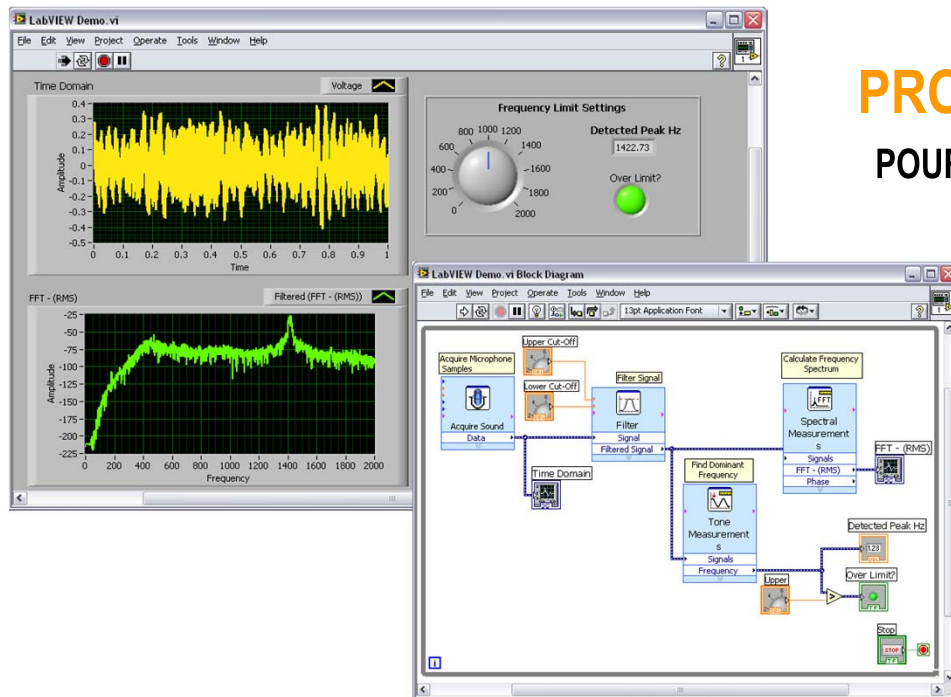


# MEC6405 - Introduction à LabVIEW



**PROGRAMMATION GRAPHIQUE**  
**POUR LES INGÉNIEURS ET LES SCIENTIFIQUES**



**COURS DE 3 HEURES AVEC APPLICATION PRATIQUE AU LABORATOIRE NO. 5**  
**Automne 2012**

# Qu'est-ce que LabVIEW ?

---

- *Laboratory Virtual Instrument Engineering Workbench*
- Logiciel de développement d'applications appelées "Instruments Virtuels" car leur apparence et fonctionnement ressemblent aux instruments réels
- Ces applications principales sont la prise de mesures, le contrôle de procédés, l'analyse des données
- C'est un langage de programmation graphique (langage G) qui fait appel à des symboles (icônes) pour décrire les opérations
- La programmation avec LabVIEW est **intuitive** et s'apprend rapidement. **Aucun pré requis** n'est nécessaire.

Un instrument virtuel (VI) peut reproduire les fonctions de ces appareils à l'aide d'un carte d'acquisition de données



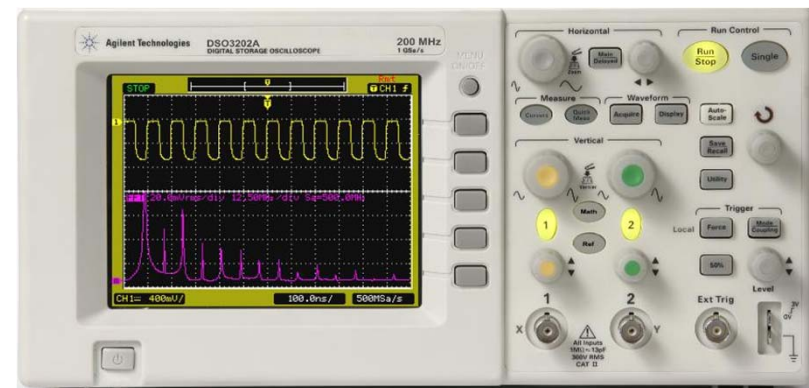
**Indicateur de déformation**



**Générateur de signal**

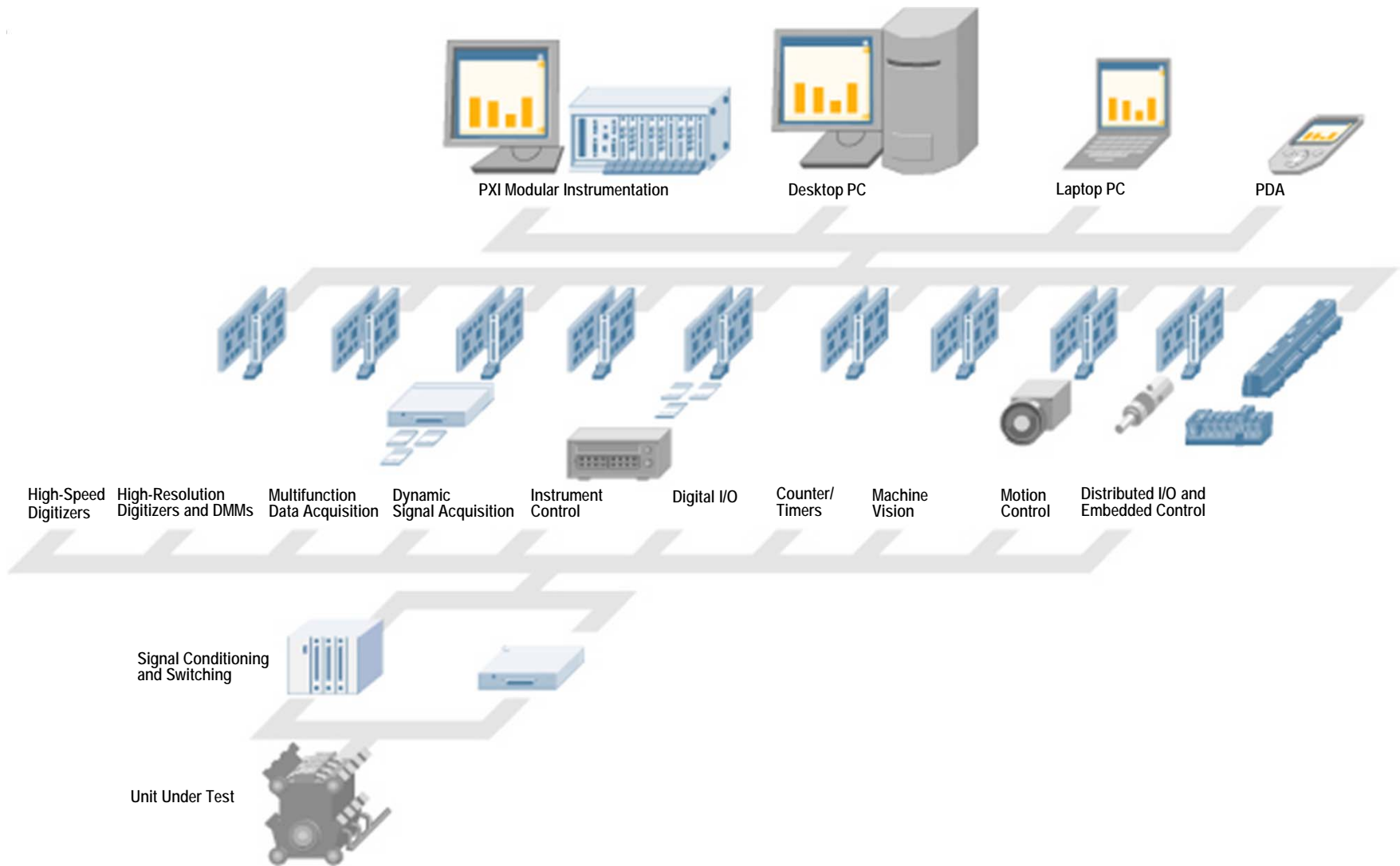


**Multimètre**



**Oscilloscope**

# Intégration de la technologie informatique pour la mesure et le contrôle à l'aide de LabVIEW



# Objectifs du cours

---

- Se familiariser avec l'environnement LabVIEW
- Comprendre la base de la programmation graphique
- Concevoir un programme d'acquisition de données simple (Partie II, Labo. 5)

# Références

---

## En français

- [1] "Initiation à LabVIEW", National Instruments, 2010.
- [2] "Principes de base LabVIEW ", National Instruments, 2007.
- [3] "Carte de référence LabVIEW ", National Instruments, 2010.

## En anglais

- [1] "LabVIEW - Getting Started ", National Instruments, 2010.
- [2] "LabVIEW - Fundamentals", National Instruments, 2007.
- [3] "LabVIEW - Quick Reference Card", National Instruments, 2010.

**Pour compléter votre formation**, lire et faire les exercices de la référence [1]

# Références (suite)

---

## Livres

- "LabVIEW 9 – Student Edition", Robert S. Bishop, Pearson Prentice Hall, 2009. Livre avec ou sans le logiciel LabVIEW 9.
- "Hands-On Introduction to LabVIEW for Scientists and Engineers", John Essick, Oxford University Press, 2009, Recommandé, ~35\$.
- "Introduction to Data Acquisition with LabVIEW", Robert H. King, McGraw Hill Higher Education, 2009, comprend le logiciel LabVIEW 8.5.

## Site Internet de National Instrument pour l'éducation

- [http://www.ni.com/academic/learn\\_labview/f/](http://www.ni.com/academic/learn_labview/f/)
- <http://www.ni.com/academic/students/learnlabview/> LabVIEW 101: Video Instruction for Students, I. LabVIEW Basics Concepts

# Exécution de LabVIEW

---

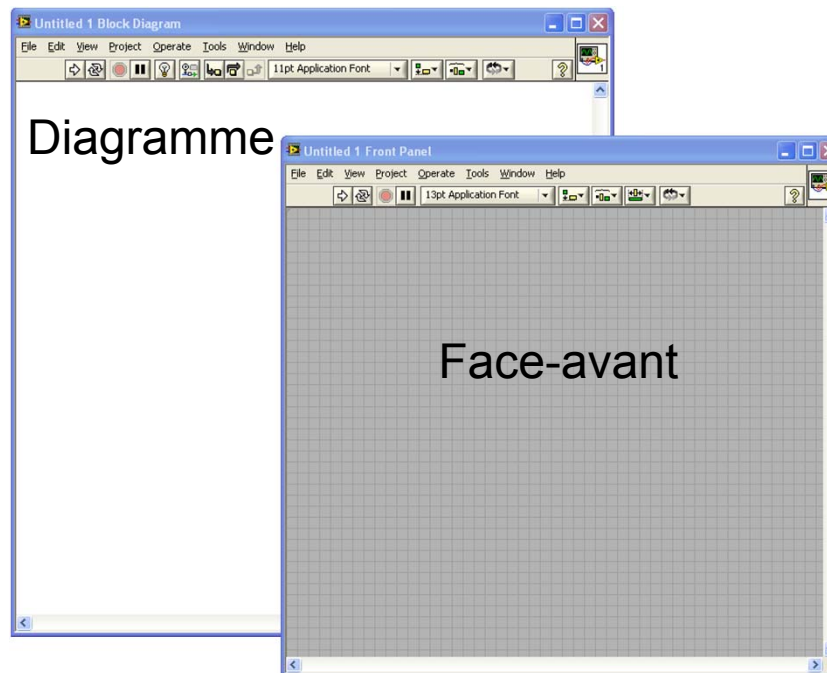
- LabVIEW 2011 version anglaise, est installé dans les salles d'informatique L-6613, L-6624, L-6655, L-6657 et L-6659
- Lancer LabVIEW en ouvrant le répertoire "Logiciels" qui se trouve sur le bureau. Ensuite, cliquer sur LabVIEW puis LabVIEW 2011
- Sauvegarder vos VI sur votre disque réseau ou sur une clé USB. Ne pas sauvegarder sur le disque local.



# Environnement LabVIEW

---

- Deux fenêtres principales
  - Face-avant (Front Panel)
  - Diagramme (Block Diagram)

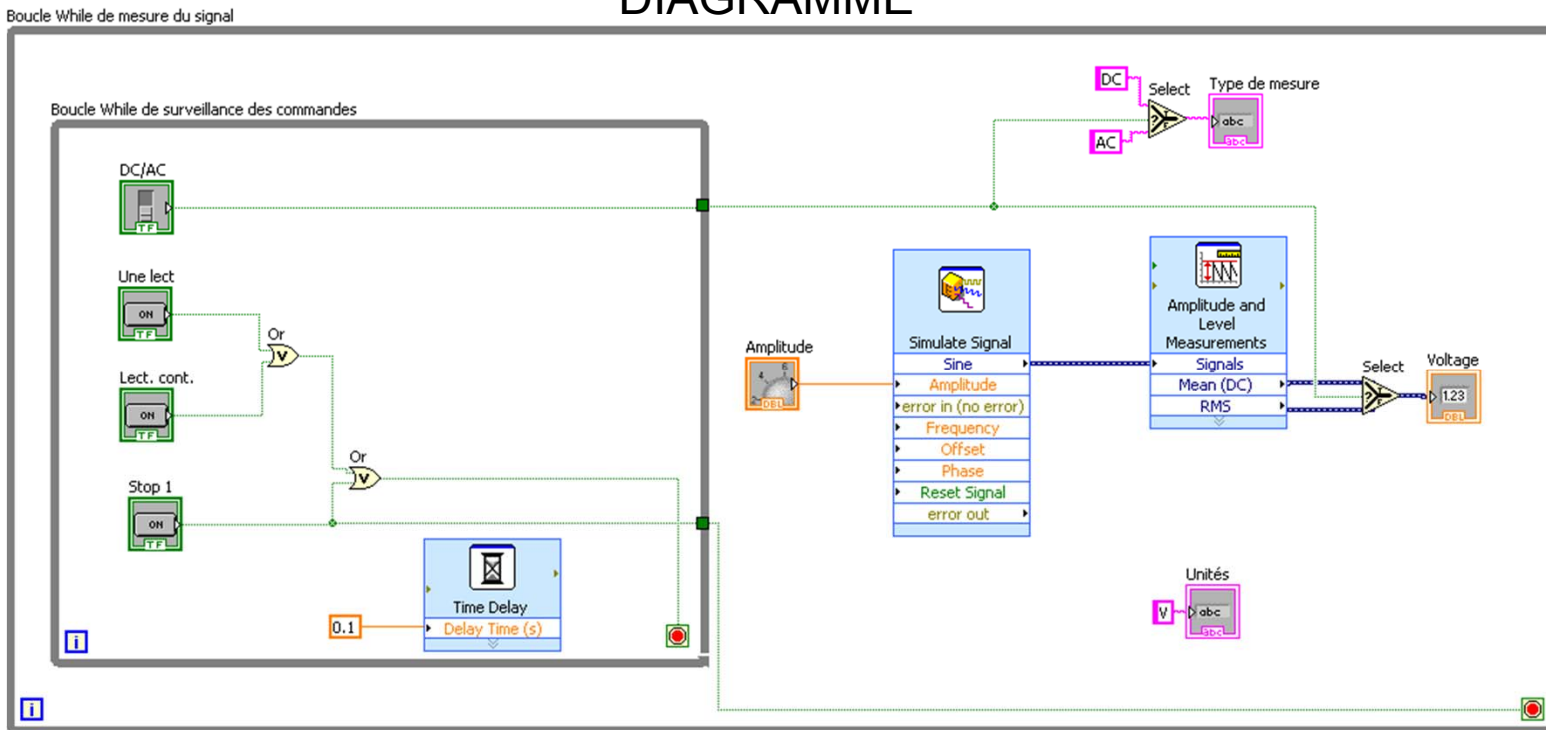


### Exemple: FACE-AVANT d'un voltmètre (simulation)



Voltmètre\_num\_A11.vi

### DIAGRAMME



# Face-avant

---

- Interface avec l'utilisateur
- Contient trois types d'objets
  - Commandes (Control) qui servent à entrer des valeurs (input)
  - Indicateurs (Indicator) qui reçoivent leurs valeurs du programme (output)
  - Décorations qui n'ont qu'une fonction esthétique

# Objets typiques de la face-avant

The image displays a variety of front-panel UI objects arranged on a grid background. On the left, two waveform plots are shown: 'Waveform Chart' (top) and 'Waveform Graph' (bottom). In the center, there are several control elements: a 'Numeric' input field, a 'Knob' with a dial, a 'Thermometer' with a vertical scale, a 'Meter' with a semi-circular scale, a 'Boolean' indicator (a small oval with a green dot), and a 'String' input field. On the right, there is an 'Array' control with four buttons labeled 1, 2, 3, and 4, and another 'Boolean 2' indicator (a small circle with a green dot). Arrows from text labels on the right point to these objects.

Indicateur Graphe déroulant

Indicateur Graphe

Commandes numériques

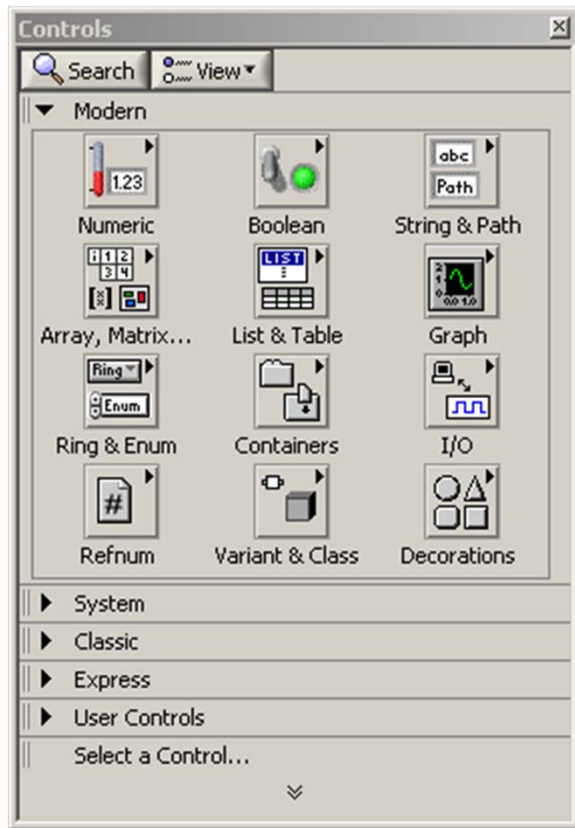
Indicateurs numériques

Commande et indicateur booléens

Commande et indicateur de type chaîne (String)

# FACE-AVANT

## Palette des commandes et indicateurs



(Cliquer-droit dans un endroit vide de la face-avant)

- Commandes et indicateurs sont placés directement sur la face-avant à partir de la palette des commandes ("controls")
- Ils ont plusieurs formes:
  - glissières
  - boîte
  - menu déroulant, etc.
- Ils sont associés à des types de données particuliers.

# Palettes de commandes très utilisées

---

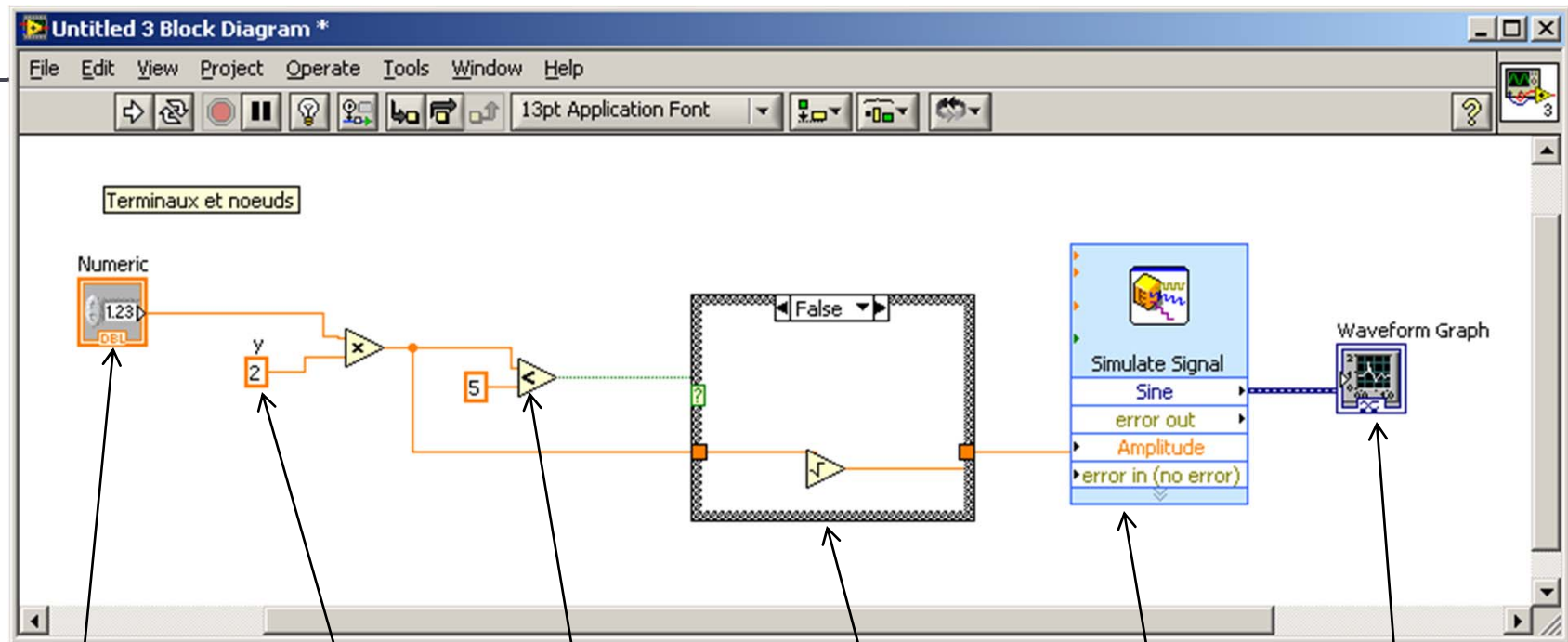
- Express
  - Contient les commandes et indicateurs les plus couramment utilisés
  - Choix limités
- Moderne et classique
  - Nombre de couleurs élevés
  - Objets stylisés
  - Plus grand choix

# Le diagramme

---

- Contient le code du programme sous forme graphique
- Principaux objets qu'on y retrouve
  - Icônes des commandes et des indicateurs
  - Fils de liaison entre les objets
  - Constantes
  - Fonctions, VI-Express, Sous-VI
  - Boucles (FOR et WHILE)
  - Structures (CASE, SEQUENCE, etc.)

# Objets typiques d'un diagramme



Commande

Constante

Fonction

Structure  
Condition  
(Case)

VI-Express

Indicateur  
Graphe



# Autres fenêtres utiles de LabVIEW

---

- Palette des outils
- Démarrage (Getting Started)
  - Ouverture de nouveaux et d'anciens VI
  - Trouver des exemples de VI
  - Manuels et aide en ligne
- Aide contextuelle (Context Help)
- Erreur (Error List)

MCours.com

# Palette outils (Menu: View → Tools Palette)

Sélection automatique



outil Doigt : permet de positionner des éléments des palettes **Controls** et **Functions** sur la face-avant et dans le diagramme



outil Flèche : permet de positionner, redimensionner et sélectionner les objets



outil Texte : permet de modifier du texte et d'en créer



outil Bobine : permet de câbler des objets entre eux dans le diagramme



outil Menu local : fait apparaître un menu local pour un objet



outil Main : fait défiler toute la fenêtre sans avoir recours aux barres de défilement



outil Point d'arrêt : permet de définir des points d'arrêt dans les VIs, les fonctions, les boucles, les séquences et les structures



outil Sonde : permet de créer des sondes sur les fils



outil Pipette : copie les couleurs pour les coller à l'aide de l'outil Pinceau



outil pinceau: permet de colorer l'avant plan et l'arrière plan des objets

# Exercices F.1 à F.3 (sur site Internet)

---

- Ouvrir un VI (face-avant et diagramme)
- Exécuter et arrêter un VI
- Créer une face-avant
  - Ajouter des commandes et des indicateurs
- Créer un diagramme
  - Ajouter une constante, une fonction math.
  - Créer les fils de liaison qui représente le flux de données entre les objets
  - Configurer un VI Express
- Sauvegarder le VI

# Exercices F.4 à F.7 (sur site Internet)







---

- Le sous-VI
  - Modifier son icône pour créer des bornes d'entrée et de sortie
  - Utiliser un sous-VI dans un VI
- Changer les propriétés d'une commande (ou indicateur)
- Commandes et indicateurs Booléens
- Boucle While (Créer, arrêter, ralentir son exécution)
- Sauvegarder les données dans un fichier

**Complétez votre formation** en terminant les exercices F1 à F7

# Types de données les plus courants



























- Numérique (Entier, réel, complexe)
- Booléen (Vrai ou Faux)
- Chaîne (Caractères ASCII)

Exemples	
<u>Commande</u>	<u>Indicateur</u>
Numeric 	Numeric 2 
Boolean 	Boolean 2 
String 	String 2 

**Note:** On ne peut pas échanger de données entre des commandes et des indicateurs qui n'ont pas le même type de données. Par ex. on en peut pas envoyer une donnée numérique à un chaîne.

# Tous les types de données dans LabVIEW













## Types de données des terminaux

<b>Entiers signés</b> (0)	8 bits		<b>Booléen</b> (Faux)			<b>Waveform</b>					
	16 bits			<b>Chaîne</b> (chaîne vide)					<b>Waveform numérique</b>		
	32 bits									<b>Chemin</b> (<Pas un chemin>)	
<b>Entiers non signés</b> (0)	8 bits		<b>Refnum</b>								
	16 bits			<b>Type énumération</b>					<b>Horodatage</b>		
	32 bits									<b>Cluster</b>	Numérique
<b>Nombres à virgule flottante</b> (0)	Simple précision		Données mixtes								
	Double précision			<b>Tableau</b>			1D				
	Précision étendue									2D	
<b>Complexes à virgule flottante</b> (0 + 0i)	Simple précision										
	Double précision										
	Précision étendue										



**Remarque** : les valeurs par défaut apparaissent entre parenthèses.

# Couleurs et formes des fils les plus courants (câblage)

Type de données	Scalaire	Tableau 1D	Tableau 2D	
Numérique				(Réels)
				(Entiers)
Booléen				
Chaîne (caractères)				

Des fils brisés sont générés lorsqu'on tente de relier des objets dont les types de données sont différents et incompatibles (ex. relier une commande numérique à un indicateur chaîne)

# Constantes

---

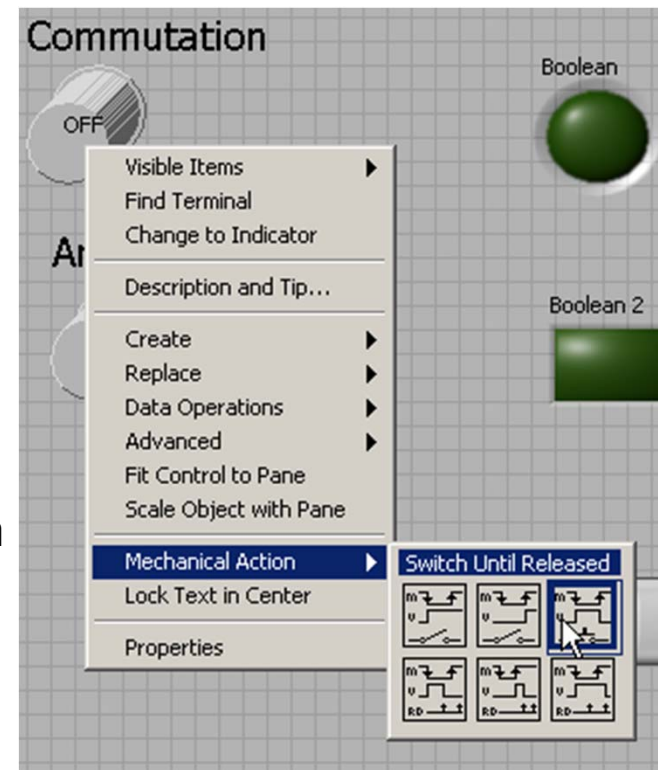
- Les constantes n'existent que dans le diagramme et ont des valeurs fixes
  - Constantes universelles ( $\pi$ ,  $\mathbf{R}$ ,  $\mathbf{e}$ , etc.)
  - Constantes définies par l'utilisateur
- Il y a des constantes pour tous les types de données

123 $\pi/2$ Tabcd



# Commandes Booléennes

- N'ont que deux valeurs: **vrai** ou **faux**
- Agissent comme des interrupteurs et ont **six** actions possibles:
  - Commutation (*Switch*)
    - à l'appui
    - au relâchement
    - jusqu'au relâchement
  - Armement (*Latch*)
    - Même 3 actions que la commutation
    - Revient à la valeur par défaut une fois lue par le VI



LV\_Cours2\_VI\_3\_H10.vi

# Graphe Déroulant (Waveform Chart)

---

- **Indicateur numérique** particulier qui affiche graphiquement une ou plusieurs courbes acquises à une vitesse constante ( $\Delta t$  constant entre les données)
- Le traçage se fait au fur et à mesure que les données sont reçues par le graphe
- Conserve un historique des données dans un "buffer" (1024 points par défaut)

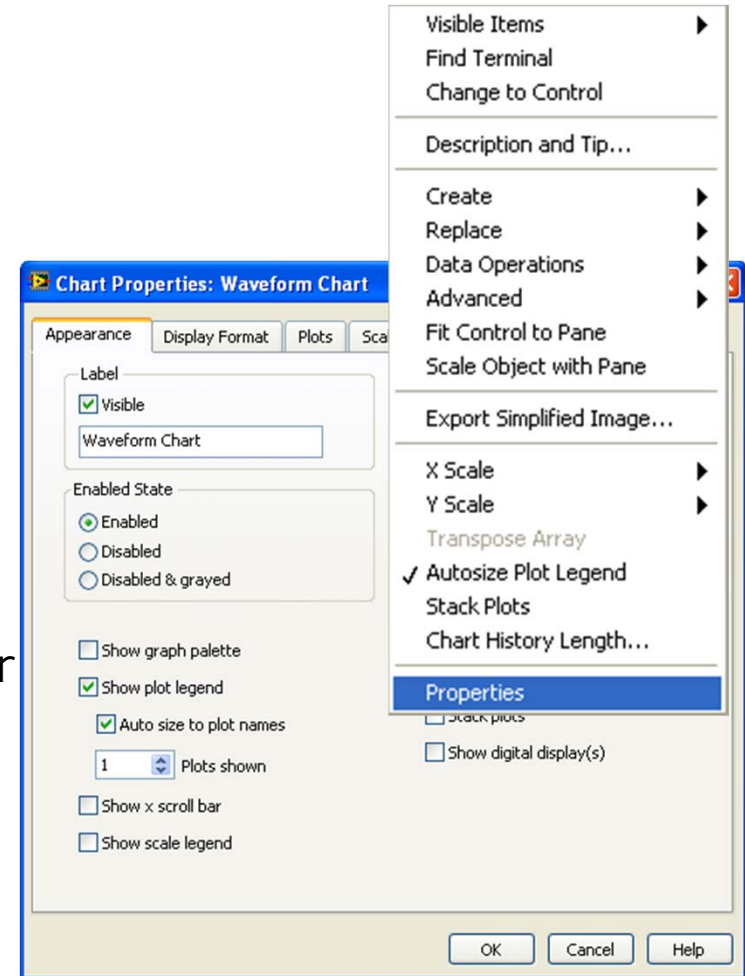
# Graphe (Waveform Graph)

---

- **Indicateurs** qui affichent graphiquement les données d'un ou plusieurs tableaux (tableaux 1D ou 2D) **d'un seul coup**
- Pour le graphe les points sont échantillonnés régulièrement ( $\Delta x = cte$ ). Il utilise le type de données Waveform (Temps de départ,  $\Delta t$  et données)
- **À utiliser dans le VI du laboratoire no.5**
- Exemple: LV\_cours3\_Graphe3\_A11.vi

# Configuration des objets de la face-avant

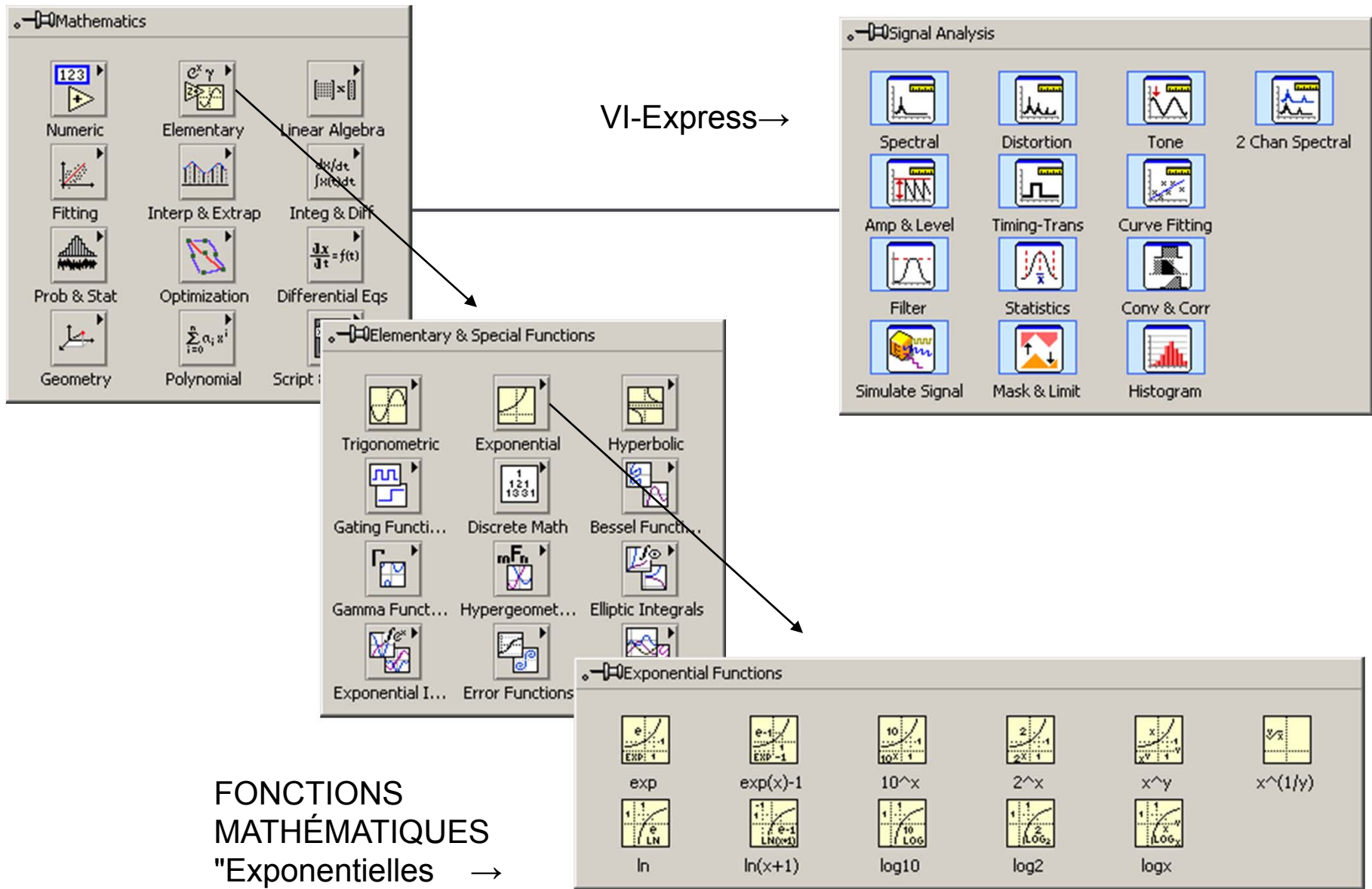
- À l'aide du menu local de l'objet (Cliquer droit sur l'objet), on peut modifier l'aspect et le comportement de tout objet, par exemple:
  - Afficher ou masquer des éléments optionnels
  - Changer le format d'affichage
  - Accéder à toutes les propriétés pour les modifier (Properties)



# Fonctions et VI-express

---

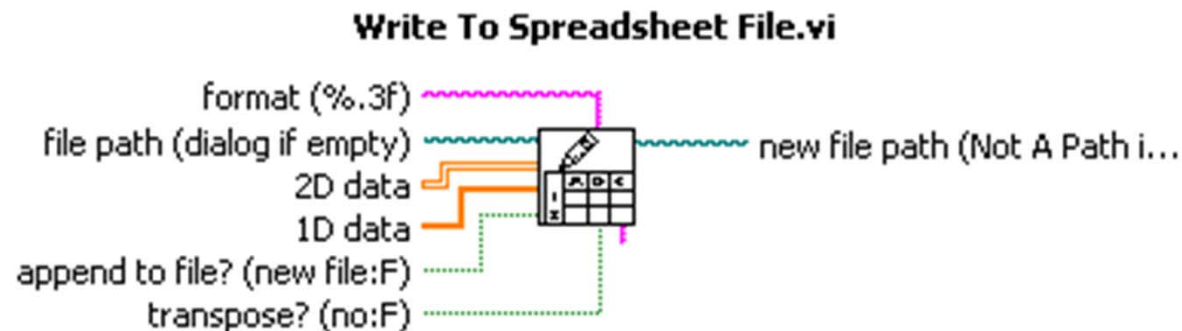
- Les fonctions sont accessibles à partir de la palette des fonctions
  - N'ont ni diagramme ni face-avant (non-éditable)
  - Possèdent des connecteurs (terminaux) d'entrée et de sortie
- Les VI-Express exécutent des tâches courantes
  - Sont configurés à l'aide d'un boîte de dialogue
  - Nécessitent un minimum de câblage
  - Accessibles à partir de la palette des fonctions
  - Ne peuvent être modifiés que si on les transforment en sous-VI



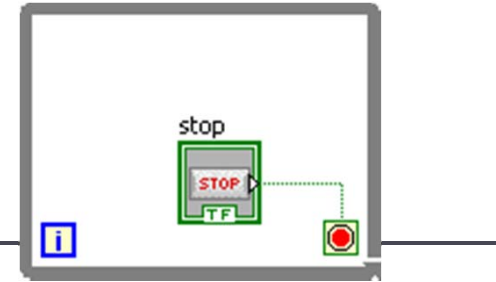
FONCTIONS  
MATHÉMATIQUES  
"Exponentielles" →

# Sous-VI

- LabVIEW possède une très grande bibliothèque de sous-VI qui effectuent des tâches spécialisées
- On peut aussi créer ses propres sous-VI en ajoutant à l'icône d'un VI, des terminaux qui serviront à faire entrer ou sortir des données (voir l'exercice F3)
- Les commandes et indicateurs d'un sous-VI échangent des données avec un autre programme via ses terminaux



# Boucle WHILE



- Structure délimitée par un cadre qui contient un sous-diagramme, un terminal de condition et un terminal compteur des itérations.
- Le sous-diagramme est exécuté jusqu'à ce que le terminal de condition reçoive une valeur booléenne particulière (VRAI ou FAUX).
- Par défaut, la boucle arrête sur la condition **VRAI** (peut être changé pour **FAUX** avec le menu local)
- **Il faut absolument** que la condition d'arrêt soit lue à **l'intérieur de la boucle**, sinon la boucle ne s'arrêtera pas.
- Le terminal **i** donne le nombre d'itérations achevées. Il démarre à zéro.

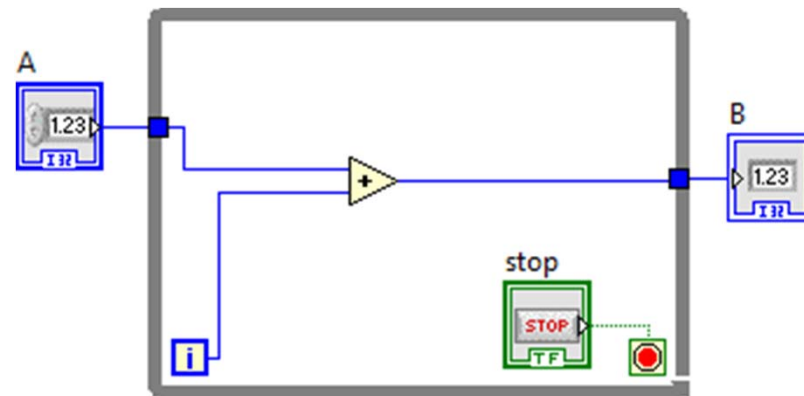




# Boucles WHILE (et FOR)

## Tunnels d'entrée et de sortie des données

- Les données d'entrée et de sortie passent par des tunnels localisés sur le cadre de la boucle
  - La valeur à un tunnel d'entrée est lue lors de la 1<sup>er</sup> itération seulement
  - La valeur envoyée à un tunnel de sortie est celle de la dernière itération



# Nœuds du diagramme

---

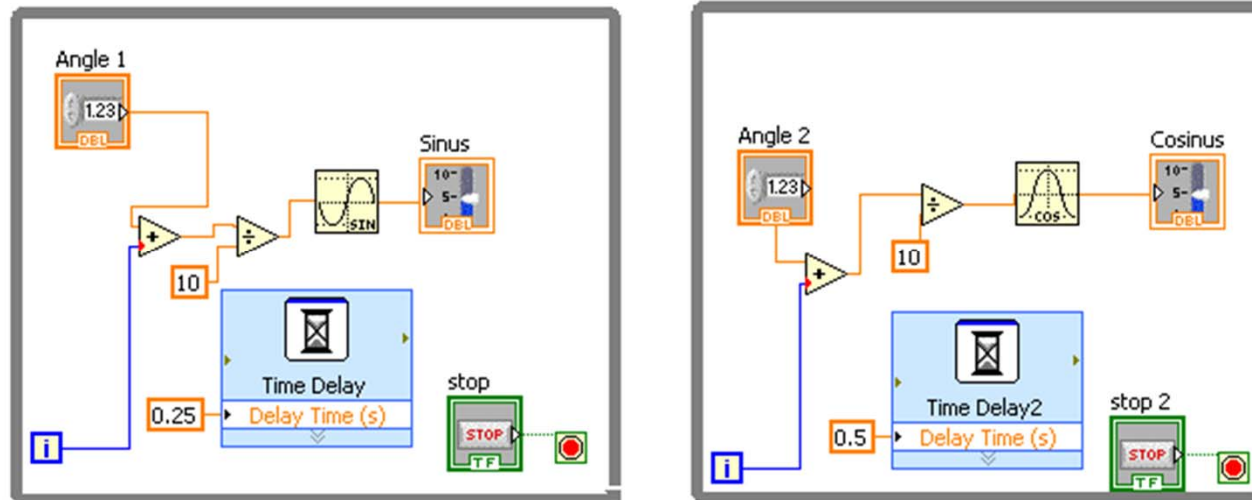
- Possèdent des entrées et des sorties
- **Et réalisent des opérations**
- Exemples de nœuds:
  - Fonctions
  - Sous-VI
  - VI-Express
  - Boucles WHILE, FOR
  - Structures (ex. CASE, SEQUENCE, etc.)

# Flux de données dans le diagramme

---

- Un nœud s'exécute lorsque toutes ses entrées sont disponibles
- Lorsqu'un nœud s'exécute, il produit des données de sortie qui sont dirigées vers le nœud suivant via les fils de liaison
- C'est le flux de données qui détermine l'ordre d'exécution des éléments du diagramme.

Exemple dans lequel deux boucles *WHILE* s'exécutent simultanément en l'absence de lien de dépendance entre les données (parallélisme d'exécution)



LV\_cours5\_VI2.vi

# Pour contrôler l'ordre d'exécution

---

- Si une dépendance naturelle des données existe:
  - Câbler le flux en fonction de la séquence d'évènements désirée
- S'il n'y a pas de dépendance naturelle
  - Créer une dépendance "artificielle" avec les paramètres dupliqués. Ces paramètres ont la même valeur à l'entrée qu'à la sortie du nœud (ex. Cluster d'erreur, etc.)
- Utiliser des structures SÉQUENCE (voir Sujets avancés à la fin du diaporama)

# Acquisition de données avec LabVIEW

MCours.com

# Sujets

---

- Périphériques DAQ
- Voies physiques et virtuelles
- NI-DAQmx et MAX
- Création d'un périphérique simulé
- Configuration d'une tâche (pour Labo. 5)

# Périphérique DAQ

---

- Un périphérique DAQ (*Data AcQuisition*)
  - acquiert ou génère des signaux
  - peut contenir plusieurs voies (canaux) et périphériques de conversion et de conditionnement



# Périphériques DAQmx et DAQmx simulés

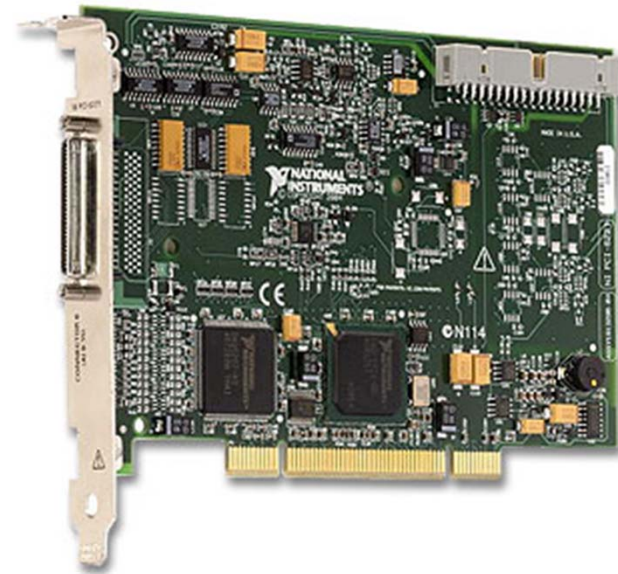
---

- Périphériques DAQmx, fabriqués par National Instruments
  - Comprennent les cartes enfichables sur bus d'ordinateur ainsi que les périphériques qui se connectent aux ports USB, "FireWire", Ethernet (Filaire et Wi-Fi), GPIB, etc.
- Périphérique DAQmx simulé
  - Réplique logicielle d'un périphérique DAQ
  - Se comporte de la même façon qu'un périphérique physique. Les programmes l'utilisant peuvent donc être vérifiés de manière exhaustive.

# Carte multifonction NI PCI-6221 (installée dans les ordinateurs du C407.14)

- **Entrée analogique:**
  - 16 Mono-connexion/8 Différentielle
  - 250 kéch./s
  - Résolution de 16 bits
- **Sortie analogique:**
  - 2 voies à 833 kéch./s
  - Résolution de 16 bits
- **E/S numériques:**
  - 24 E/S N · 1 MHz
- **Compteurs/timers:**
  - 2 · 32 bits · 80 MHz

NI PCI-6221



# Module externe d'acquisition de données NI USB 9234 (Installé au C407.14)

## 4 voies d'entrée analogiques

- Fréquence d'échantillonnage maximale de 51,2 kéch./s par voie
- Entrée de  $\pm 5$  V
- Convertisseur Delta-Sigma d'une résolution de 24 bits (un par voie)
- Couplage AC/DC sélectionnable par logiciel
- Conditionnement de signaux IEPE (capteurs piézo-électriques)
- Transfert de données à haute vitesse sur USB
- Alimenté par le bus USB



Laboratoire no. 5: Sert à lire les signaux à l'entrée et à la sortie des filtres.

# Voie (canal, "channel")

---

- Voie Physique
  - Terminal sur lequel vous pouvez mesurer ou générer un signal analogique ou numérique
- Voie Virtuelle
  - Ensemble des paramètres de propriétés pouvant inclure le nom, la voie physique, la configuration du terminal d'entrée, le type de mesures ou de génération, et des informations de mise à l'échelle.
  - Vous pouvez configurer des voies virtuelles d'un périphérique DAQmx avec le programme **MAX** (diapo suivante) ou bien avec un VI-Express DAQ Assistant dans LabVIEW.

# MAX



Measurement &  
Automation

---

## • MAX: Measurement & Automation Explorer

- Environnement de configuration centralisé qui vous permet de configurer tous vos périphériques National Instruments sans avoir recours au protocole de programmation des instruments
- Le **DAQ Assistant** permet de configurer des voies virtuelles et des tâches de mesure pour votre périphérique en vue de l'utiliser dans LabVIEW

# Création d'un périphérique simulé

---

- Dans **LabVIEW**, menu principal, choisir Tools et ensuite Measurement and Automation Explorer (MAX)
- Dans **MAX**
  1. Fenêtre Configuration (à gauche), choisir Périphériques et Interfaces et ouvrir le menu local (clic droit)
  2. Créer un nouvel objet, choisir Périphérique Simulé NI-DAQmx, puis terminer
  3. Dans la fenêtre Périphériques, choisir USB DAQ, NI USB-9234
  4. La création du périphérique simulé est terminée (noter le numéro du périphérique créé, ex. Dev1)

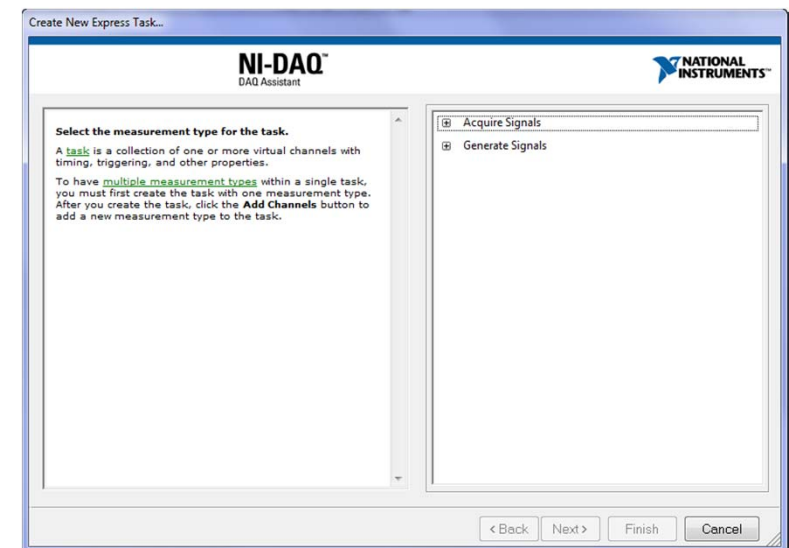
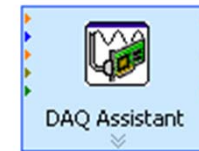
# Qu'est ce qu'une tâche?

---

- Fondamentalement, cela représente **la mesure ou la génération que l'on veut réaliser.**
- Un tâche comprend:
  - L'ensemble des paramètres de propriétés de une ou plusieurs voies
  - Le cadencement (taux d'échantillonnage), le nombre d'échantillons, le déclenchement et d'autres propriétés particulières à l'instrument

# Créer une tâche

- Dans le diagramme, insérer le VI-Express DAQ Assistant qui se trouve dans la palette Measurement I/O, sous-palette NI-DAQmx
- Lorsque vous déposer le VI DAQ Assistant dans le diagramme, le programme de configuration DAQ Assistant démarre automatiquement





# Créer une tâche

---

- Pour fin d'exemple, nous lirons deux signaux sur le module NI USB 9234 (simulé)
- Dans la fenêtre du VI-Express DAQ Assistant
  1. Choisir "Acquire Signals", puis "Analog Input" et finalement "Voltage"
  2. Voie physique, choisir USB-9234 simulé, les voies *ai0* et *ai1* et ensuite terminer
  3. Configurer de la voie *ai0* (voir diapo. suivante)
  4. Configurer la voie *ai1*
  5. Tester la tâche ("Run" dans la barre du haut)
  6. Fermer la fenêtre de l'assistant DAQ
- Le VI express est généré sur le diagramme

# Assistant DAQ, configuration de la voie

(étape 3 de la diapo. précédente)

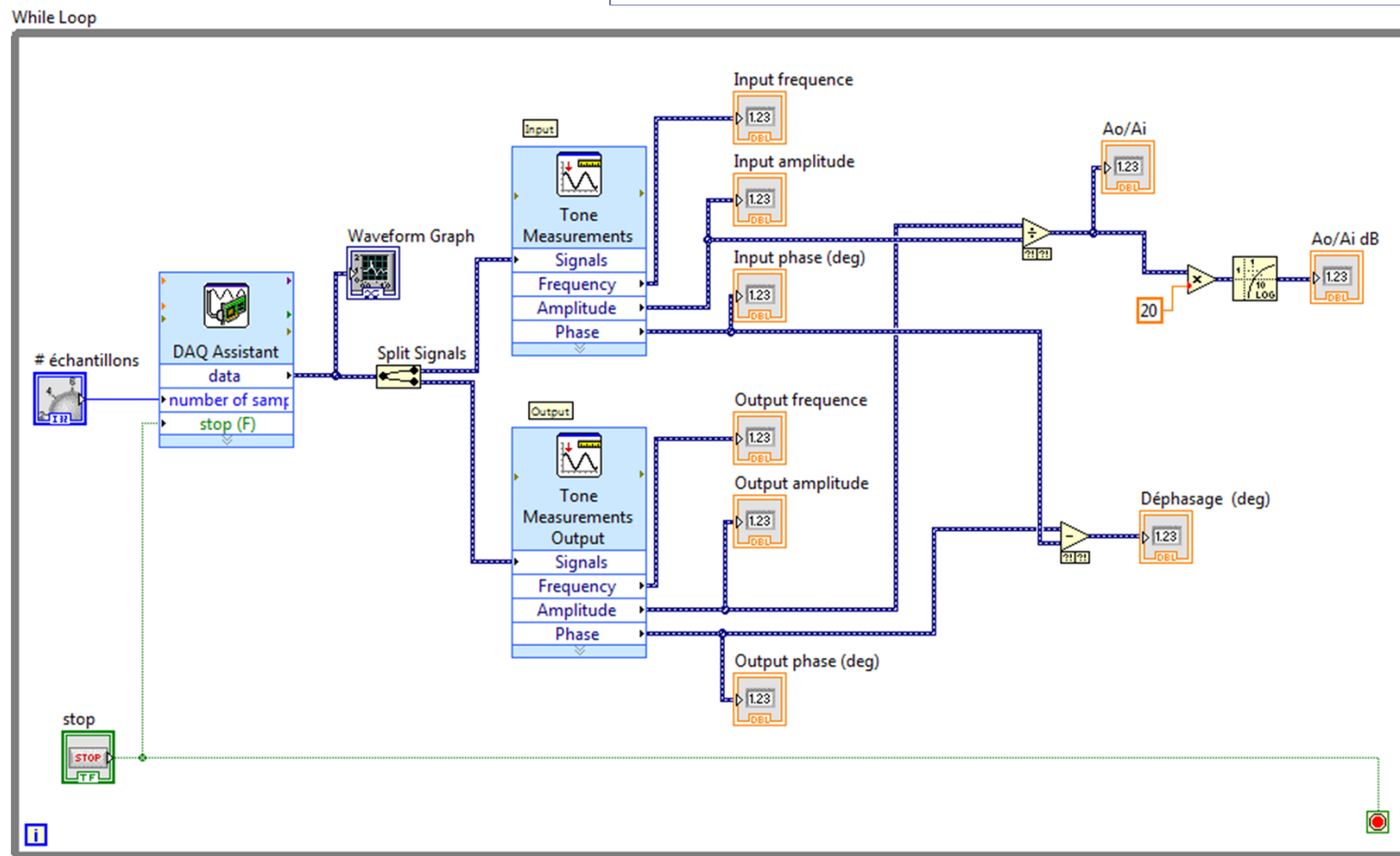
---

- a) Renommer la voie ai0 "Signal d'entrée" (clé F2)
- b) Modifier
  - Gamme du signal d'entrée (*Signal Input Range*):  $\pm 5$  V
  - Configuration du terminal (*Terminal Configuration*): Pseudo Diff
  - Modes d'acquisition (*Acquisition Mode*): Continuous Samples
  - Échantillons à lire (*Samples to Read*): 1000
  - Fréquence (*Rate Hz*): 51200 (éch./s)
  - La Fréquence et le nombre d'échantillons sont modifiables durant l'exécution de votre VI
- c) Onglets Déclenchement et Cadencement Avancé (*Triggering and Advanced Timing*)
  - NE RIEN MODIFIER

# Exemple d'un VI pour lire les signaux à l'entrée et à la sortie du filtre (Labo. 5)

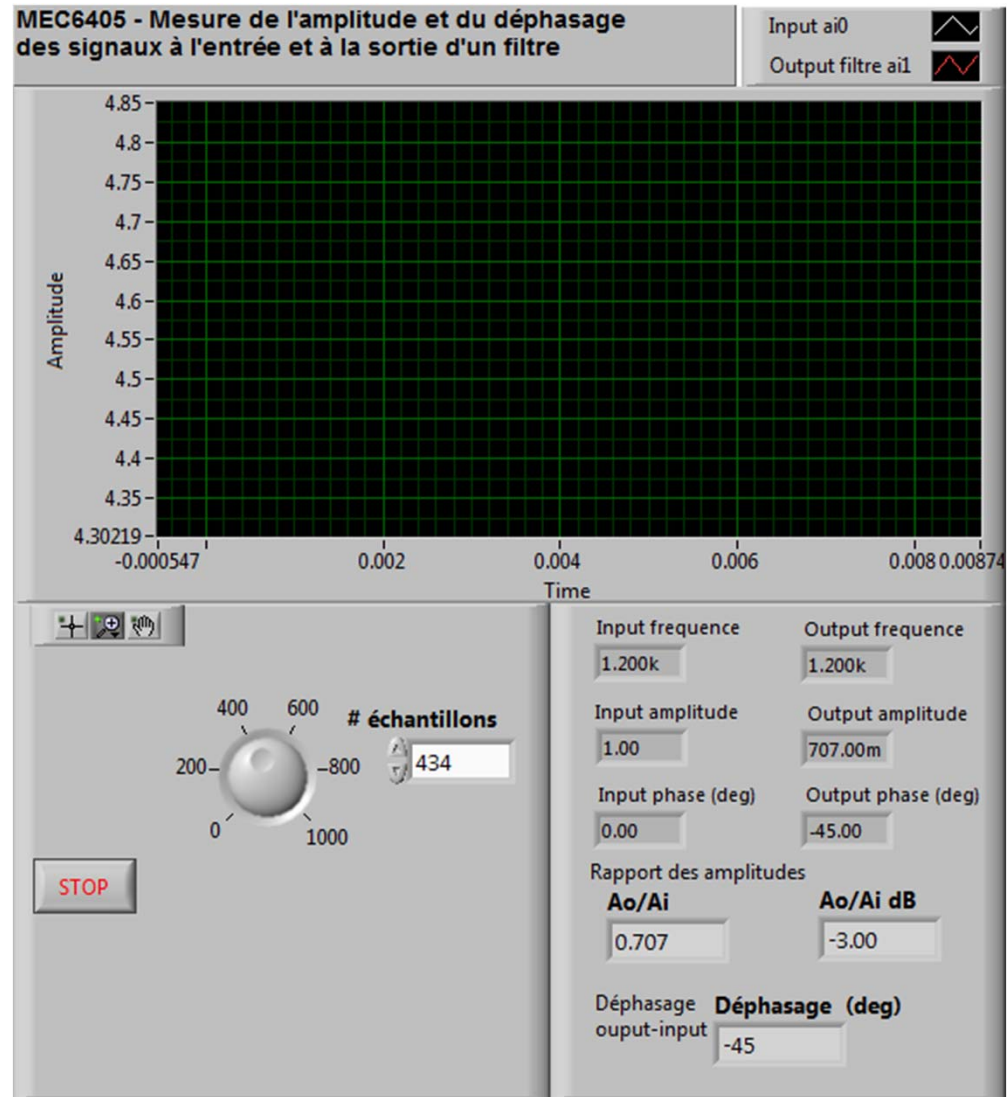
DIAGRAMME DU VI

Acquisition\_Labo5\_MEC6405\_A12.vi



# Exemple de FACE-AVANT DU VI

Acquisition\_Labo5\_  
MEC6405\_A12.vi

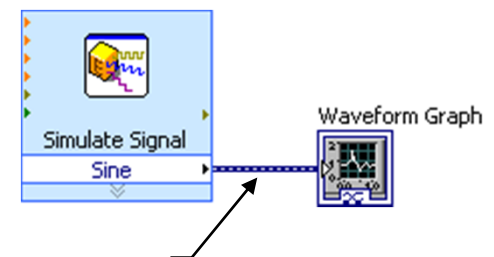


# Données Dynamiques des VI-Express

- Les VI-Express utilisent le type de données *dynamique* qui contient un ou plusieurs *Waveform*
- Un *Waveform* est un *Cluster* qui comprend:
  - *t0*: temps de départ
    - Temps absolu ou,
    - Temps relatif au départ de la mesure
  - *dt*: incrément de temps entre les données
  - *Y*: tableau 1D des données
- Les graphes (*Waveform Graph*) peuvent tracer directement les données *dynamiques* provenant d'un VI-Express

Waveform		<i>t0</i> absolu
<b>t0</b>	Y	0
15:52:55		0.063418
2010-02-08		0.12658
<b>dt</b>		0.189233
0.001000		0.251123

Waveform		<i>t0</i> relatif au départ
<b>t0</b>	Y	0
00:00:00		0.063418
YYYY-MM-DD		0.12658
<b>dt</b>		0.189233
0.001000		0.251123



Fil de données  
Dynamiques

# SUJETS AVANCÉS

---

- Structure CONDITION (CASE)
- Boucle FOR
- Tunnels auto-indexés
- Tableaux
- Clusters
- Structure SEQUENCE
- Entrée et sortie sur fichier

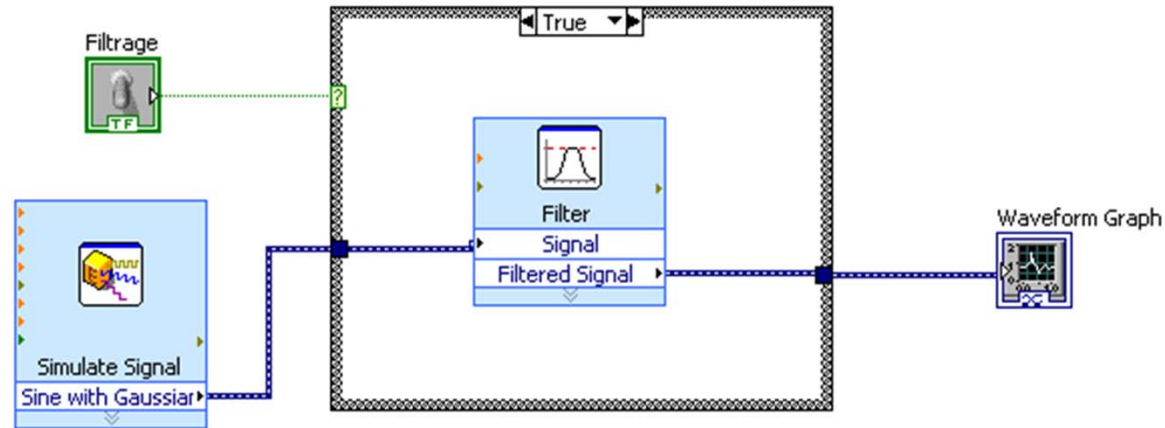
# Structure CONDITION (CASE)

---

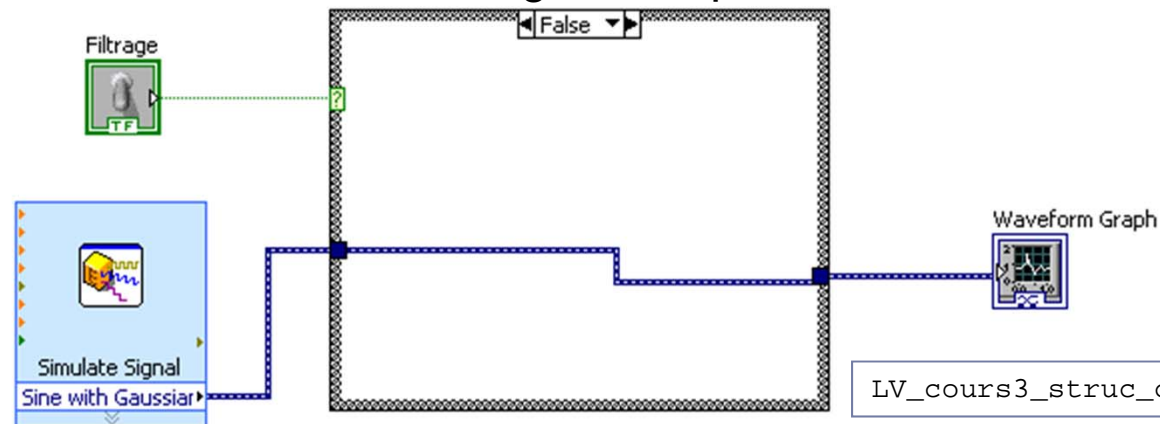
- Exécute un sous-diagramme selon la valeur d'entrée transmise à la Structure Condition
- Similaire à "if...then...else"
- Valeurs d'entrée possibles:
  - Booléen
  - Entier
  - Chaîne
  - Type énumération
- Il n'y a pas de limite au nombre de sous-diagrammes (conditions) que peut contenir une Structure Condition

Exemple: Filtrer ou non un signal à l'aide d'une structure condition avec valeur d'entrée booléenne

Sous-diagramme pour la condition "Vrai"



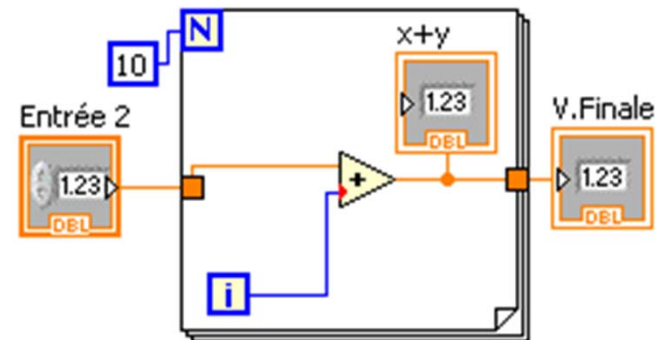
Sous-diagramme pour la condition "Faux"



LV\_cours3\_struc\_cond1\_A11.vi



# Boucle FOR

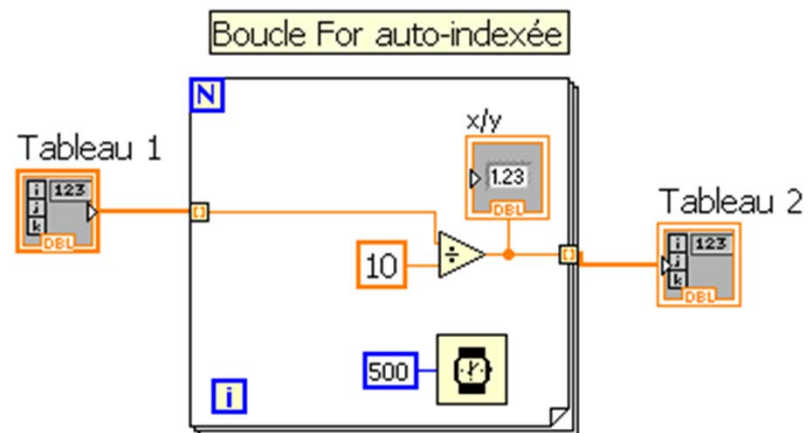
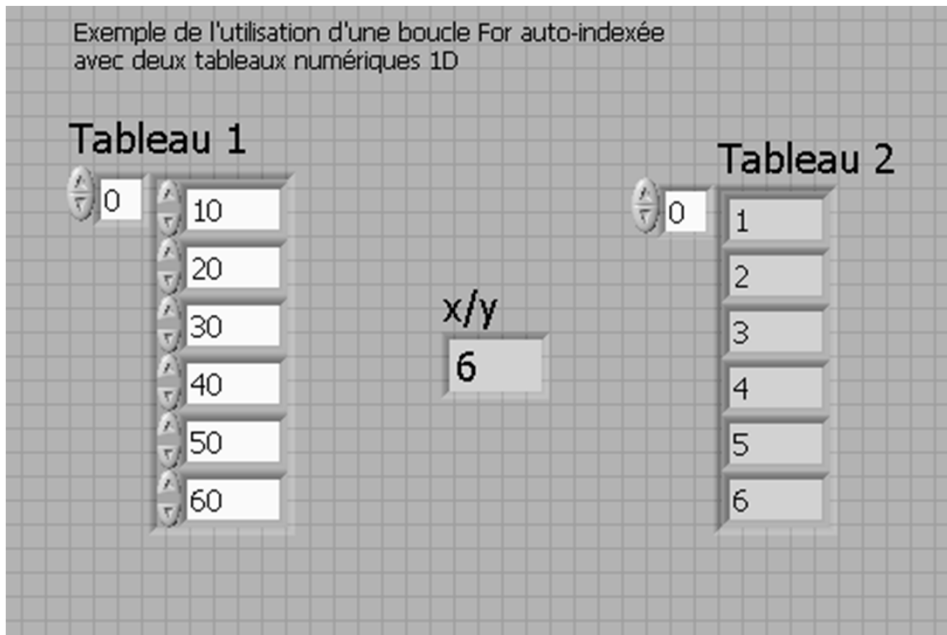


- Une boucle FOR exécute le sous-diagramme le nombre de fois défini par le terminal de comptage **N** (un terminal d'entrée)
- Le terminal d'itération **i** (un terminal de sortie), contient le nombre d'itérations achevées. Le comptage démarre toujours à zéro
- Les données d'entrée et de sortie passent par des tunnels situés sur le pourtour du cadre de la boucle

# Tunnels auto-indexées dans les boucles

---

- Si un tableau est connecté à un tunnel d'entrée auto-indexé d'une boucle, on lit alors chaque élément du tableau à raison d'un élément par itération, en commençant par le premier élément du tableau.
- Si un tableau (indicateur) est connecté à un tunnel de sortie auto-indexé, il reçoit un élément par itération. Ces éléments sont accumulés dans un tableau qui sera transféré hors de la boucle lorsque celle-ci termine sa dernière itération.
- Si l'auto-indexation est désactivé, un tableau relié à un terminal d'entrée (ou de sortie) est lu (ou écrit) en entier lors de la:
  - première itération pour un tableau à l'entrée
  - dernière itération pour un tableau à la sortie



LV\_cours4\_FOR\_autoindex1.vi

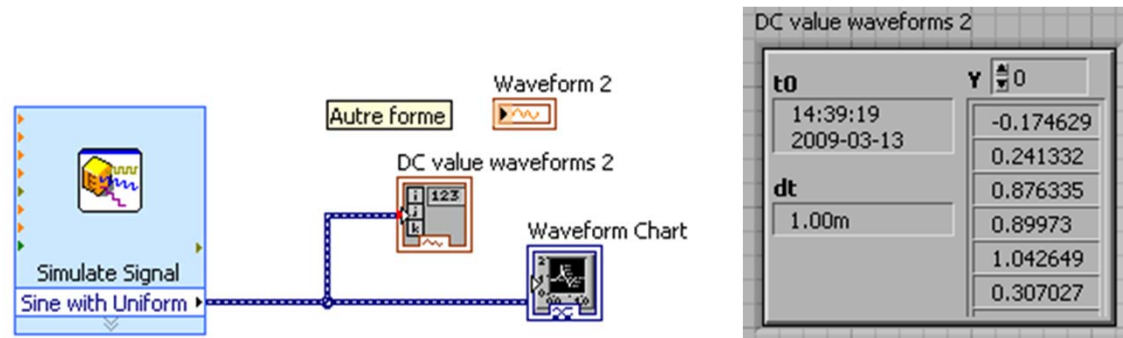
# Groupage des données

---

1. Tableaux (données du même type)
2. Clusters (données de types différents)

# Tableaux (*Arrays*)

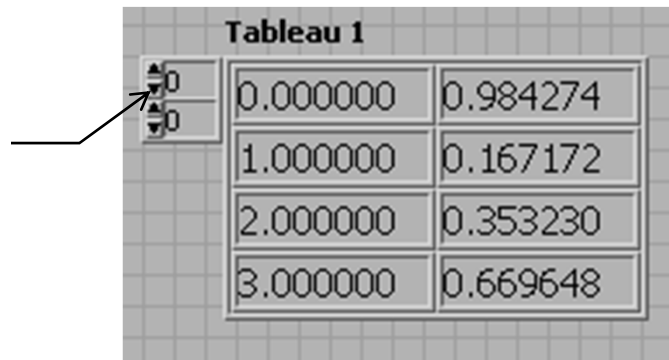
- Un tableau est défini par des éléments et des dimensions
- Vous pouvez construire des tableaux de données numériques, de booléens, de chemins, de chaînes de caractères, de waveforms et de clusters.
  - **Waveforms:** Type de données qui contient des lectures multiples  $Y_i$  effectuées à une fréquence de lecture particulière ( $dt$ ) à partir d'un temps de départ ( $t0$ ). Il s'agit en fait d'un cluster particulier.



# Indices

- Un tableau utilise des indices pour que vous puissiez accéder facilement à tout élément particulier.
- L'indice d'une dimension démarre à zéro

Indices:  
Haut: Ligne  
Bas: Colonne



The diagram shows a 2D array titled "Tableau 1" on a grid background. To the left of the array, there are two vertical arrows: the top one points upwards and is labeled "0", and the bottom one points downwards and is also labeled "0". An arrow points from the text "Haut: Ligne" to the top arrow, and another arrow points from "Bas: Colonne" to the bottom arrow. The array itself is a 4x2 grid of cells containing the following values:

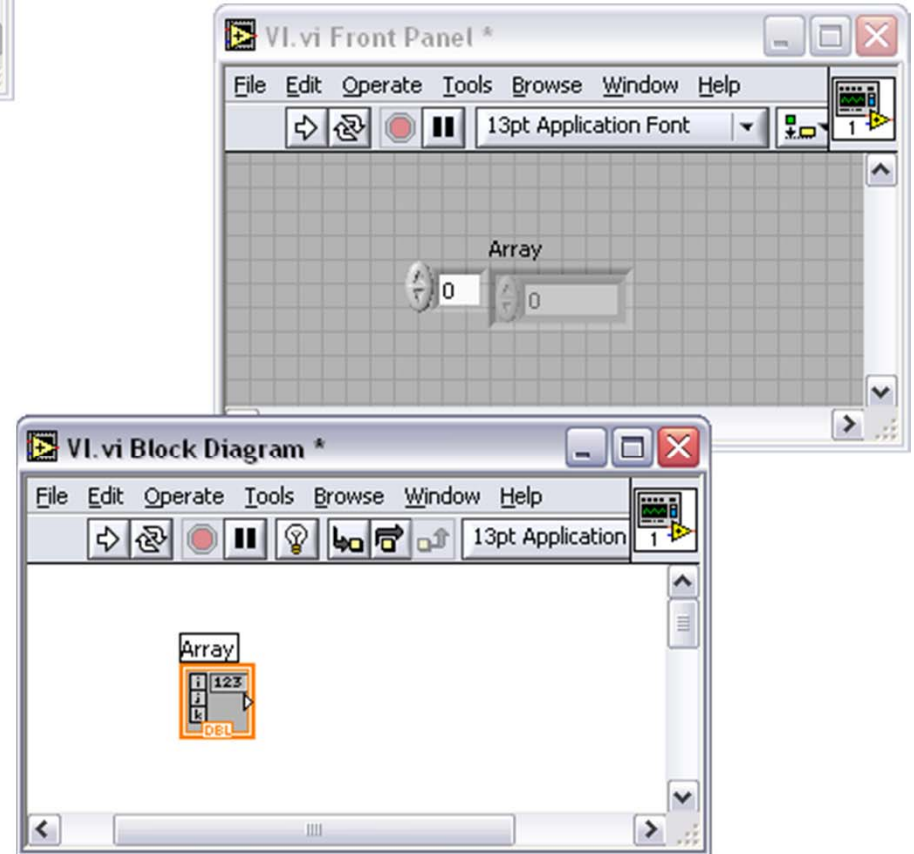
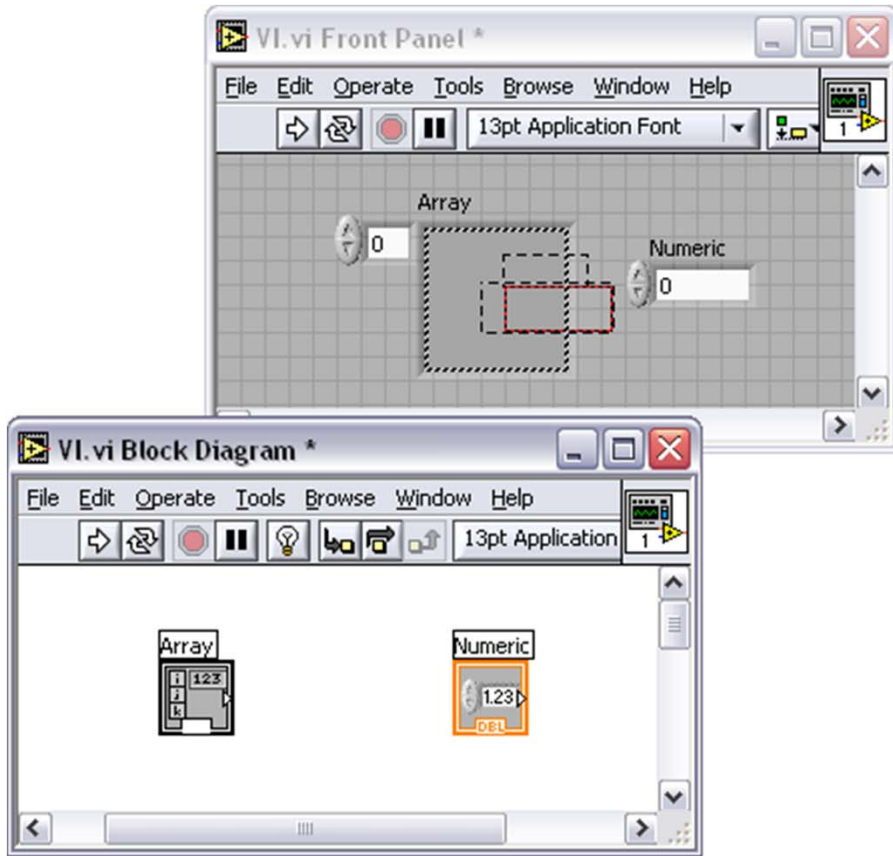
0	0
0.000000	0.984274
1.000000	0.167172
2.000000	0.353230
3.000000	0.669648

Ex. de tableau 2D (2 indices)

# Création d'un tableau

---

- Ajoutez un tableau (*array*) sur la face-avant. À cette étape, le tableau n'est pas associé à aucun type d'élément.
- Choisir l'élément de commande ou d'indicateur de votre choix (numérique, booléen, chaîne, etc.) et le glisser à l'intérieur du tableau
- Selon le type d'élément inséré, le tableau devient une commande ou un indicateur
- Par défaut, le tableau aura une seule dimension. Ajouter d'autres dimensions (ou indices) à l'aide du menu local du tableau (*add dimension*).

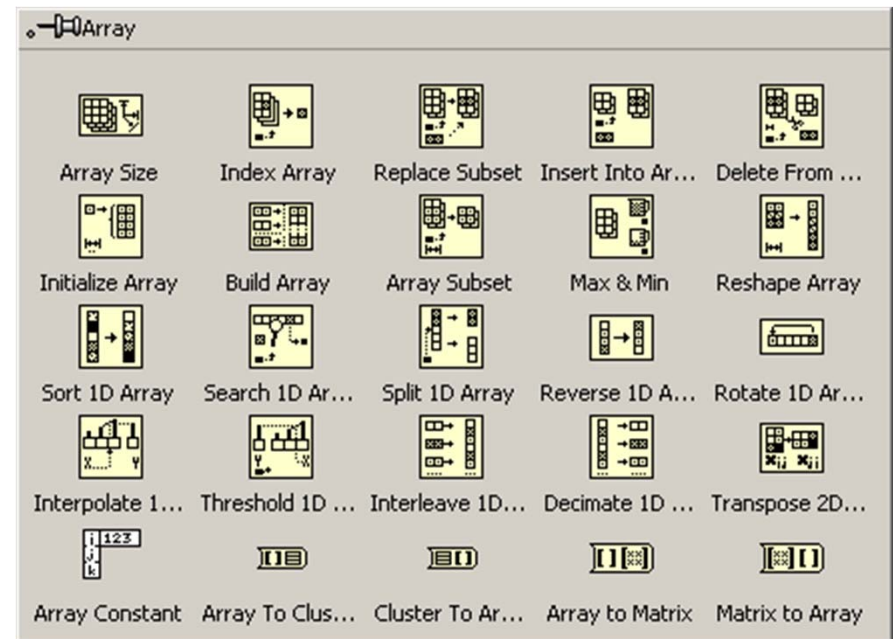




# Fonctions de tableau (*Array functions*)

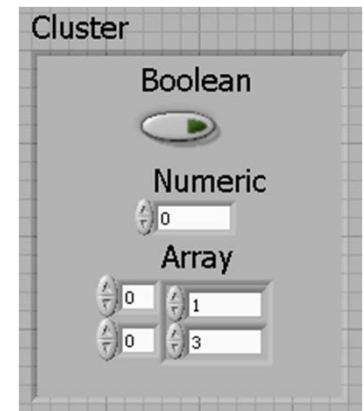
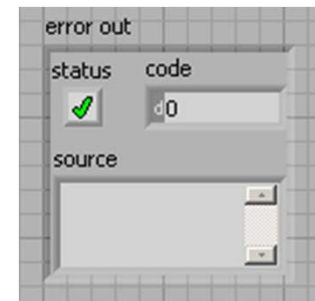
Servent à:

- Extraire des éléments d'un tableau
- Insérer, supprimer ou remplacer des éléments dans un tableau
- Créer, diviser des tableaux



# Cluster (*Cluster*)

- Un cluster regroupe des éléments de données de types différents
- Par exemple, le cluster d'erreur LabVIEW comprend une valeur booléenne, une valeur numérique et une chaîne
- L'assemblage de plusieurs éléments de données dans des clusters :
  - élimine l'encombrement des câbles sur le diagramme
  - réduit le nombre de terminaux de connexion nécessaires pour les sous-VIs



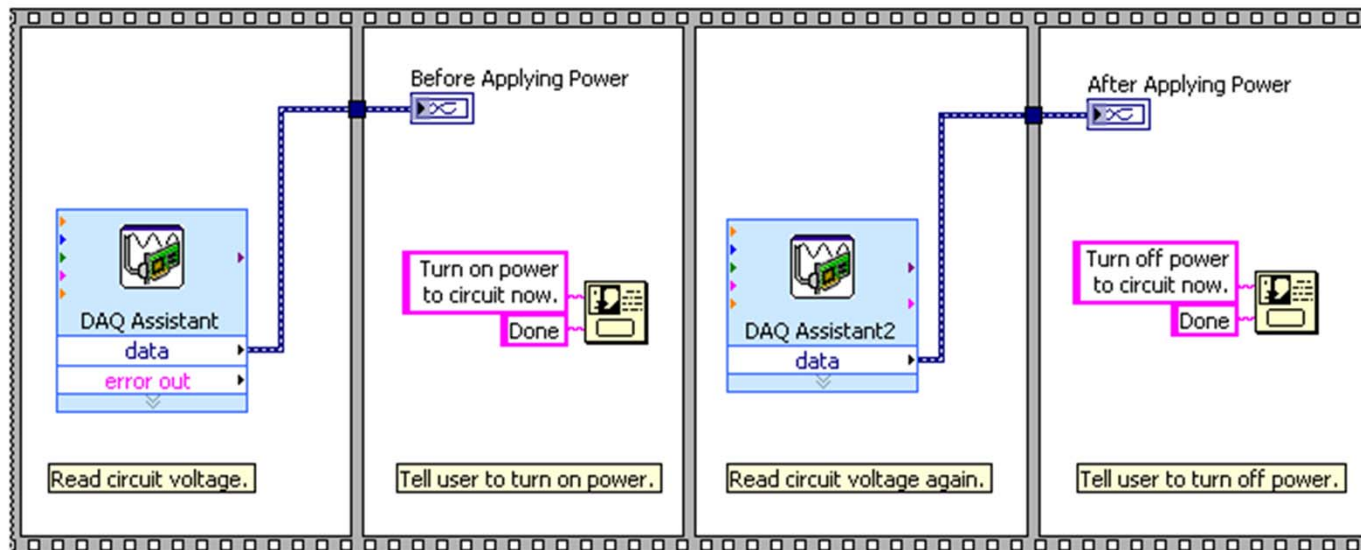
# Structures SÉQUENCE

---

- Contiennent un ou plusieurs sous-diagrammes (étapes) qui s'exécutent dans un ordre séquentiel
- Dans chaque sous-diagramme, comme dans tout autre diagramme, la dépendance des données détermine l'ordre d'exécution des nœuds
- Les structures Séquence garantissent l'ordre d'exécution et interdisent l'exécution en parallèle de leurs différents sous-diagrammes

# Structure Séquence déroulée

- Affiche tous les sous-diagrammes (étapes) et les exécute de gauche à droite
- Les valeurs de sortie quittent chaque sous-diagramme lorsque celui-ci finit de s'exécuter



# Entrée/Sortie sur fichiers (*File I/O*)

---

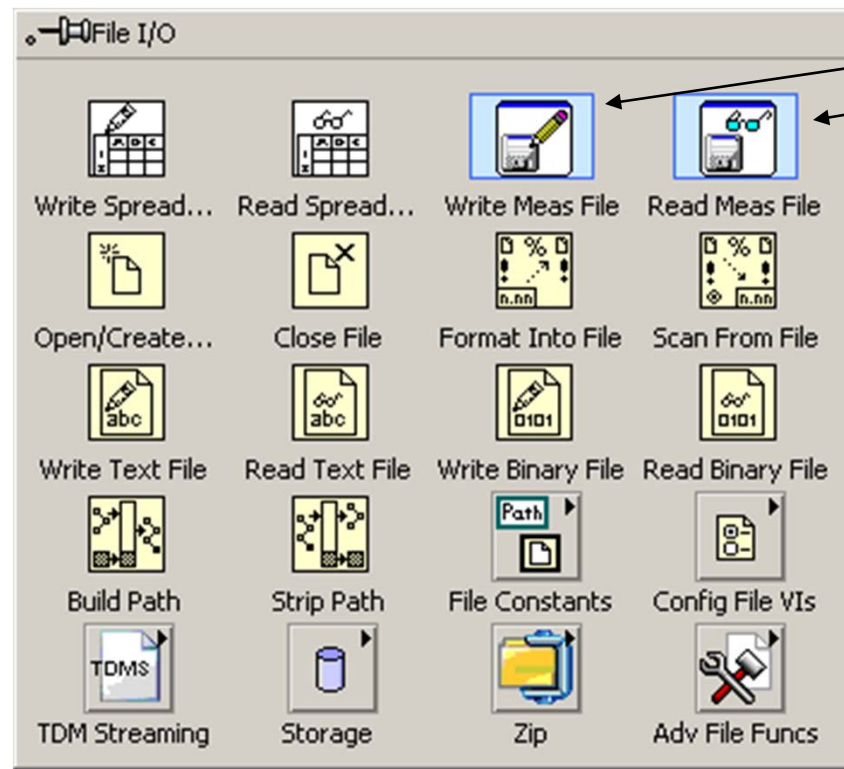
- Vous pouvez lire ou écrire des données dans les fichiers avec trois formats :
  - texte
  - binaire
  - journal (Datalog)

# Choix du format des fichiers

---

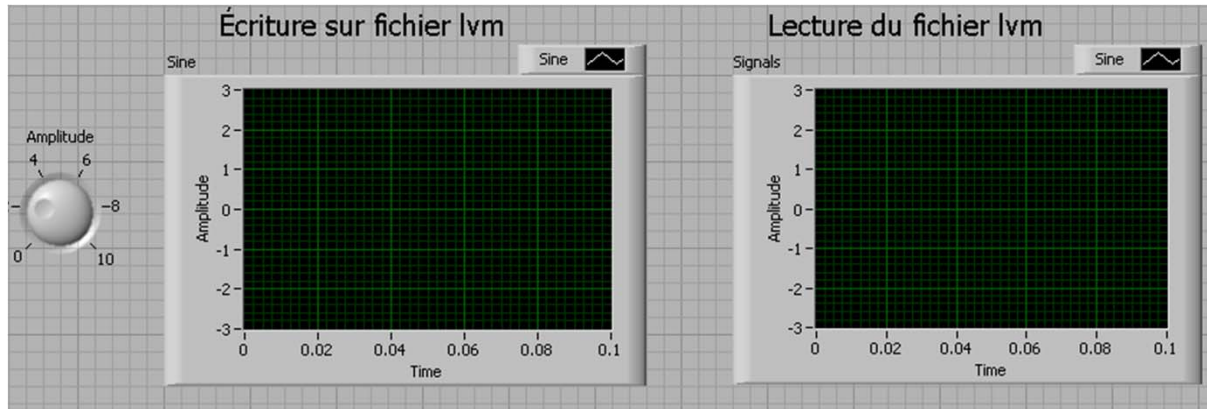
- Pour manipuler des enregistrements de **données complexes** ou des types de données différents dans LabVIEW, utilisez le fichier **journal (.lvm)** qui représente le meilleur moyen de stocker des données dans des fichiers **texte ou binaire**.
- Les fichiers journal sont spécialement conçus pour être écrits et relus avec un VI Express. On peut aussi relire les fichiers avec d'autres logiciels (traitement de texte, tableur).
  - Pour rendre vos données **accessibles** à d'autres applications, par exemple Excel, utilisez des fichiers **texte** qui sont les plus portables.
  - Pour un **accès aléatoire** utilisez des fichiers **binaires** qui sont plus performants que les fichiers texte en termes d'espace disque et de rapidité.

# Fonctions d'E/S (*File I/O*)



Pour fichier journal

# Fichier journal (.lvm)



Utilisation des VI express pour l'écriture et la lecture avec fichier .lvm (Labview Measurements)



LV\_cours5\_lvm\_A11.vi

```

LabVIEW Measurement
Writer_Version    0.92
Reader_Version    1
Separator         Tab
Multi_Headings   Yes
X_Columns         One
Time_Pref         Relative
Operator          lumarc
Date              2009/03/13
Time              17:26:00.872031
***End_of_Header***

Channels          1
Samples           100
Date              2009/03/13
Time              17:26:00.872031
X_Dimension       Time
X0                0.0000000000000000E+0

```

```

Delta_X           0.001000
***End_of_Header***

X_Value           Sine          Comment
0.000000         0.000000
0.001000         0.138924
0.002000         0.277289
0.003000         0.414537
0.004000         0.550117
0.005000         0.683482
0.006000         0.814095
0.007000         0.941431
0.008000         1.064977
0.009000         1.184235
0.010000         1.298726
0.011000         1.407988
0.012000         1.511582
0.013000         1.609091
0.014000         1.700122
0.015000         1.784308

```