

## TP

### Communication série et Bus de terrain

## I. Objectifs

Suite aux différents travaux menés cette semaine, vous devrez être capable :

- De comprendre les notions de base d'une communication série.
- D'analyser et résoudre des problèmes simples de communication.
- De développer un protocole de communication fiable entre plusieurs équipements connectés sur un bus RS485.

## II. Découverte et étude du matériel

### A. Analyse de la carte RS485

A partir du schéma de la carte et de la documentation technique du FT231X, analyser et décrire son principe de fonctionnement.

### B. Etude dynamique

Afin de vérifier votre analyse, interconnecter deux cartes et utiliser deux émulateurs de terminaux (putty, teraterm,...) pour envoyer et recevoir des caractères.

→ Utiliser un oscilloscope ou un analyseur logique pour relever les différents signaux utiles.

Envoyer des caractères « U » (en majuscules). Relever et expliquer la forme du signal observé.

## III. Développement d'un protocole de communication

### A. Librairie de communication

A partir de la documentation « D2XX Programmer's Guide », compléter la classe « CSerial » du projet Visual Studio « communication en ajoutant des méthodes (fonctions) pour initialiser la vitesse, le format d'émission, le mode de contrôle de flux, envoyer et recevoir des trames de caractères,...

**Important :** cette classe sera la base de votre communication. Ne négligez pas la qualité de votre code, la gestion des erreurs et les messages associés. Cela vous permettra de gagner du temps dans les étapes suivantes en détectant plus facilement la source des erreurs.

## IV. Echo + 1, de a à z

En utilisant deux cartes et deux programmes, développer le principe suivant :

- Lors de l'appui sur une touche, le premier programme envoie la lettre 'a'.
- Le second programme la reçoit, l'affiche et renvoie le code reçu +1 (donc 'b').
- Le premier programme, réagit de façon similaire et le cycle continue jusqu'à la lettre 'z'.
- Un appui sur une touche permet de relancer le cycle.
- Prévoir une touche pour fermer les deux programmes avec un seul appui.

**Important :** enrichissez votre classe CSerial au fur et à mesure de vos développements sans supprimer les méthodes que vous avez mis au point.

## V. Echo de trame + 1, de a à z

Dupliquer votre projet et le modifier pour émettre avec trames de 100 caractères identiques au lieu d'un seul caractère.

Pour ne pas trop ralentir l'application, n'afficher que le premier caractère de chaque trame.

## VI. Communication multipoints RS485

On souhaite réaliser une communication fiable entre plusieurs systèmes connectés sur un bus 2 fils de type RS485.

Les caractéristiques souhaitées sont les suivantes :

- Connexion de 2 à 32 équipements.
- Communication possible du maître vers un esclave ou d'un esclave vers le maître.
- Aucun message ne doit être perdu en cas de perturbation ou coupure de la ligne de communication. Lors du rétablissement de la liaison, le message est transmis automatiquement.
- Possibilité de diffuser un message à l'ensemble des esclaves (broadcast), mais sans garantie de réception.

Au niveau logiciel, les attentes sont les suivantes :

- Un seul programme pour l'ensemble des esclaves. Vous utiliserez les paramètres de ligne de commande pour la configuration.
- Utilisation de machines à états pour développer le protocole.

Suivant votre vitesse d'avancement, vous pourrez :

- Simuler un système réel (machine, véhicule, réseau domotique, système autoroutier, ...).
- Utiliser la console ou une interface graphique.
- Détecter automatiquement l'ajout de nouveaux systèmes sur le bus (configuration automatique).

**Important :** la réussite de votre projet dépend principalement de votre analyse et de votre réflexion de départ. Il est donc indispensable de définir le format des trames et de réfléchir à la machine à états « sur papier » avant commencer la programmation.