Mini-Projet (GL3 – 2024/25)

- Déposer (sur classroom) un fichier compressé qui contient aussi bien le projet keil que le compte rendu.
- Ne pas oublier d'indiquer sur la page de garde du compte rendu, les noms des membres ainsi que le groupe.
- Ne déposer qu'un seul fichier par équipe .
- **Date limite : 21/12/2024**

Remarques :

- Pour mieux comprendre le projet, il est possible de consulter la manip « **Manip OOP** »
- Pour le développement, utiliser comme template, le projet « manipOOP_Nuclo1_OOLayer » qui se trouve dans le répertoire « Manips ».

Sujet :

On veut développer une classe **Serial** permettant la communication série en utilisant l'une des interfaces USART disponibles sur le STM32F1.

Cette Classe devrait permettre de configurer l'un des ports séries (USART1, USART2 ou USART3) et d'envoyer et recevoir les données.

L'initialisation et la configuration du port série est réalisée dès la création d'un objet de la classe **Serial**. La déclaration complète d'un objet de classe Serial se fait de la manière suivante :

Serial MyPortCOM **(uint8_t** COMPORT, **uint8_t** PinTX, **uint8_t** PinRX ,

uint16_t Speed, **uint8_t** Config, **uint8_t** Mode)

Chacun des paramètres pouvant prendre l'une des valeurs suivantes (les **valeurs en bleu** étant celles choisies par défaut, si aucune valeur n'est passée en paramètre pour le paramètre correspondant)

COMPORT	speed	Config		Mode
COM1	**9600**	**8N1**	9N1	TxMode
COM2	19200	8N2	9N2	RxMode
COM3	115200	8E1	9E1	**TxRxMode**
		8E2	9E2	
		8O1	9O1	
		8O2	9O2	

Pour le paramètre **Config** : Le premier chiffre représente le nombre de bits par mot. La lettre indique la parité (N : none, E : Even, O : Odd) et le dernier chiffre représente le nombre de bits de stop.

Pour les pins : il faut reprendre la même représentation que celle vue en cours. PX_i (X représentant le GPIO (A, B, C ou D) et i le numéro du pin (0 à15).

Pour la déclaration de l'objet, il est possible d'avoir ces formats :

- *Serial MyPortCom ( )* : dans ce cas, il s'agira du COM2 avec les pins PA_2 (Tx) et PA_3 (Rx).
- *Serial MyPortCom (COMPORT, PinTX, PinRX)*
- *Serial MyPortCom (COMPORT, PinTX, PinRX, Speed)*
- *Serial MyPortCom (COMPORT, PinTX, PinRX, Speed, Config)*
- *Serial MyPortCom (COMPORT, PinTX, PinRX, Speed, Config, Mode)*

En plus, on veut avoir au niveau de la classe, 2 méthodes

- **printchaine** : qui permet d'envoyer une chaine de caractères en mode polling. L'appel se fait :

  *MyPortCom.**printchaine*** (unechaine, length)
- **readchaine** : qui permet de recevoir une chaine de caractères en mode polling. L'appel se fait :

  unechaine = *MyPortCom.**readchaine*** (length)

Travail demandé

Partie1 : Développement d'une classe Serial [14 pts]

Cette partie est à concevoir, développer et tester sur la carte (ou simulateur)

A- Conception et compte-rendu [10 points]

1- Commencer par donner les pins qui peuvent être utilisés par chacun des interfaces séries (USART1, USART2 et USART3) des microcontrôleurs STM32 pour la transmission et la réception.

2- Sachant qu'on peut utiliser les fonctions de la librairie de ST (ST HAL Library), donner les fonctions dont vous aurez besoin pour l'implémentation de la classe.

3- Donner l'architecture générale de la solution (les couches OO, Wrapper et HAL) et préciser ***les fichiers*** au niveau de chaque couche, les fonctions de chaque fichier ainsi que les dépendances entre fichiers (appels).

4- Donner le **rôle** de chacune des fonctions à implémenter au niveau de la couche OO et présenter le **code**.

5- Donner le rôle de chacune des fonctions à implémenter au niveau de la couche Wrapper et présenter le code.

B- <u>Implémentation et test du code [4 pts]</u>

1- Développer le code et le tester (en mode simulation) avec des exemples d'envoi et de réception de chaines de caractères. *(Pour la simulation de la communication série, consulter le dossier « simulation série ».*

2- Ajouter au compte rendu des imprime écrans qui démontrent le bon fonctionnement du système.

Pour le développement, utiliser le projet « manipOOP_Nucleo_1_OOLayer » (qui se trouve dans le répertoire « Manips ».

Partie2 : Exploitation des interruptions pour l'envoi/réception [6 pts]

Cette partie est à concevoir uniquement sans la déployer et la tester.

On veut ajouter à cette classe, 2 méthodes qui permettent d'utiliser les interruptions pour l'envoi et la réception de chaines de caractères :

- MyPortCom. **IT*printchaine*** (unechaine, length)
- unechaine = *MyPortCom*.**IT*readchaine*** (length)

1- Donner le code à placer au niveau de chacune des 2 fonctions.

2- Faut-il ajouter un autre code ? à quel niveau ? le donner si oui.