

Alpesview

Fabien Marco

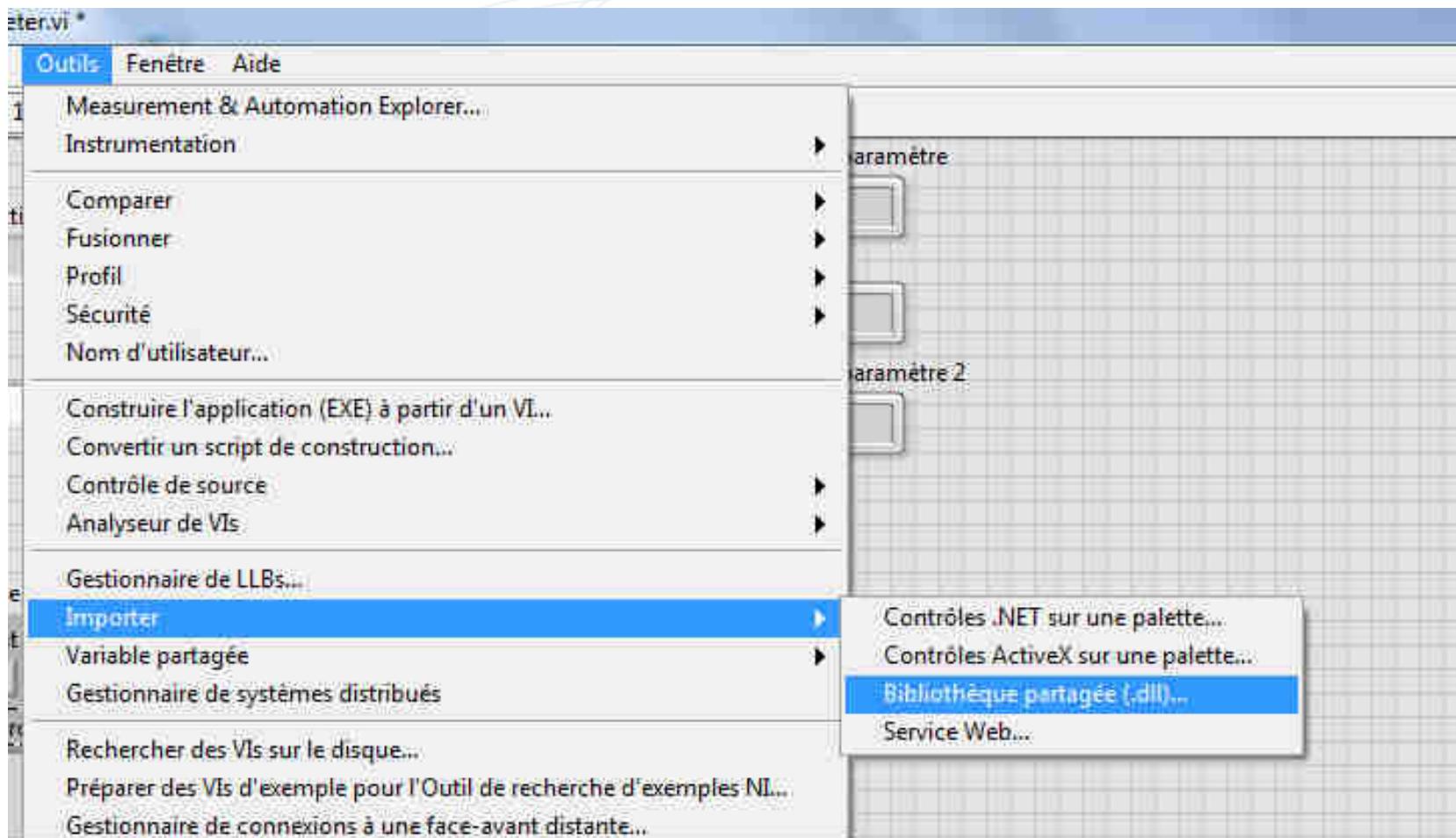
Création d'un pilote

- Il est possible de créer automatiquement un pilote, afin d'utiliser un instrument sous l'environnement de développement Labview.
- La présentation qui suit, a été créée pour une carte humusoft.

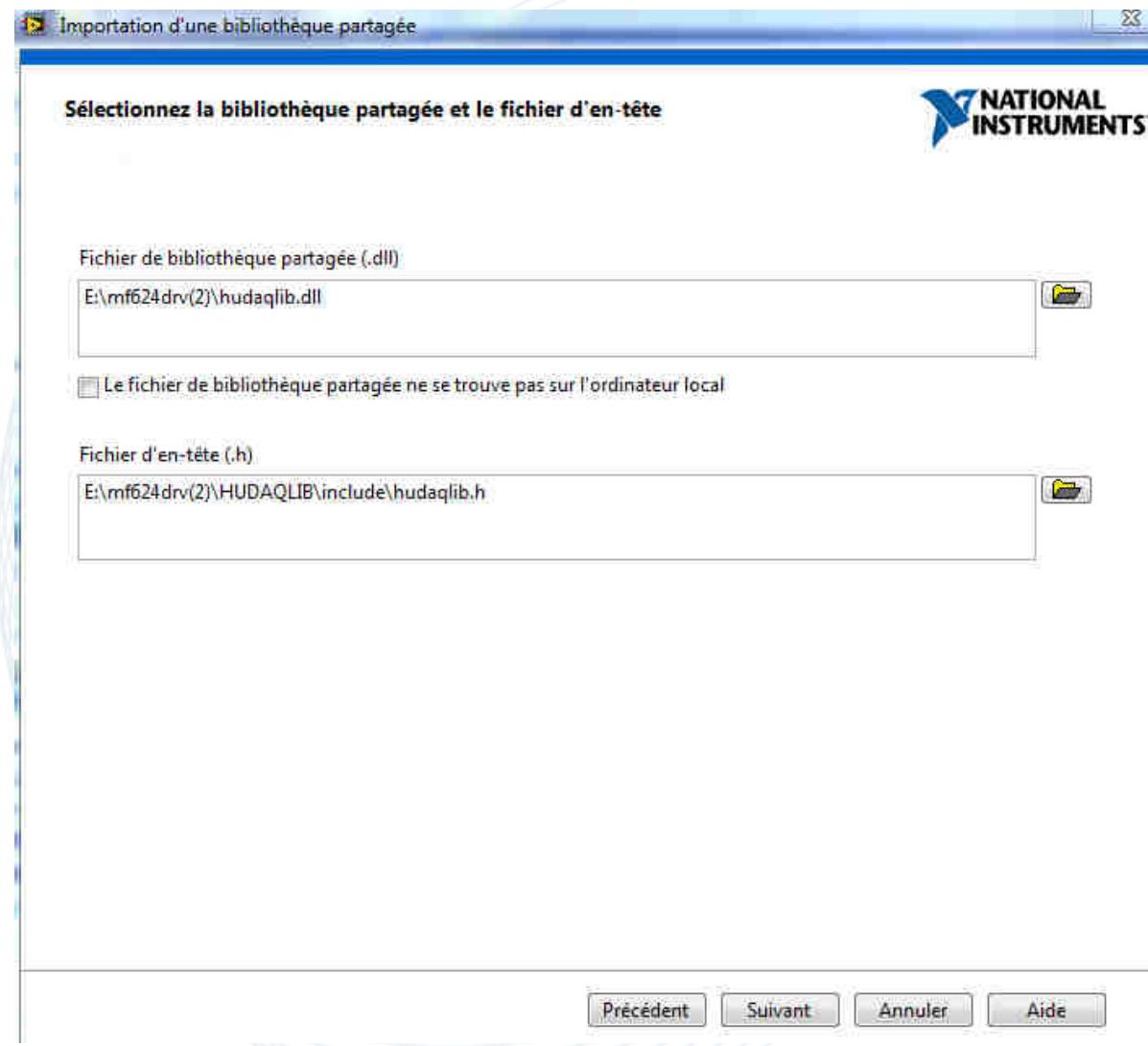
Document nécessaire pour générer des Vis d'instrumentation

- La dll compilé pour son système (32 ou 64 bit)
- L'entête *.h

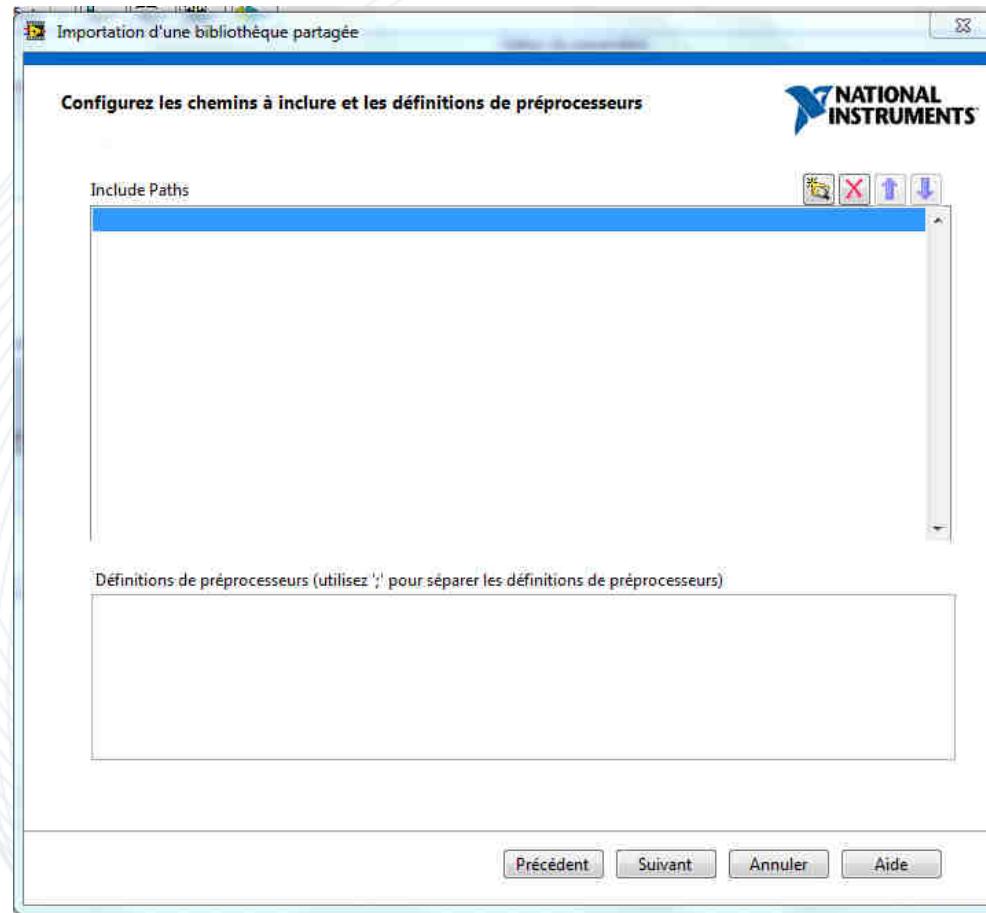
Dans labview ?



Que doit t-on faire ? ^_^, enfin si tout va bien ... ici, c'est une carte humusoft ad622



Directive de compilation, une étape pas forcément nécessaire..



Génération des Vis du périphérique (ici notre carte d'acquisition)..... Echec, on ne peut sélectionner aucun vi..

Sélectionnez les fonctions à convertir



La bibliothèque partagée contient 29 fonctions. Cependant aucune fonction n'est trouvée et reconnue dans le fichier d'en-tête. Les fonctions suivantes ne peuvent pas être encapsulées. Si vous voulez importer ces fonctions, consultez les messages de mise en garde à côté des fonctions ci-dessous. Vous devrez corriger les problèmes pour pouvoir continuer dans l'assistant.

- X ClassInstaller()
- X GetBoardIOAddress()
- X HdaaqAIRead()
- X HdaaqAIReadMultiple()
- X HdaaqAOWrite()
- X HdaaqAOWriteMultiple()
- X HdaaqCloseDevice()
- X HdaaqCtrLoad()
- X HdaaqCtrRead()
- X HdaaqCtrReset()
- X HdaaqCtrStart()
- X HdaaqCtrStop()
- X HdaaqDILoad()
- X HdaaqDIReadBit()
- X HdaaqDIReadMultiple()
- X HdaaqDOWrite()
- X HdaaqDOWriteBit()
- X HdaaqDOWriteMultiple()
- X HdaaqDOWriteMultipleBits()
- X HdaaqEncRead()
- X HdaaqEncReset()
- X HdaaqGetDeviceResources()
- ✓ HdaaqGetParameter()

 Tout cocher Tout décocher Précédent Suivant Annuler Aide

Utilisation des préprocesseurs, afin de guider labview.

- Labview ne reconnaît que la biblio standard.h
- Il faut parfois préciser le système sur lequel vous travaillez :
 - _WIN32 Pour un système 32
 - _WIN64 Pour un système 64
- Si la biblio est écrite en c++, on peut ajouter la balise suivante : --cplusplus
- On sépare chaque balise, par des « ; »

Allons vérifier notre fichier d'entête ... Voir si un détail nous avait échappé.

General Purpose Constants and Data Types

This section documents constants and data types used throughout the Humusoft Data Acquisition Library. [More...](#)

Typedefs

`typedef size_t HUDAQHANDLE`
The HUDAQ device handle data type.

Enumerations

`enum HUDAQSTATUS { HUDAQSUCCESS = 0 }`
Return codes for HUDAQ functions. [More...](#)

Detailed Description

This section documents constants and data types used throughout the Humusoft Data Acquisition Library.

Enumeration Type Documentation

enum HUDAQSTATUS

Return codes for HUDAQ functions.

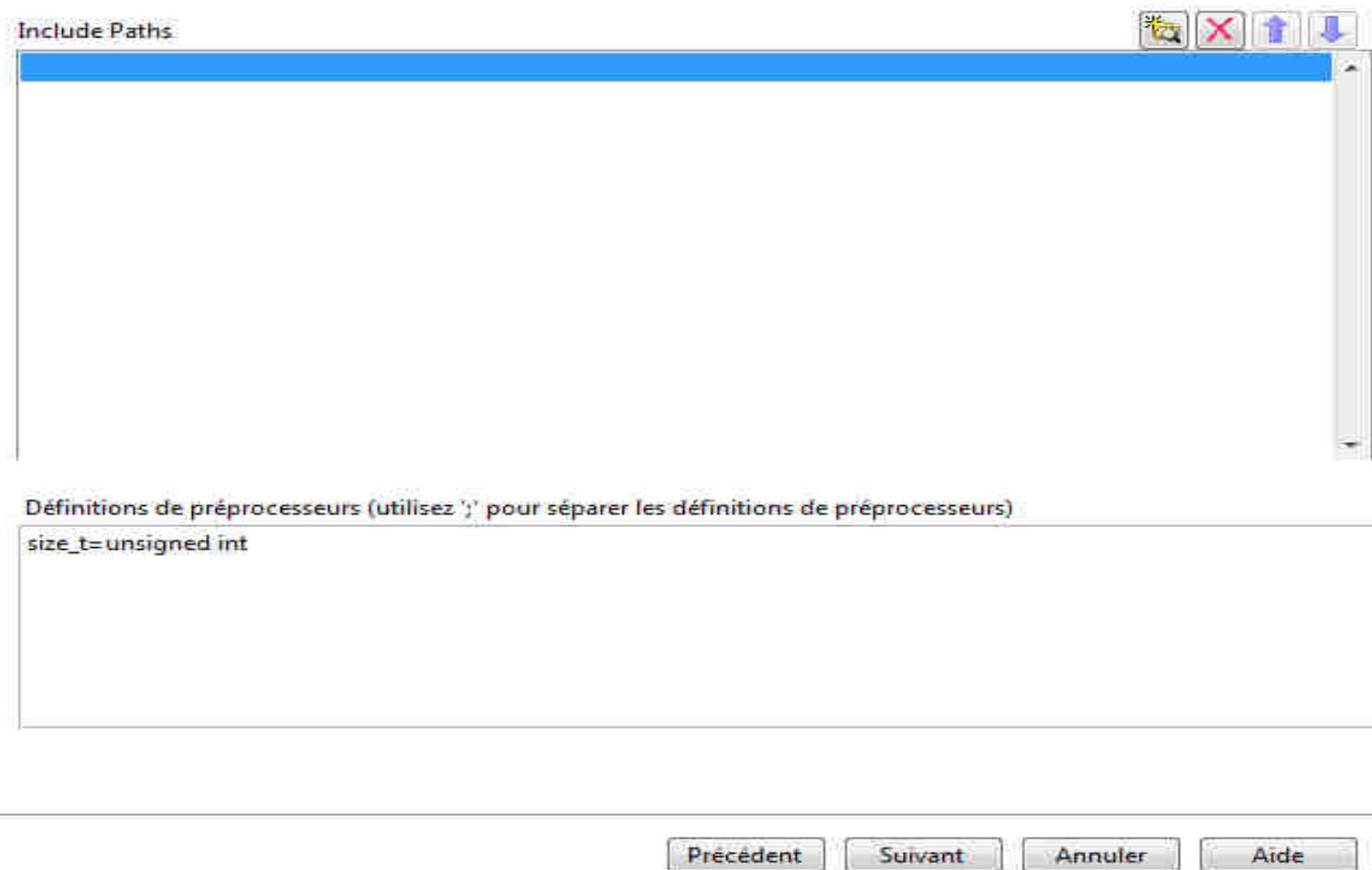
Enumerator:

`HUDAQSUCCESS` Success, All other values mean failure.

Handle ?

- C'est un numéro d'identification qui va permettre (d'où le t en entier non signé) au programme de manipuler un évènement (ici la carte d'acquisition). Il est donc nécessairement implémenter dans tous le code de la DLL.

Nous allons ajouter ici notre t, qui définit « handle » (nécessaire pour chaque prototype de la fonction)



Et ? Ca fonctionne...

Sélectionnez les fonctions à convertir



La bibliothèque partagée contient 29 fonctions. Les déclarations de 27 fonctions sont présentes et reconnues dans le fichier d'en-tête, et ces fonctions peuvent être encapsulées. Les autres fonctions ne peuvent pas l'être. Si vous voulez importer ces fonctions, consultez les messages de mise en garde à côté des fonctions ci-dessous. Vous devrez corriger les problèmes pour pouvoir continuer dans l'assistant.

- HudaqCtrReset ()
- HudaqCtrStart ()
- HudaqCtrStop ()
- HudaqDILRead ()
- HudaqDILReadBit ()
- HudaqDILReadMultiple ()
- HudaqDOWrite ()
- HudaqDOWriteBit ()
- HudaqDOWriteMultiple ()
- HudaqDOWriteMultipleBits ()
- HudaqEncRead ()
- HudaqEncReset ()
- HudaqGetDeviceResources ()
- HudaqGetParameter ()
- HudaqOpenDevice ()
- HudaqPWMWrite ()
- HudaqQueryRange ()
- HudaqResetDevice ()
- HudaqSetParameter ()
- HudaqStepWrite ()
- ClassInstaller ()
- GetBoardIOAddress ()

HudaqAIRead
double * HudaqAIRead(unsigned long handle, unsigned long channel);

Tout cocher

Tout décocher

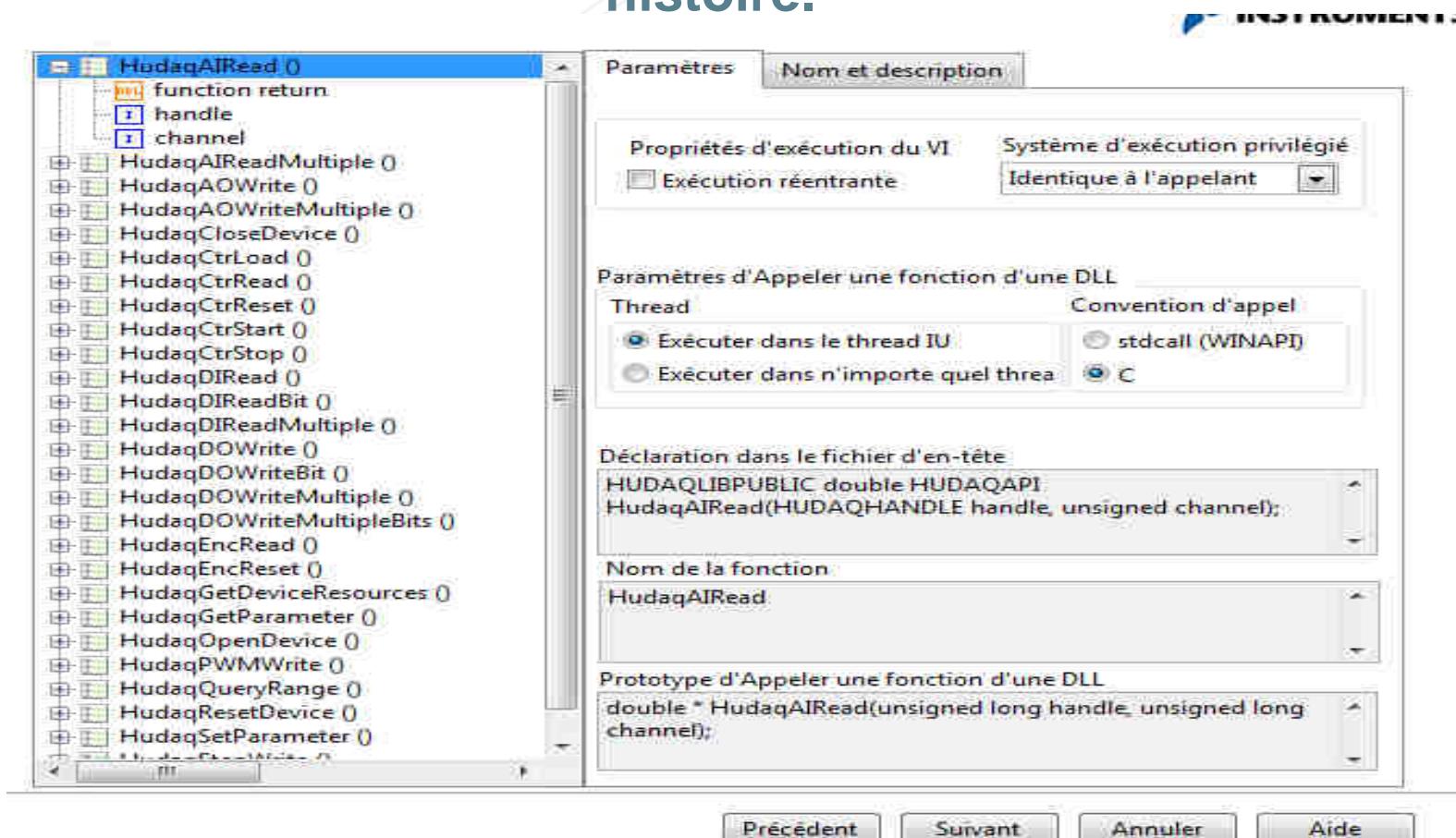
Précédent

Suivant

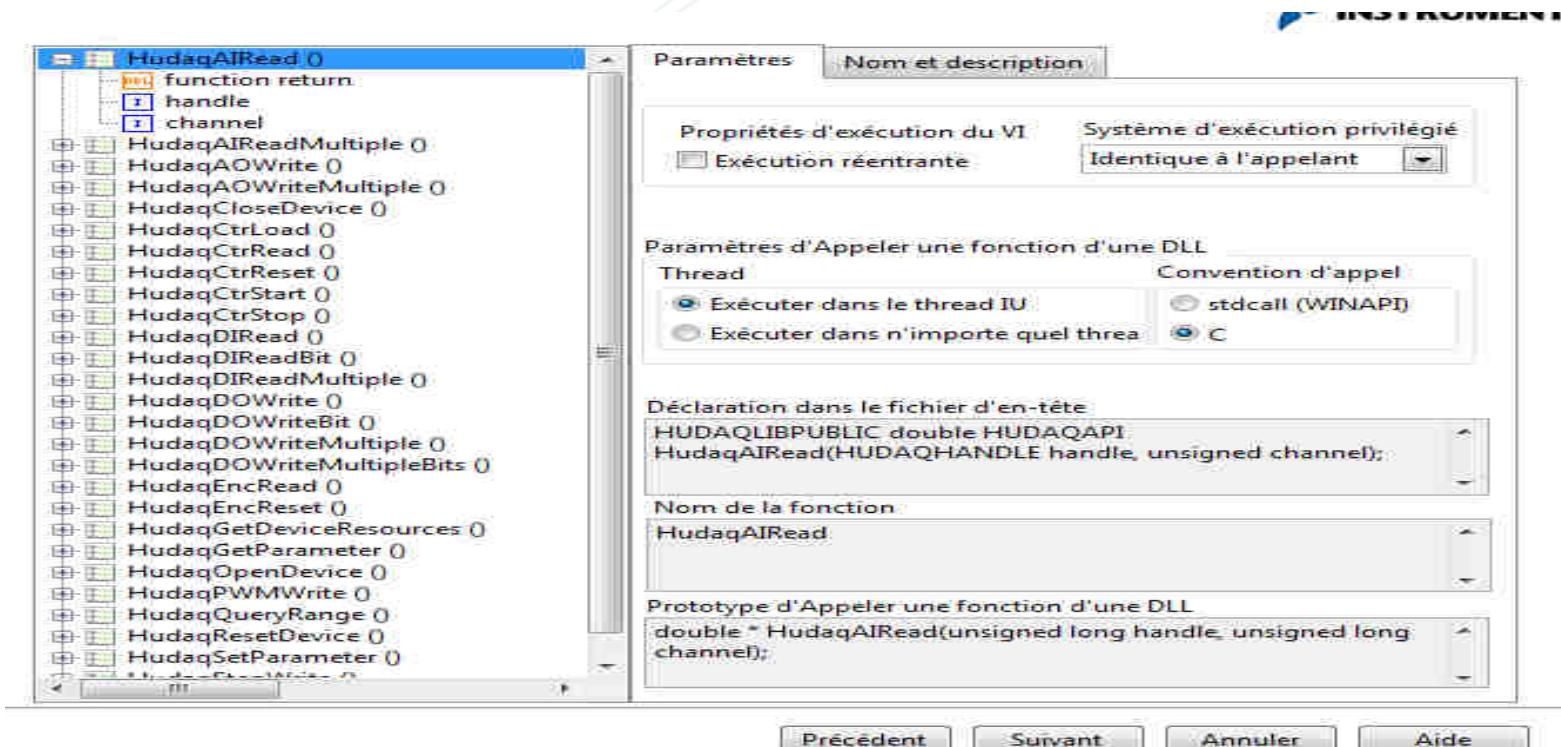
Annuler

Aide

Ici on va sélectionner chaque vi, tous ne sont peut être pas nécessaires pour nos besoins, mais cela est une autre histoire.



Reste à configurer les fonctions que l'on a choisies d'importer.



Functions

double	HudaqGetParameter (HUDAQHANDLE handle, unsigned channel, HudaqParameter param)
HUDAQSTATUS	Reads single channel configuration for a given subsystem.
HUDAQSTATUS	HudaqSetParameter (HUDAQHANDLE handle, unsigned channel, HudaqParameter param, double value)
const HudaqRange *	Configures single channel of a given subsystem.
const HudaqRange *	HudaqQueryRange (HUDAQHANDLE handle, HudaqSubsystem S, unsigned item)
const HudaqRange *	Query voltage ranges by their indices.

Choix à retenir

- La convention d'appel de base est C, mais cela peut entraîner un plantage de labview (au même titre que Winapi,stdcall). Il faut tester ...
- Nom de la fonction, si celui-ci généré ne vous paraît pas jolie, utile, ou bien parce que c'est comme ça.
- Le prototype provient directement du fichier d'entête.

Normalement à l'issu de l'importation, un rapport est émis par Labview afin de vérifier que toute l'exportation a fonctionné. Mais parfois tous ne fonctionne pas 😞

Rapport d'importation de bibliothèque partagée

mercredi 29 janvier 2014 20:00:39

Les fichiers générés sont installés dans le dossier suivant :

C:\Program Files (x86)\National Instruments\LabVIEW 2013\user.lib\hudaqlib

Mises en garde d'analyse du fichier d'en-tête :

Aucune erreur ni mise en garde à l'analyse du fichier d'en-tête.

Les erreurs et mises en garde suivantes sont survenues lors de la génération des VIs wrapper de cette bibliothèque partagée.

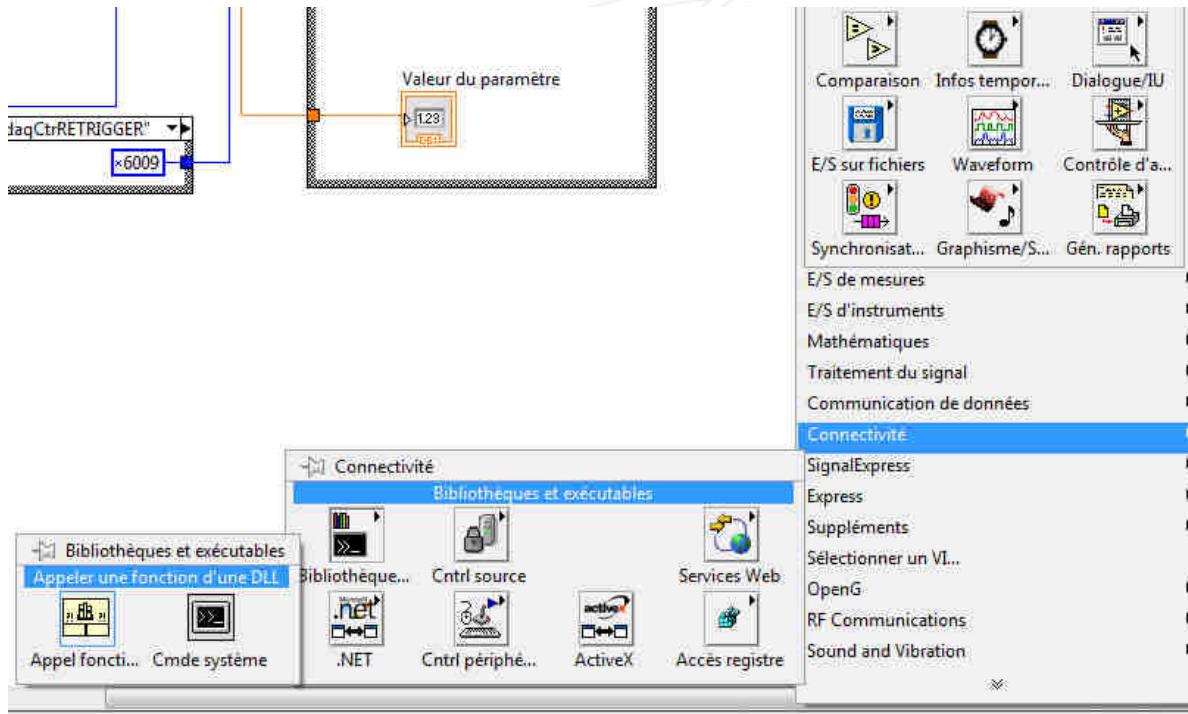
Impossible de créer le VI

Le VI suivant ne peut pas être créé. Ceci peut indiquer que la fonction associée comporte des paramètres dont le type de données ne peut pas être converti directement. Pour corriger ce problème, vous pouvez créer une commande ou une commande de définition de type personnalisée pour représenter une structure complexe ou un tableau multidimensionnel puis relancer l'Assistant Importer une bibliothèque partagée et sélectionner Mettre à jour les VIs d'une bibliothèque partagée. Avec cette bibliothèque partagée, passez de nouveau par les étapes de l'assistant. Sur la page Configurez les VIs et les commandes, assignez la commande personnalisée ou la définition de type aux paramètres associés. Les structures complexes comportent des structures imbriquées, des structures qui contiennent des tableaux, des tableaux de chaînes et des tableaux multidimensionnels.

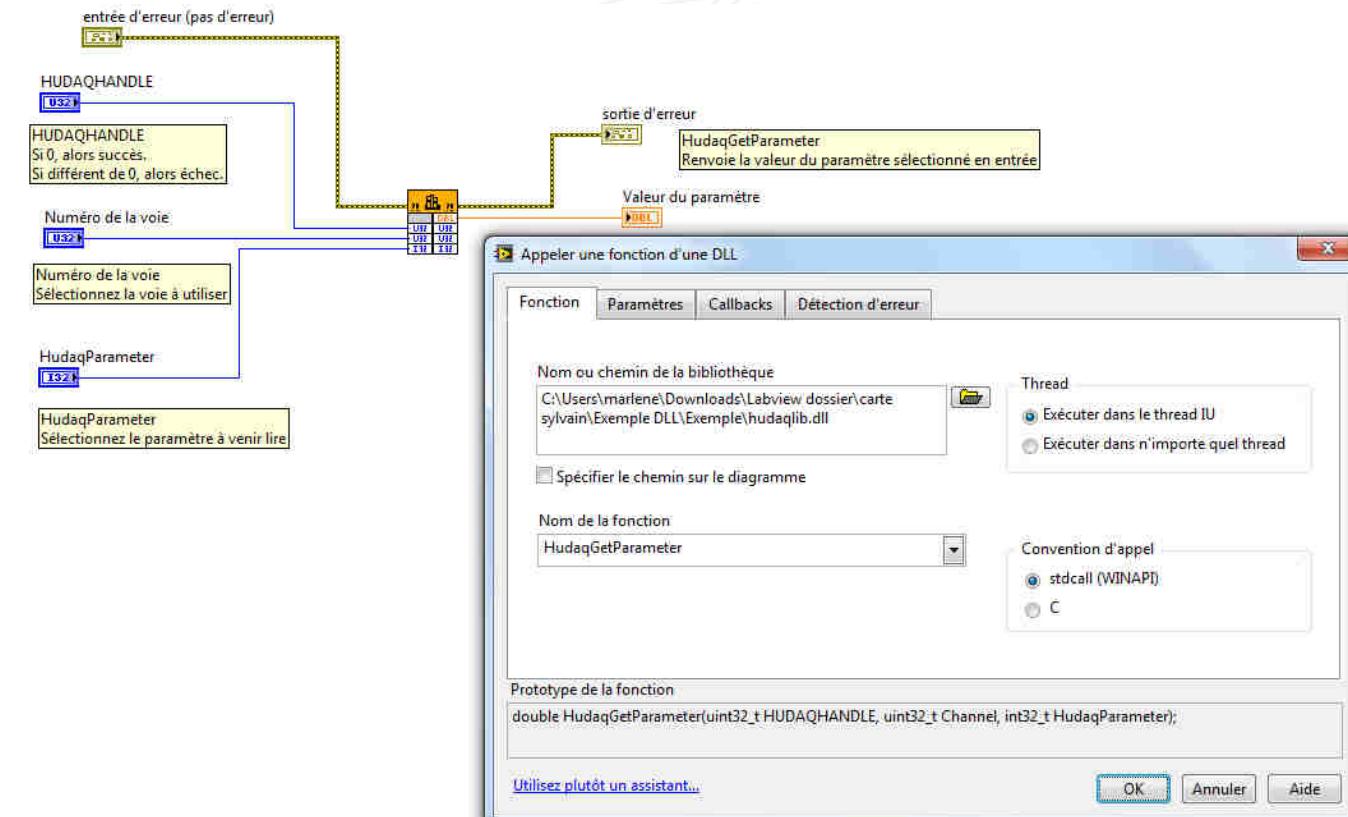
Hudaq Get Parameter.vi

Hudaq Set Parameter.vi

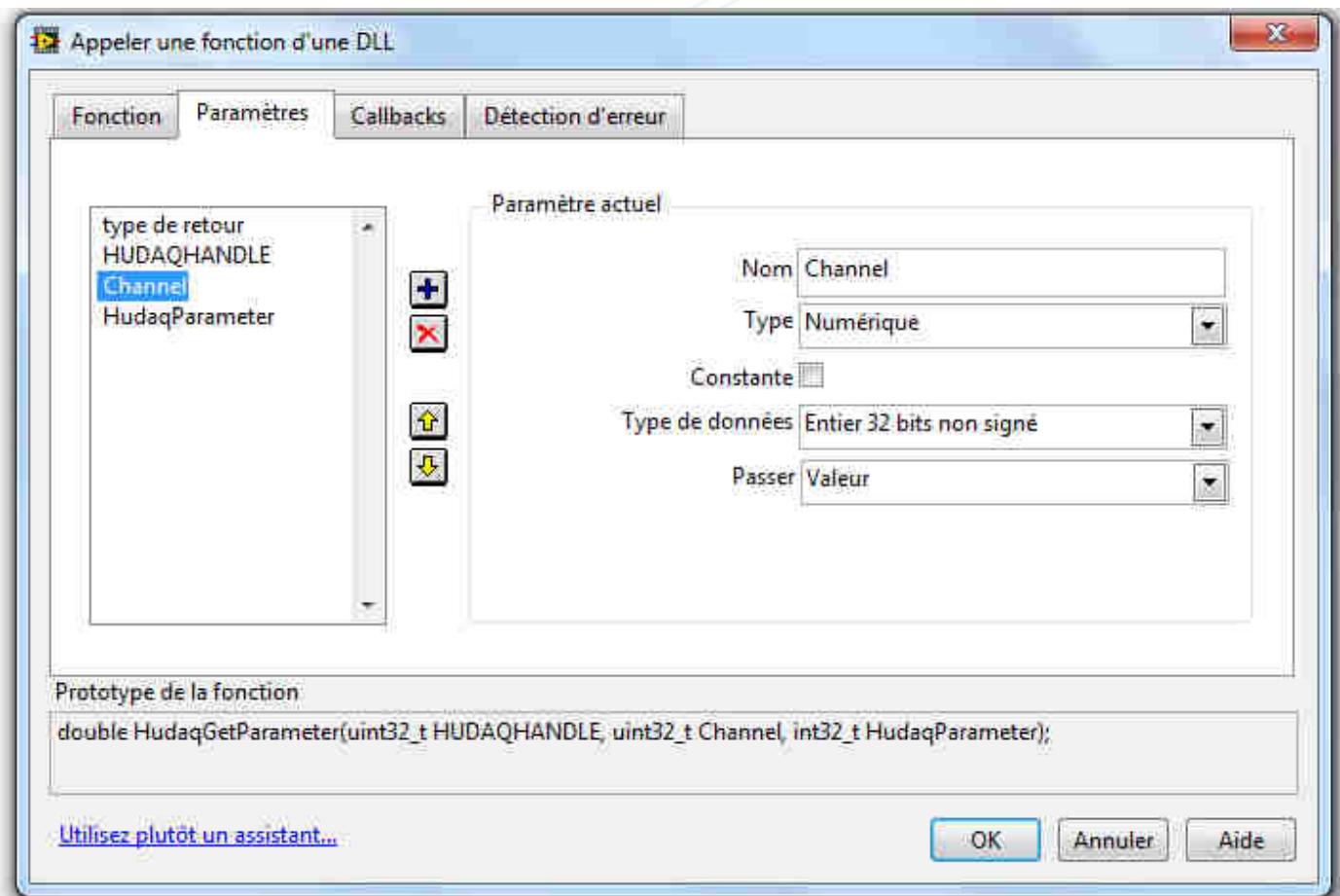
Il faut donc créer notre dll pour getparameter et set parameter...



Ouverture de Getparameter



Configuration du prototype.



Regardons plus en avant le prototype de la fonction Get parameter.

```
double HudaqGetParameter ( HUDAQHANDLE handle,  
                           unsigned      channel,  
                           HudaqParameter param  
                         )
```

Reads single channel configuration for a given subsystem.

Not all devices support all the parameters.

Parameters:

[in] **handle** Device handle.

[in] **channel** Channel number. The channel type is determined by the parameter identifier **param**.

[in] **param** Parameter identifier; see [HudaqParameter](#) for possible values.

Returns:

value Value of the parameter; see individual parameter descriptions for description of the values.

- Il nous faut donc définir le paramètre **hudaqParameter** qui est ici une liste d'énumération.
- Une liste d'énumération pourra être remplacer par un contrôle personnalisé.

Que faut il mettre dans notre liste ?

enum HudaqParameter

Parameter identifiers.

They are used as the third parameter to [HudaqGetParameter](#) and [HudaqSetParameter](#) functions to specify which parameter should be configured. Please note that a particular device does not necessarily support all functions.

Enumerator:

HudaqAI RANGE

Select analog input voltage range.

The range is selected by its index; the actual voltages corresponding to the range can be obtained by calling [HudaqQueryRange](#).

HudaqAO RANGE

Select analog output voltage range.

The range is selected by its index; the actual voltages corresponding to the range can be obtained by calling [HudaqQueryRange](#).

HudaqEncRESETONREAD

Automatically reset encoder pulse count after it is read; possible values are 0 (off) or 1 (on).

HudaqEncFILTER

Filter encoder inputs with a lowpass filter; possible values are 0 (off) or 1 (on).

HudaqEncMODE

Encoder mode; for possible values see [HudaqEncMode](#).

HudaqEncCOUNTCONTROL

Encoder count control; for possible values see [HudaqEncCountControl](#).

HudaqEncRESETMODE

Encoder reset mode; for possible values see [HudaqEncResetMode](#).

HudaqCtrRESETONREAD

Automatically reset counter pulse count after it is read; possible values are 0 (off) or 1 (on).

HudaqCtrCLOCKSOURCE

Counter clock source; for possible values see [HudaqCtrClockSource](#).

HudaqCtrOUTPUTCONTROL

Counter output signal; for possible values see [HudaqCtrOutputControl](#).

HudaqCtrREPETITION

0 - counter stops after terminal count; 1 - counter reloads and continues counting.

HudaqCtrLOADTOGGLE

0 - always load from register A; 1 - alternate load registers A and B.

HudaqCtrDIRECTION

0 - counter counts down; 1 - counter counts up.

HudaqCtrOUTTOGGLE

0 - output is directly connected to TC; 1 - use flip flop that is toggled on every TC.

HudaqCtrTRIGSOURCE

Counter trigger source; for possible values see [HudaqCtrTrigSource](#).

HudaqCtrTRIGTYPE

Counter trigger edge; for possible values see [HudaqCtrTrigType](#).

HudaqCtrRETRIGGER

Counter can be retriggered: 0 - only when stopped; 1 - anytime.

HudaqCtrGATESOURCE

Counter gate source; for possible values see [HudaqCtrGateSource](#).

HudaqCtrGATEPOLARITY

Counter gate polarity: 0 - gate low disables counting; 1 - gate high disables counting.

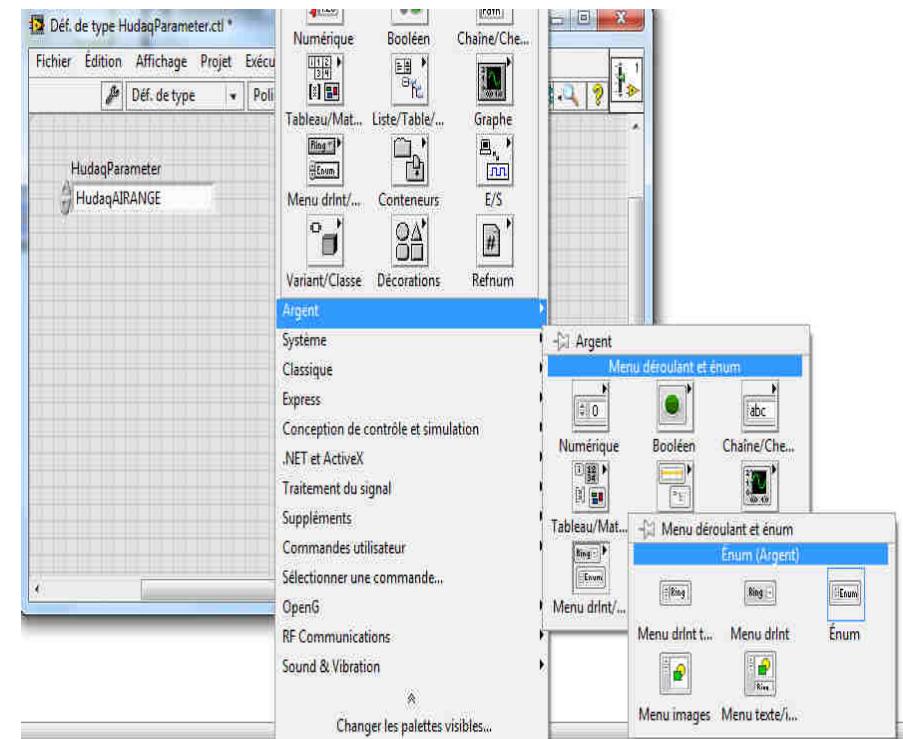
- Voici les éléments qui peuvent placer dans Hudaqparameter.
- Tous n'est pas obligatoire, il faut prendre ceux que l'on a besoin

2 choix, pour recréer cette liste, existent :

- Par liste d'enum déroulante en définition de type
- Par liste d'enum classique associé à une structure de choix

Création de la liste d'enum

- 1 Mettre une liste n'enum
- 2 ouvrir la définition de type
- 3 enregistrer



Dans un driver d'instrumentation chacun des paramètres sont en faites des adresses mémoires (un registre) du périphérique

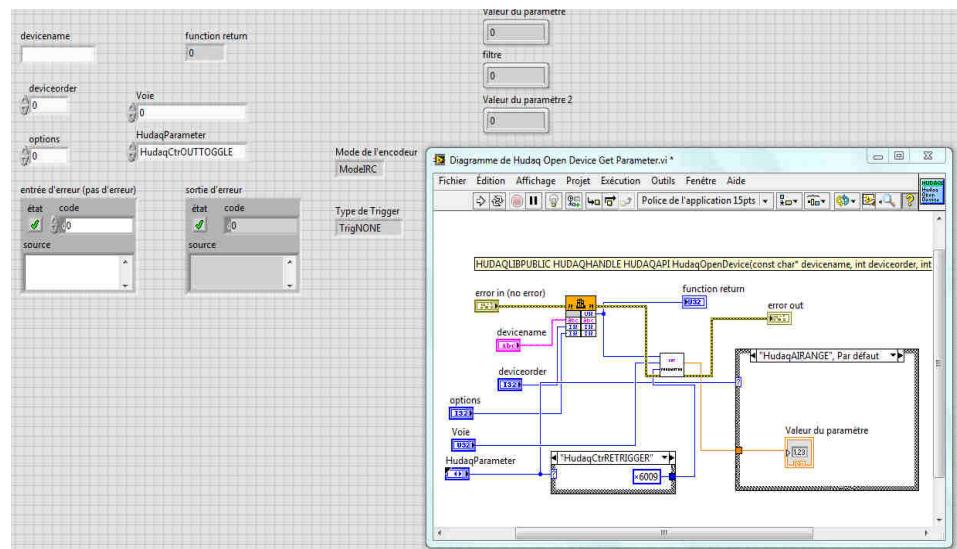
- Les adresses sont souvent données en Hexadécimal (on le reconnaît aisément au Ox....)

enum

```
HudaqSubsystem {  
    HudaqAI = 0x1000,  
    HudaqAO = 0x2000,  
    HudaqDI = 0x3000,  
    HudaqDO = 0x4000,  
    HudaqEnc = 0x5000,  
    HudaqCtr = 0x6000,  
    HudaqPWM = 0x7000  
}  
Subsystem identifiers. More
```

Présentation de Open device, qui se sert lui-même de Get parameter.

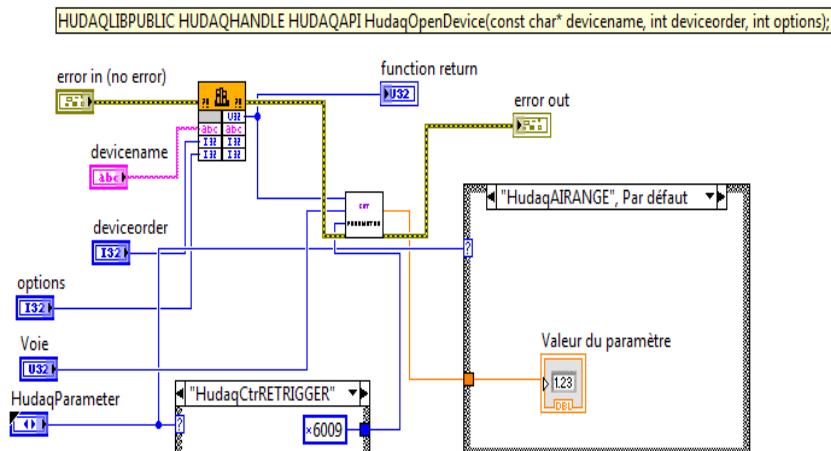
- La liste d'énum est mise en entrée d'une structure de choix, celle-ci va donc créer automatique « un choix » pour chaque enum de la liste *.ctrl



Exemple d'utilisation de hudaqparameter (c'est presque fini...)

Functions

double	HudaqGetParameter (HUDAQHANDLE handle, unsigned channel, HudaqParameter param) Reads single channel configuration for a given subsystem.
HUDAQSTATUS	HudaqSetParameter (HUDAQHANDLE handle, unsigned channel, HudaqParameter param, double value) Configures single channel of a given subsystem.
const HudaqRange *	HudaqQueryRange (HUDAQHANDLE handle, HudaqSubsystem S, unsigned item) Query voltage ranges by their indices.



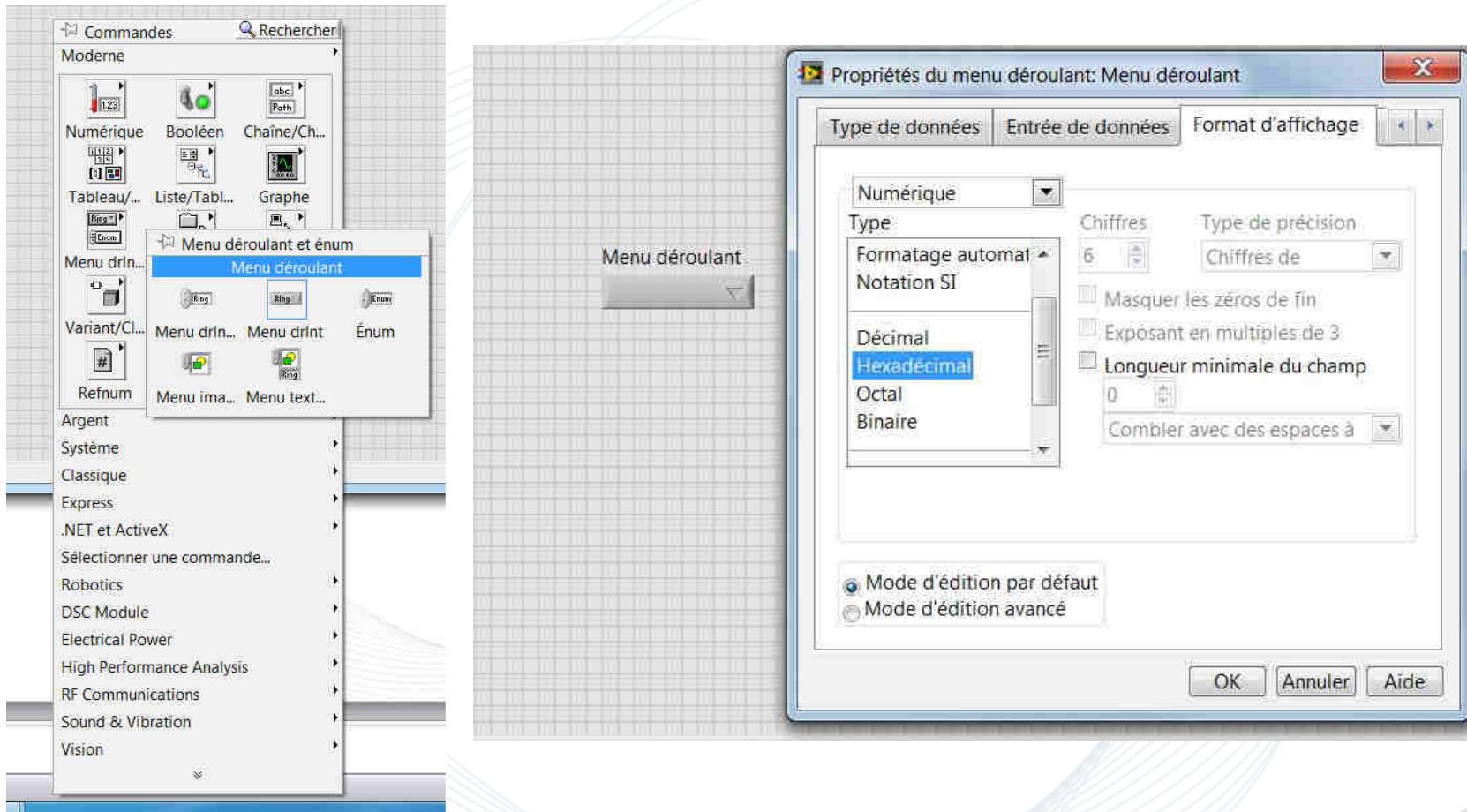
- Les entrées de hudaqparameter sont donc :

Le handle

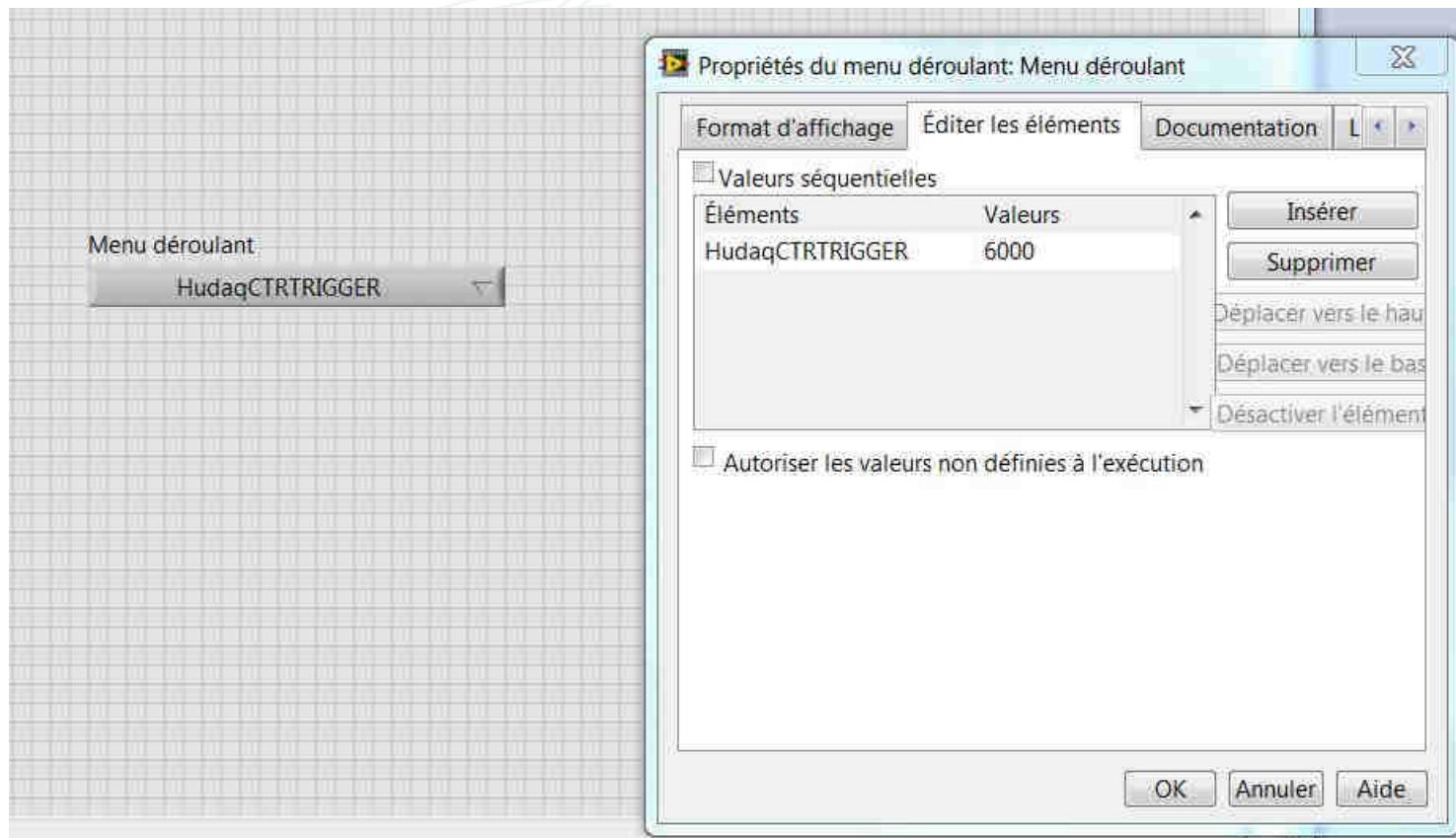
Les voies

Et hudaqparameter

Avec une liste déroulante d'élément hexadécimal



Modification nécessaire dans Editer élément



C'est la chute !!

- **Quelques liens utiles**

Forum lava <http://lavag.org/>

Forum Ni <http://forums.ni.com/t5/Discussions-au-sujet-de-NI/bd-p/4171>