**TP Camera groupe Garcia-Sene-Clermont**

**Sommaire**

**-Principe de fonctionnement………………………………………………………….page 1**

**-But…………………………………………………………………………….page 1**

**-Questions Préliminaires…………………………………………….page 2**

**-Programme C++…………………………………………………………page 2-7**

**-Conclusion…………………………………………………………………page 8**

**Principe :**

La caméra SONY EVI-D31 est une caméra qui se contrôle via des trames RS-232.

Chaque mouvement et fonctionnalité est affilié à une trame spécifique en hexadécimal.

**But :**

Nous avions pour objectif d’effectuer une liaison série avec la caméra, puis d’envoyer des trames via le langage cpp et la librairie ‘’winsock’’.

A partir de notre application, nous devions avoir la possibilité de déplacer la caméra dans diverses positions,de zoomer et de créer un mode de balaye automatique (latéralement).

**Questions Préliminaires :**

1. La caméra peut être déplacée latéralement (gauche et droite) à 180°, verticalement (de haut en bas). Elle peut donc être contrôlée à distance via deux ordinateurs un client et un serveur, le serveur communique avec la caméra via le protocole RS-232.
2. Elle est pilotable via des trames que l’on envoie en RS-232.

Une trame se compose d’au maximum 9 codes hexadécimales codable chacun sur un octet (ici, 2 paquets de 4bits). Le premier octet définit l’adresse de la caméra (0x81 car l’adresse de la caméra est 1).

3)  L'une des différences majeures est que dans la transmission synchrone, l'émetteur et le récepteur doivent avoir des horloges synchronisées avant la transmission des données. Alors que la transmission asynchrone ne nécessite pas d'horloge mais ajoute un bit de parité aux données avant la transmission.

4)  Le format de transmission de la caméra est de 9600 bps.

5)  le premier octet sert à spécifier l’adresse de la caméra (dans cet octet il y a le bit de

start).

Le second spécifie la catégorie du message.

Du troisième au huitième c’est le message en lui même (data).

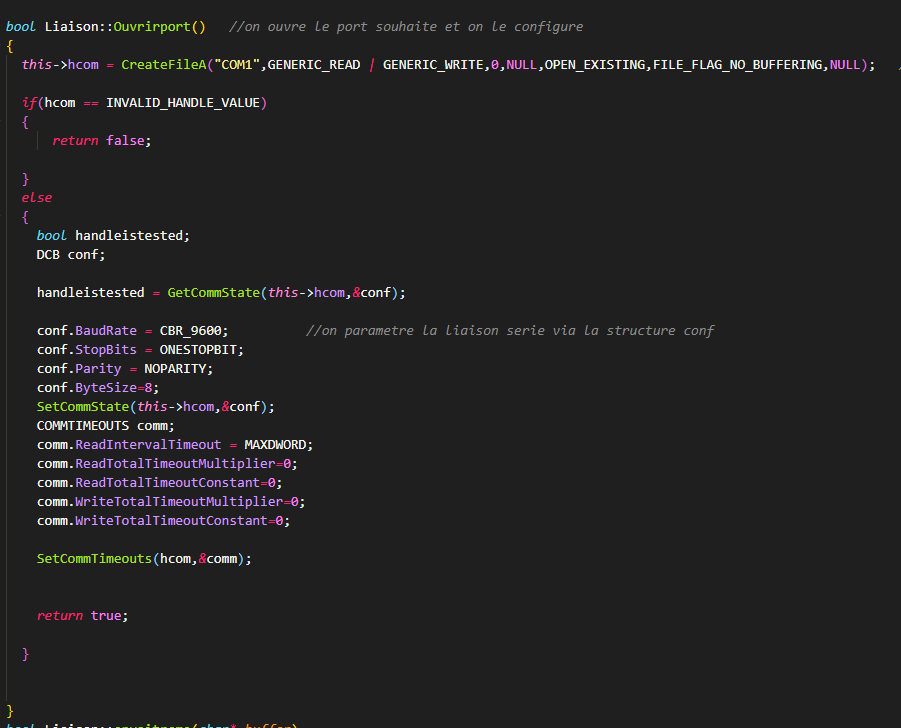
Le dernier octet signifie la fin de la communication (Terminator)

6)  Allumer : 81 01 04 00 02 FF

Eteindre : 81 01 04 00 03 FF

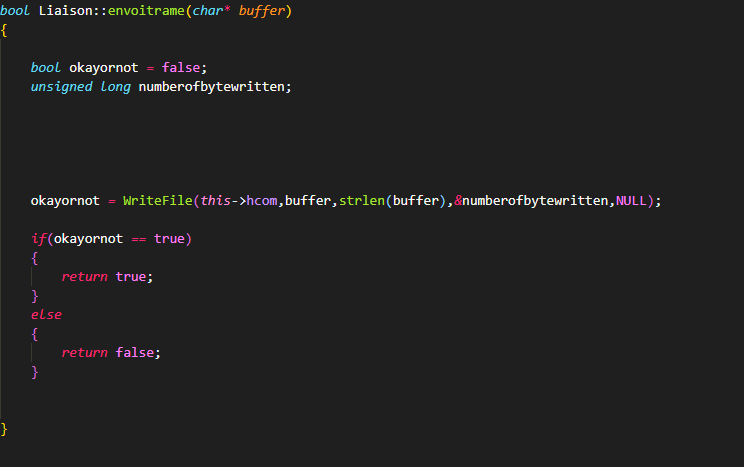
**Programme C++ :**

*ci-dessous, la partie du code permettant la liaison série*

**

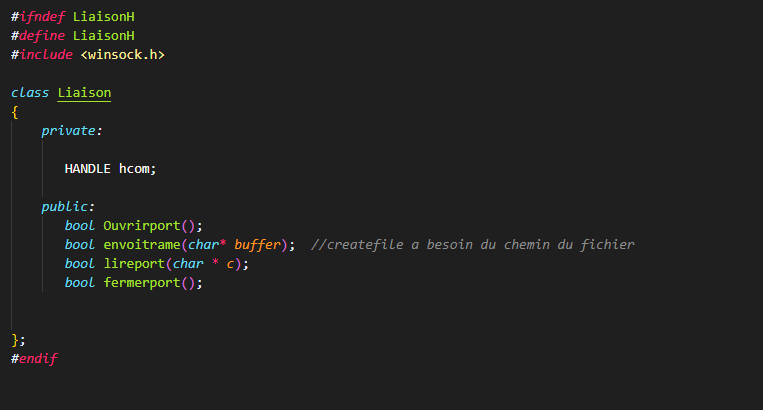
On se connecte à la camera via notre port COM1.

*ci-dessous, la partie du code permettant l’envoi d’une trame*

**

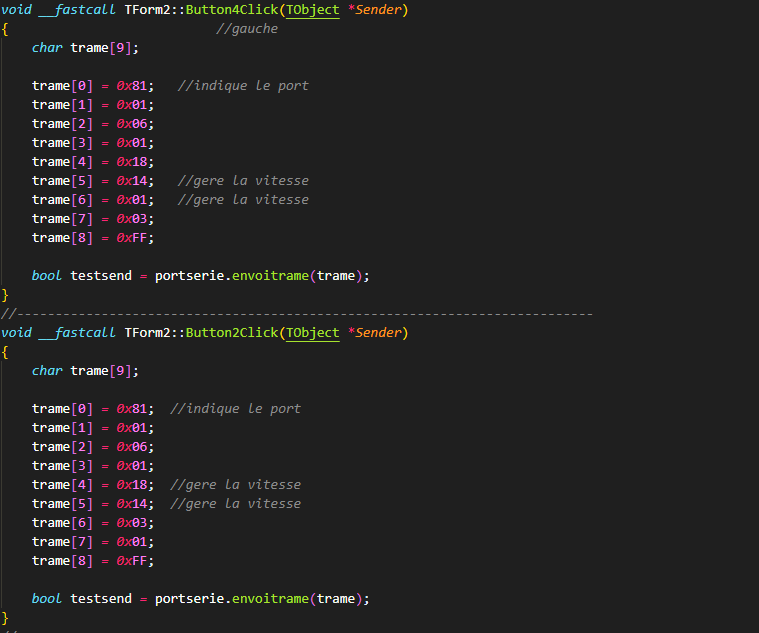
On pointe sur un tableau de caractères qu’on envoie via la fonction **WriteFile** de la librairie.

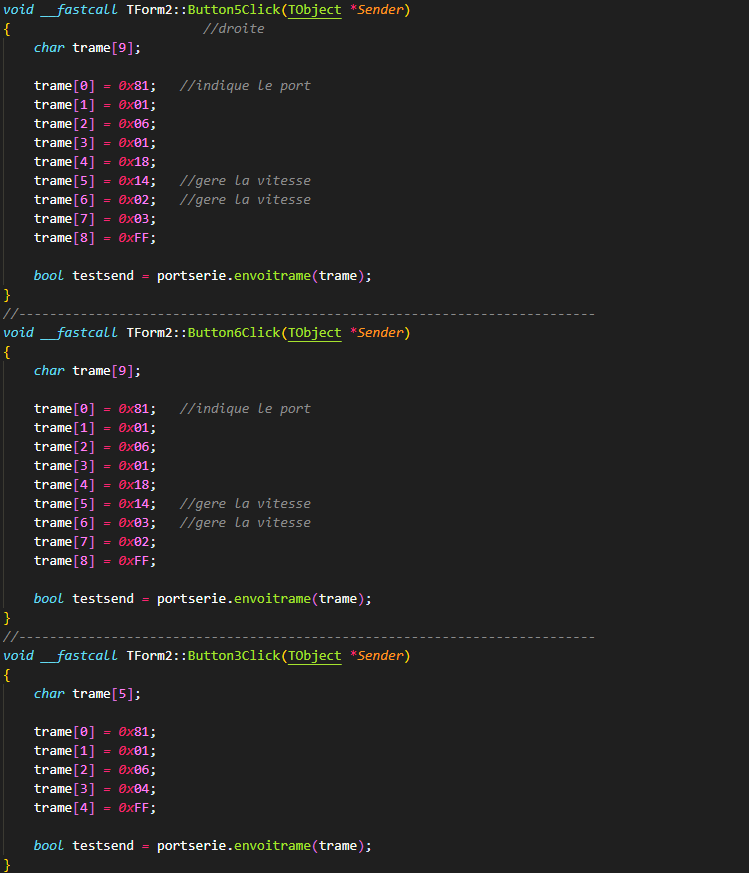
*ci-dessous, la déclaration de notre classe*

**

La plupart des fonctions de la librairie Windows renvoie une variable de type **HANDLE**, c’est pour cela que la variable hcom est attribut de notre classe.

*ci-dessous, une partie du code appelé via l’IHM*

**

**

Chaque commande de la documentation correspond à une trame. Pour appliquer cela à notre programme, nous avons crée plusieurs boutons en IHM qui sont chacun relié à un bout de code qui va envoyer une trame spécifique dans chaque cas.

Ici dans les différents tableaux de caractères, on affecte chaque octet à un indice du tableau.

La trame est donc préconstruite à chaque fois, ce qui est très pratique.

**Conclusion:**

Ce TP nous a beaucoup plu, nous avons trouvé qu’il était très concret et qu’il permettait une bonne compréhension du protocole RS-232 (avec le découpage de la trame).

La caméra SONY EVI-D31 est remplie de fonctionnalités, cependant nous avons eu du mal à nous repérer dans la documentation, parfois assez succincte…

En guise d’amélioration, nous avons pensé au fait de zoomer et dézoomer selon une valeur définie et choisie par l’utilisateur, et non jusqu’au maximum ou minimum que peut attendre l’appareil.