

4 – Transmission série UART

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

Préparation: lire la partie **1.1** Description et **1.** Transmission d'un caractère du sujet, ainsi que STM32F4 Series Reference Manual p. 742-744 et p. 782-793.

A quelle horloge est connecté le port GPIOB? A quelles broches du GPIO peuvent être connecté les signaux Tx/Rx de l'USART USART3? A quelle horloge sont connectées ces broches? Justifiez vos réponses en mentionnant les pages de la documentation qui indiquent ces informations. Voir User Manual (STM32F4-Discovery).

Réalisation : les parties **1.2**, **1.3** et **1.4** doivent être validées par l'encadrant au cours de la séance.

1.1 Description

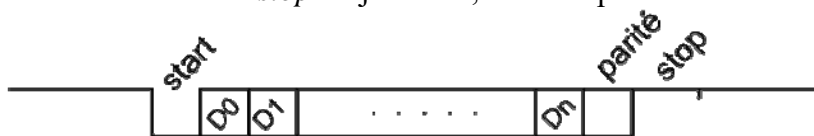
L'objectif de cette séance est de mettre en œuvre une transmission série.

Nous utiliserons pour cela l'USART USART3 de la carte STM32F4-Discovery pour transmettre sur la broche Tx.

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 (RS232). Pour transmettre une donnée (dont la taille est programmable de 5 à 9 bits), il suffit de l'écrire dans le registre de transmission (en général 8-bit, 16-bit ou 32-bit). L'UART transforme ces données pour les faire passer à travers une liaison série, qui utilise un seul fil, et indique que la transmission est terminée par un bit indicateur dans le registre d'état. L'USART (Universal Synchronous & Asynchronous Receiver Transmitter) est une variante de l'UART qui peut communiquer de manière synchrone.

Une trame UART est composée de la façon suivante :

- Un bit de *start* toujours à 0, utilisé pour la synchronisation du récepteur.
- Les données, dont 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).

L'objectif est de mettre en œuvre une transmission de données 8 bit par l'UART3 vers le port série (COM1) de la station de travail. Les données envoyées sur la broche Tx seront affichées sur la station de travail à l'aide d'un utilitaire de communication (Tera Term Pro).

Configuration du projet en C:

- Ouvrir Keil μ Vision.
- Project \rightarrow Create new μ Vision project

Sélectionner le répertoire C:\users\elec3\TP_microprocesseurs\TP4 (à créer s'il n'existe pas). **!/ ne jamais travailler dans Mes Documents, ni sur le Bureau Windows.**

Nom du fichier : TP4_USART

- Dans la fenêtre Select Device for Target 'Target 1', cliquer dans STMicroelectronics et sélectionner STM32F4 Series / STM32F407 / STM32F407VG.
- Dans la fenêtre Manage Run-Time Environment, sélectionner: Board Support → STM32F4-Discovery (à sélectionner dans le menu déroulant), CMSIS → CORE, Device → Startup
- Pour pouvoir utiliser les bibliothèques CMSIS de l'USART, cocher aussi Device → StdPeriphDrivers → USART (qui nécessite RCC et Framework) et Device → GPIO dans le menu, puis faites OK.
- Dans le menu projet à gauche, clic droit sur Source Group 1, puis Add New Item to Group 'Source Group 1' / C File (.c) / Name: main_uart.c, puis Add.
- Essayez de compiler, si vous avez l'erreur « function assert param declared implicitly », clic droit dans Target1 → Options for Target 'Target 1'... → C/C++ → Preprocessor Symbols → Define : USE_STDPERIPH_DRIVER
- Remplacer le fichier system_stm32f4xx.c par celui qui se trouve sur la page <http://users.polytech.unice.fr/~bilavarn/> rubrique Systèmes à Microprocesseurs.
- Clic droit dans Target1 → Options for Target 'Target 1'... → C/C++ → Preprocessor Symbols → Define et ajouter : HSE_VALUE=8000000.

Configuration du debugger ST LINK:

- Dans le menu Options → Debug → ST Link Debugger → Settings → Port: SW.

1.2 Transmission d'un caractère

Dans notre application, nous allons opérer à une vitesse de transmission de 9600 bauds, qui correspond en transmission numérique à 9600 bits par seconde.

Nous utiliserons le composant USART3. Il faut d'abord configurer l'UART avec les paramètres de transmission (vitesse, contrôle de flux, mode émission/réception, parité, bit de stop, taille des données transmises), puis configurer les broches émission/réception (Tx/Rx) de façon à les rediriger sur des broches prédéfinies du GPIO, ce qui permettra de les connecter au câble RS232.

A l'aide des fonctions de la bibliothèque CMSIS (stm32f4xx_rcc.h, stm32f4xx_usart.h), écrire les deux fonctions suivantes qui permettront de manipuler l'USART USART3:

- void USART3_Initialize(void) qui permet d'initialiser l'USART USART3. Les fonctions à utiliser sont:
 - RCC_APB1PeriphClockCmd pour activer l'horloge périphérique sur lequel est connectée l'USART USART3.
 - USART_Init pour configurer les paramètres de transmission: 9600 bauds, pas contrôle de flux, mode émission/réception, pas de parité, 1 bit de stop, taille des données 8 bit.
 - USART_Cmd pour activer l'USART USART3.
- GPIO_Initialize (void) qui permet de configurer et d'activer les broches Tx/Rx:
 - RCC_AHB1PeriphClockCmd pour activer l'horloge périphérique dont dépendent les broches Tx/Rx de l'USART USART3.
 - GPIO_PinAF pour activer le mode *alternate function* / USART3 du GPIO pour les broches correspondant à Tx/Rx.
 - GPIO_PinConfigure pour configurer ces broches en mode: Alternate Function, Output Push Pull, Output Speed 100MHz, et GPIO Pull up.

Ecrire ensuite un programme principal qui transmet un caractère 'A' en utilisant la fonction USART_SendData. Testez le programme.

Annexe : utilisation d'un utilitaire de communication (TeraTerm Pro)

La transmission se fait vers le port série des stations de travail via un câble RS232 et un composant MAX232 qui permet la connexion et l'adaptation des niveaux TTL – RS232.

Pour visualiser les messages sur les stations de travail, on utilisera un utilitaire de communication: Tera Term Pro, à télécharger sur <http://users.polytech.unice.fr/~bilavarn/>
Configurer le port COM1 du port série de votre station via Tera Term Pro, avec les mêmes paramètres que l'USART USART3. Configuré ainsi, les caractères envoyés par la carte STM32 sont reçus par la station et affichés dans la fenêtre Tera Term Pro.

1.3 Transmission d'un message

Transmettre un message (une chaîne de caractère) consiste à parcourir la chaîne de caractère et à transmettre chaque caractère avec la fonction `USART_SendData`. Il faut néanmoins s'assurer que la transmission du caractère précédent est bien terminée avant d'en transmettre un nouveau. Cette information est indiquée dans le registre d'état de l'USART par le bit `TXE` du registre d'état `USART_SR`.

A l'aide des fonctions de la librairie CMSIS (`stm32f4xx_usart.h`), écrire les deux fonctions suivantes qui permettront la transmission d'un message par l'USART USART3:

- `int USART3_transmitter_empty(void)` qui vaut 1 (return 1) si le registre de transmission est vide, 0 (return 0) sinon. Les fonctions à utiliser sont:
 - `USART_GetFlagStatus` pour lire le flag `TXE` du registre d'état d'une USART.
- `void USART3_puts(char *s)` qui transmet le message passé par le pointeur `*s` en utilisant les fonctions `USART_SendData` et `USART3_transmitter_empty`.

Testez ensuite le fonctionnement du programme dans son ensemble en vérifiant la transmission d'un message.

1.4 Réception d'un caractère

Recevoir un caractère consiste à attendre que l'USART signale qu'elle a reçu une donnée, puis à lire le registre de réception avec la fonction `USART_ReceiveData`.

A l'aide des fonctions de la librairie CMSIS (`stm32f4xx_usart.h`), écrire la fonction suivante:

- `int USART3_receiver_ready(void)` qui vaut 1 (return 1) si le registre de réception n'est pas vide, 0 (return 0) sinon. Les fonctions à utiliser sont:
 - `USART_GetFlagStatus` pour lire le flag `RXNE` du registre d'état d'une USART.

Ecrire ensuite un programme principal qui attend la réception d'un caractère en utilisant la fonction `USART3_receiver_ready`, qui lit le registre de réception avec la fonction `USART_ReceiveData` et enfin qui renvoie le caractère lu par la fonction `USART_SendData` pour affichage par Tera Term Pro. Testez le programme.