sous-programmes suivants :

- `void uart0_putc(char c)` permettant la transmission d'un caractère.
- `int transmitter0_empty(void)` permettant de tester si le registre de transmission `U0THR` est vide.
- `void uart0_puts(char *s)` permettant la transmission d'un message et qui utilisera la procédure `uart0_putc` et `transmitter0_empty`.

Tester la transmission d'un message de la carte LPC106 vers l'Hyperterminal Windows.

1.5 Mise en forme d'une trame