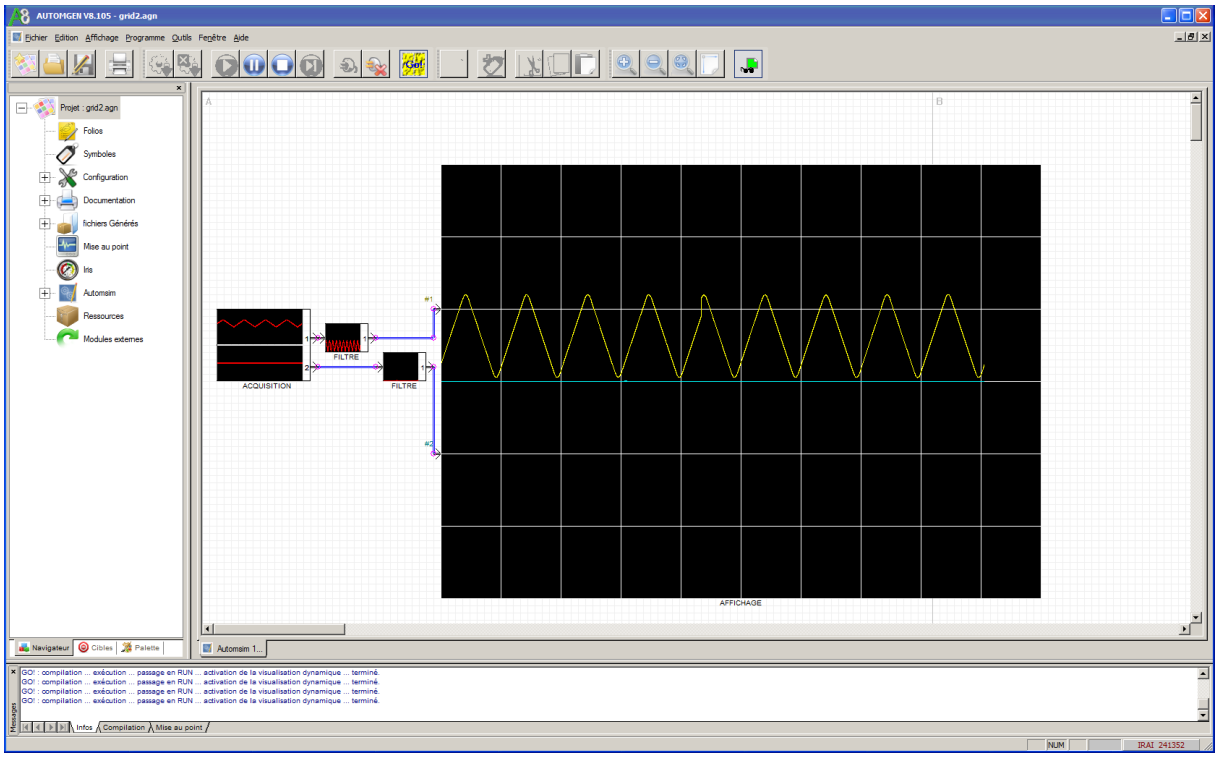
AUTOMDAQ

révision 2



Sommaire

Généralités

AUTOMDAQ est un module d'AUTOMGEN permettant de réaliser des acquisitions, du traitement de données ainsi que l'affichage de celles-ci.

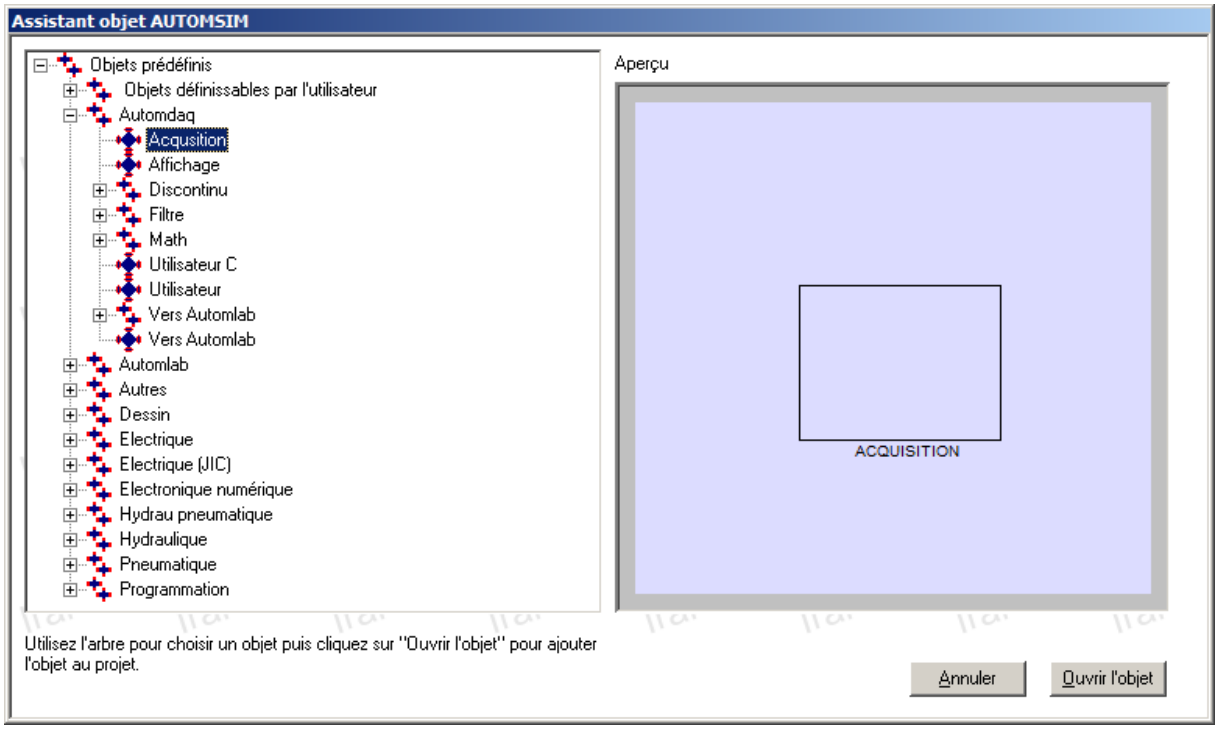
Pré requis

AUTOMDAQ nécessite AUTOMGEN V>=8.105.

AUTOMDAQ peut fonctionner avec les cartes d'acquisitions LabJack U12, National Instrument (6008,6009 par exemple) ou avec tout système capable d'envoyer des données sur une liaison série (une carte Arduino par exemple).

Création des schémas

La création des schémas est réalisée sur des folios AUTOMSIM d'AUTOMGEN. Les schémas ainsi réalisés peuvent cohabiter avec les autres éléments d'une application AUTOMGEN. Les blocs sont accessibles par un clic droit sur le folio AUTOMSIM puis la sélection de « Ajouter un objet » puis la rubrique « Automdaq ».



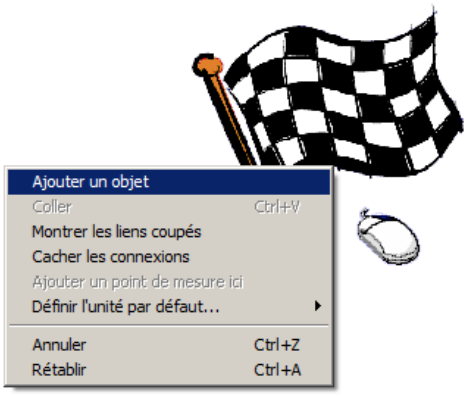
Principe de fonctionnement

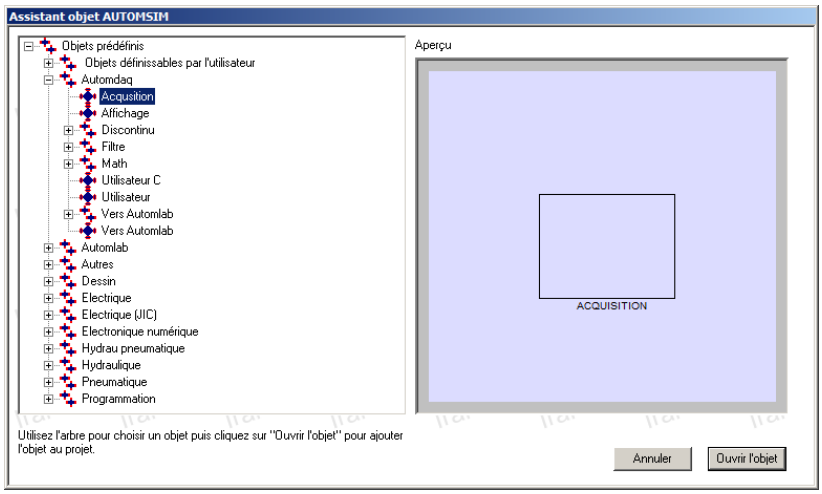
Chaque bloc peut posséder une ou plusieurs entrées sur la partie gauche du bloc et une ou plusieurs sorties sur la partie droite. Les blocs peuvent également contenir des paramètres.

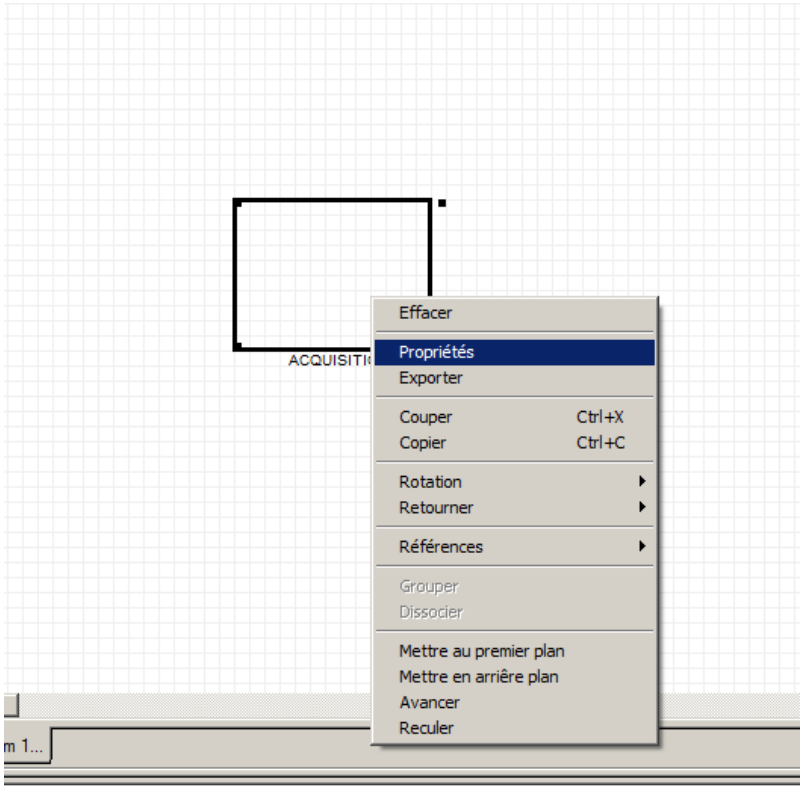
L'objet "Acquisition" permet de recevoir des valeurs provenant d'un ou plusieurs canaux d'une source d'acquisition (cartes LabJack U12,National Instrument ou liaison série).

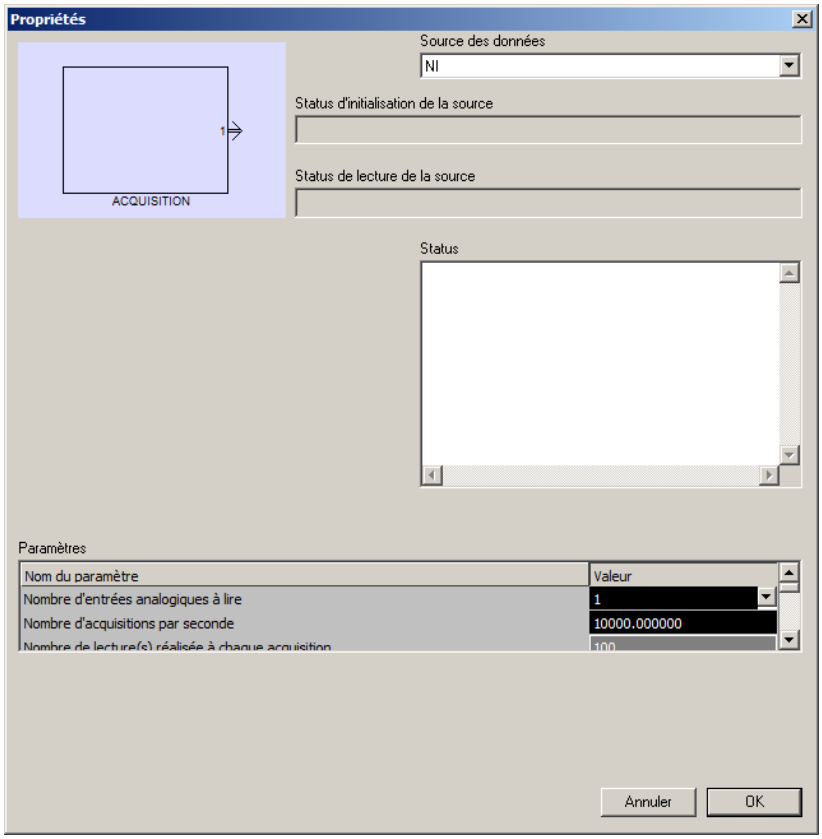
L'objet "Affichage" permet d'afficher les données collectées et éventuellement de les exporter vers Microsoft Excel.

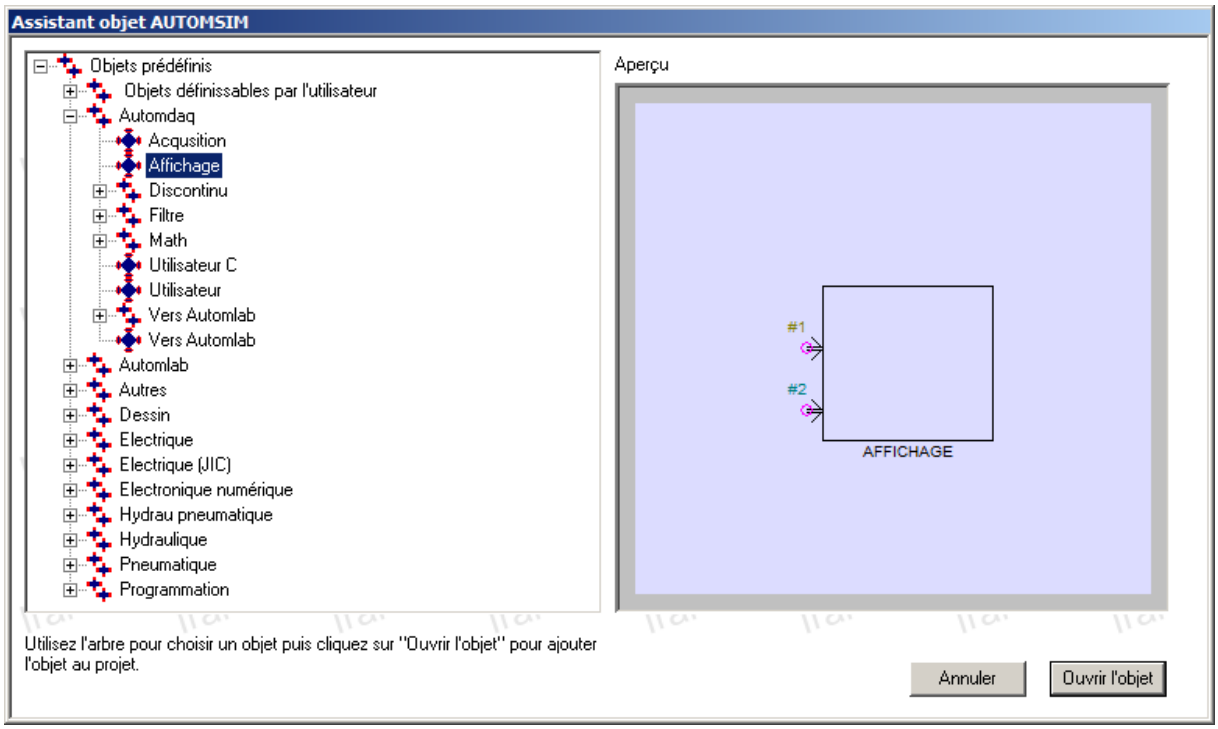
Exemple : création d'un oscilloscope affichant un signal reçu sur le canal 1 d'une carte National Instrument 6009 avec un temps d'échantillonnage de 10 Khz.

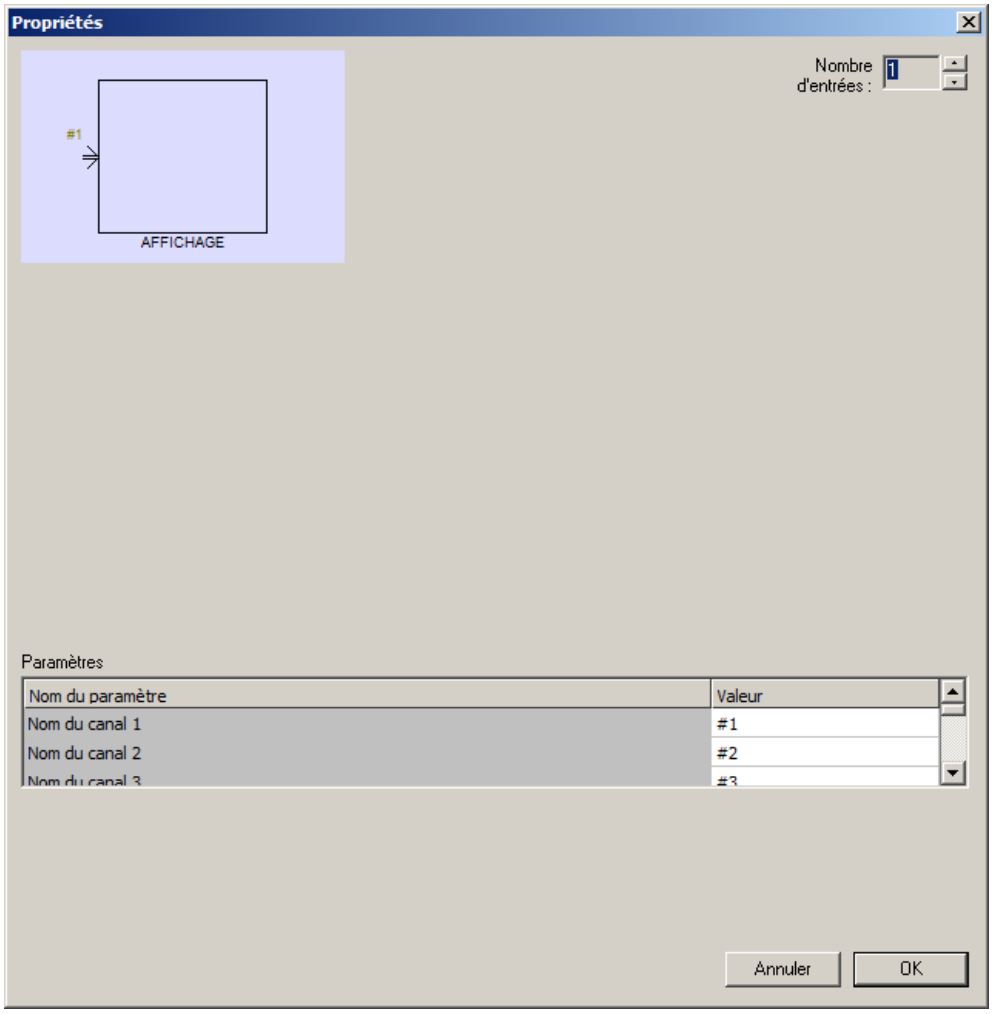




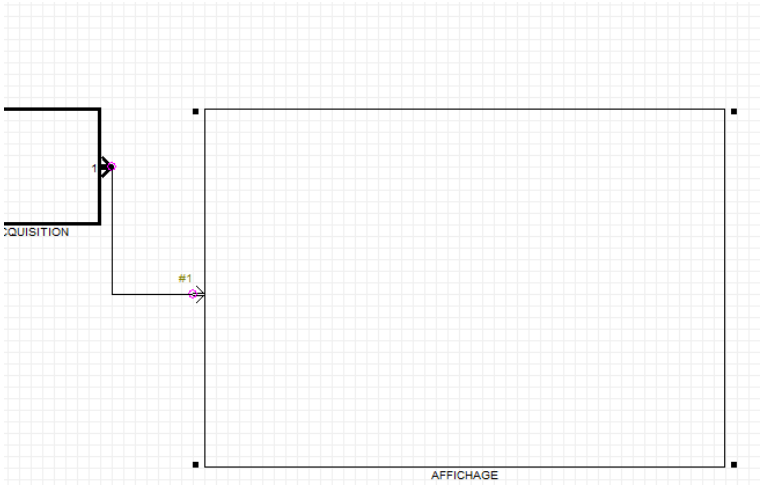

ACQUISITI
Effacer
Propriétés
Exporter
Couper Ctrl+X
Copier Ctrl+C
Rotation
Retourner
Références
Grouper
Dissocier
Mettre au premier plan
Mettre en arrière plan
Avancer
Reculer
m 1...

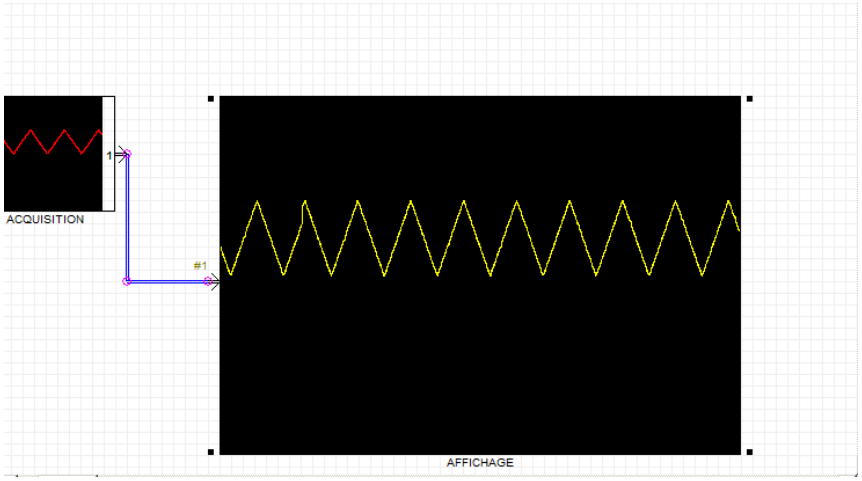

Propriétés
Source des données
NI
Status d'initialisation de la source
Status de lecture de la source
Status
ACQUISITION
Paramètres
Nom du paramètre
Valeur
Nombre d'entrées analogiques à lire
1
Nombre d'acquisitions par seconde
10000.000000
Annuler
OK


Assistant objet AUTOMSIM
Objets prédéfinis
Objets définissables par l'utilisateur
Automdaq
Acqusition
Affichage
Discontinu
Filtre
Math
Utilisateur C
Utilisateur
Vers Automlab
Vers Automlab
Automlab
Autres
Dessin
Electrique
Electrique (JIC)
Electronique numérique
Hydrau pneumatique
Hydraulique
Pneumatique
Programmation
Aperçu
#1
#2
AFFICHAGE
Utilisez l'arbre pour choisir un objet puis cliquez sur "Ouvrir l'objet" pour ajouter l'objet au projet.
Annuler
Ouvrir l'objet


Propriétés
Nombre d'entrées :
1
#1
AFFICHAGE
Paramètres
Nom du paramètre
Valeur
Nom du canal 1
#1
Nom du canal 2
#2
Annuler
OK

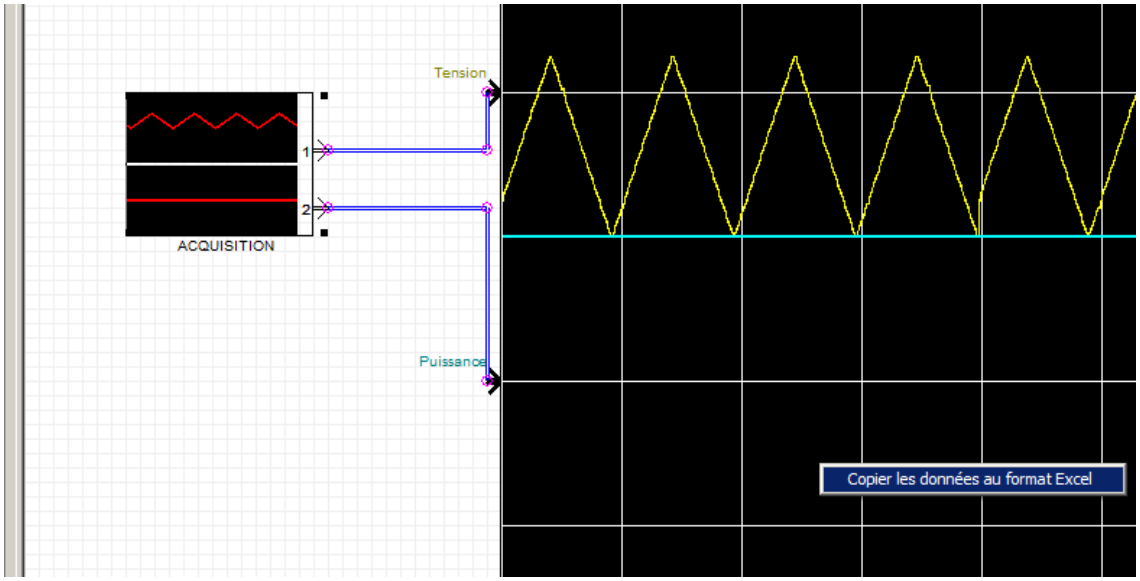

1
ACQUISITION
#1
AFFICHAGE

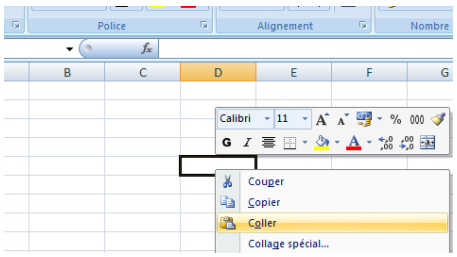

#1
AFFICHAGE

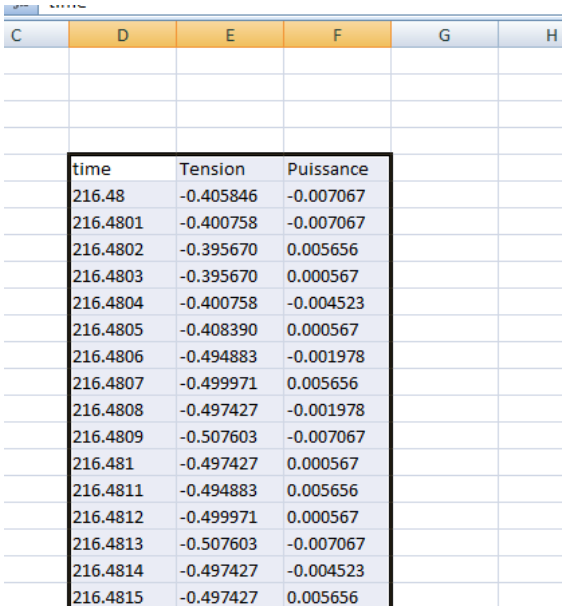

1
ACQUISITION
#1
AFFICHAGE

Copie des données vers Excel

En cliquant avec le bouton droit de la souris sur un bloc affichage, un menu permet de copier les données vers Excel. Exemple :





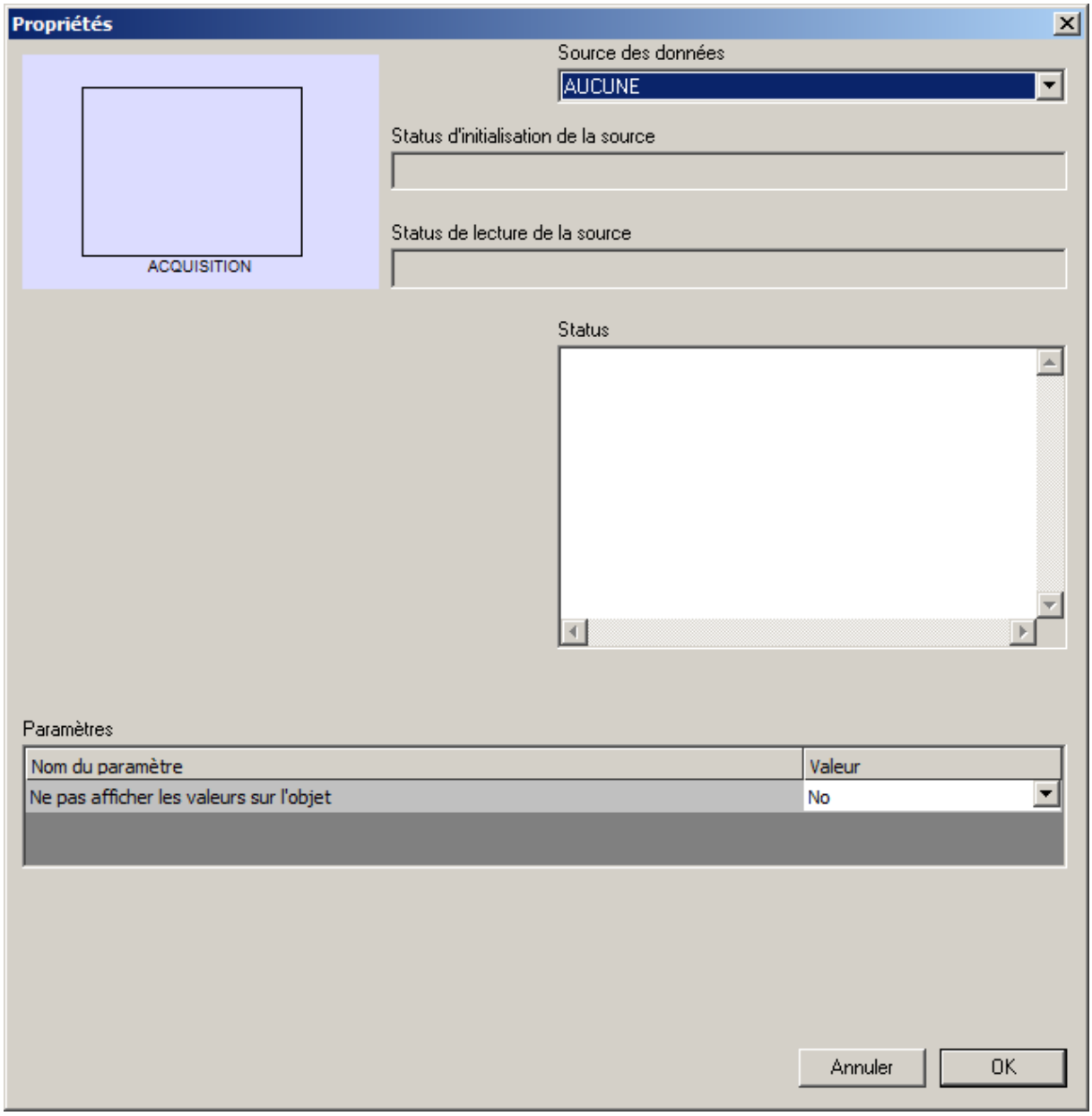


time	Tension	Puissance
216.48	-0.405846	-0.007067
216.4801	-0.400758	-0.007067
216.4802	-0.395670	0.005656
216.4803	-0.395670	0.000567
216.4804	-0.400758	-0.004523
216.4805	-0.408390	0.000567
216.4806	-0.494883	-0.001978
216.4807	-0.499971	0.005656
216.4808	-0.497427	-0.001978
216.4809	-0.507603	-0.007067
216.481	-0.497427	0.000567
216.4811	-0.494883	0.005656
216.4812	-0.499971	0.000567
216.4813	-0.507603	-0.007067
216.4814	-0.497427	-0.004523
216.4815	-0.497427	0.005656

Liste des blocs

Acquisition

Ce bloc permet de réaliser l'acquisition de données.



"Source de données" permet de sélectionner le type de système d'acquisition : LabJack U12, NI (National Instrument), COM PORT (liaison série).

Les zones "Status" permettent d'observer d'éventuelles erreurs lors de l'initialisation du système d'acquisition (paramètres non supportés par exemple) ou pendant l'acquisition.

Par défaut, un aperçu des données est affiché sur l'objet pendant l'acquisition, l'option "Ne pas afficher les valeurs sur l'objet" permet de désactiver cet affichage.

Paramètres spécifiques à la carte LabJack U12

Nombre d'entrées analogiques à lire : choix du nombre de canaux à utiliser : 1, 2 ou 4

Nombre de lectures réalisées à chaque acquisition : nombre d'échantillons transférés pour chaque canaux de la carte LabJack U12 vers le PC. Ce paramètre peut être modifié en accord avec la fréquence d'échantillonnage.

Nombre de valeurs stockées dans l'objet pour chaque canal : l'objet mémorise en interne le nombre de valeurs spécifié par ce paramètre pour chaque canal. Exemple : pour une fréquence d'échantillonnage de 1000 Hz, si ce paramètre est égal à 10000, on stockera des échantillons correspondant à 10 secondes de temps d'acquisition.

Nombre d'acquisitions par seconde : la fréquence d'échantillonnage pour chaque canal. Cette fréquence est limitée par constitution par la carte LabJack12 : entre 400 et 8192 échantillons / seconde.

Valeurs mini et maxi : la plus petite et la plus grande valeur pouvant être acquise par la carte.

Paramètres spécifiques aux cartes NI

Nombre d'entrées analogiques à lire : choix du nombre de canaux à utiliser : 1, 2 ou 4

Nombre d'acquisition par seconde : la fréquence d'échantillonnage pour chaque canal. Cette fréquence est limitée par constitution pour chaque type de cartes National Instrument (se référer aux informations fournies par National Instrument).

Nombre de lectures réalisées à chaque acquisition : nombre d'échantillons transférés pour chaque canal de la carte NI vers le PC. Ce paramètre peut être modifié en accord avec la fréquence d'échantillonnage.

Nombre de valeurs stockées dans l'objet pour chaque canal : l'objet mémorise en interne le nombre de valeurs spécifié par ce paramètre pour chaque canal. Exemple : pour une fréquence d'échantillonnage de 1000 Hz, si ce paramètre est égal à 10000, on stockera des échantillons correspondant à 10 secondes de temps d'acquisition.

Nom des canaux : doit être en accord avec la syntaxe NI du canal associé à chaque voix de la carte. Par exemple Dev1/ai0 = première voix analogique de la carte Dev1.

Valeurs mini et maxi : la plus petite et la plus grande valeur pouvant être acquise par la carte.

Paramètres spécifiques l'acquisition sur liaison série

Numéro de port COM : numéro du port de communication, par exemple 2 pour utiliser COM2.

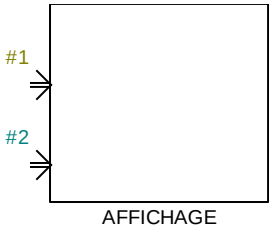
Vitesse en baud : la vitesse en bits/seconde, par exemple 115200 pour 115200 bauds.

Nombre d'entrées analogiques à lire : choix du nombre de canaux à utiliser : 1, 2, 3 ou 4

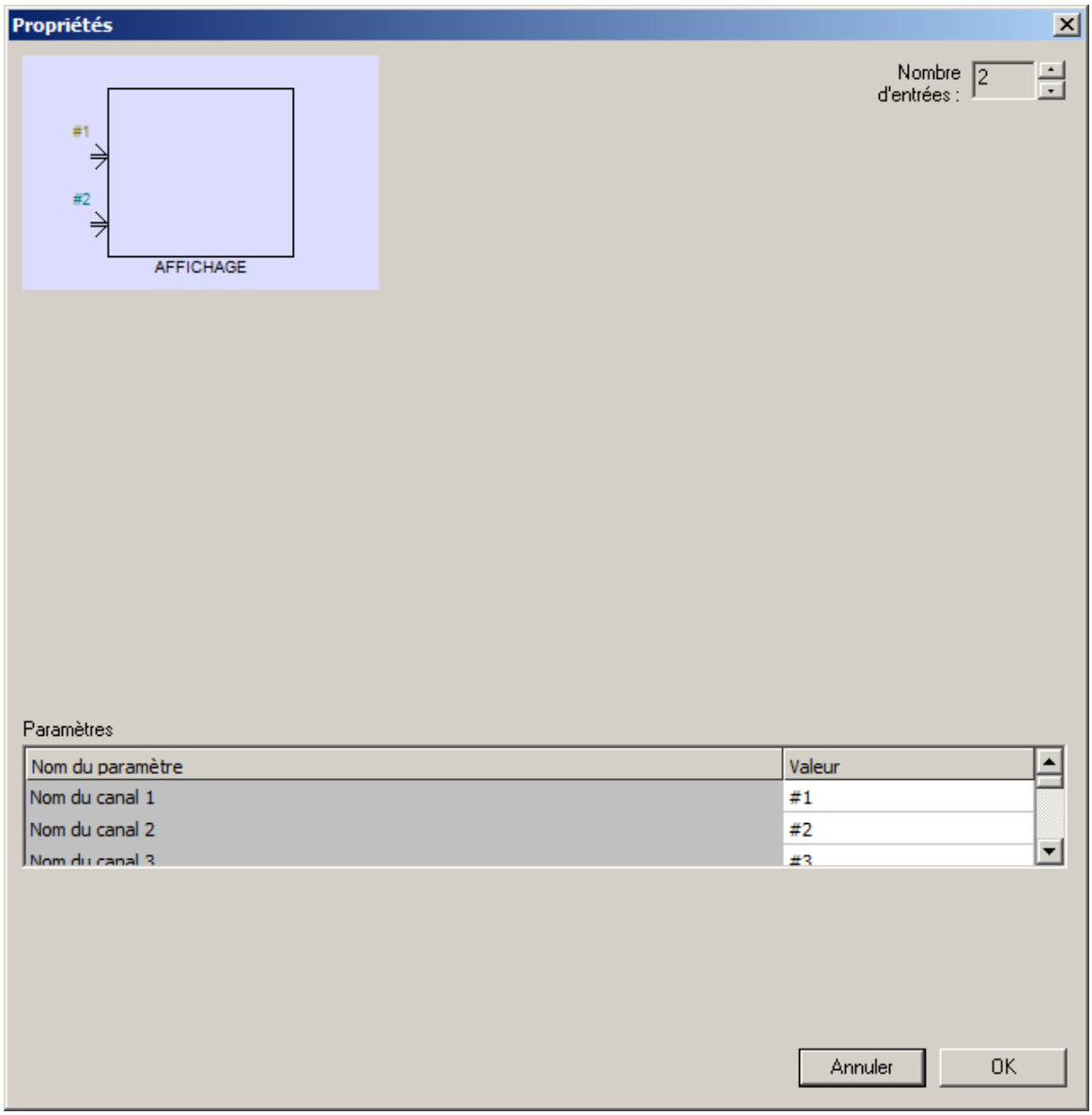
Nombre de lectures réalisées à chaque acquisition : nombre d'échantillons composant les paquets de données transférés pour chaque canal de l'objet acquisition vers les objets connectés. Ce paramètre peut être modifié en accord avec la fréquence d'échantillonnage.

Nombre de valeurs stockées dans l'objet pour chaque canal : l'objet mémorise en interne le nombre de valeurs spécifié par ce paramètre pour chaque canal. Exemple : pour une fréquence d'échantillonnage de 1000 Hz, si ce paramètre est égal à 10000, on stockera des échantillons correspondant à 10 secondes de temps d'acquisition.

Affichage



Cet objet permet d'afficher les données, il peut être utilisé à la manière d'un oscilloscope ou d'un afficheur de courbes de tendance.



"Nombre d'entrées" permet de spécifier le nombre de canaux à afficher (maximum 4).

Paramètres

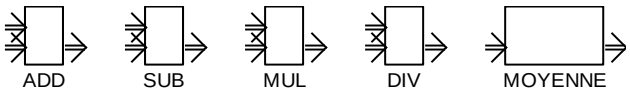
Nom des canaux : modifie le nom affiché pour chaque canal ainsi que le libellé de la colonne associée lorsqu'on exporte les données vers Excel.

Valeurs mini et maxi : défini les valeurs pour une mise à l'échelle des données affichées. Si par exemple mini=0 et maxi=5 alors 0 correspondra au bas de la zone d'affichage et 5 au haut de celle-ci.

Nombre de valeurs stockées : défini la nombre de valeurs qui seront stockées dans l'objet et affichées. Le côté droit de l'affichage affiche les données acquises le plus récemment.

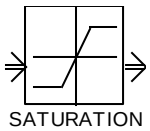
Synchronisation : permet de définir un des canaux pour la synchronisation des données, une valeur et le type de front. L'affichage est synchronisé (à la manière de la fonction trigger sur un oscilloscope) lorsque la valeur acquise sur le canal sélectionné franchi le niveau spécifié avec le type de front spécifié. Le paramètre "Espacement des valeurs" permet de tester une variation sur des échantillons espacés afin d'éviter une fausse synchronisation sur un signal parasité.

Math



Ces blocs réalisent des opérations mathématiques entre le ou les canaux d'entrée et le canal de sortie.

Saturation



Ce bloc réalise une saturation par rapport à ses paramètres min et max.

Filtre

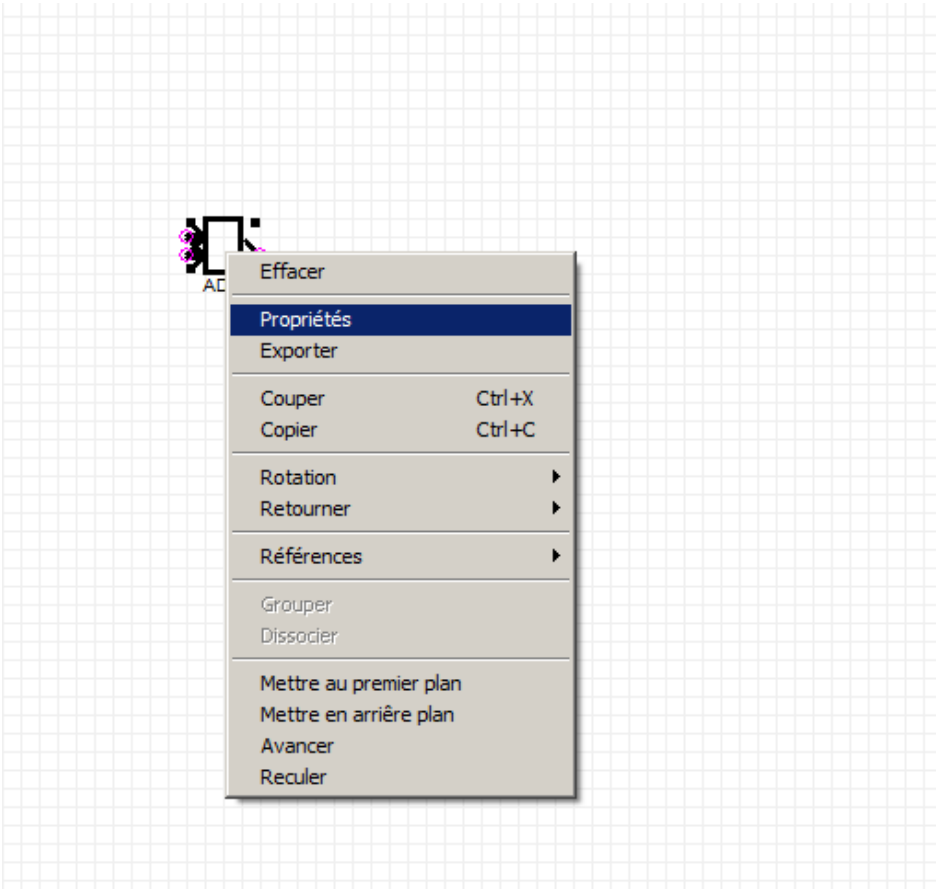
Ce bloc réalise un filtrage par calcul de moyenne sur le nombre d'échantillons défini dans ses paramètres.

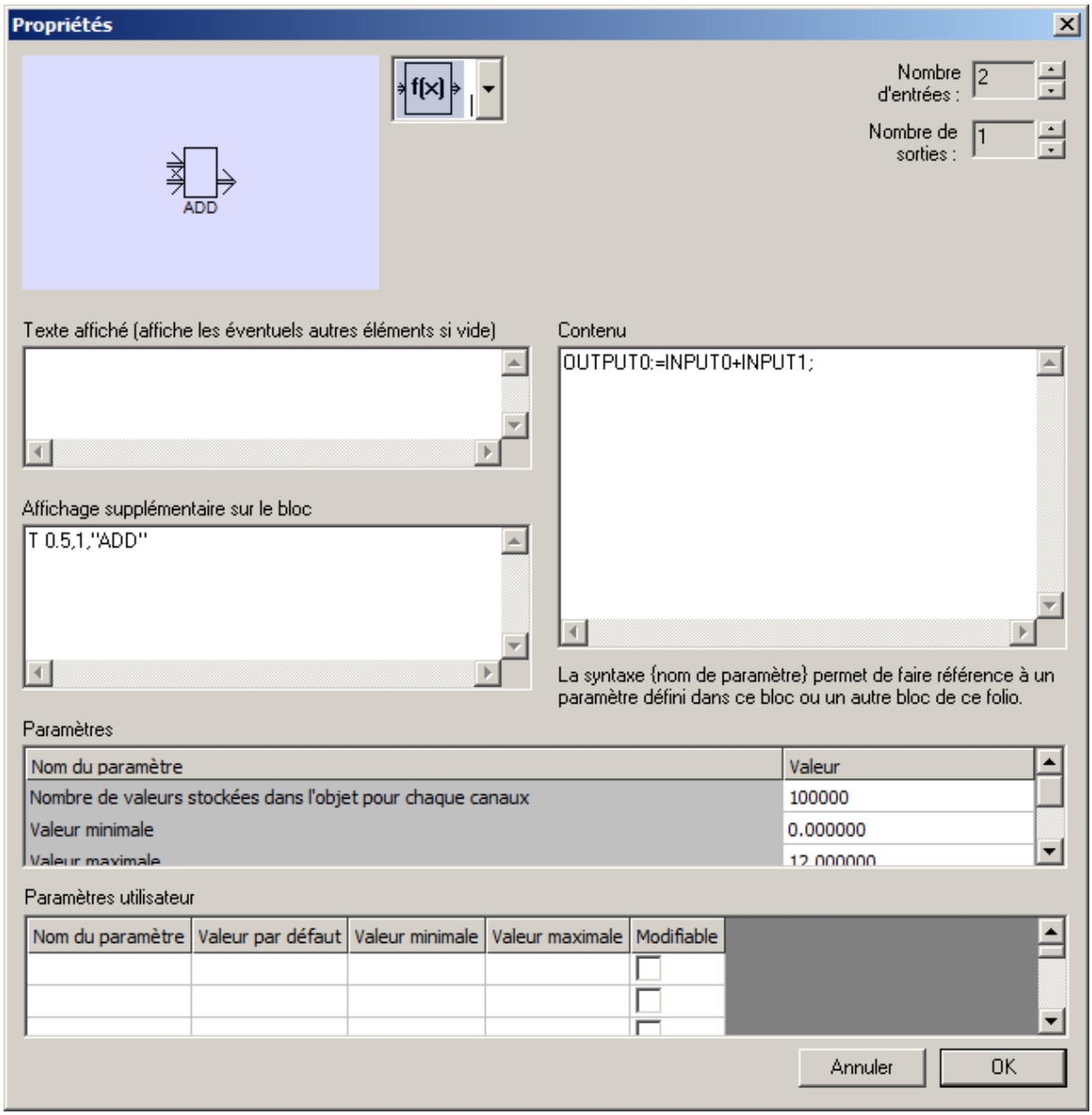
Vers Automlab / Moyenne

Calcul la moyenne sur le nombre d'échantillons mémorisé dans l'objet et place le résultat sur un connecteur Automlab.

Blocs utilisateurs

Les blocs utilisateurs permettent de créer des blocs personnalisés programmés en langage littéral ST (seules les équations numériques sont supportées) ou en langage C. Les blocs utilisateurs possèdent de 1 à 4 canaux en entrées et de 1 à 4 canaux en sortie. Les blocs proposés dans la bibliothèque utilisent les blocs utilisateurs, pour observer l'ensemble de leurs paramètres, y compris le code, laissez la touche Shift enfoncée lorsque vous accédez aux propriétés de l'objet. Exemple :


AD
Effacer
Propriétés
Exporter
Couper Ctrl+X
Copier Ctrl+C
Rotation
Retourner
Références
Grouper
Dissocier
Mettre au premier plan
Mettre en arrière plan
Avancer
Reculer



Flux de données

Le principe de fonctionnement des flux de données est le suivant dans Automdaq : à chaque acquisition d'un paquet de données sur l'ensemble des canaux d'un système d'acquisition par un objet "Acquisition", le paquet est transmis à chaque objet connecté sur chaque connexion de sortie, cette transmission est réalisée en cascade sur les objets connectés à cet objet et ainsi de suite jusqu'à la fin de la chaîne. Chaque objet possède un buffer interne dont la longueur est modifiable dans les paramètres des objets. Si ce buffer ne peut pas contenir les données transmises alors les données les plus anciennes sont supprimées.

Le code écrit dans les blocs utilisateurs est exécuté à chaque fois qu'un paquet de donnée arrive.

Pour accéder au code des objets utilisateurs, maintenez la touche Shift du clavier enfoncée lors de l'accès aux propriétés de ceux-ci.

Bloc utilisateur langage ST

Le code littéral écrit pour ce type de bloc défini une équation pour mettre en relation un ou plusieurs flux de sortie avec un ou plusieurs flux d'entrée. Le code s'applique uniquement au paquet de données qui vent d'être reçu par l'objet. Le code ne peut pas accéder aux valeurs présentent dans le buffer de l'objet. L'objet Bloc utilisateur Langage C ne possède pas cette limitation (voir plus loin).

L'équation est de la forme :

flux de sortie=F(flux d'entrée)

Exemple d'équation :

OUTPUT0:=INPUT0+INPUT1;

qui signifie : Premier flux de sortie = premier flux d'entrée + deuxième flux d'entrée

La syntaxe OUTPUTx avec x compris entre 0 et 3 permet de référencer les flux de sorties 1 à 4, la syntaxe INPUTx avec x compris entre 0 et 3 permet de référencer les flux d'entrée 1 à 4.

La syntaxe ETIME permet de référencer la valeur du temps écoulé entre 2 scrutations exprimée en seconde (valeur décimale).

La syntaxe TIME permet de référencer la valeur du temps écoulé depuis la première acquisition.

Bloc utilisateur langage C

Le code écrit en langage C pour ce type de bloc défini une fonction qui reçoit en paramètre des pointeurs sur chacun des flux d'entrée et de sortie du bloc ainsi que la longueur du paquet de données arrivées et la longueur des données dans le buffer de l'objet.

Le prototype de la fonction est :

```
__declspec(dllexport) void compute2(
float *fi0,
float *fi1,
float *fi2,
float *fi3,
float *fo0,
float *fo1,
float *fo2,
float *fo3,
unsigned len,
```

unsigned fullen,
float ETIME,
float TIME,
int b0)

fi0 à fi3 : pointeurs sur le début de chacun flux de données possibles correspondant au 4 entrées possibles du bloc. Si un canal n'est pas défini, le pointeur est NULL.

len : longueur du paquet de données arrivé en nombre de valeurs. On assume que la même quantité de données vient d'arriver sur chaque canal.

fo0 à fo3 : pointeurs sur le buffer correspondant au flux de sortie associé à chaque sortie du bloc. Si un canal de sortie n'est pas défini, le pointeur est NULL. Chaque buffer de sortie à une taille de 'len' valeurs (il sort autant de données du bloc que celles qui entrent).

fullen : longueur des données du buffer interne pour chaque canal.

ETIME : temps écoulé en seconde entre deux appels

TIME : temps écoulé depuis le début des acquisitions

b0 : 1 au premier appel, 0 ensuite

Remarques :

- lorsque la fonction est appelée, le paquet de données nouvellement arrivé est comptabilisé dans fullen.

Exemple de l'évolution des valeurs de len et fullen pour une taille de paquet de données de 100 et un buffer interne de 500 :

Appel numéro	Len	fullen
1	100	100
2	100	200
3	100	300
4	100	400
5	100	500
6 et suivants	100	500

- les valeurs du buffer interne sont accessibles en décrémentant les pointeurs fi0 à fi3.

Paramètres

La zone "paramètres utilisateurs" de l'objet permet de définir des paramètres accessibles à l'utilisateur du bloc. La syntaxe {nom de paramètre} permet de référencer ceux-ci.

Exemple de code : génération de la moyenne sur un flux de sortie à partir d'un flux d'entrée :

```
__declspec(dllexport) void compute2(float *fi0,float *fi1,float *fi2,float *fi3,float *fo0,float *fo1,float *fo2,float *fo3,unsigned len,unsigned fullen,float ETIME,float TIME,int b0)

{

unsigned count;

// Boucle de génération de chaque valeur du flux de sortie

for(count=0;count<len;count++)

  {

  float sum; // Pour totaliser les n valeurs précédentes

  int count2;

  int flen;

  // Calcul de ma longueur des données disponibles pour le calcul

  // de la moyenne

  flen=count+(fullen-len);

  sum=fi0[count];

  for(count2=1;count2<=flen;count2++)

    {

    sum+=fi0[count-count2];

    }

  // Calcul de la moyenne : total/nombre de valeurs

  fo0[count]=sum/(flen+1);

  }

}
```

Bloc utilisateur hybride "vers Automlab"

Ce bloc permet de générer des valeurs sur des sorties (1 à 4) sur lesquels ont peut connecter des blocs AUTOMLAB à partir des entrées de flux Automdaq. Ces blocs sont programmés en langage C. Les mots clés OUTPUT0 à OUTPU3 permettent de référencer les sorties Automlab du bloc.

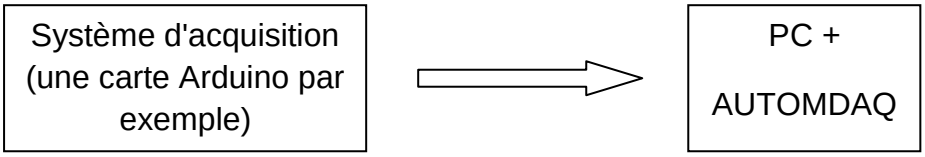
Exemple de code d'un bloc de calcul d'une moyenne.

```
float sum=0;

BEGINLOOP

sum=sum+INPUT0;

ENDLOOP

OUTPUT0=sum/LOOPLEN;
```

Informations spécifiques à l'acquisition de données par liaison série

Le mode "COM PORT" de l'objet acquisition permet d'utiliser une simple liaison série pour acquérir des données.

Principe



Dans ce mode, la fréquence d'acquisition est automatiquement calculée en fonction de la fréquence des données reçues par la liaison série.

Le système d'acquisition doit émettre des données de façon continue et régulier sur la liaison série. La vitesse d'acquisition est automatiquement calculée par l'objet AUTOMDAQ en fonction de la fréquence des données reçues.

Format liaison série

Le format des données au niveau liaison série est 8 bits de données, 1 bit de stop, sans parité. La vitesse est paramétrée dans les propriétés de l'objet acquisition.

Format des données

Chaque acquisition est représentée par 4 octets codant un flottant IEEE 754 par voie. Si plusieurs voies sont utilisées, les données sont envoyées alternativement et dans l'ordre pour chaque voie. Exemple pour 4 voies :

Voie 1	Voie 2	Voie 3	Voie 4	Voie 1	Voie 2	Voie 3	Voie 4

Synchronisation

Le protocole actuel ne prévoit pas d'octet de synchronisation (seules les données sont transmises), il convient donc que l'application AUTOMDAQ soit en mode RUN avant que les données commencent à être émises pour éviter une désynchronisation.

Exemple avec une carte Arduino

Programme côté Arduino (exemple pour une voie):

```
unsigned int convertedValue;  // variable to store the converted value
float voltageValue;           // variable to store the voltage applied to the analog pin
byte analogPin = 0;           // analog input pin

void setup(void) {
  Serial.begin(115200);
}

void loop(void){
  convertedValue = analogRead(analogPin);
  voltageValue = 5.0 * convertedValue / 1023;
  Serial.write((byte *)&voltageValue,4);
}
```

Ce programme fait simplement l'acquisition d'une valeur analogique, calcule la valeur en volt et émet la valeur codée flottant IEEE sur la liaison série avec une vitesse de 115200 bauds (vitesse maximale compatible avec la liaison série de base de la carte Arduino et le PC).

Programme côté Arduino (exemple pour deux voies):

```
unsigned int convertedValue; // variable to store the converted value
float voltageValue;          // variable to store the voltage applied to the analog pin
byte analogPin = 0;          // analog input pin

void setup(void) {
  Serial.begin(115200);
}

void loop(void){
  convertedValue = analogRead(analogPin);
  voltageValue = 5.0 * convertedValue / 1023;
  Serial.write((byte *)&voltageValue,4);
```

```
  convertedValue = analogRead(analogPin+1);
  voltageValue = 5.0 * convertedValue / 1023;
  Serial.write((byte *)&voltageValue,4);
}
```

Calcul de la vitesse d'acquisition : l'action significative en terme de temps est l'émission des caractères sur la liaison série. Avec une vitesse de 115200 bauds, la durée d'émission de 4 caractères (une mesure sur une voie) est :

115200 / 10 (8 bits de données, 1 bit de start, 1 bit de stop) = 11520 octets par secondes.

La fréquence maximale d'acquisition pour une voie est donc : 11520/4 4 (4 octets pour une valeur) = 2880 Hz.

Pour 2 voies, cette fréquence sera 1440 Hz, pour 3 : 960Hz, pour 4 : 720 Hz.

Des fréquences supérieures peuvent être envisagées en utilisant les liaisons séries supplémentaires d'une carte Arduino pouvant permettre d'utiliser des vitesses de communication plus élevées en adéquation avec les vitesses de communications disponibles sur certains port série de PCs.