

Import de données sérielles sous MatLab

Et

Programmation Multi Thread de data login synchrone.

Cette note d'application va répondre à la question suivante:
Comment acquérir et dater des données en provenance d'un port
série RS232 sur un PC ?

Introduction :

Les deux principales problématiques abordées dans cette note d'application sont l'accès aux ports série sous un environnement de travail Microsoft (en l'occurrence Windows XP SP2) et l'obtention sous ce même système d'exploitation des informations de synchronisation temporelle plus précises qu'à la seconde près. La principale difficulté réside dans le fait que l'accès à ces ressources est protégé par le noyau du système d'exploitation et donc il nous est nécessaire d'utiliser les bonnes API.

Accès au(x) port(s) série du PC :

Il est possible d'accéder aux ports séries d'un PC directement avec un logiciel du type Matlab.

```
% Créer le port série
obj1 = serial('COM4');
% Sélectionner le débit
obj1.BaudRate = 4800;
% Se connecter au port créé
fopen(obj1);

% Il est ensuite possible de réceptionner les trames reçues sur le port série
data = fscanf(obj1);

% Pour finir, ne pas oublier de libérer le port
fclose(obj1);
```

Cependant, je déconseille vivement cette méthode pour une application de data login d'un grand nombre de données car cette méthode n'est pas très « optimisée », elle provoque entre autres des blocages du logiciel, il est même indispensable d'installer un patch pour l'utiliser sur certaines versions de Matlab (cf: premier lien de la bibliographie).

Pour accéder aux ports série même sous un environnement de type Windows, il est donc aussi rapide et efficace de développer une application (en C/C++) personnelle (Ou utiliser un logiciel de data login du commerce ... Excepté pour une simple visualisation ou un envoi de trames où, un logiciel du type « hyper-terminal » est suffisant). L'accès aux ports se fait par le biais de fonctions nécessitant d'inclure les fichiers <windows.h> et <winbase.h> dans le programme réalisé. Toutes les informations pour utiliser les fonctions disponibles sont présentées à l'adresse suivante :

[http://msdn.microsoft.com/en-us/library/aa363196\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa363196(VS.85).aspx)

Cette méthode fonctionne en développant avec l'IDE Code::Blocks et le compilateur MinGW qui est une adaptation du compilateur GCC.

Datation des données :

Une fois les données réceptionnées, il faut pouvoir les dater. Cependant, par défaut la datation n'est précise qu'à la seconde près. Pour obtenir une datation plus précise, il est là encore nécessaire de faire appel aux bonnes API. Pour ce faire, il faut utiliser la bibliothèque « winmm ». Il sera alors possible d'utiliser un référentiel temporel précis à la milli seconde près en utilisant par exemple la fonction « timeGetTime » (cf :[http://msdn.microsoft.com/en-us/library/ms713418\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms713418(VS.85).aspx)). La fonction précédemment citée retourne le temps depuis lequel la session courante est active, il sera donc possible de connaître la durée précise (à la milli seconde près) entre la réception de deux trames par exemple.

Programmation de l'application de data login :

Nous avons maintenant toutes les cartes en main pour développer une application de data login. Pour une application devant dater des données en provenance de différents ports, et devant synchroniser ces différentes données, il est judicieux d'utiliser une programmation Multi Threads. En effet, la création d'un Tread principal pilotant l'acquisition et d'un Thread pour chaque source de données facilite le développement et l'utilisation de l'application, dans le cas où le nombre de ports série à enregistrer peut être différent pour chaque campagne de relevés de données.

La date de référence pour chaque donnée sera le moment de son entrée sur le port série (Cette datation ne sera pas totalement correcte car cette date ne correspond pas au moment de la création de la donnée dans le périphérique, mais en considérant ce temps peu différent dans chaque périphérique, voir même en négligeant ces temps de « propagation » de l'information, cette datation peut être acceptable).

Conclusion :

Cette note d'application est le fruit de plusieurs heures de recherches sur Internet ayant proposées un grand nombre de possibilités pas toujours simple à mettre en œuvre ou incompatibles entre elles, pour créer l'application finale. Cette note d'application permet donc de mettre en œuvre facilement et rapidement une application de data login synchrone de données sérielles.

Références

Patch de mise à jour pour Matlab :

- <http://www.mathworks.com/support/solutions/data/1-1AQN3.html?solution=1-1AQN3>

IDE pour développement d'application C/C++ :

- <http://www.codeblocks.org/>

Compilateur C/C++ Windows :

- <http://www.mingw.org/>

Accès aux ports série sous Microsoft Windows :

- [http://msdn.microsoft.com/en-us/library/aa363196\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa363196(VS.85).aspx)

Accès au temps système sous Microsoft Windows :

- [http://msdn.microsoft.com/en-us/library/ms713418\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms713418(VS.85).aspx)