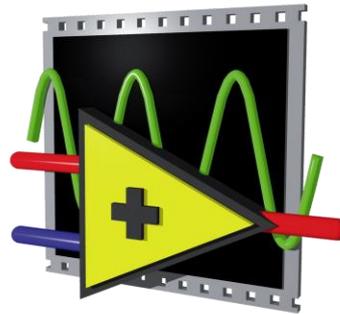
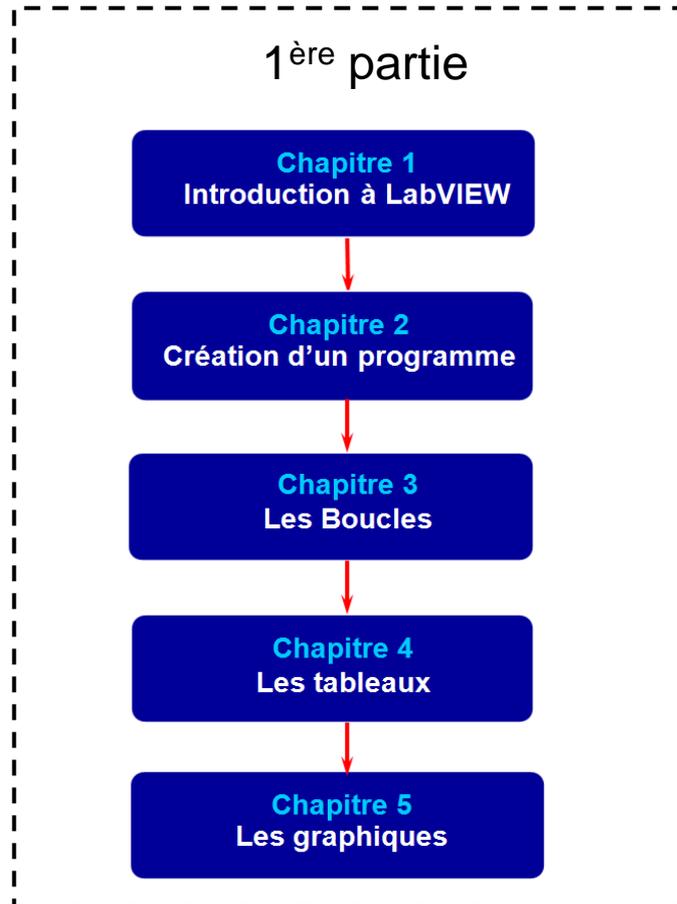


# Introduction à LabVIEW – 2<sup>ème</sup> partie



Ces slides sont en grande partie tirées du cours de  
Aldo Vaccari, Unité Power & Control / HES-SO Valais, Sion

# Introduction à LabVIEW – 2<sup>ème</sup> partie



# Chapitre 6

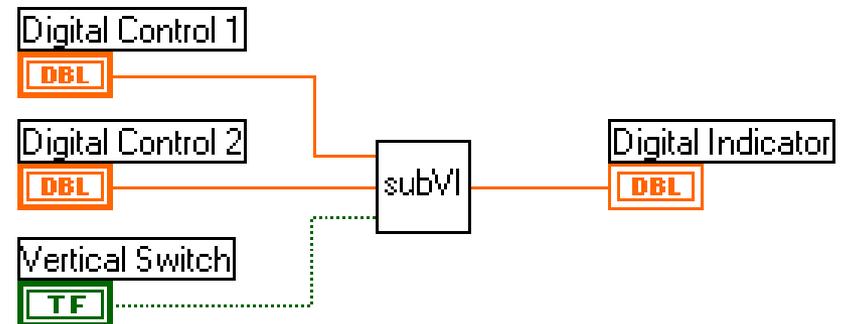
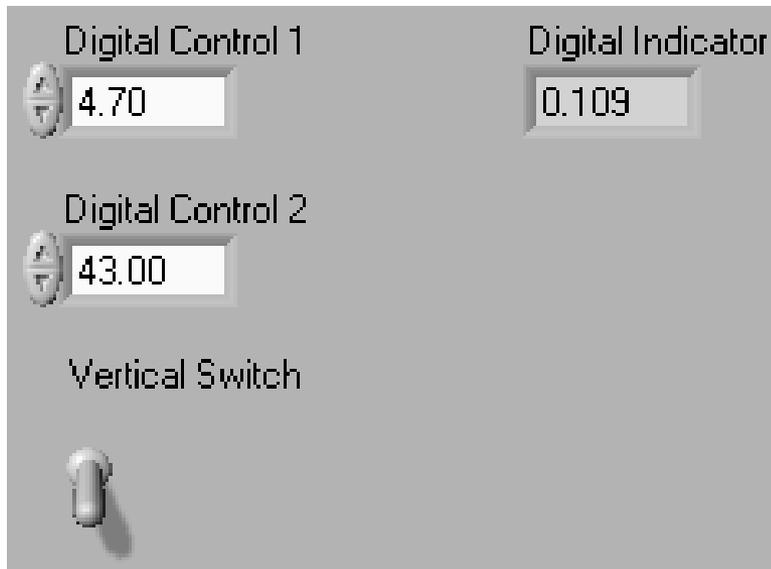
## Création d'un sous-VI

### Thèmes :

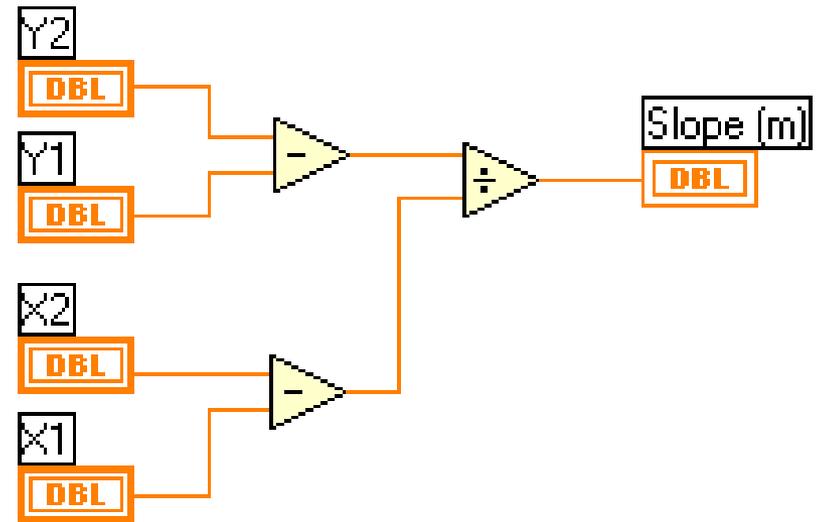
- A. Qu'est-ce qu'un sous-VI ?
- B. Comment créer un *SubVI* .
- C. Comment éditer l'icône et le connecteur ?
- D. Comment utiliser un VI comme sous-VI ?

# Sous-Vis (*subVIs*)

- Consiste à utiliser un VI dans le block diagram d'un autre VI de plus haut niveau
- Nécessite l'utilisation de l'icône et du connecteur

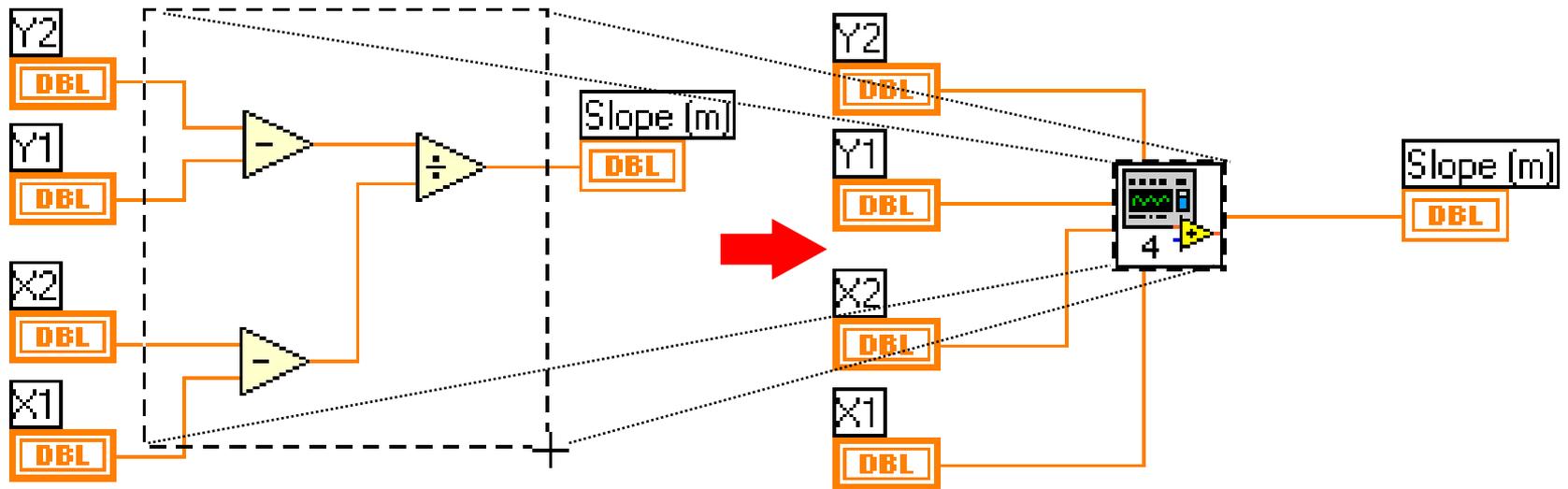


# Exemple : sous-VI de calcul de pente



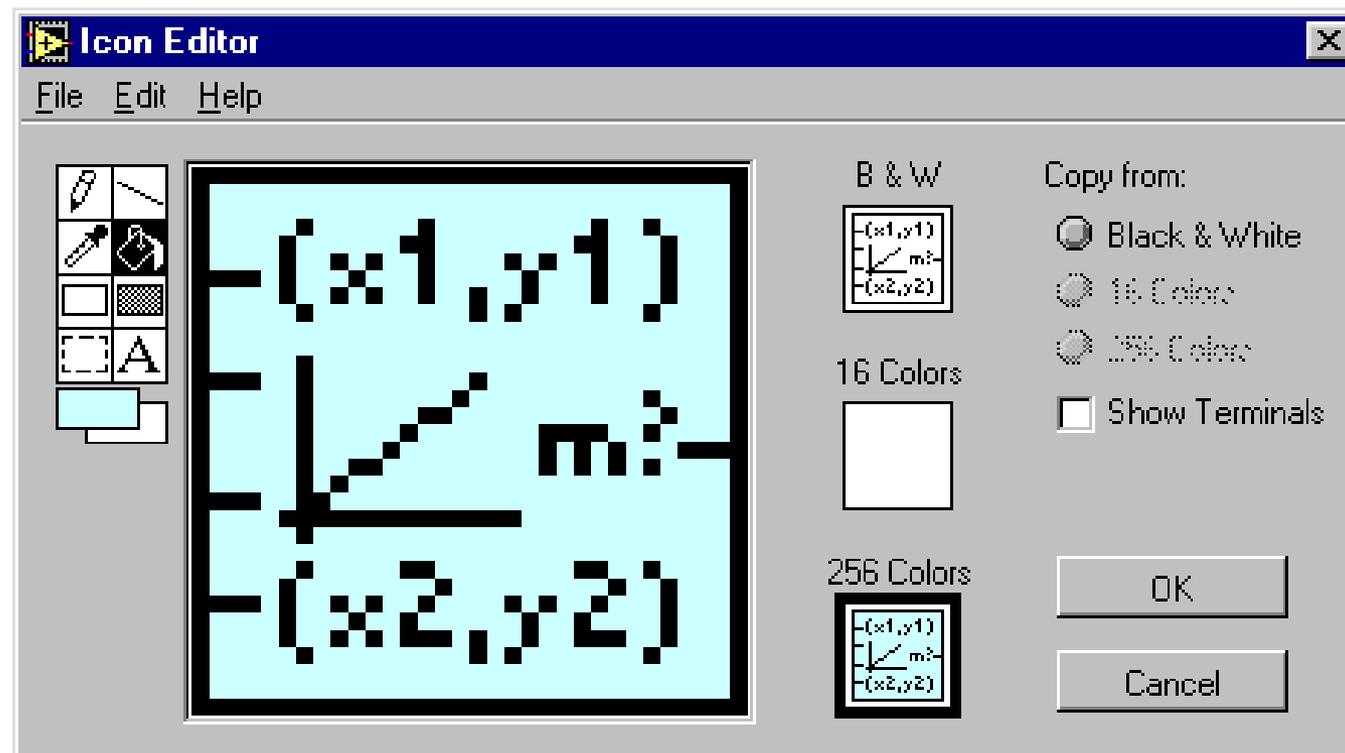
# L'option "Create SubVI"

- Sélectionner une partie de programme à convertir dans un sous-VI
- Sélectionner Create SubVI depuis le menu Edit



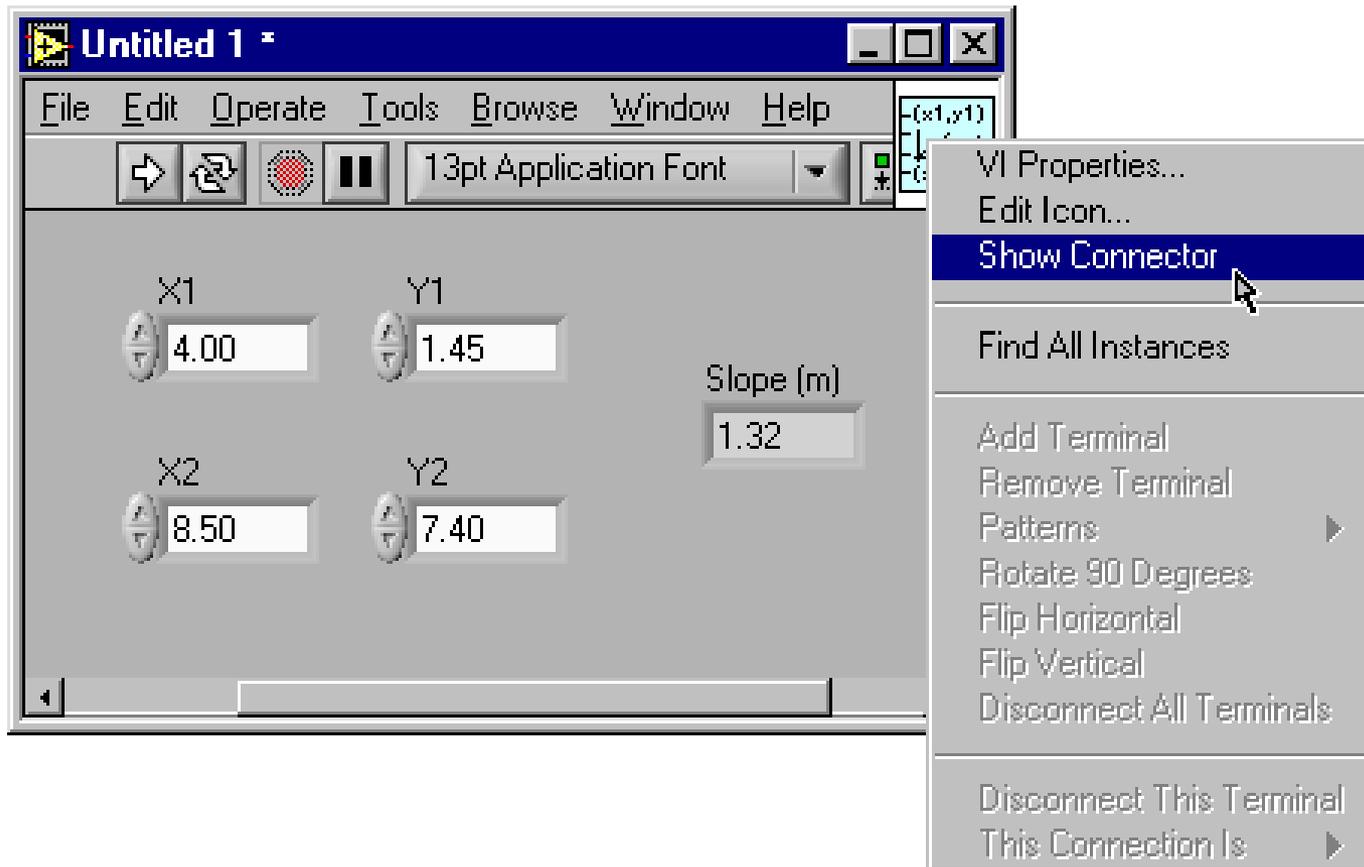
# Edition de l'icône

- Click de droite sur l'icône en haut à droite (depuis le Panel ou le Diagram)



# Edition du connecteur

Click de droite sur l'icône (depuis le Panel seulement)



# Utilisation d'un VI comme sous-VI

- Les changements faits au sous-VI sont sauvés dans la mémoire jusqu'à ce qu'il soit sauvé sur le disque

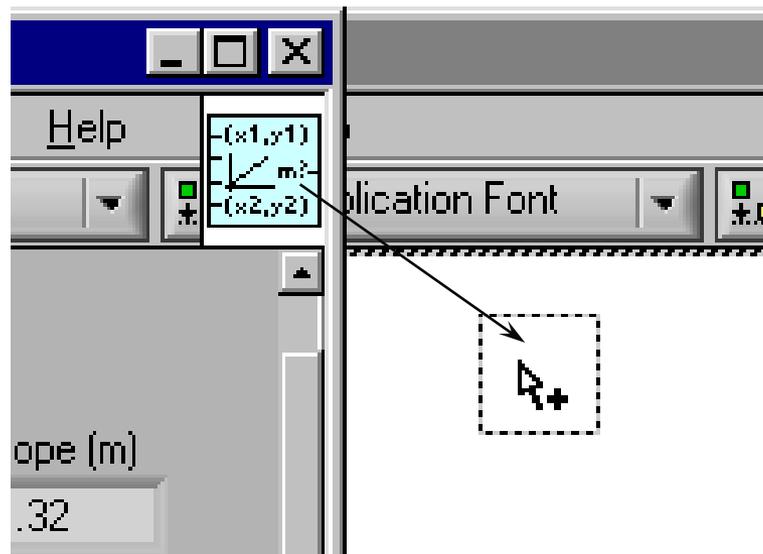


- Appel d'un sous-VI

Palette de Fonctions»Select a VI...

OU

- Tirer l'icône (à droite en haut) **sur le diagramme du sub-VI** vers sur le diagramme du VI .



# Résumé du chapitre

- Les VIs peuvent être utilisés comme sousVIs après avoir réalisés son icône et son connector
- Utilisation de l'option Create SubVI pour modulariser le block diagram d'un Vi compliqué et faire des sous-programmes
- L'icône est créée avec l'Icon Editor
- Le connecteur est défini par le choix du nombre de terminaux
- Chargement du sous-VIs en utilisant l'option Select a VI...  de la palette de fonctions ou en tirant l'icône sur le nouveau diagram

# Exercices : à vous de jouer !

- Modifier l'exercice 1 (slide 39 de la première partie) pour réaliser le calcul dans un sous-programme réutilisable ailleurs.
- Faire ensuite un programme principal qui effectue dans une boucle While trois conversions de température simultanément en utilisant le même sous-programme de calcul

exercice\_subVI.vi Front Panel

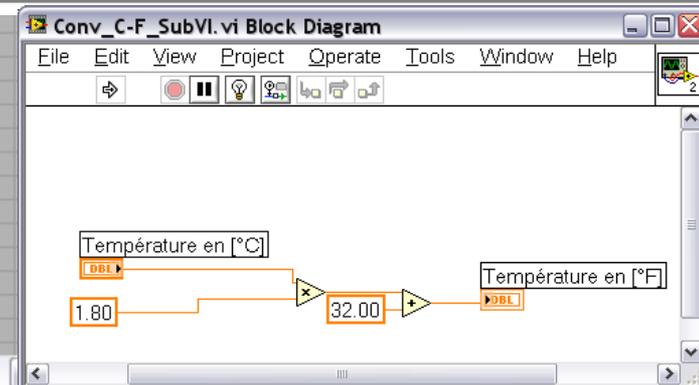
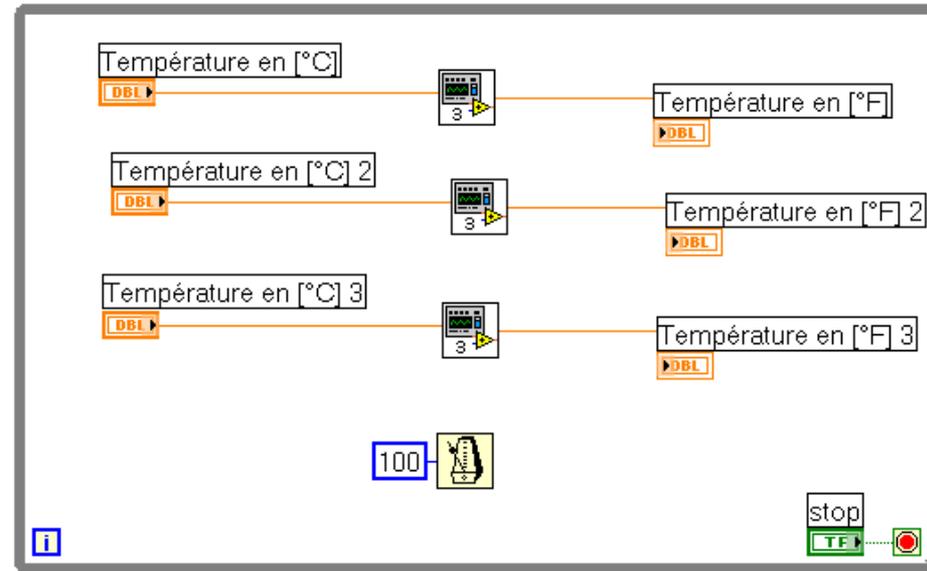
File Edit View Project Operate Tools Window Help

18pt Application Font

Conversion degrés Celsius [°C] en degrés Farenheit [° avec des sous-VI

Température en [°C]	Température en [°F]
0.00	0.00
Température en [°C] 2	Température en [°F] 2
0.00	0.00
Température en [°C] 3	Température en [°F] 3
0.00	0.00

STOP





# Chapitre 7

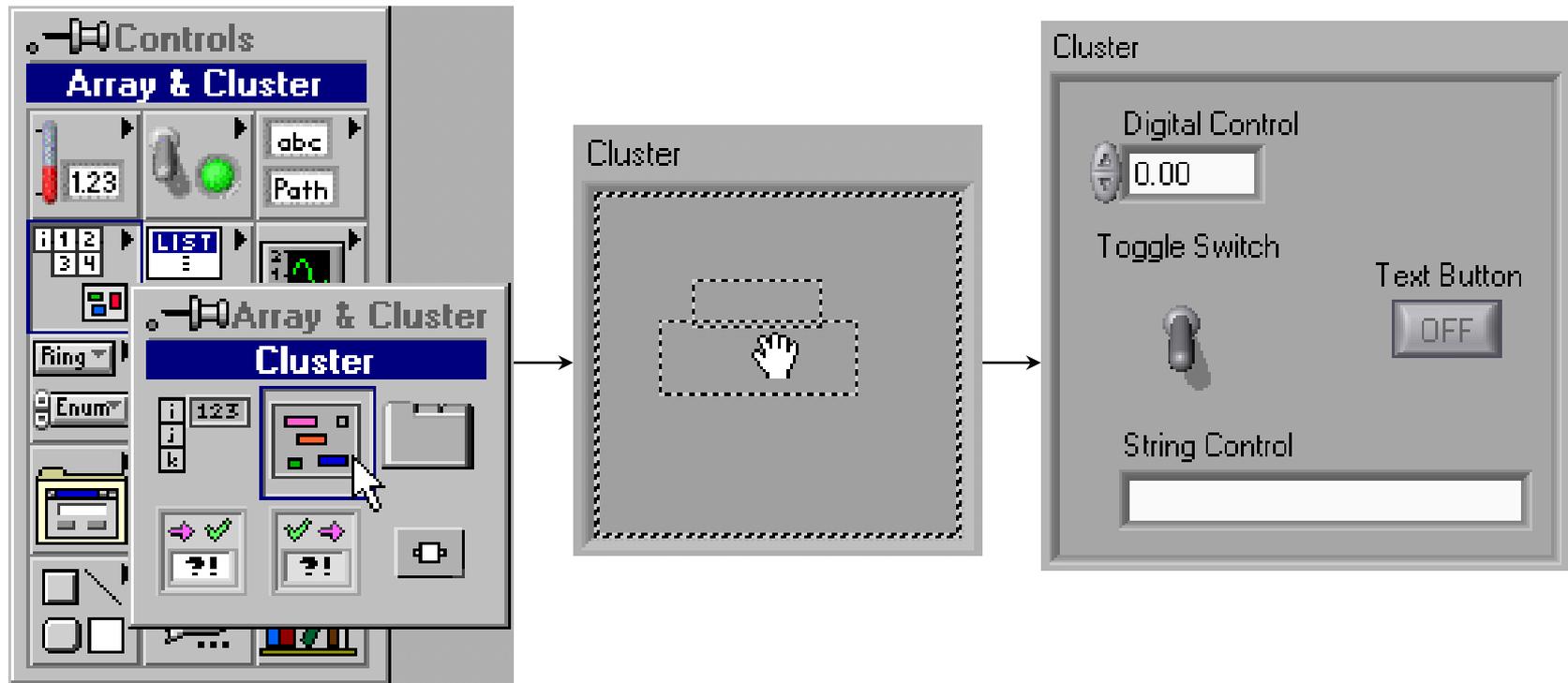
## Les structures de données (Clusters)

- Structure de données qui groupe des éléments de type différent.
- Analogue à *record* en Pascal ou *struct* en C
- Les éléments doivent être soit tous des contrôles ou tous des indicateurs
- Comme des fils regroupés dans un unique câble



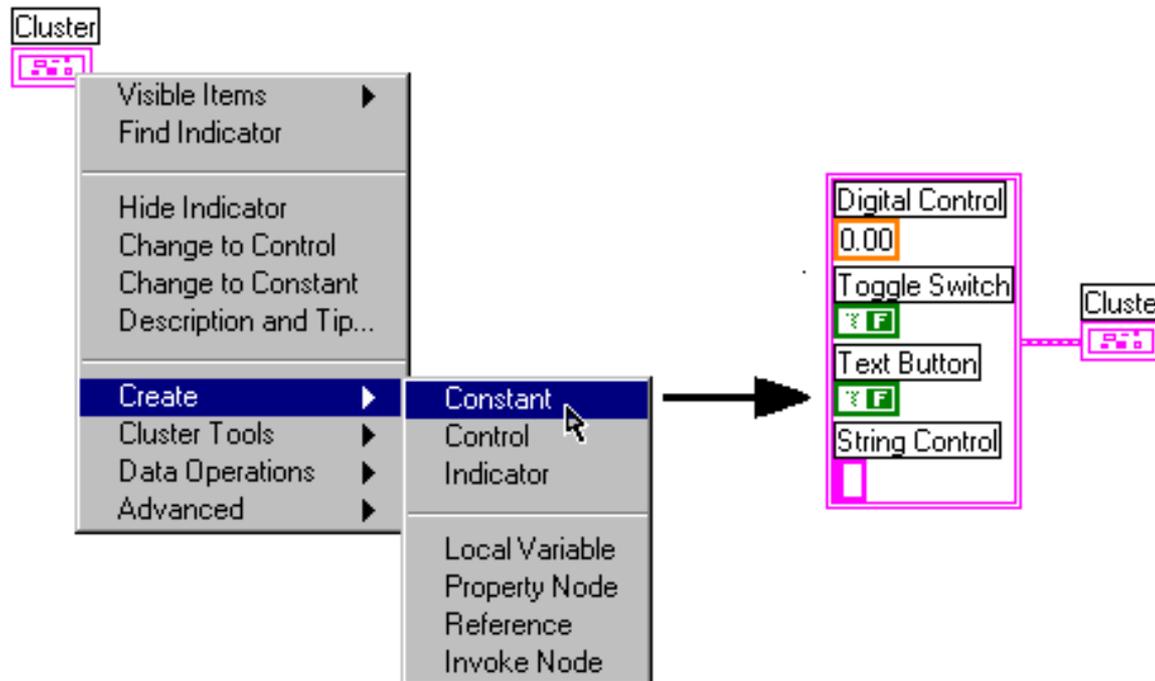
# Les Contrôles et Indicateurs de type cluster

1. Sélectionner le contrôle de type Cluster depuis la palette Array & Cluster
2. Placer les éléments de différents types dans le contrôle cluster pour définir sa structure



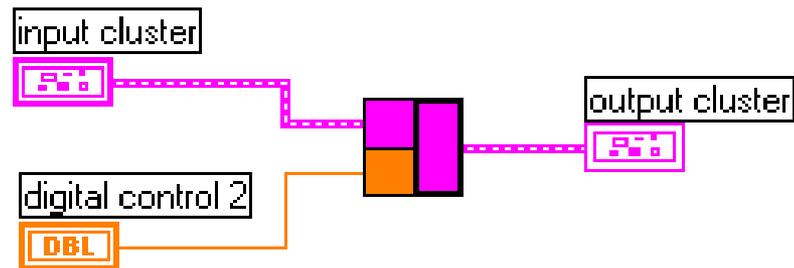
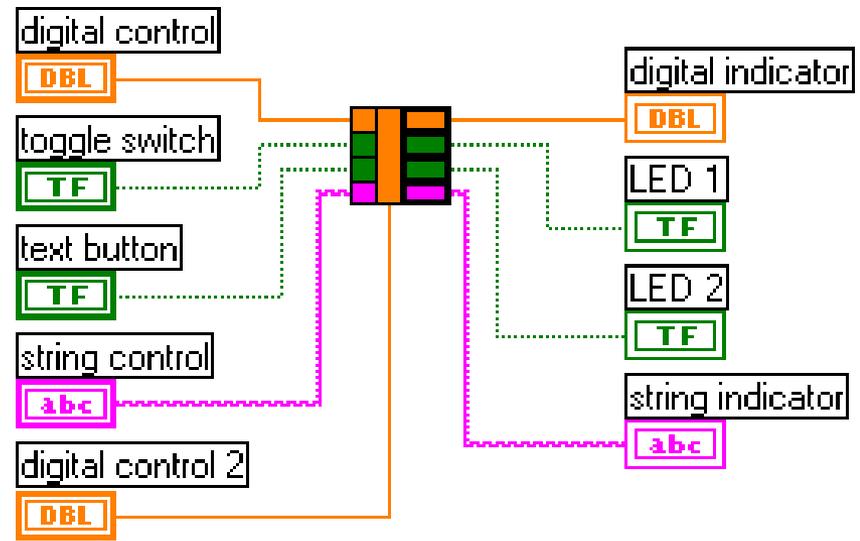
# Les constantes de type Cluster

- Sélectionner une constante de type Cluster depuis la palette de fonction Cluster et placer les différents composants à l'intérieur.
- Click de droite sur un indicateur cluster existant et sélectionner Create » Constant .



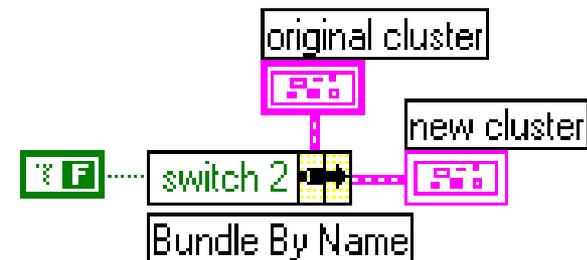
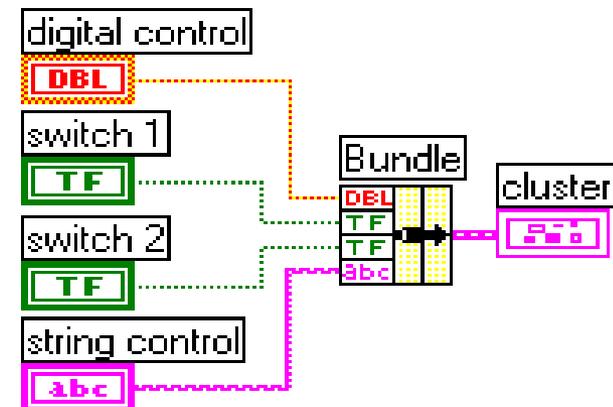
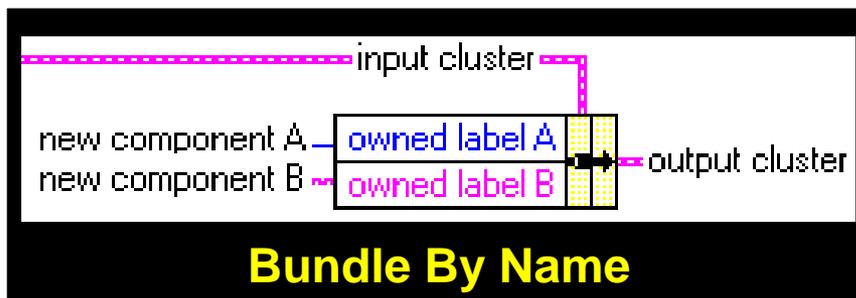
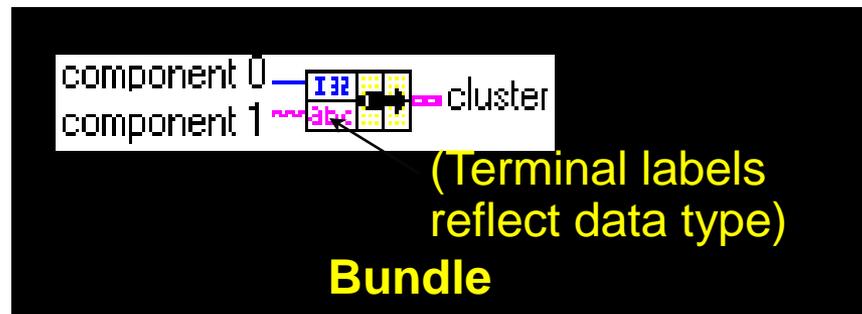
# Utilisation de Clusters pour passer des données à des sous-VI

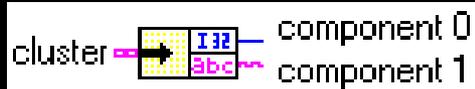
- Permet de passer plusieurs valeurs dans un seul terminal
- Permet de dépasser la limite des 28-terminaux/VI
- Simplifie le câblage



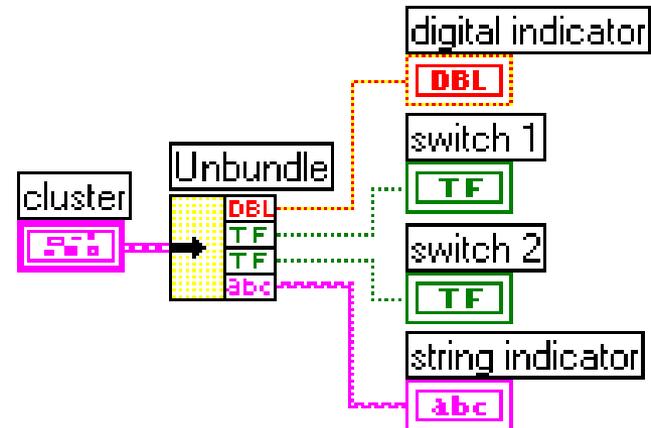
# Les Fonctions sur les Clusters

- Ces fonctions se trouvent dans la sous-palette Cluster de la palette de fonction.
- Les fonctions sont accessibles aussi par un click de droite à partir d'un terminal de type cluster.

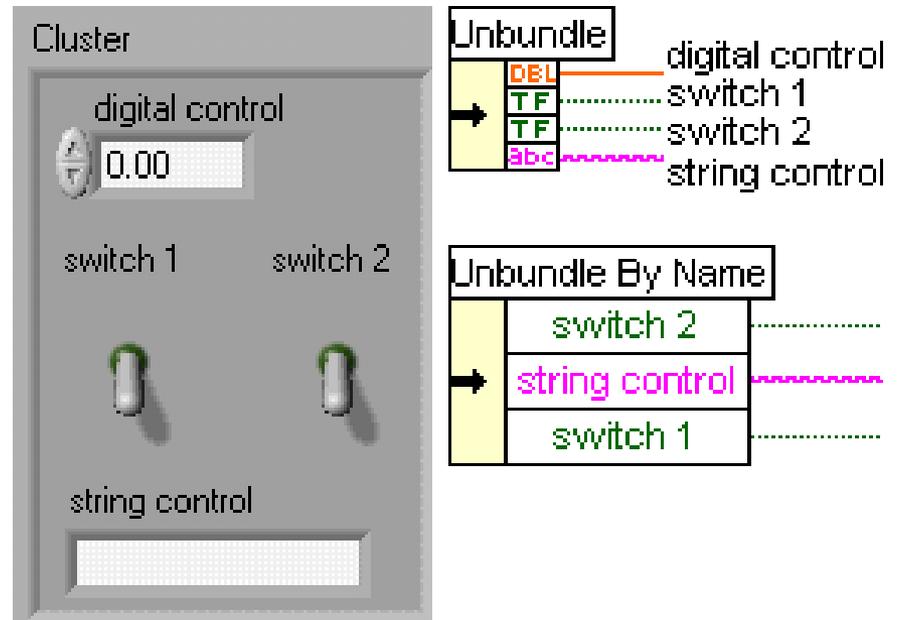




## Unbundle

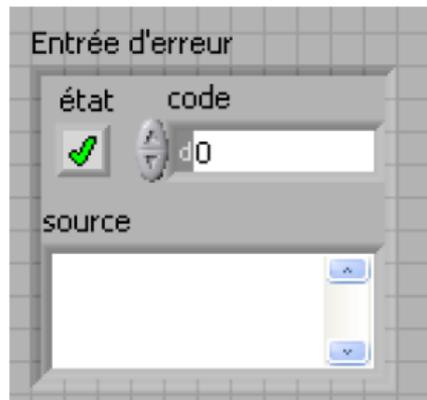


## Unbundle By Name



# Cluster d'erreurs

- Un cluster d'erreur permet la visualisation d'un problème lors de l'exécution d'un VI.
- Un cluster d'erreur contient les informations suivantes :
  - **État** rapporte si une erreur se produit (valeur booléenne).
  - **Code** rapporte le code spécifique de l'erreur (Entier 32 bits).
  - **Source** donne des informations sur l'erreur (Chaîne de caractères).



**Pas d'erreur**

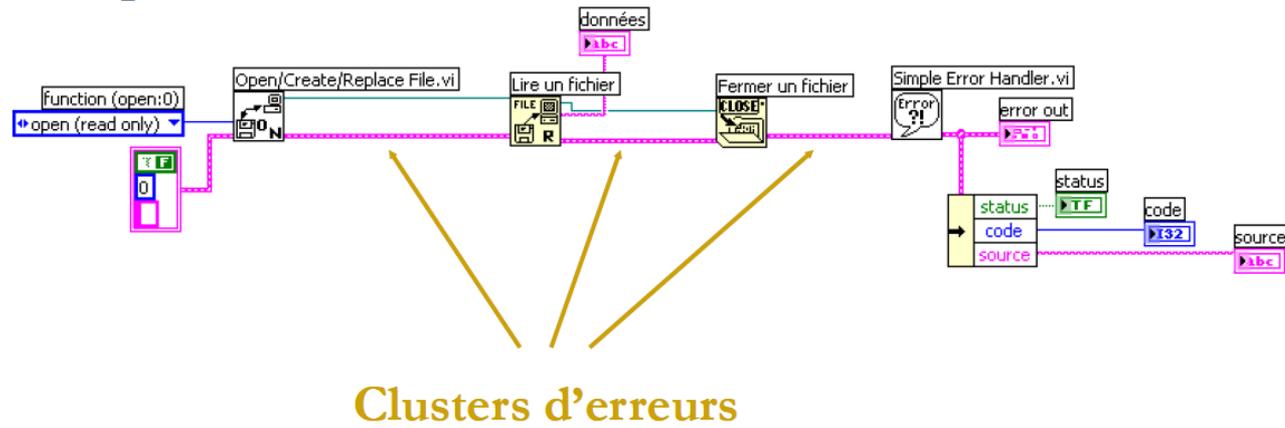


**Erreur**

# Exemple de manipulation des erreurs

- L'information d'erreur est passée d'un sous VI au suivant.
- Si une erreur se produit dans un sous VI, tous les sous VI suivants ne sont pas exécutés de la façon habituelle.
- Gestion d'erreur automatique.

## Exemple :

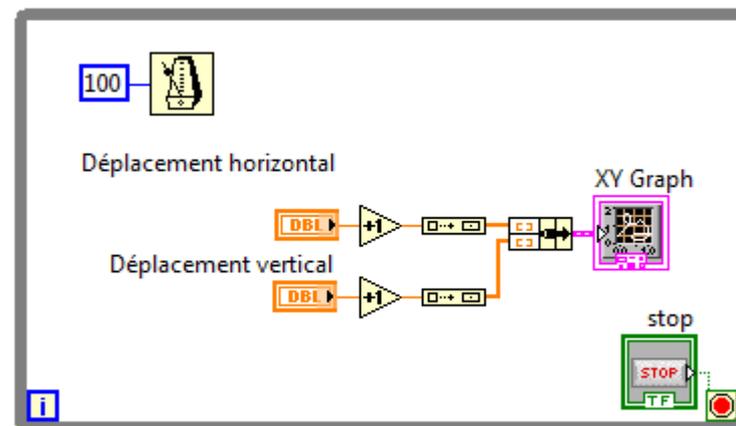
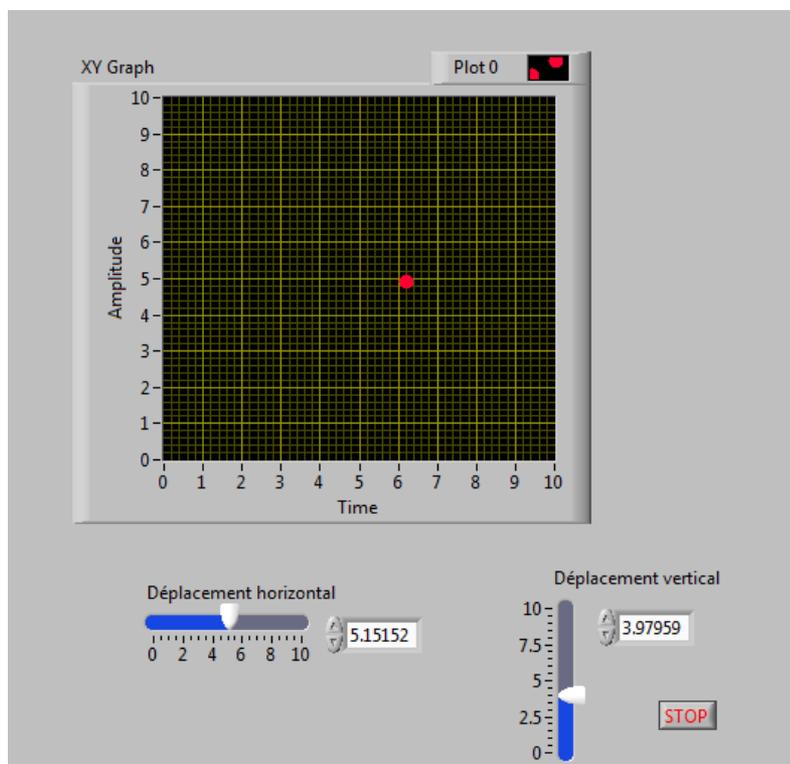


# Résumé du chapitre

- Un tableau est une collection d'éléments de même type: numérique, Booléen, string, etc..
- La création de contrôle/ indicateurs de type Array ou Cluster se fait en deux étapes:
  1. Placer un contrôle array ou cluster depuis la palette .
  2. Placer un contrôle/indicateur dans le contrôle array pour définir le type du tableau /du cluster.
- Les boucles peuvent générer des tableaux à l'aide de l'option auto-indexing.
- Les Clusters regroupent des données de types différents.
- Les fonctions Bundle et Unbundle permettent de créer et de désassembler les clusters.

# Exercices : à vous de jouer !

- Générez un VI qui simule le déplacement d'un pointeur laser sur une photodiode 4 quadrant.







# Chapitre 8

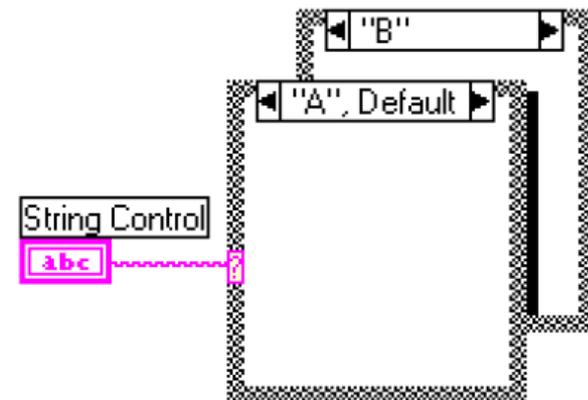
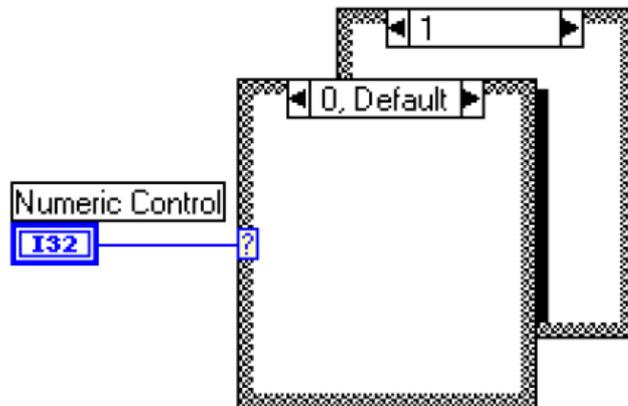
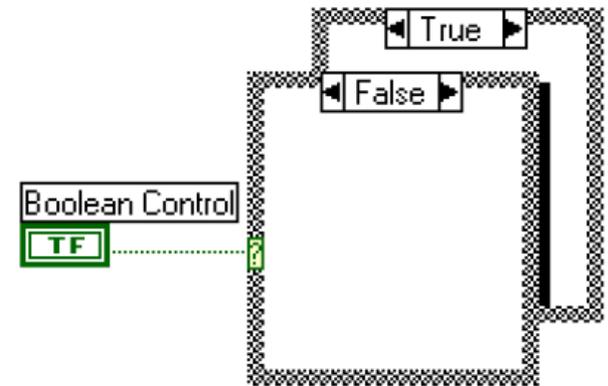
## Séquences

Thèmes :

- A. La fonction de choix Case (rappel)
- B. Les séquences
- C. Les Formula Nodes

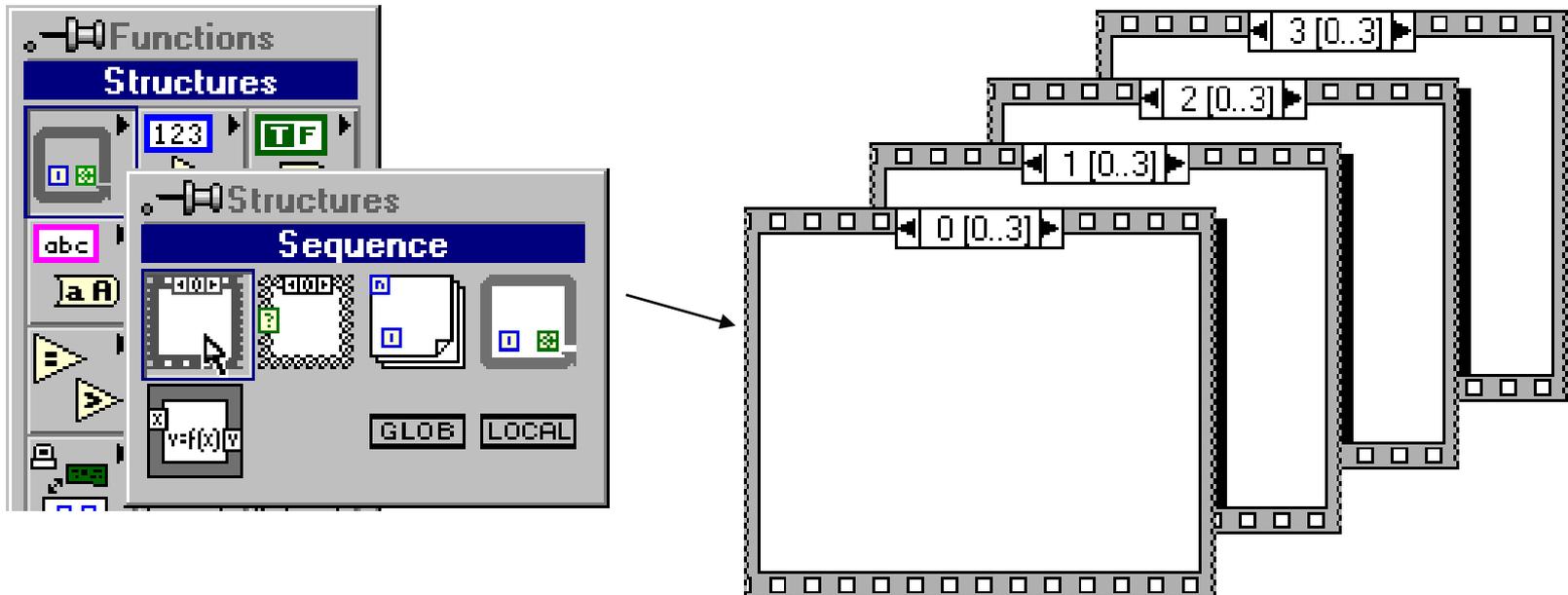
# Case

- **Correspond** à “if” en C, Matlab, etc.
- Se trouve dans la sous-palette de fonction ‘Structures’
- Entoure le programme à effectuer dans le cas traité
- Les cas sont superposés, un seul cas visible à la fois



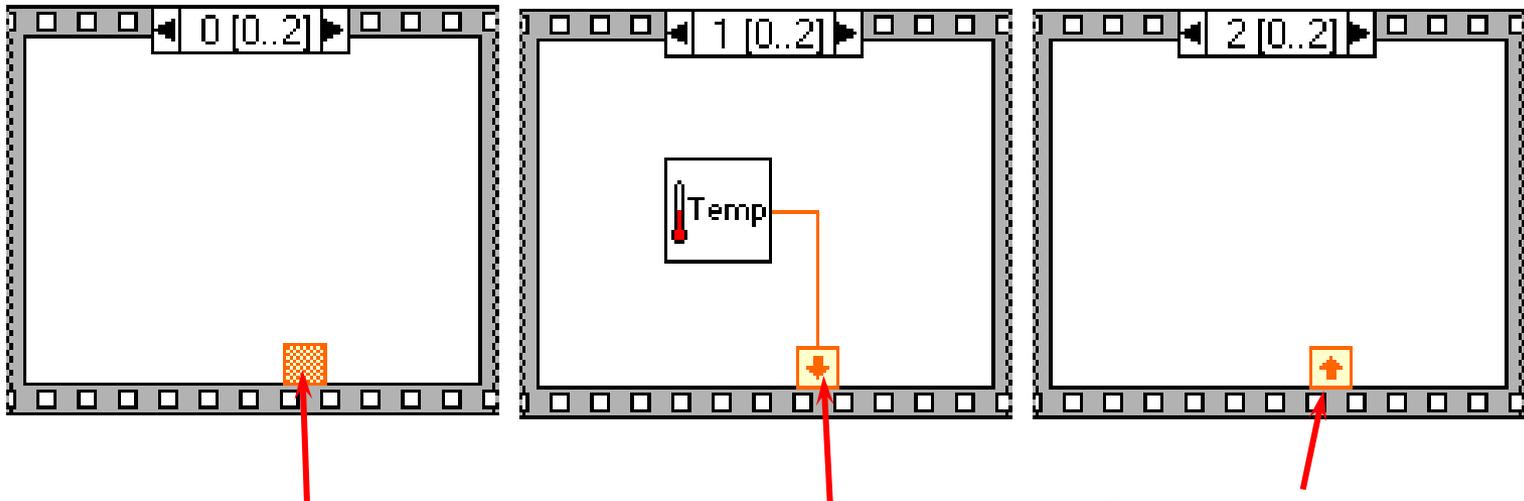
# Séquences

- Se trouve dans la sous-palette de fonctions 'Structures'.
- Permet d'ordonner un programme de manière séquentielle:  
Etape 0 (0..x), ou x est le nombre total d'étapes (frames).  
Une étape doit être complètement finie pour passer à la suivante.
- Les étapes sont superposées, seulement une étape visible à la fois.



# Séquences locales

- Passe des données d'une étape à une future étape.
- Creation par click de droite sur le bord de la séquence.



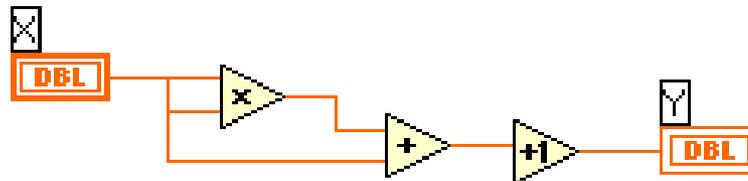
Pas de data  
À disposition

Séquence locale  
Crée dans le  
Frame 1

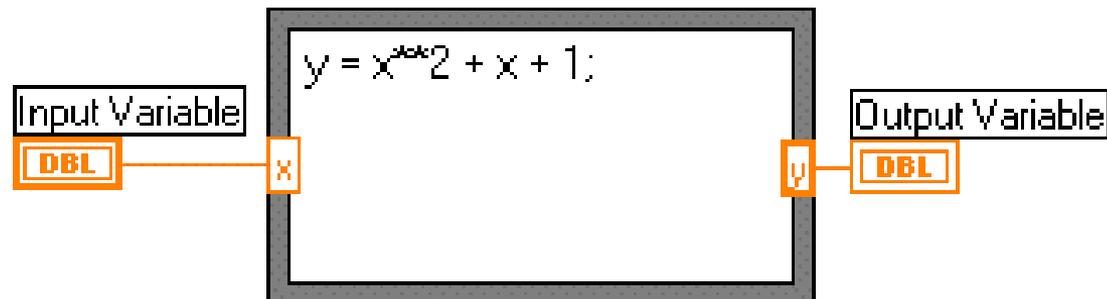
Data  
Disponible depuis  
Le Frame 1

# Formula Node

- Se trouve dans la sous-palette de fonctions 'Structures'.
- Permet de réaliser des équations compliquées.
- Les variables d'entrée/sorties sont créés sur les bords.
- Les noms des variables sont 'case-sensitive'.
- Chaque ligne de code doit se terminer par un pt virgule (;).
- L'aide contextuelle montre les fonctions disponibles.
- Exemple: au lieu de:



- On peut avoir:

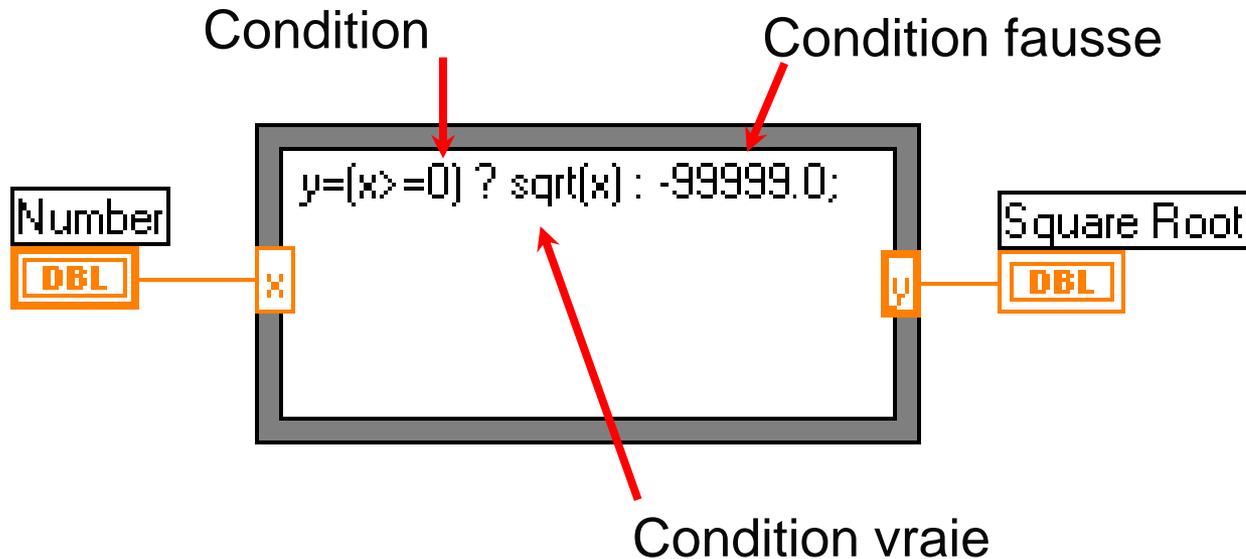


# Exécuter des conditions dans un Formula Node

- Par exemple, pour un code de type Matlab suivant:

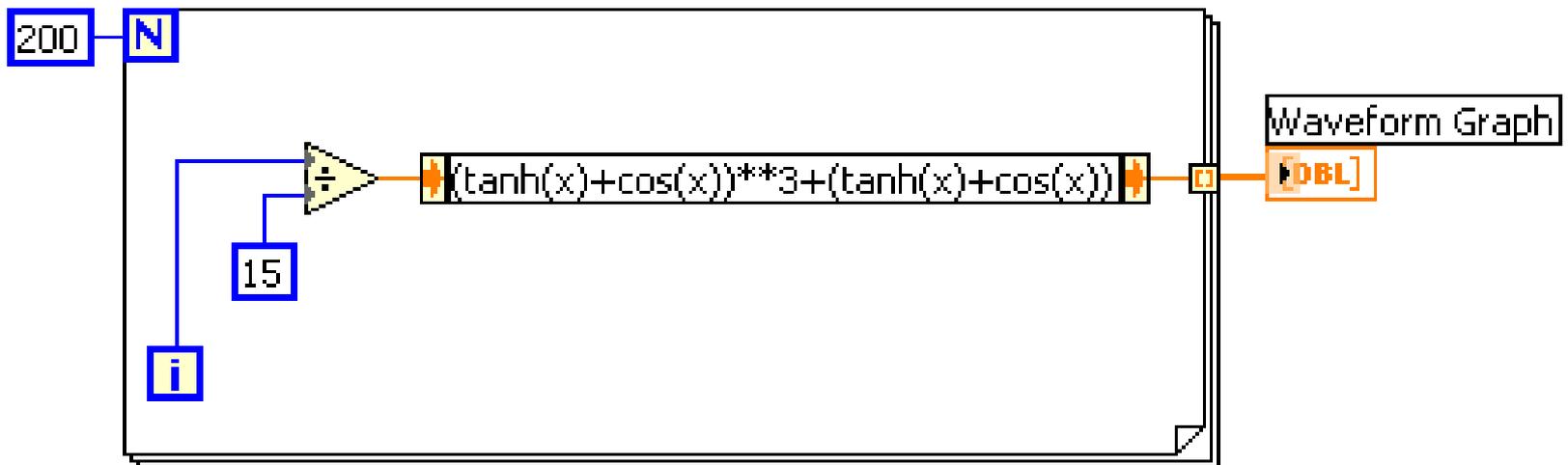
```
if (x >= 0) then
    y = sqrt(x)
else
    y = -99999.0
end if
```

- On peut faire:



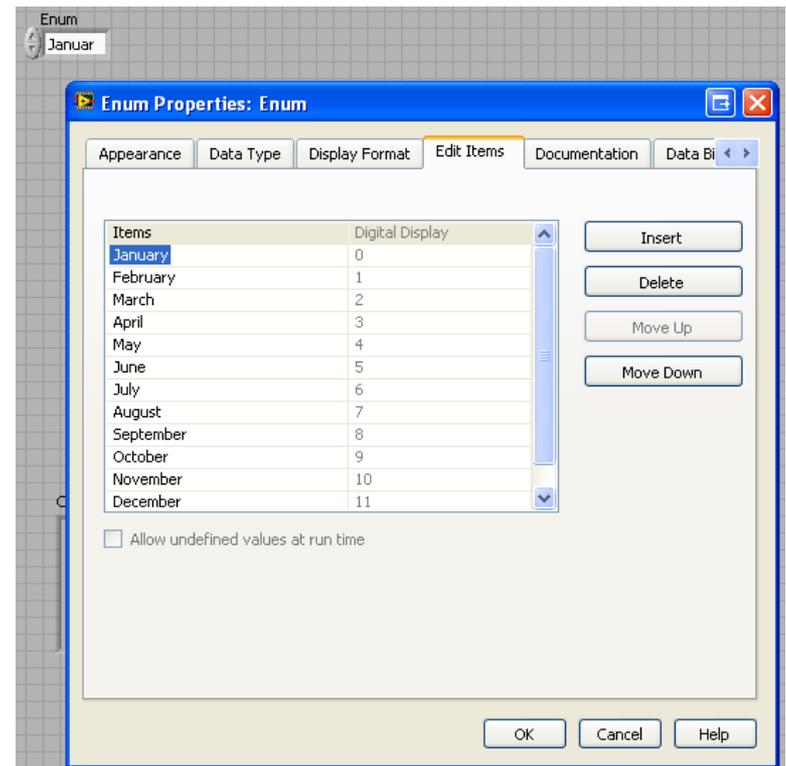
# Expression Nodes

- Utilisation d' Expression Nodes pour résoudre des équations simples mono-variables



# Les Enums

- Combinaison de deux types de data ( string et numérique).
- Les enums sont très pratiques car il est plus simple de manipuler des nombres que du texte dans le block diagram.

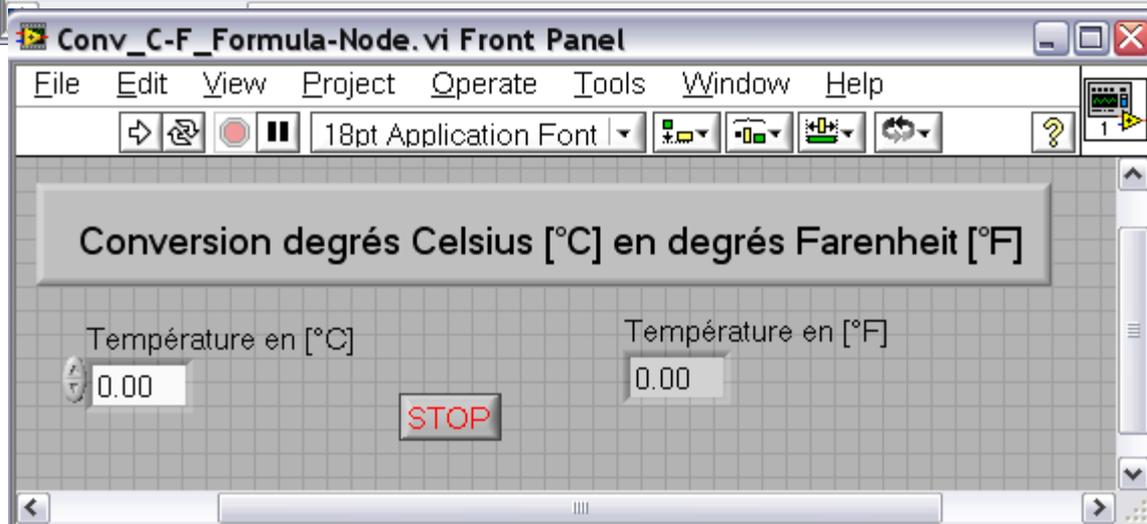
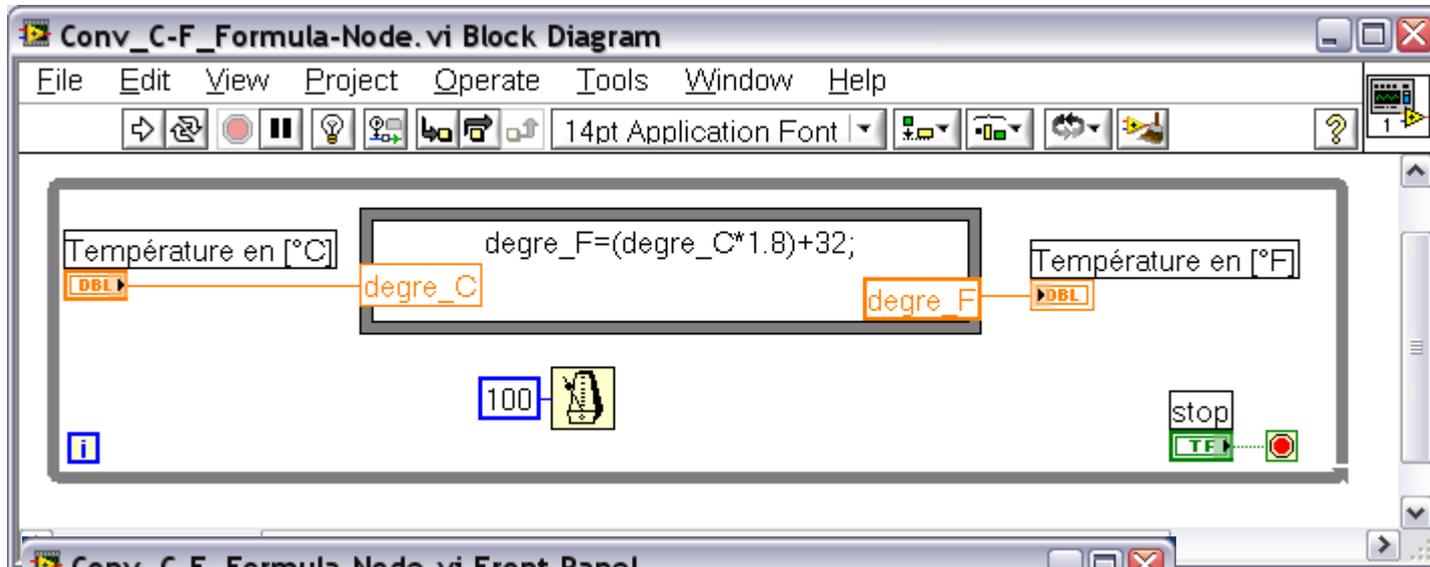


# Résumé du chapitre

- Deux structures pour contrôler l'exécution d'un programme
  - structure 'Case'
  - structure 'Sequence'
- Case structure
  - 'Case' numériques ou booléens – le sélecteur détermine le type
  - Le sous-diagramme est placé dans le 'Case' correspondant
  - La sortie de chaque cas possible doit être définie
- Une Séquence exécute des sous-diagrammes séquentiellement
- Une 'Sequence locale' passe les données entre les étapes
  - Création depuis le bord de la structure avec le click de droite
  - Les données sont disponibles pour les étapes futures
- Un 'Formula Node' permet d'entrer directement une expression ou une équation dans le diagramme.

# Exercices : à vous de jouer !

- Reprenez le programme de conversion de température de degrés Celsius en degrés Farenheit, mais réalisez-le à l'aide d'un Formula Node.









# Chapitre 9

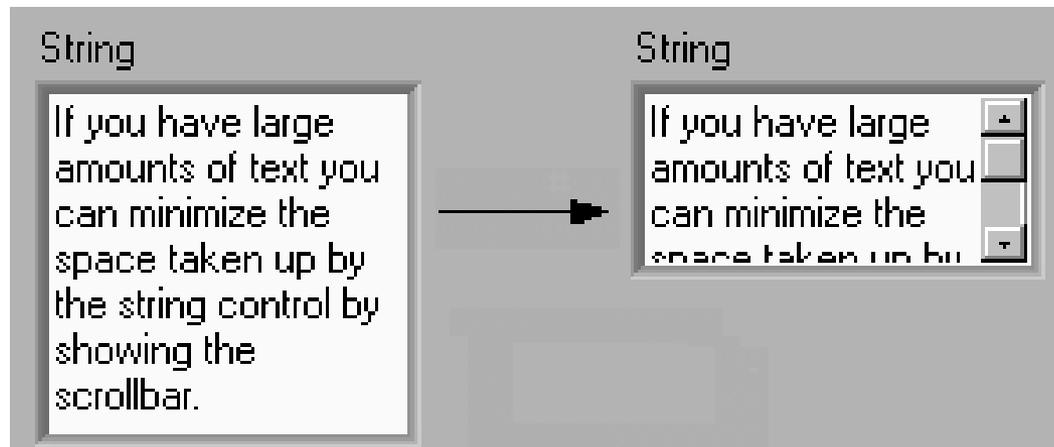
## Chaînes de caractères et opérations sur fichiers

### Thèmes :

- Comment créer un contrôle/indicateur de type chaîne de caractères.
- Comment faire une lecture/écriture sur un fichier texte.

# Les chaînes de caractères (string)

- Une chaîne de caractères est une suite de caractères visibles ou invisibles (ASCII).
- Beaucoup d'applications – Affichage de messages, contrôle d'instruments , I/O sur des fichiers.
- Les contrôles / indicateurs de type String se trouvent dans la sous-palette : Controls » String .



# Modes d'affichage d'une chaîne de caractères

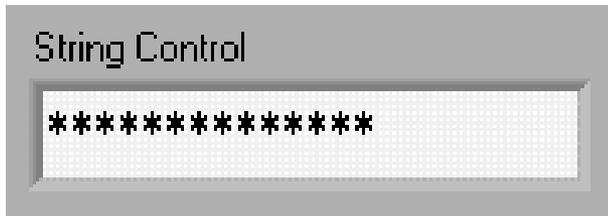
- Affichage Normal



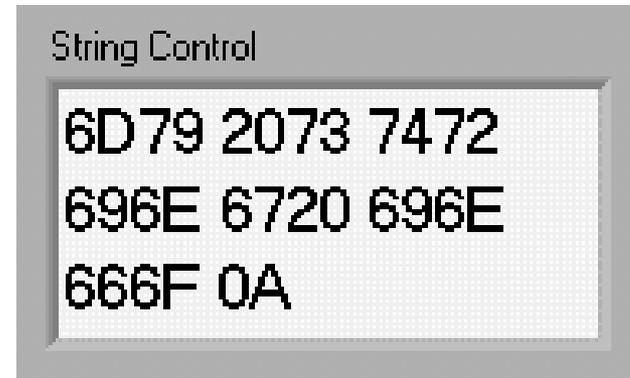
- Affichage \code



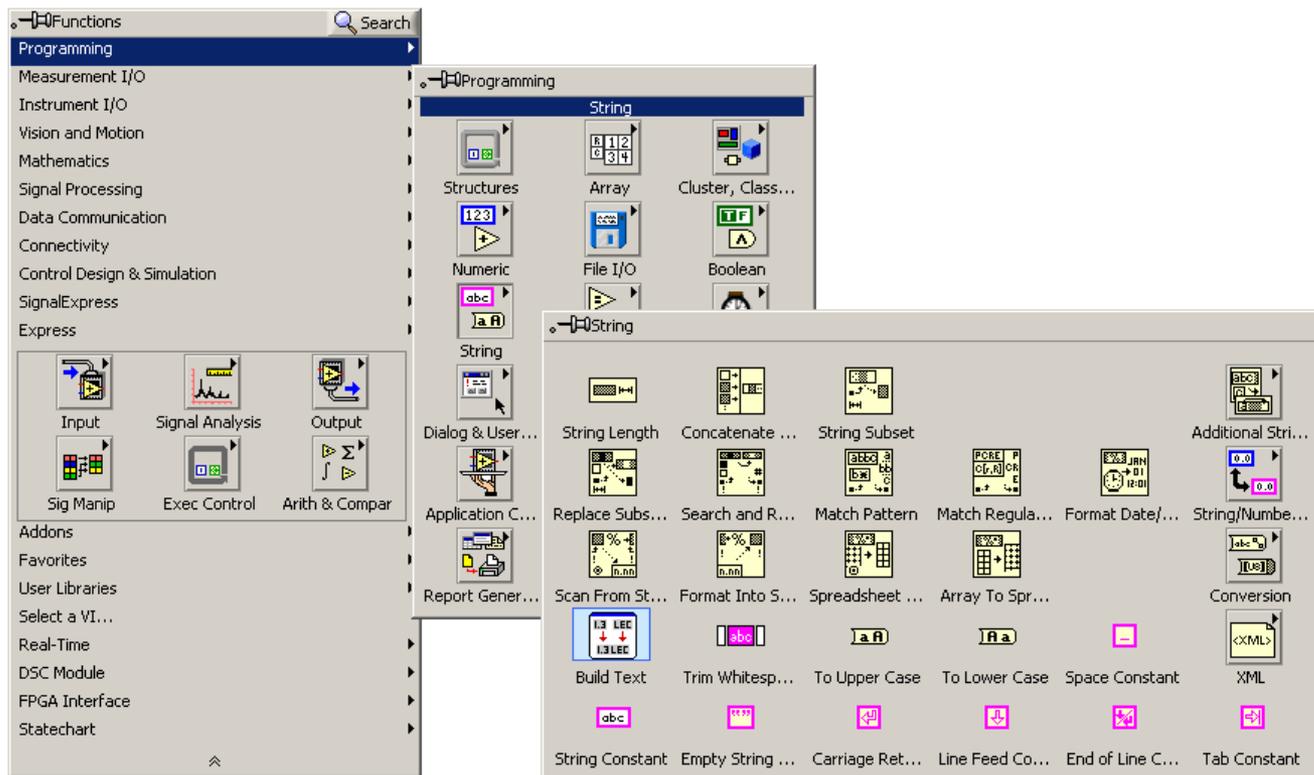
- Affichage 'Password'



- Affichage Hexadécimal



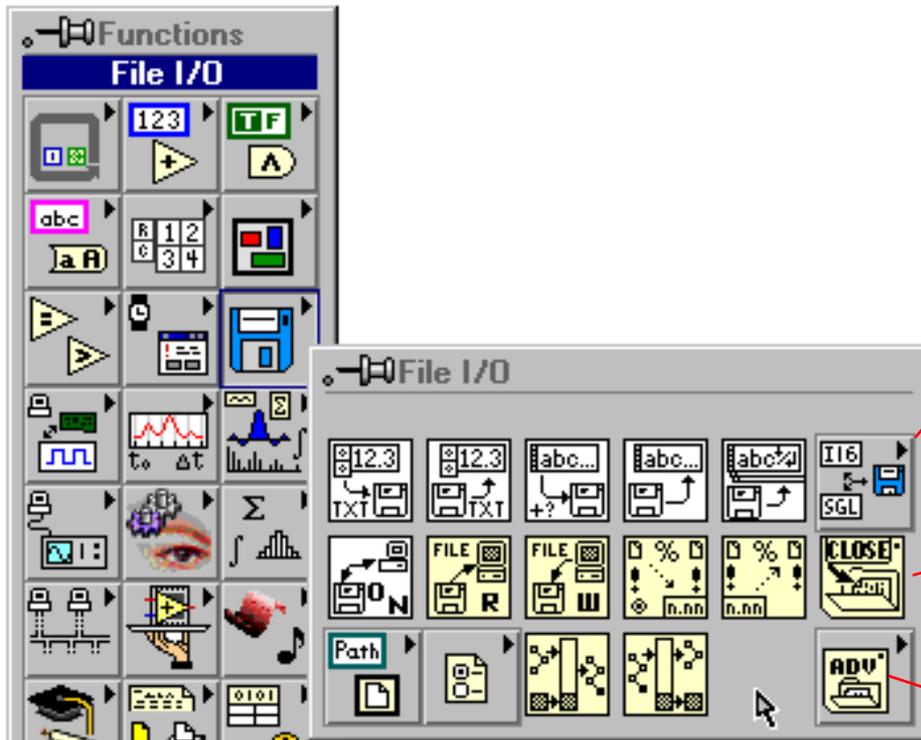
# Opérations sur chaînes de caractères



# Opérations sur fichiers

Trois niveaux de hiérarchie :

- *High-level file VIs*
- *Intermediate file I/O VIs*
- *Advanced file I/O functions*



High Level File VIs

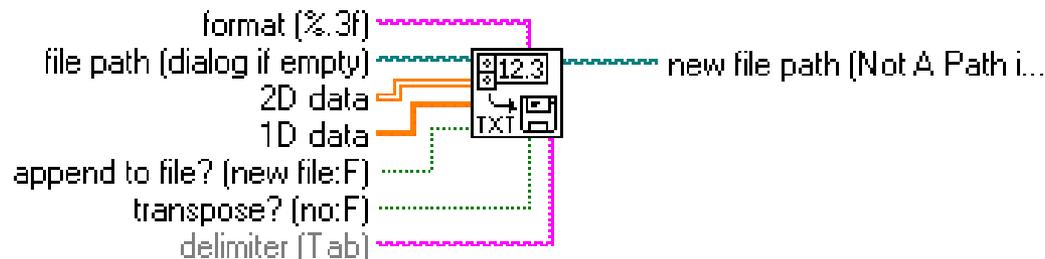
Intermediate File VIs and Functions

Advanced File Functions

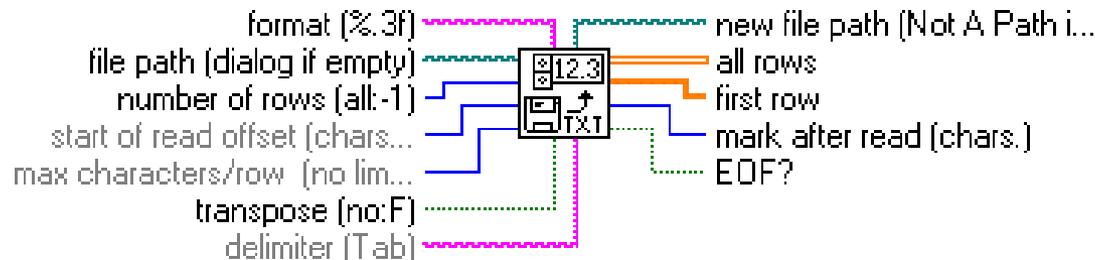
# Vis de haut-niveau pour opérations sur fichiers



- Write to Spreadsheet File (tableur)
- Read from Spreadsheet File (tableur)
- Write Characters to File (texte)
- Read Characters from File (texte)
- Read Lines from File



**Write To Spreadsheet File.vi**



**Read From Spreadsheet File.vi**

# VIs intermédiaires pour I/O fichiers



- **Open/Create/Replace file** – ouvre, crée, ou remplace un fichier



- **Read File** – Lire un certain nombre de caractères (bytes) depuis un fichier



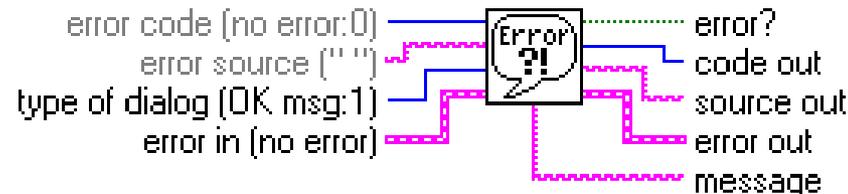
- **Write File** – écrire les données dans un fichier



- **Close File** – fermer l'accès au fichier

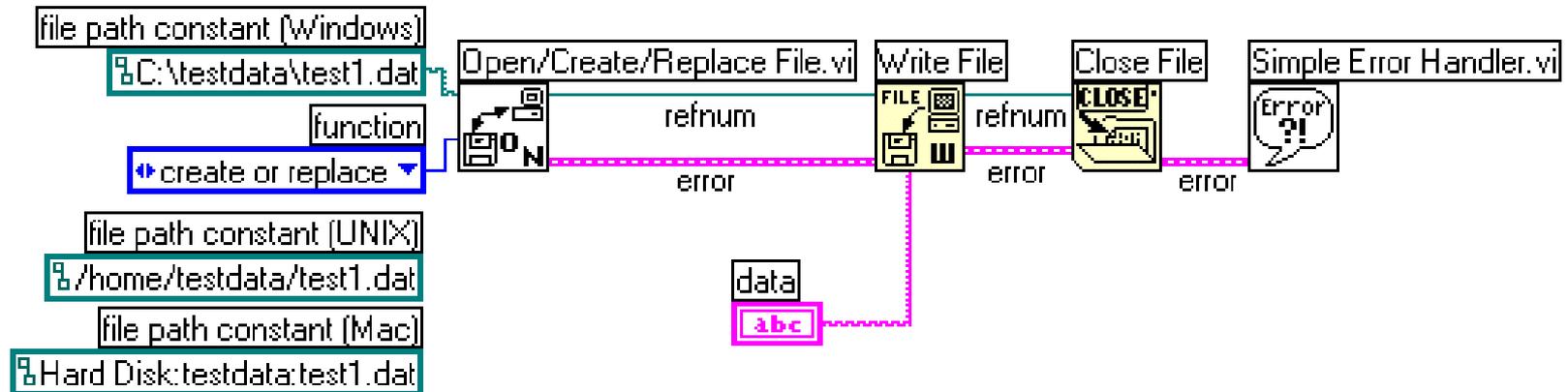
- Gestion des erreurs dans les fonctions fichier:  
se trouve dans la sous-palette *'Time & Dialog'*

– Affiche un message d'erreur le cas échéant



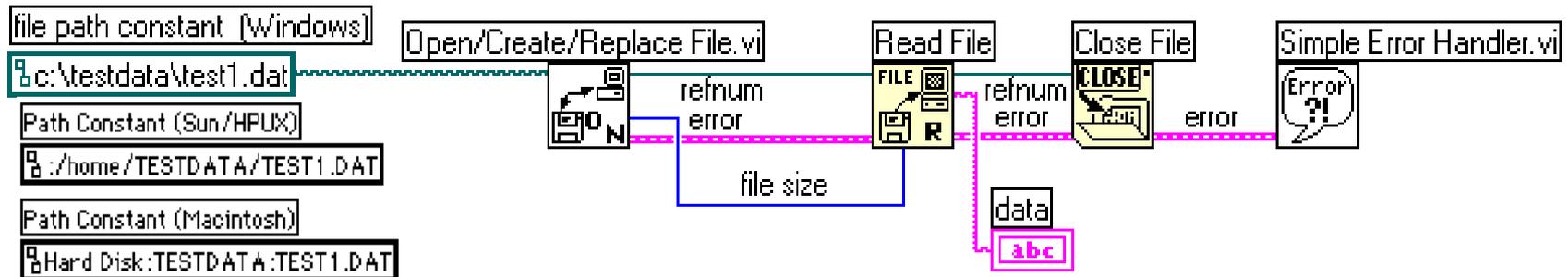
**Simple Error Handler.vi**

# Sauver des données dans un fichier



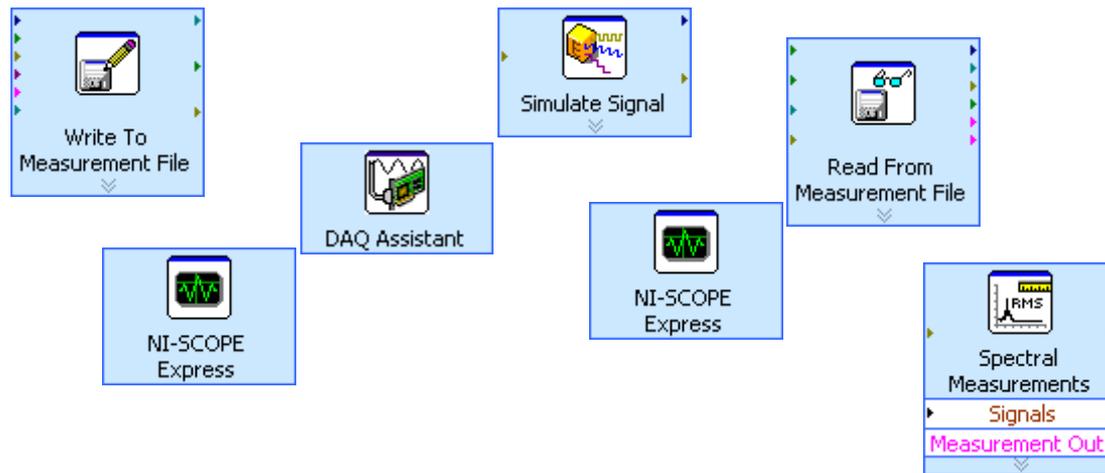
- **Open/Create/Replace**: ouvre le fichier existant TEST1.DAT et génère un cluster `refnum` et un cluster `error` pour les blocs suivants
- **Write File** : écrit les datas dans le fichier
- **Close File** : ferme l'accès au fichier
- **Simple Error Handler** : affiche un message en cas d'erreur

# Lire des données depuis un fichier



- **Open/Create/Replace** : ouvrir le fichier
- **Read File** : Lit le nombre de caractères spécifiés (bytes) dans le fichier
- **Close File** : ferme le fichier
- **Simple Error Handler** : affiche un message en cas d'erreur

# Express VIs = Configurer à la place de programmer



- Possibilité de générer automatiquement le code LabVIEW équivalent

# Express VIs pour sauvegarde dans des fichiers de mesures

The image displays a LabVIEW Express VI titled "Test\_Data2.vi". The front panel includes a "Simulate Signal" subVI with a "Sine" control, a "Write To Measurement File" subVI with a "Signals" control, a numeric control set to "100", and a "stop" button. A red arrow points from the "Write To Measurement File" subVI to its configuration dialog box.

The "Configure Write To Measurement File [Write To Measurement File]" dialog box is shown with the following settings:

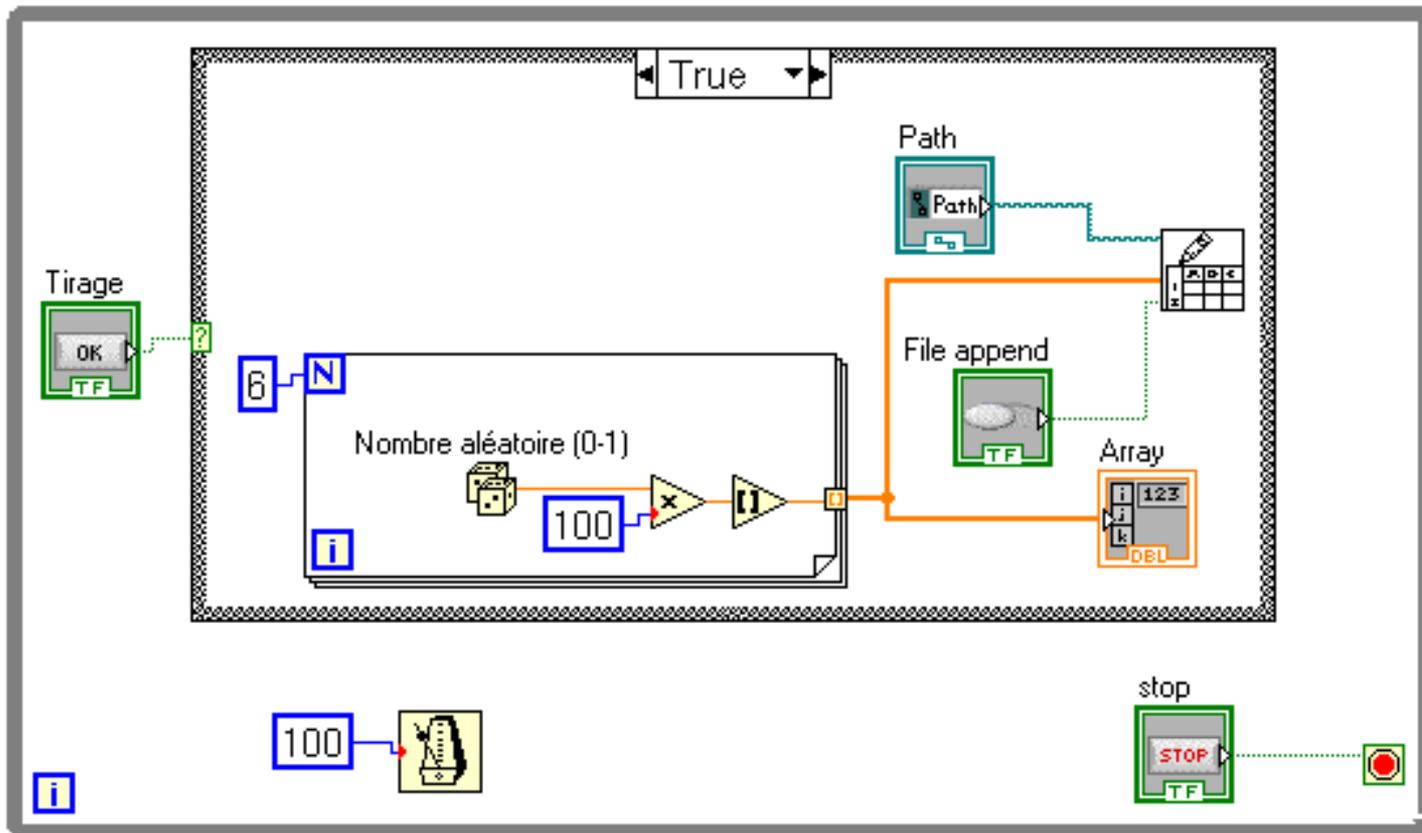
- File Name:** C:\Documents and Settings\Administrator\Desktop\Test\_Data.lvm
- File Format:**  Text (LVM),  Binary (TDM5),  Binary with XML Header (TDM),  Lock file for faster access
- Action:**  Save to one file,  Ask user to choose file,  Ask only once,  Ask each iteration
- If a file already exists:**  Rename existing file,  Use next available file name,  Append to file,  Overwrite file
- Segment Headers:**  One header per segment,  One header only,  No headers
- X Value Columns:**  One column per channel,  One column only,  Empty time column
- Delimiter:**  Tab,  Comma
- File Description:** (Empty text box)

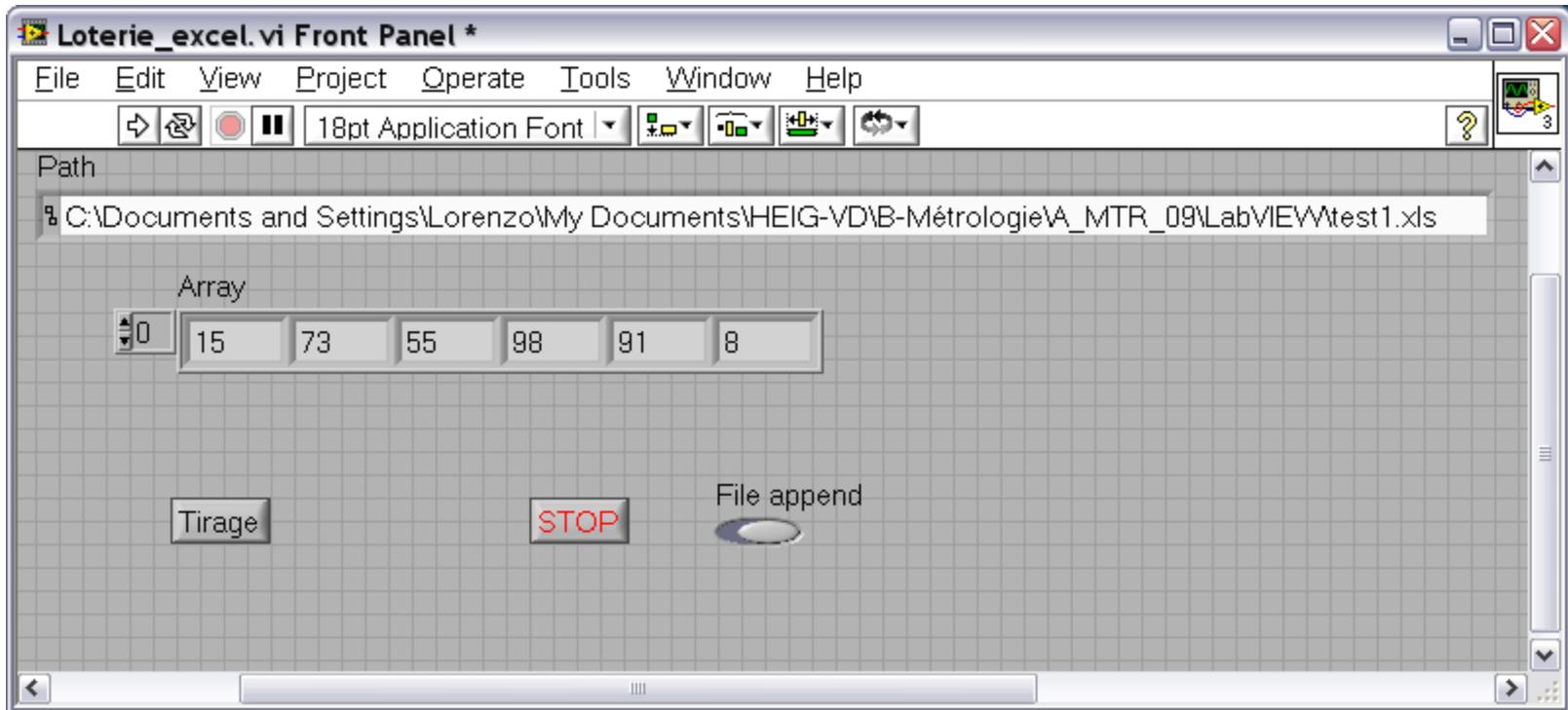
Below the dialog box, a Microsoft Excel spreadsheet is open, displaying a table of data with two columns: time (0 to 0.47) and signal amplitude (0 to -1.5). A graph window shows a sine wave plot of the data.



# Exercices : à vous de jouer !

- Reprendre le programme de loterie (slide 53 de 1<sup>ère</sup> partie) et écrire les valeurs des tirages dans un fichier Excel.  
Le block « Write To Spreadsheet File VI » recevra les six valeurs de chaque tirage. Il devra être renseigné avec le *path* du fichier ainsi que si on veut écrire en mode “append” ou non.





# Travail personnel

- Exécutez le tutoriel du chapitre 3 (page 3-1 à 3-18) de l'*Initiation à LabVIEW* (censé prendre 40 minutes).

# Travail personnel: (formula node, shift register et boucles)

## Luminance spectrale du corps noir

- Un corps noir est un radiateur thermique idéal qui n'est pas affecté par les radiations incidentes quelles que soient leur longueur d'onde, leur direction et leur état de polarisation

### Loi de Planck

$$L_{e,\lambda}(\lambda, T) = \frac{2hc^2}{\lambda^5 \left( e^{\frac{hc}{\lambda kT}} - 1 \right)}$$

### Vitesse de la lumière dans le vide

$$c = 299\,792\,458 \text{ m.s}^{-1}$$

### Constante de Planck

$$h = 6,626\,069\,3 \times 10^{-34} \text{ J.s}$$

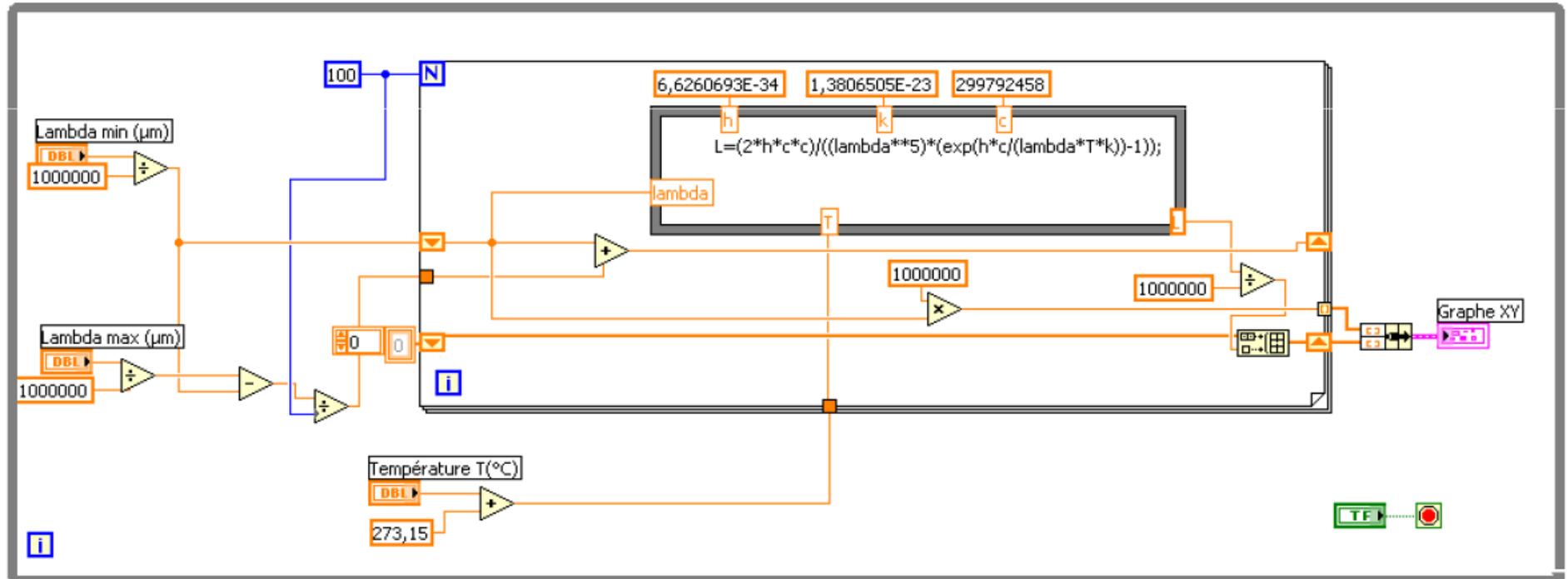
### Constante de Boltzmann

$$k = 1,380\,650\,5 \times 10^{-23} \text{ J.K}^{-1}$$

**Générez un VI qui permet de tracer la luminance spectrale du corps noir pour une température donnée. Exemple: le soleil peut être assimilé à un corps noir avec une température de surface  $\approx 5800 \text{ K}$ .**



# Solution possible



N. B.

Le **corps noir** est appelé ainsi parce que son émission lumineuse (qui est bien présente !) dépend uniquement de la température, sans aucun effet de réflexion, en analogie à un *vrai* corps noir qui absorbe toute radiation incidente.