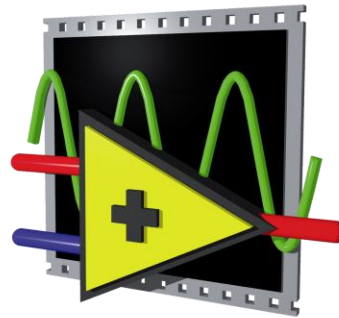
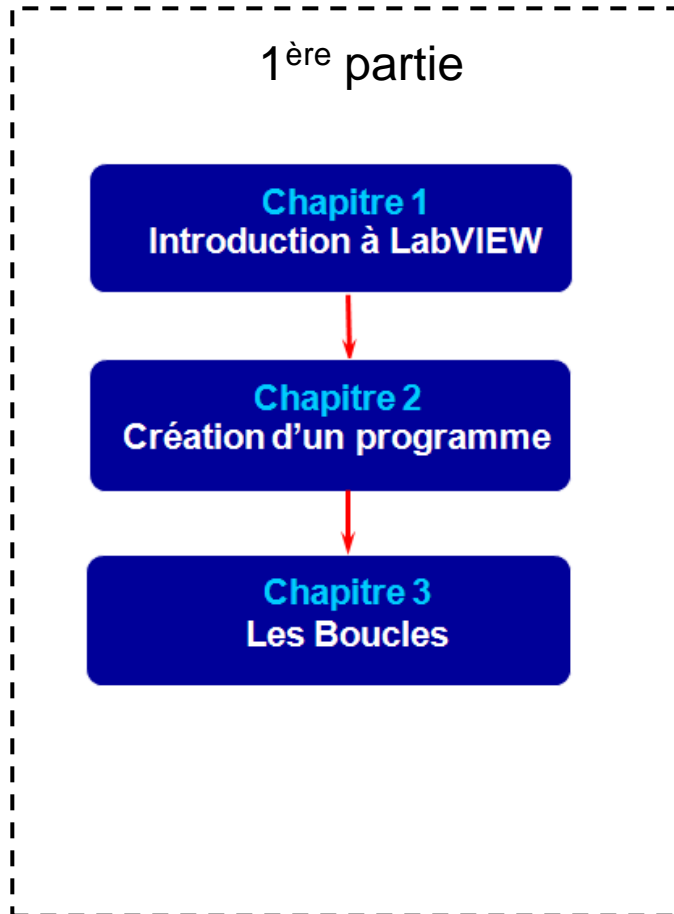


Introduction à LabVIEW – 2^{ème} partie



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

Introduction à LabVIEW – 2^{ème} partie





Chapitre 4

Les tableaux

Thèmes :

- A. A propos de tableaux
- B. Comment générer un tableau avec une boucle
- C. Fonctions de base sur des tableaux

Les tableaux

- Collection de données du même type
- Une ou plusieurs dimensions
- Les éléments sont accessibles par leur index
- Le premier élément se trouve à l'index 0

index 0 1 2 3 4 5 6 7 8 9

Tableau 1D à 10 éléments

1.2	3.2	8.2	8.0	4.8	5.1	6.0	1.0	2.5	1.7
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

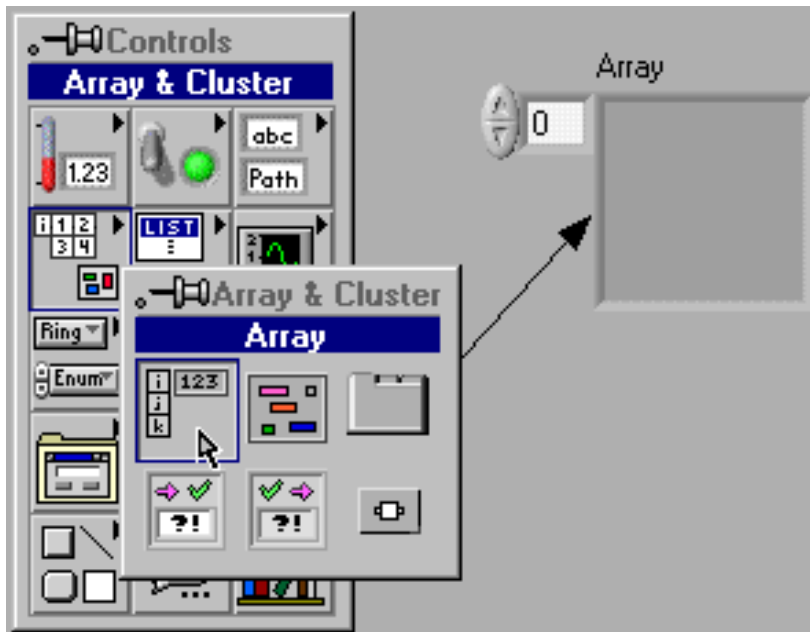
Tableau 2D

0							
1							
2							
3							
4							

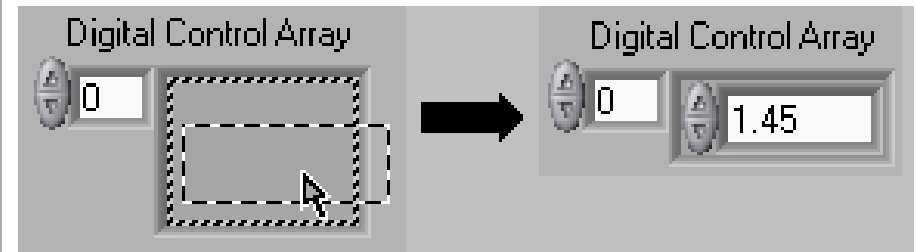
5 lignes par 7 colonnes
Tableau de 35 éléments

Les contrôles et indicateurs de tableaux

1. Selection du contrôle Tableau (Array)



2. Placer un contrôle dans le tableau pour définir le type de données du tableau (ici numérique)

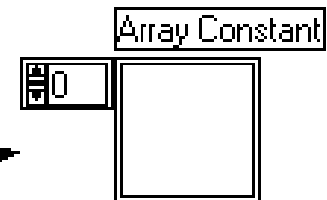
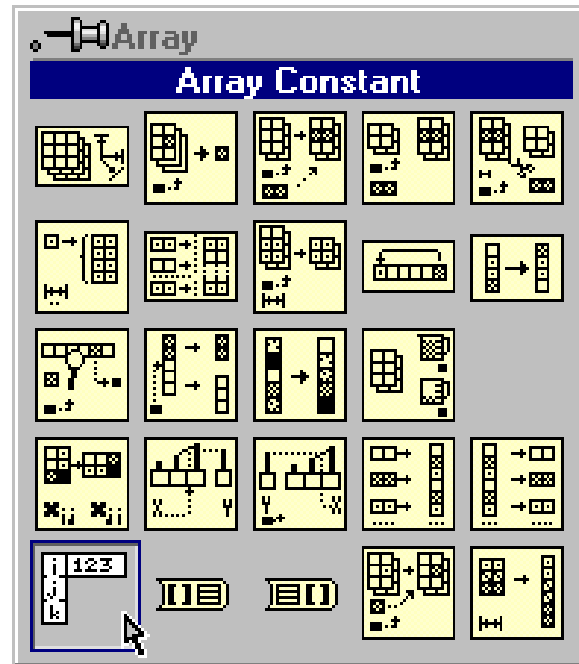


Redimensionner le contrôle pour ajouter une dimension

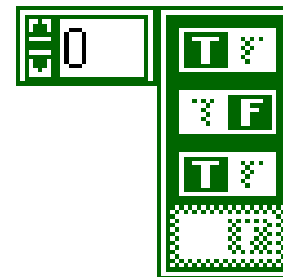


Creation de constantes tableaux

1. Choisir une constante de type tableau dans la palette de fonctions Array



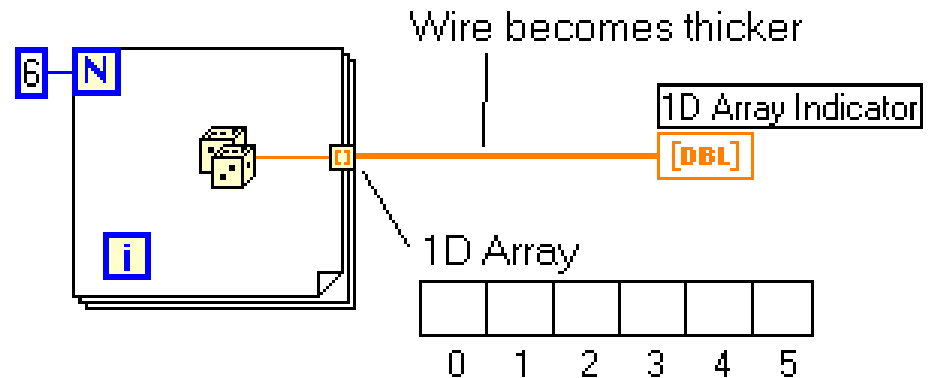
2. Placer une constante dans la constante tableau pour définir le type. (Ici, constante Booléene)



Création et utilisation de tableaux

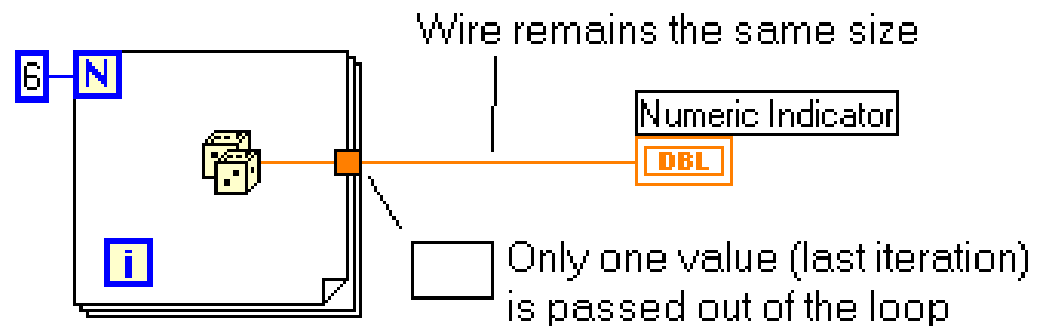
- *Sur une boucle, l'option 'Auto-Indexing' permet de construire automatiquement un tableau.*

Auto-Indexing Enabled (Default of For Loops)

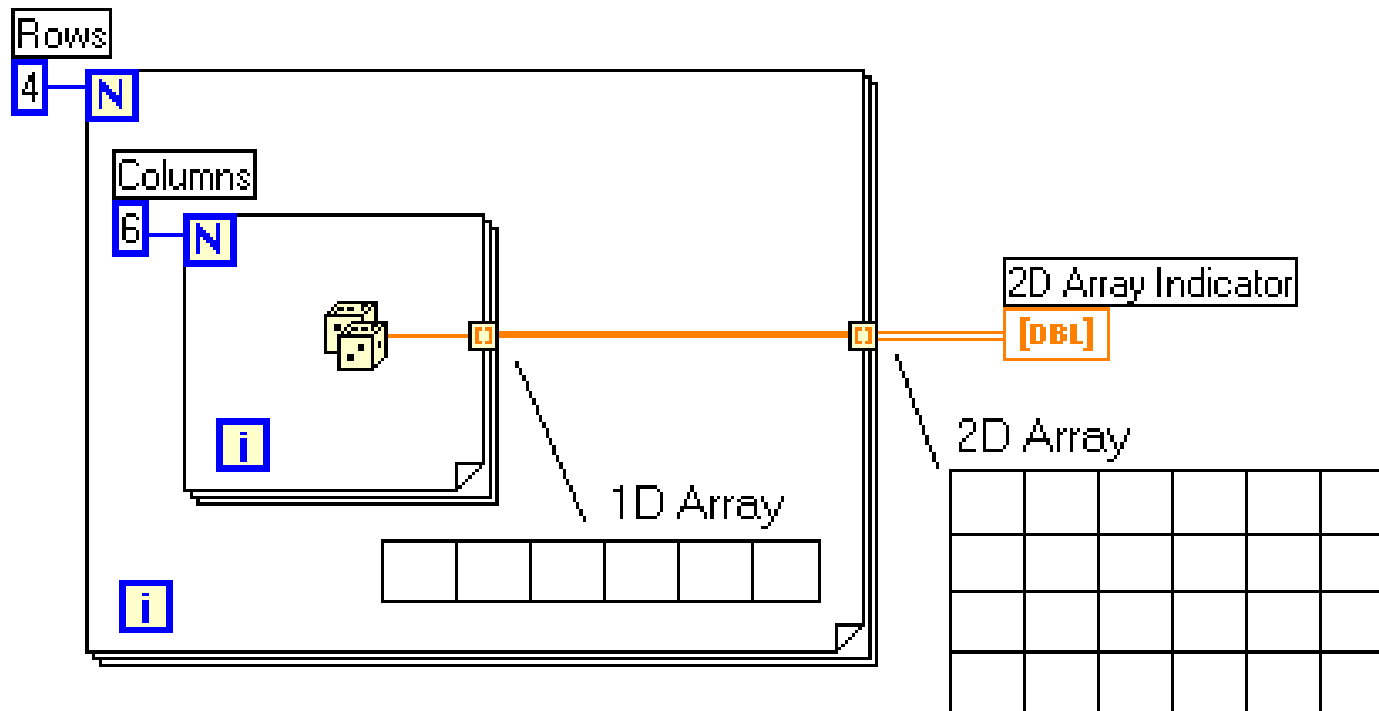


- Boucle For: auto indexing par default; Boucle While pas par défaut

Auto-Indexing Disabled (Default of While Loops)

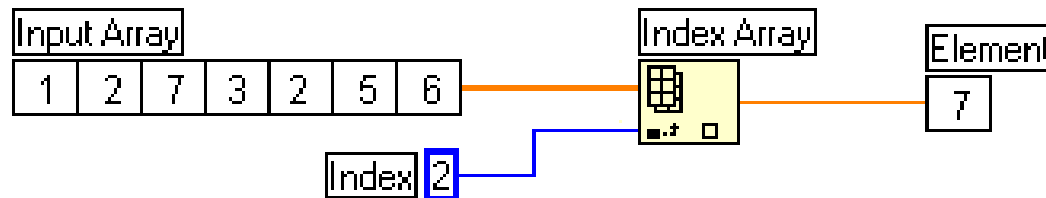


Création de tableaux 2D

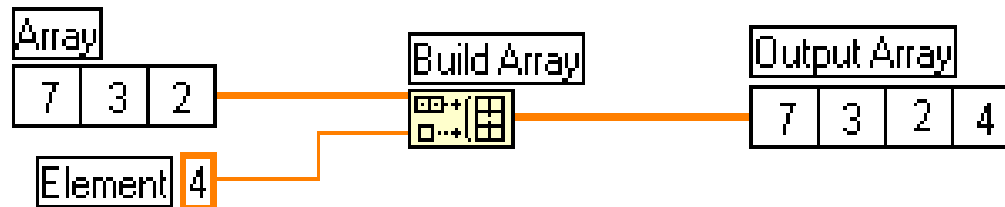


- La boucle interne crée les colonnes
- La boucle externe stocke les colonnes dans les lignes

Accès aux éléments du tableau



Extraction d'un élément



Ajouter un élément

Chapitre 5

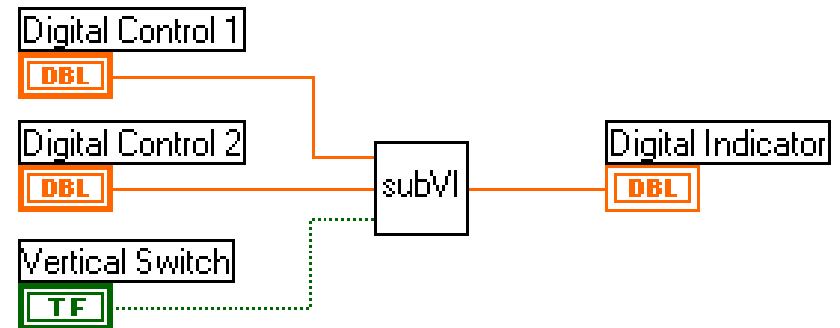
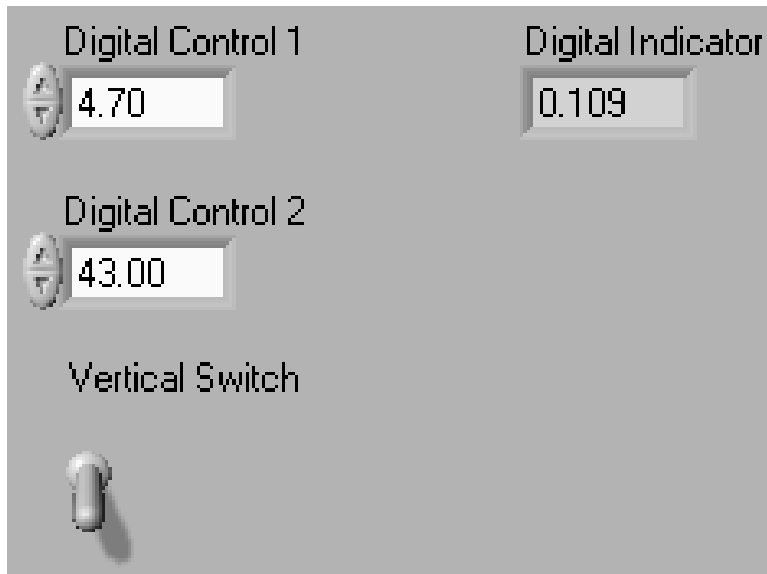
Création d'un sous-VI

Thèmes :

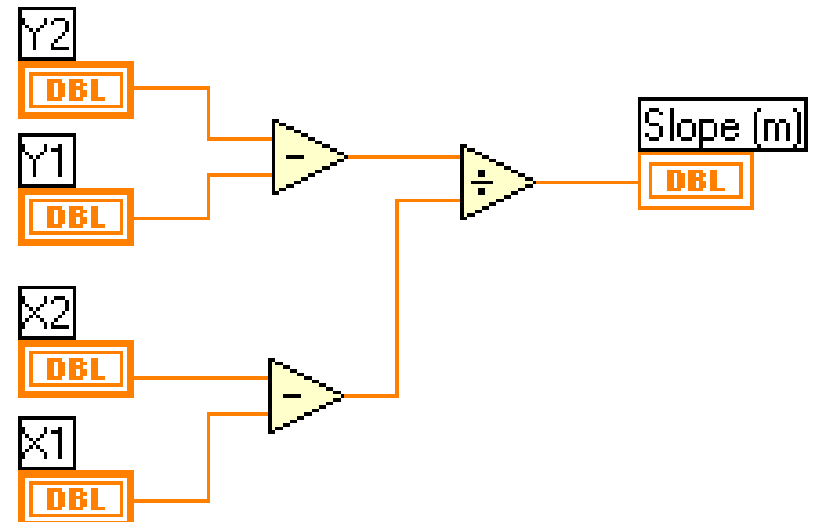
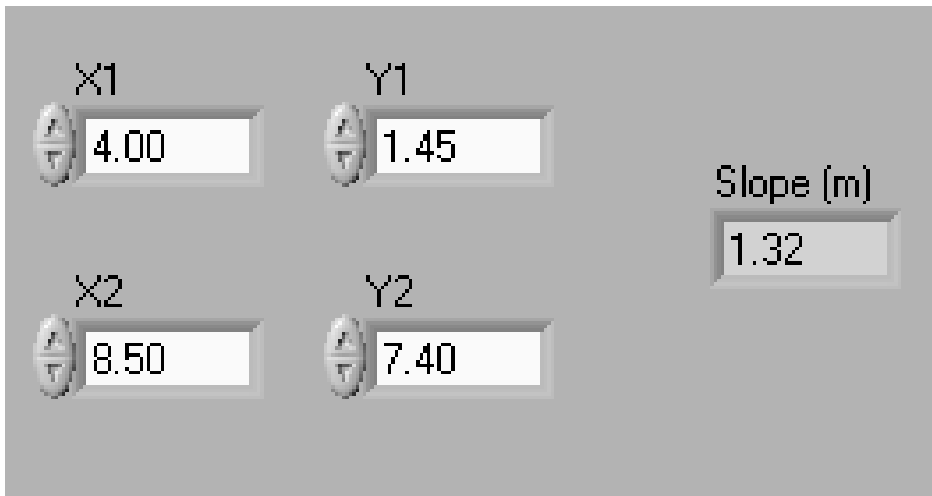
- A. Qu'est-ce qu'un sous-VI ?
- B. Comment créer un *SubVI* .
- C. Comment éditer l'icône et le connecteur ?
- D. Comment utiliser un VI comme sous-VI ?

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

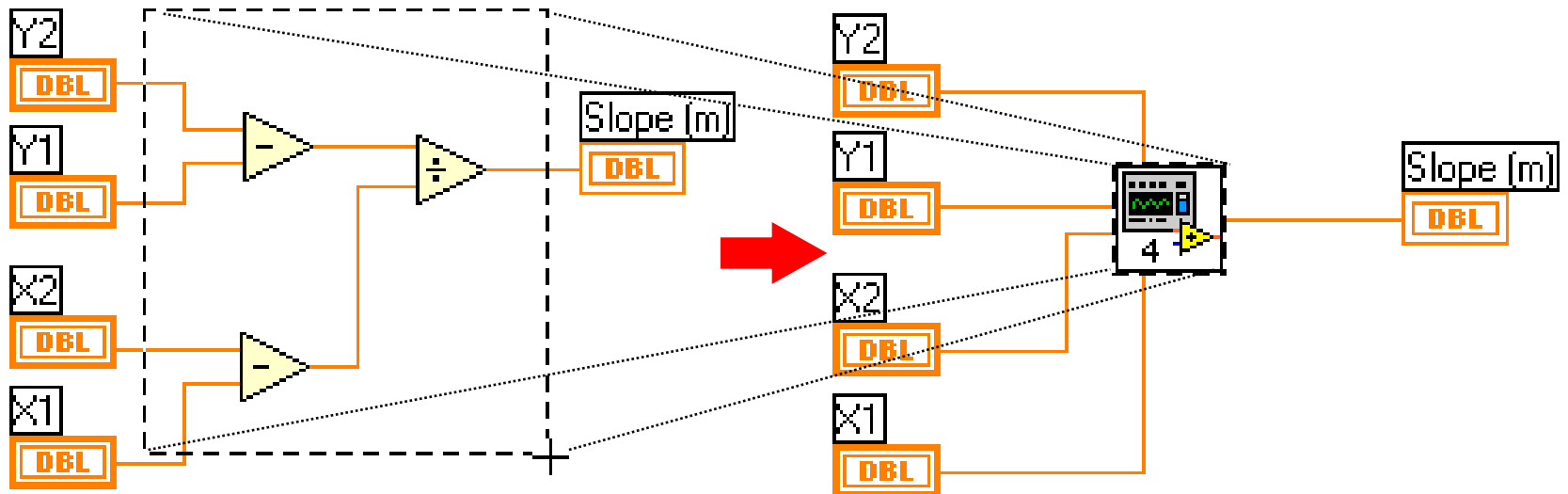


Exemple : sous-VI de calcul de pente



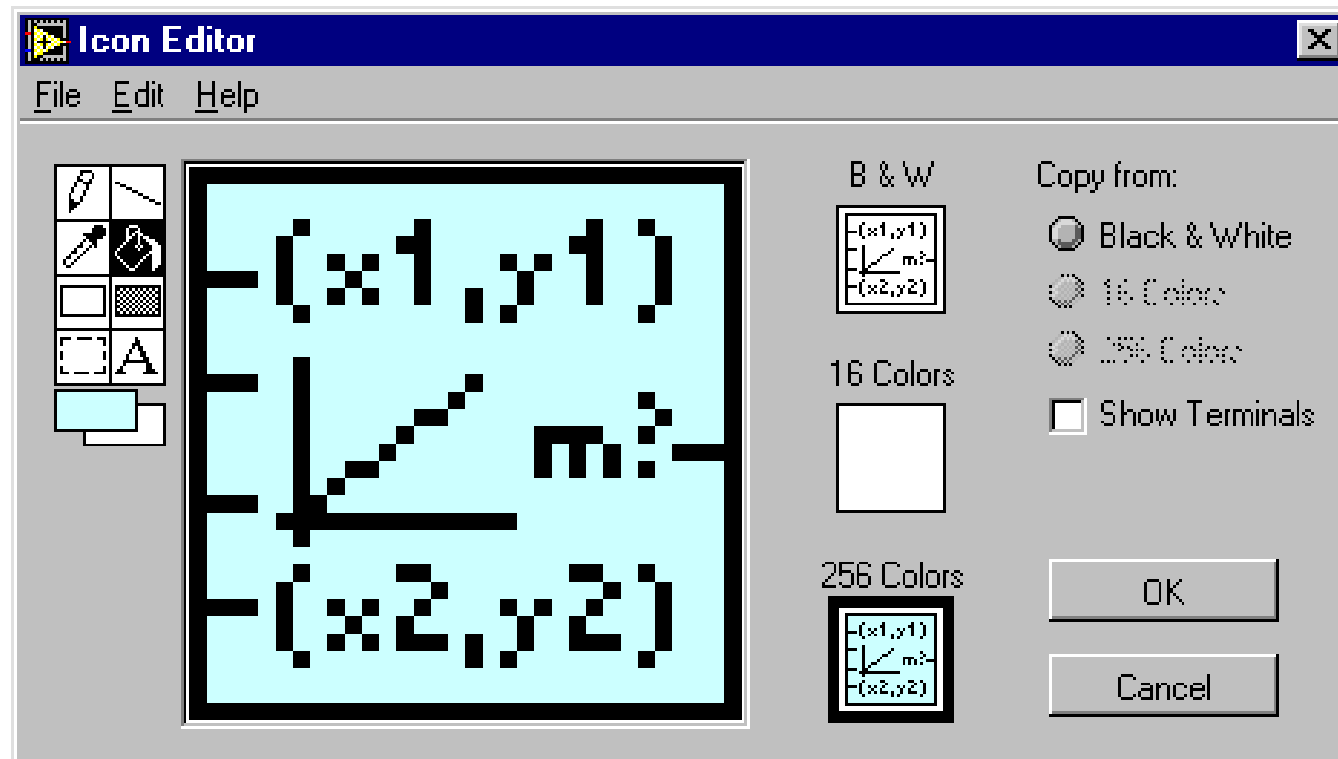
L'option "Create SubVI"

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



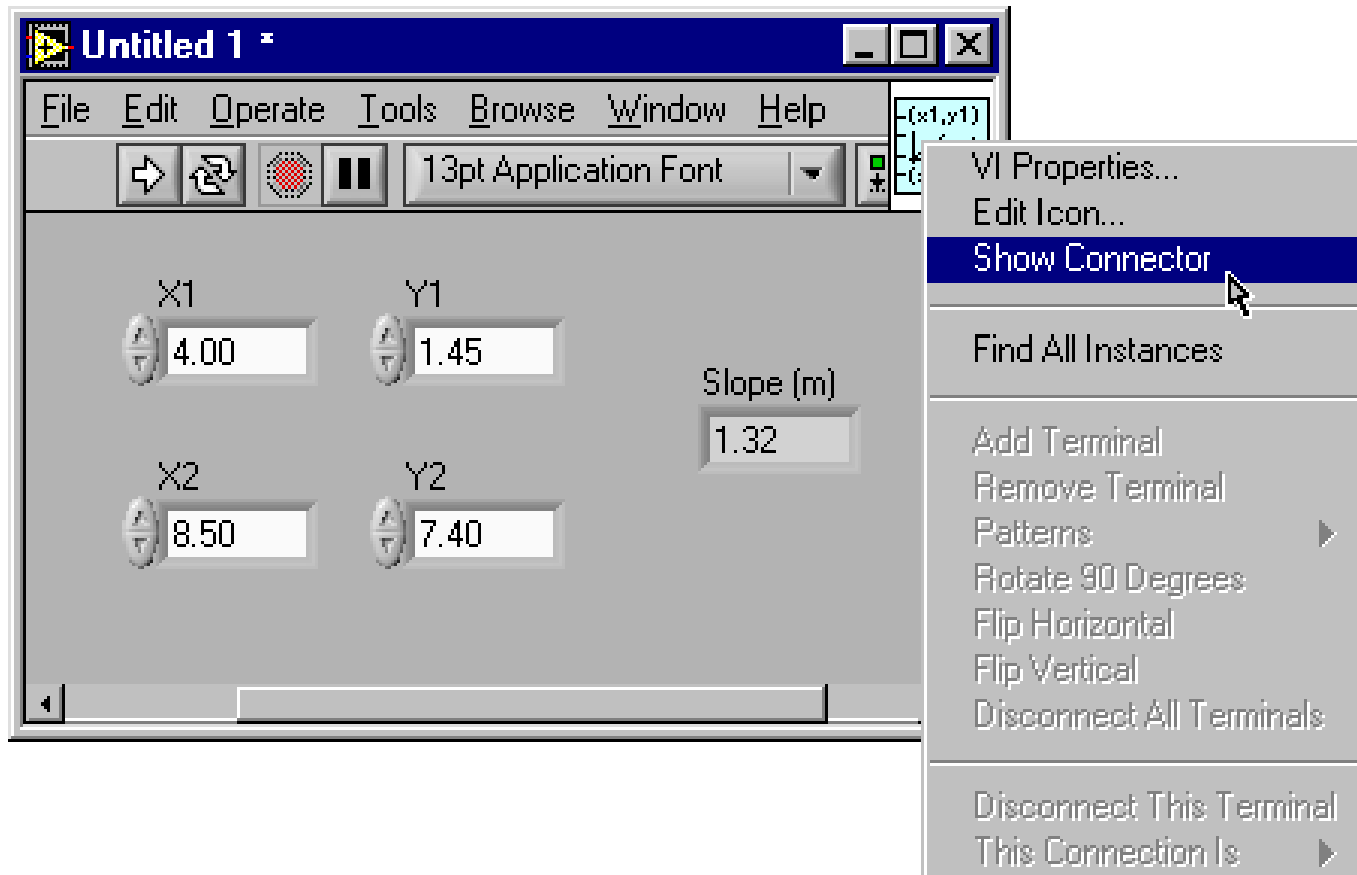
Edition de l'icône

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



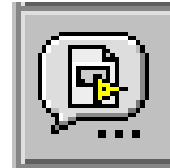
Edition du connecteur

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

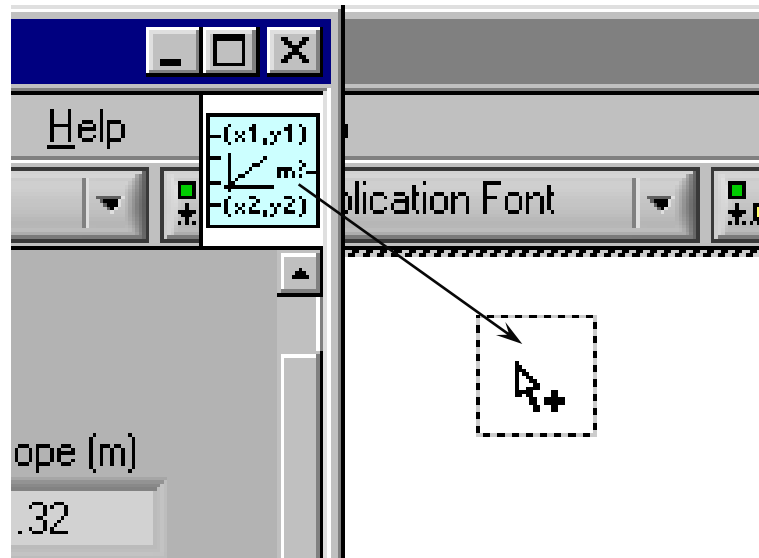


- Appel d'un sous-VI


- 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





Chapitre 6

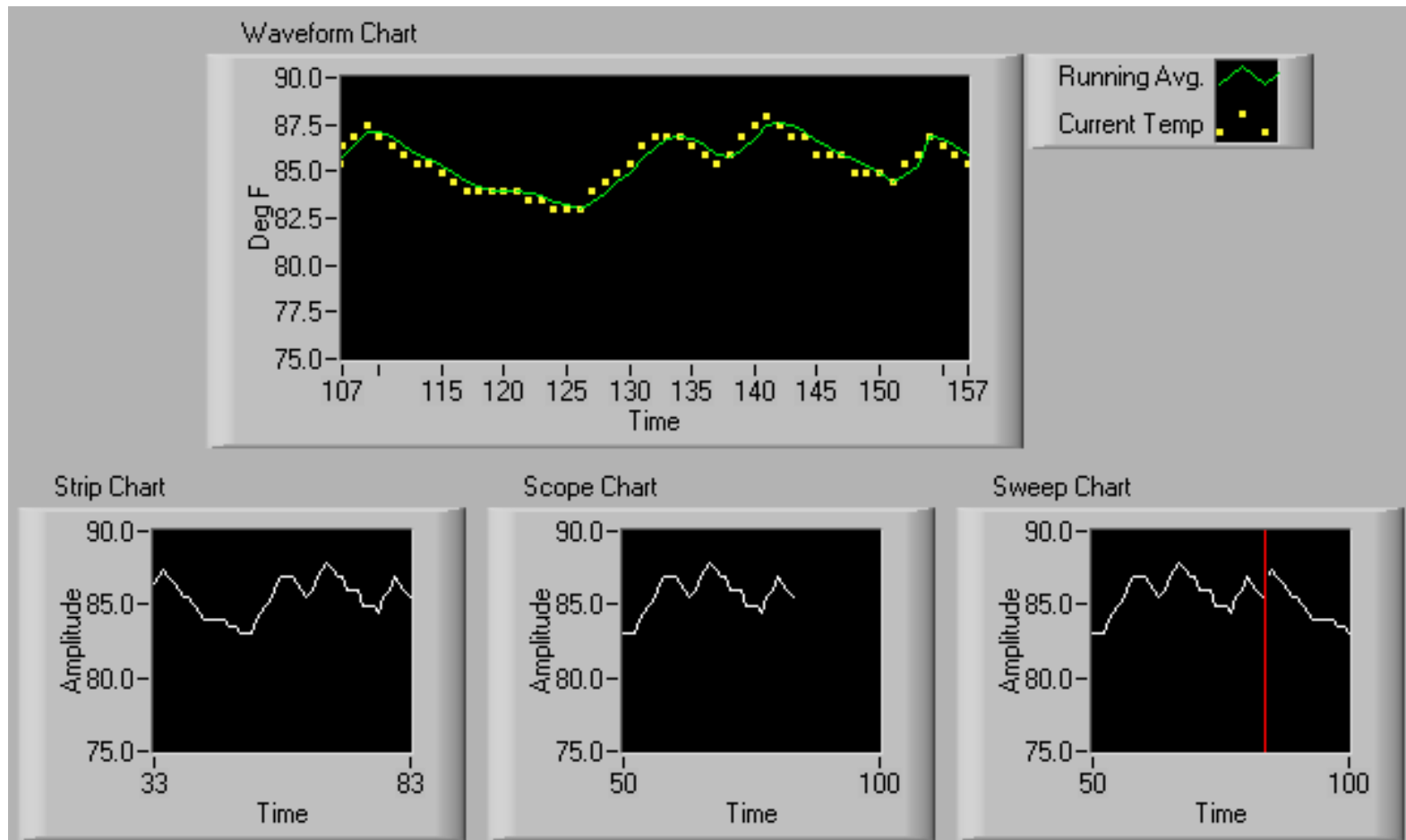
Graphs et Charts

Thèmes :

- A. Les graphiques (Graphs)
- B. Utilisation des graphs
- C. Les graphiques (Charts)
- D. Utilisation des Charts

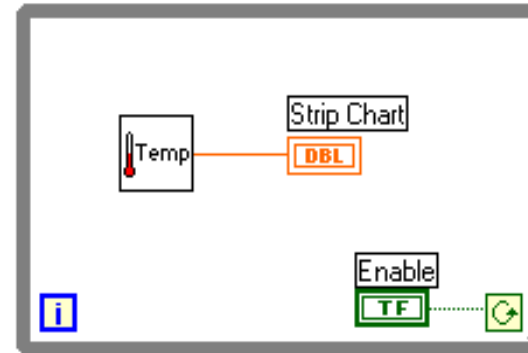
Chart

- Se trouve dans la sous-palette Controls → Graph

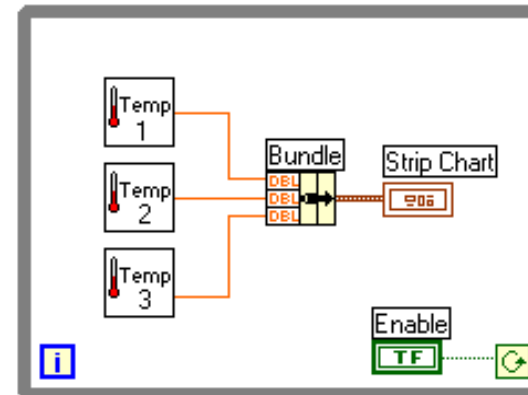


Utilisation des Charts

- Chart mono-trace



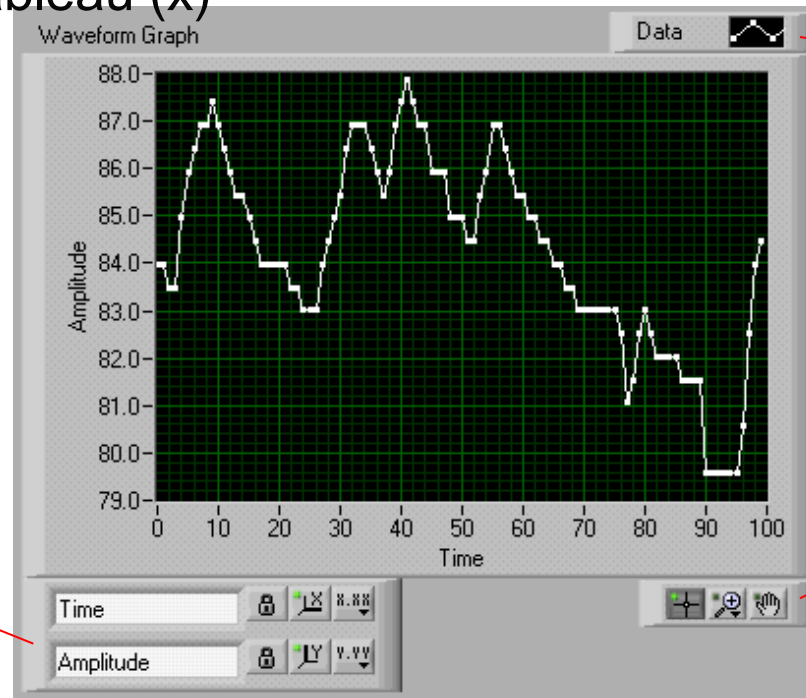
- Chart multi-traces



A chaque passage dans la boucle, le chart rajoute le dernier point à la suite de la courbe. C'est lui qui contient l'historique des points comme un enregistreur à papier déroulant

Graph

- Se trouve dans la sous-palette Graph
- Waveform Graph – Trace un tableau de points en fonction de leur index en une fois (pas d'historique)
- XY Graph – Plot un tableau de point (y) en fonction d'un second tableau (x)

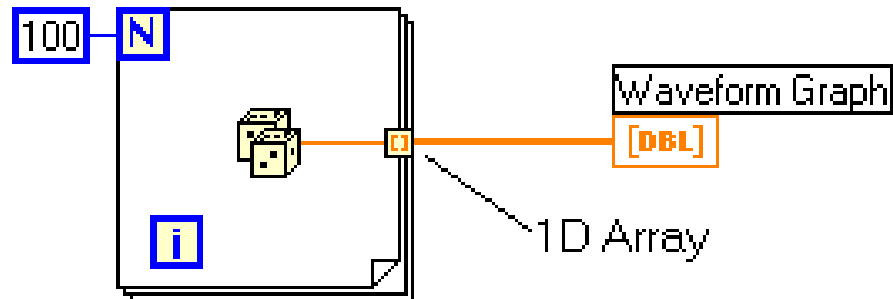


Plot Legend
(point and
line styles)

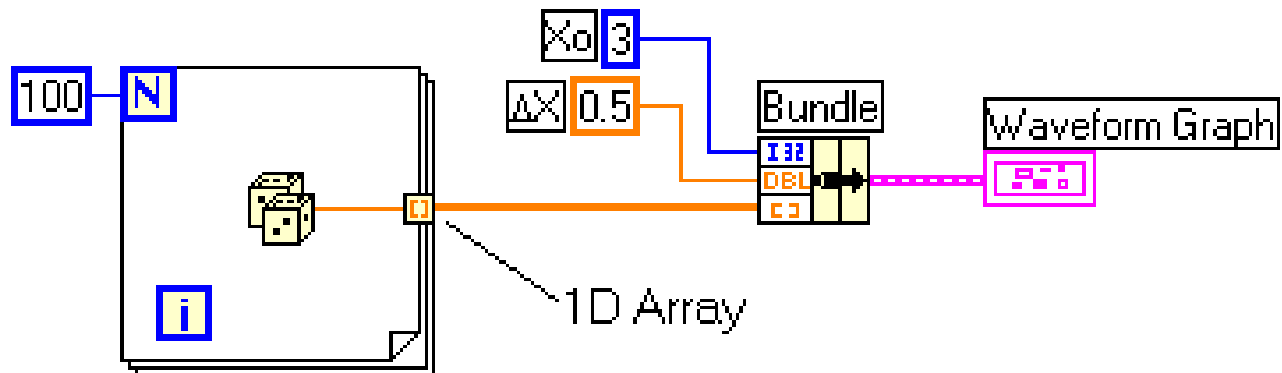
Scale Legend

Graph Palette

Graph mono-trace

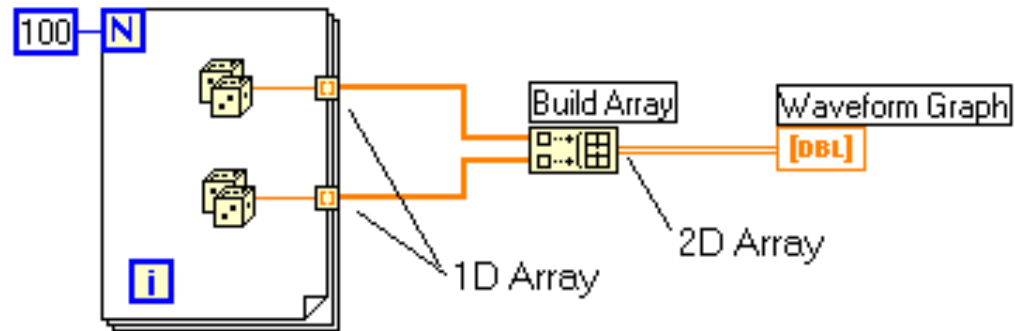


Axe X : $X_{\text{initial}} = 0.0$, $\Delta X = 1.0$ (par défaut)

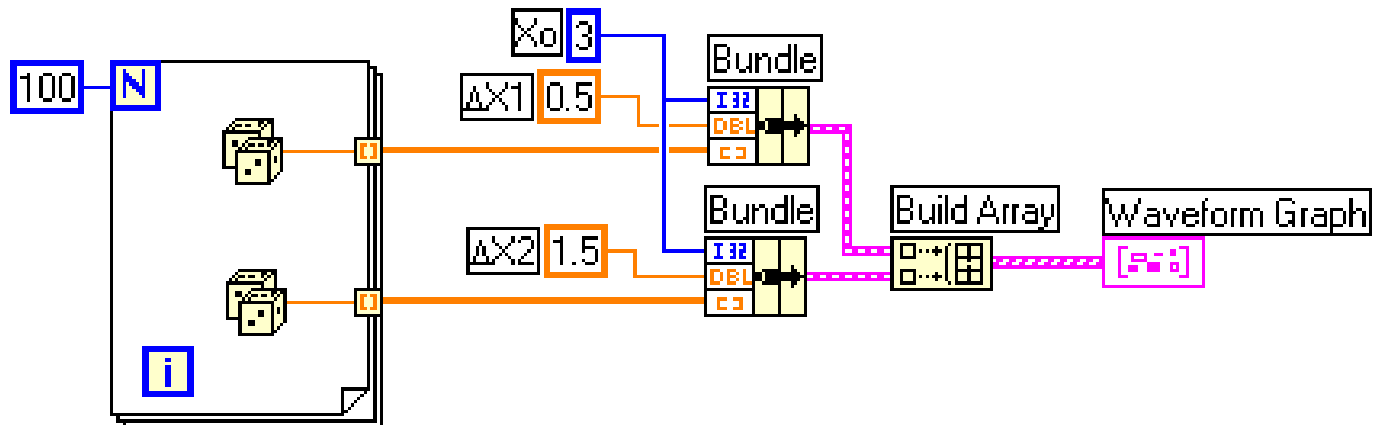


Axe X : $X_{\text{initial}} = 3$, $\Delta X = 0.5$ (défini par l'utilisateur)

Graph multi-traces

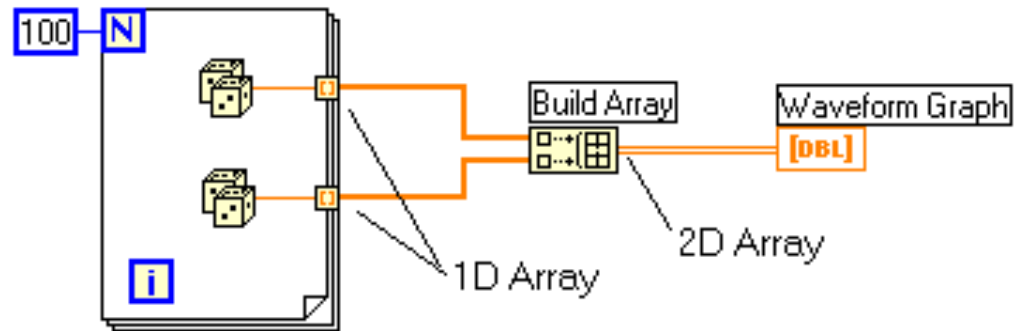


Chaque ligne est une trace séparée ($X_0 = 0$, $\Delta X = 1$ par défaut)

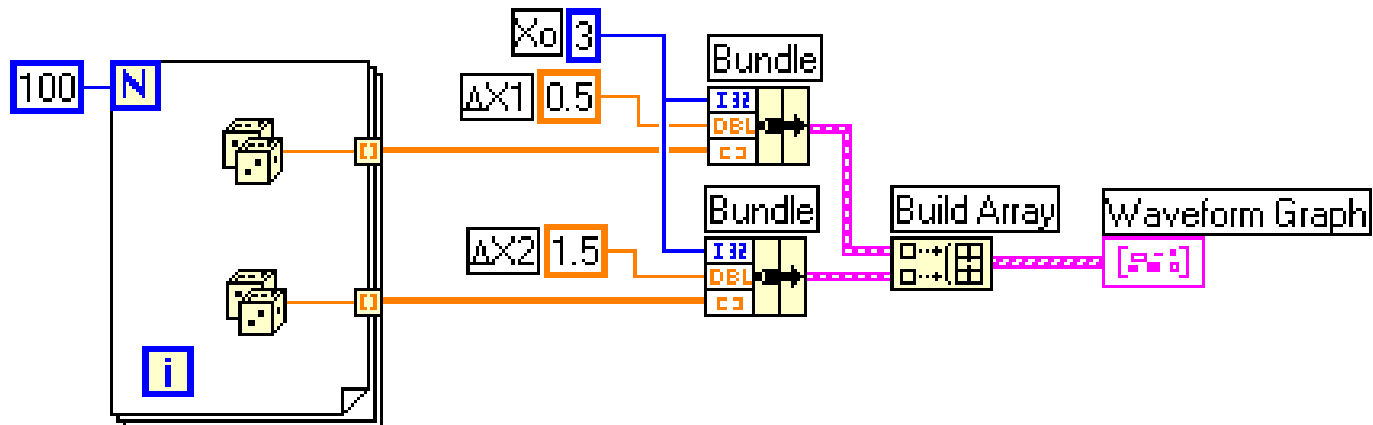


Chaque ligne est une trace séparée (X_0 et X spécifiés par l'utilisateur)

Graph multi-traces



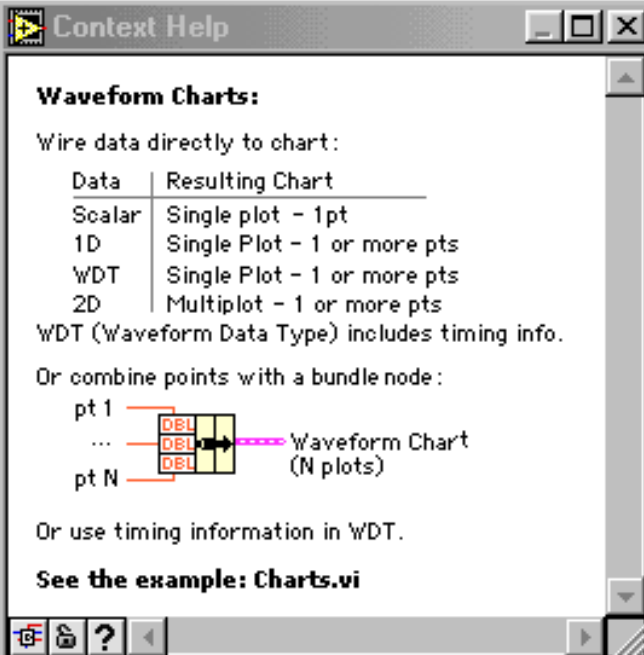
Chaque ligne est une trace séparée ($X_0 = 0$, $\Delta X = 1$ par défaut)



Chaque ligne est une trace séparée (X_0 et ΔX spécifiés par l'utilisateur)

Utilisation de l'aide contextuelle pour les Charts & Graphs

Waveform Chart



Context Help

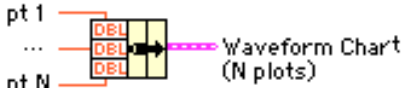
Waveform Charts:

Wire data directly to chart:

Data	Resulting Chart
Scalar	Single plot - 1pt
1D	Single Plot - 1 or more pts
WDT	Single Plot - 1 or more pts
2D	Multiplot - 1 or more pts

WDT (Waveform Data Type) includes timing info.

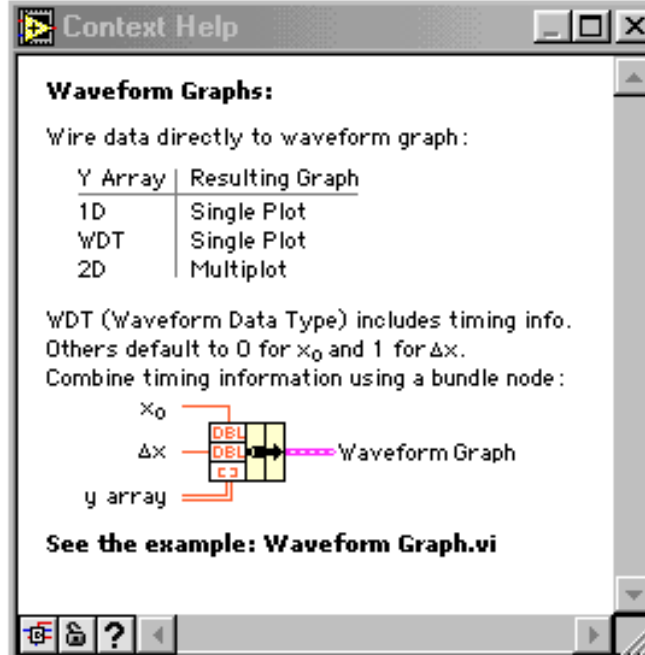
Or combine points with a bundle node:



Or use timing information in WDT.

See the example: Charts.vi

Waveform Graph



Context Help

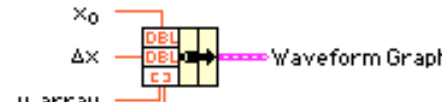
Waveform Graphs:

Wire data directly to waveform graph:

Y Array	Resulting Graph
1D	Single Plot
WDT	Single Plot
2D	Multiplot

WDT (Waveform Data Type) includes timing info. Others default to 0 for x_0 and 1 for Δx .

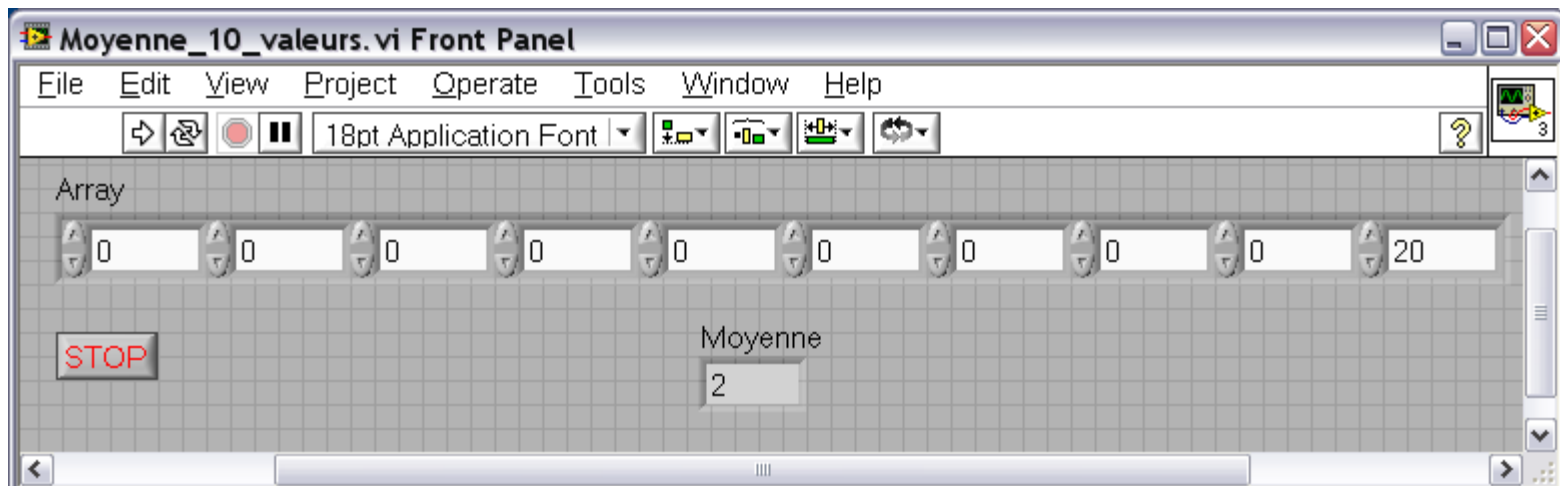
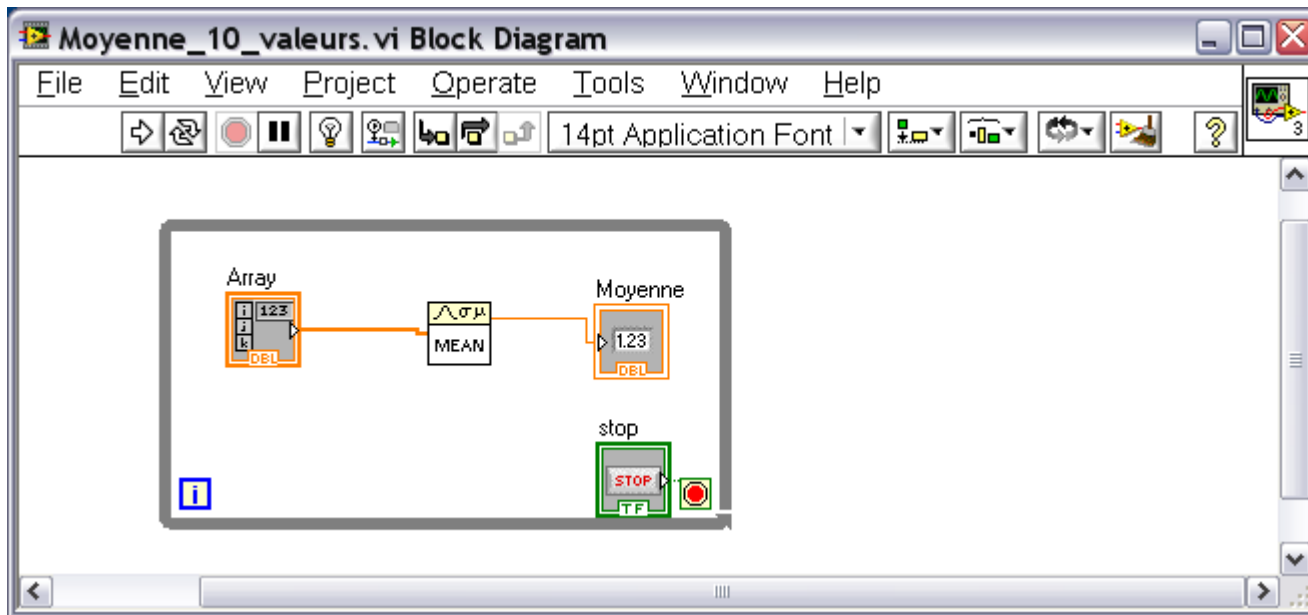
Combine timing information using a bundle node:



See the example: Waveform Graph.vi

Exercice sur les tableaux

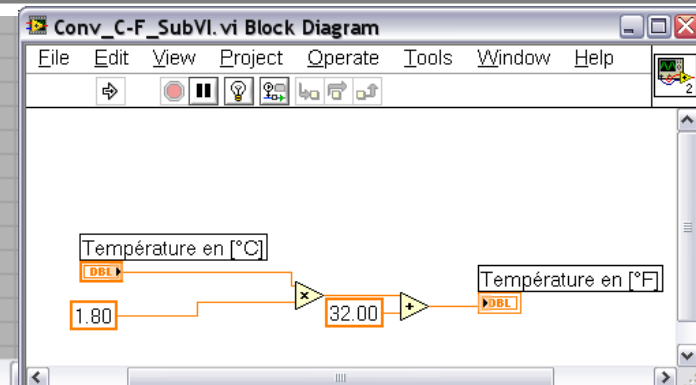
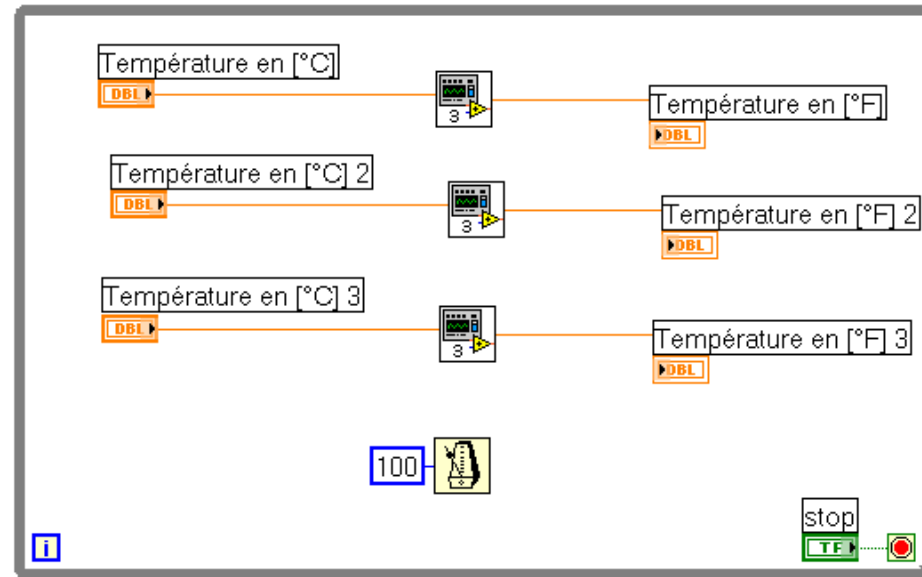
- Réaliser un programme qui calcule la valeur moyenne d'un tableau de 10 valeurs.



Exercice sur les sous-VI

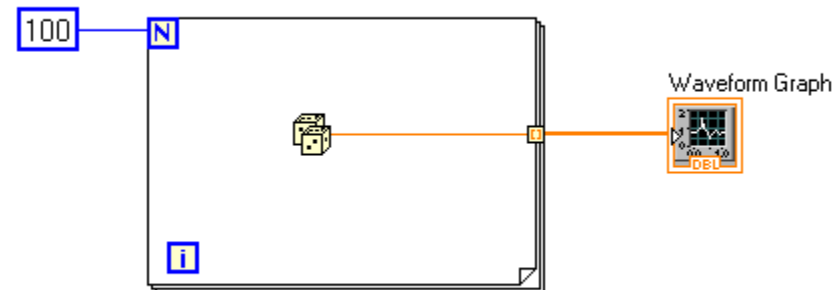
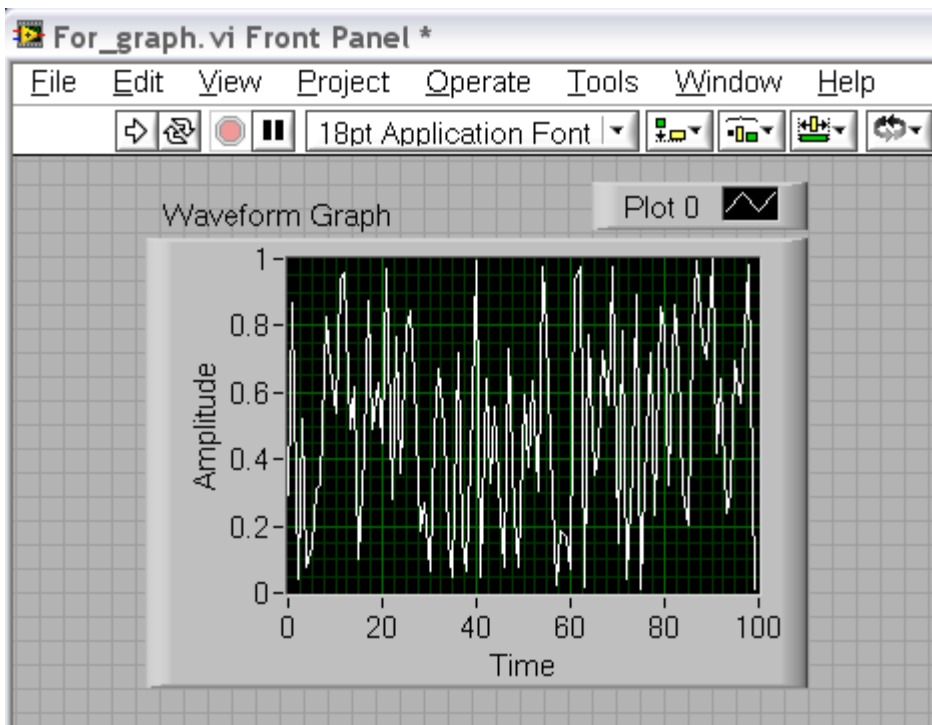
- Modifier l'exercice 1 (slide 37 de la première partie) pour réaliser le calcul dans un sous-programme réutilisable ailleurs.
- Faire ensuite un programme principal qui effectue dans une boucle While trois conversions de température simultanément en utilisant le même sous-programme de calcul

The screenshot shows the LabVIEW Front Panel for a sub-VI titled "exercice_subVI.vi". The title bar includes "exercice_subVI.vi Front Panel" and a menu with "File", "Edit", "View", "Project", "Operate", "Tools", "Window", and "Help". Below the menu is a toolbar with icons for navigation and execution, and a font size dropdown set to "18pt Application Font". The main area contains a large text box with the title "Conversion degrés Celsius [°C] en degrés Farenheit [° avec des sous-VI". Below this, there are three rows of controls. Each row has a numeric input field on the left labeled "Température en [°C]" and a numeric display field on the right labeled "Température en [°F]". All input and display fields currently show the value "0.00". At the bottom right of the panel is a red "STOP" button.



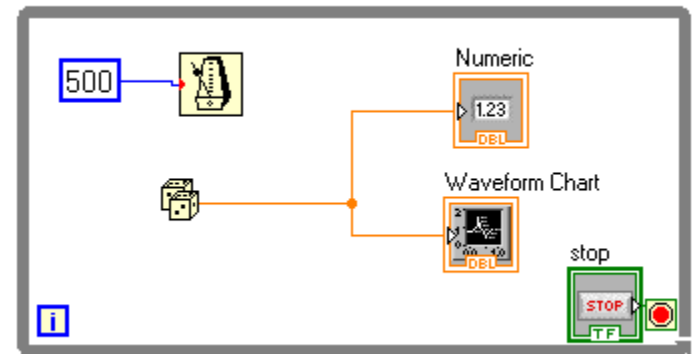
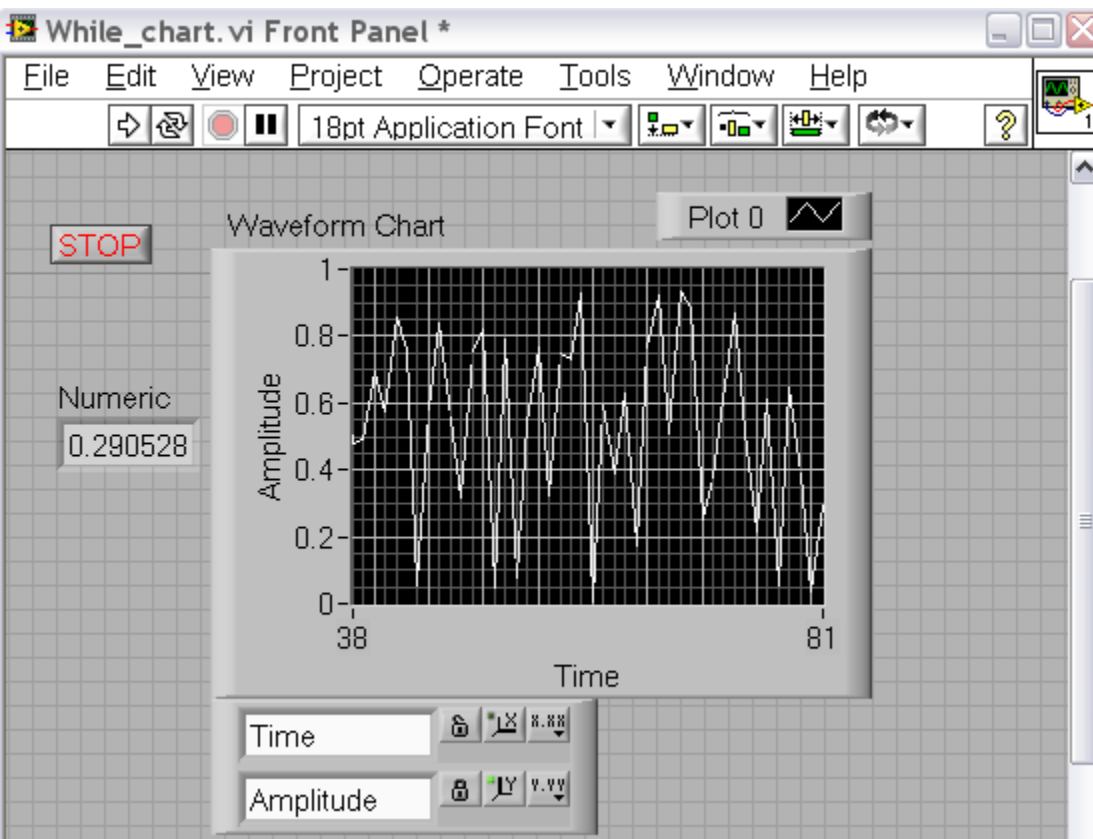
Exercice avec un Graph

- Réaliser un programme qui permet d'afficher une séquence de 100 nombres aléatoires entre 0 et 1 dans un graphique de type Graph.



Exercice avec un Chart

- Réaliser un programme qui permet d'afficher une séquence de durée indéfinie de nombres aléatoires entre 0 et 1 dans un graphique de type Chart.



Travail personnel

(à faire vraiment !)

- Exécutez le tutoriel du chapitre 2 (page 2-1 à 2-15) de l'*Initiation à LabVIEW* (censé prendre 45 minutes).



Chapitre 7

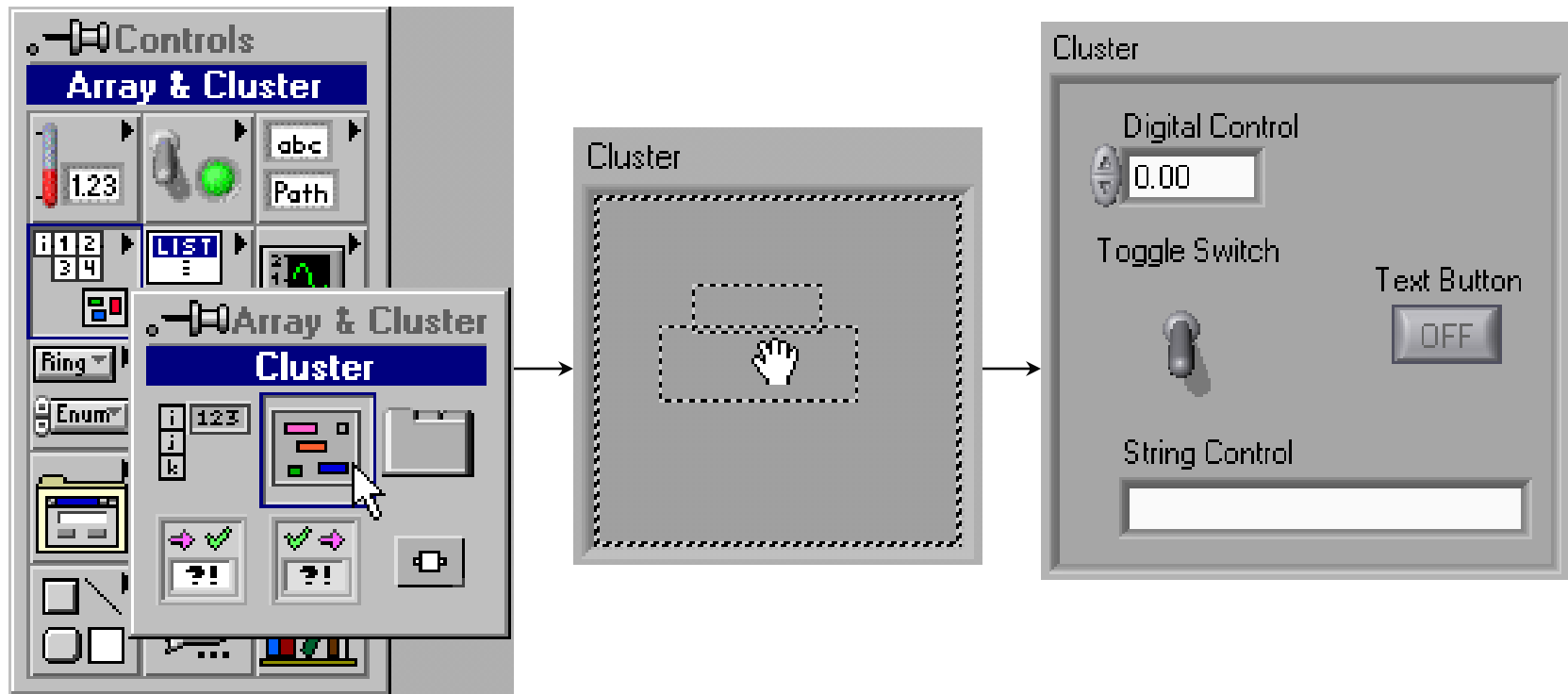
Les structures de données (Clusters)

- 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



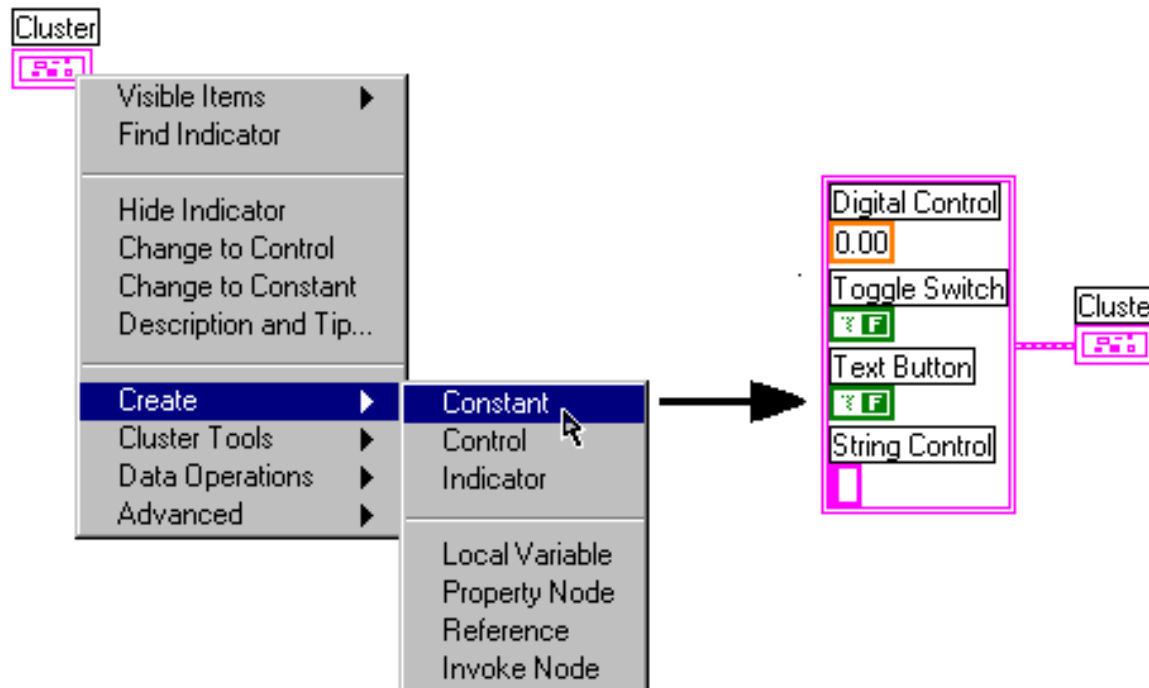
Les Contrôles et Indicateurs de type cluster

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



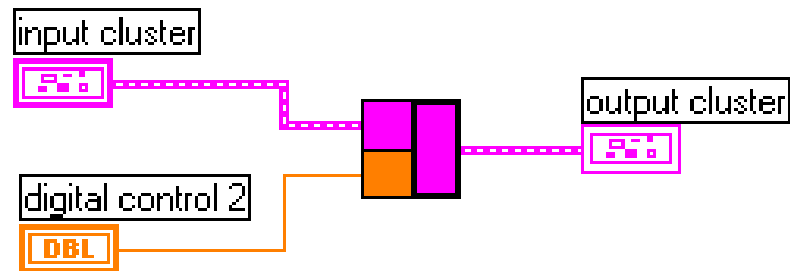
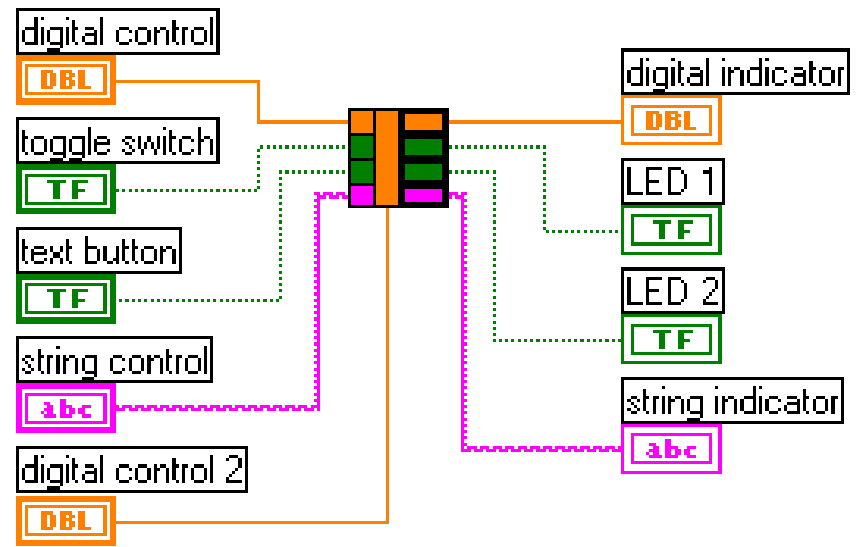
Les constantes de type Cluster

- 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 .



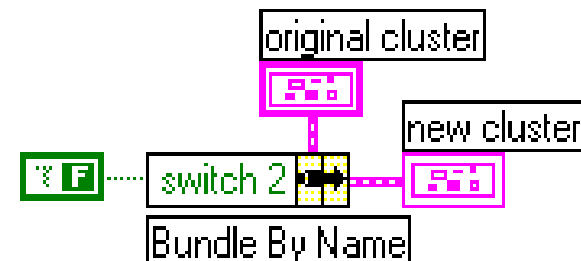
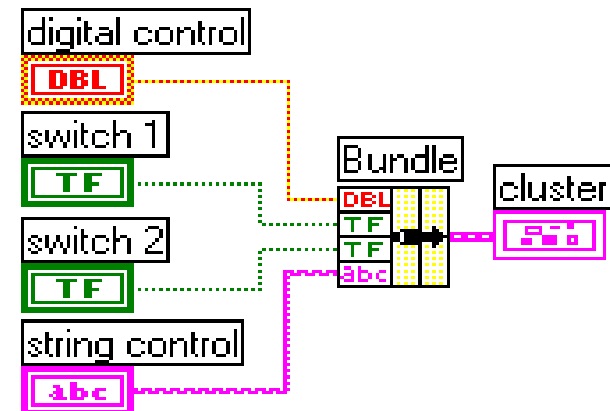
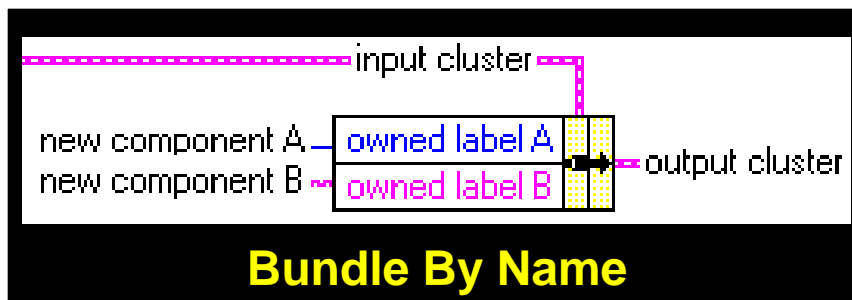
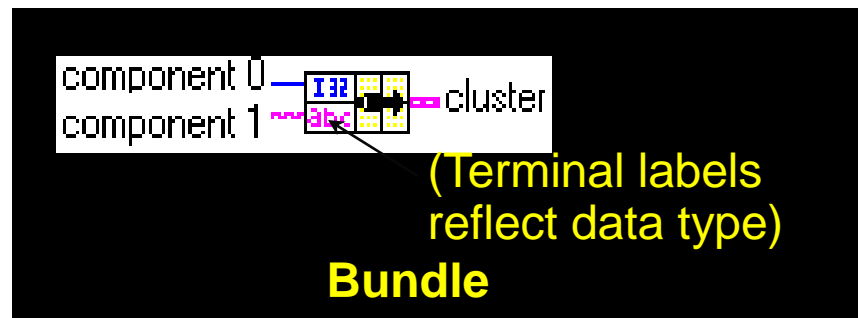
Utilisation de Clusters pour passer des données à des sous-VI

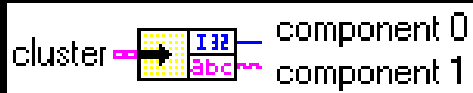
- Permet de passer plusieurs valeurs dans un seul terminal
- Permet de dépasser la limite des 28-terminaux/VI
- Simplifie le câblage



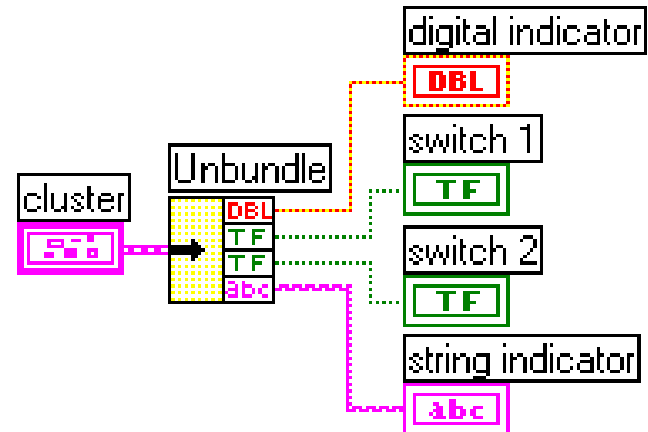
Les Fonctions sur les Clusters

- 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.

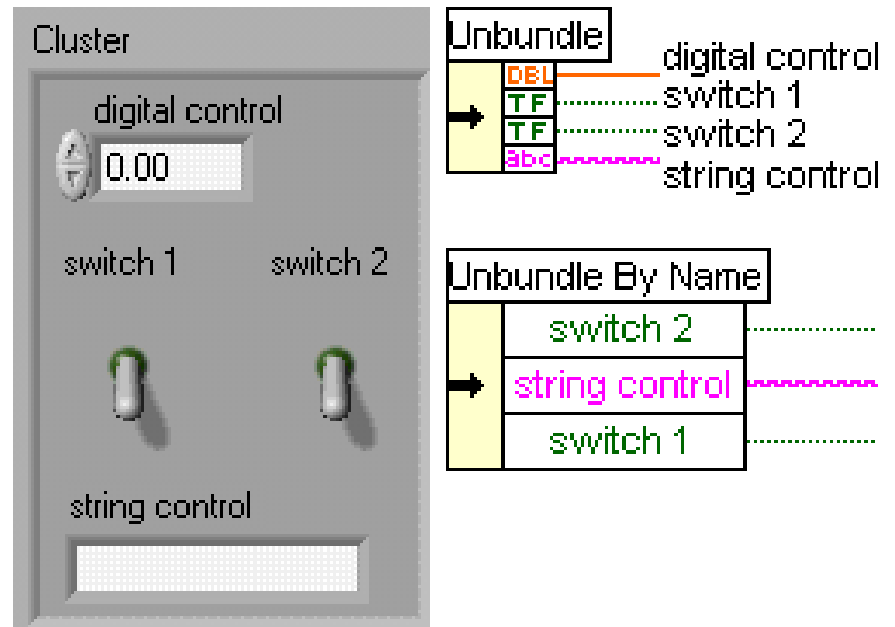




Unbundle



Unbundle By Name



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.



Chapitre 8

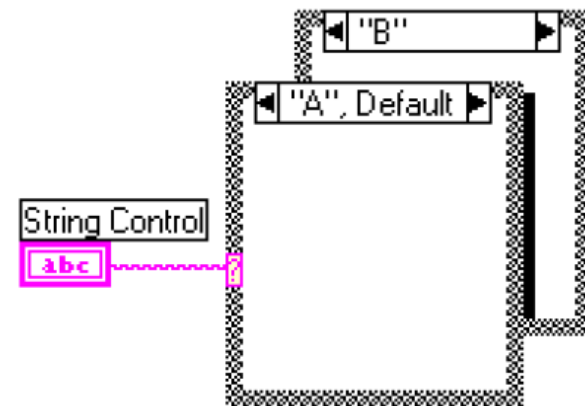
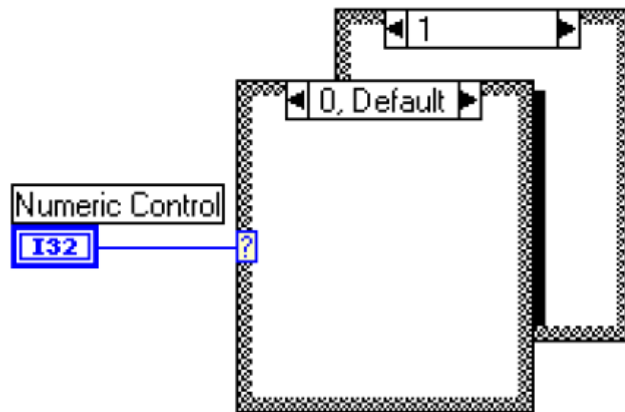
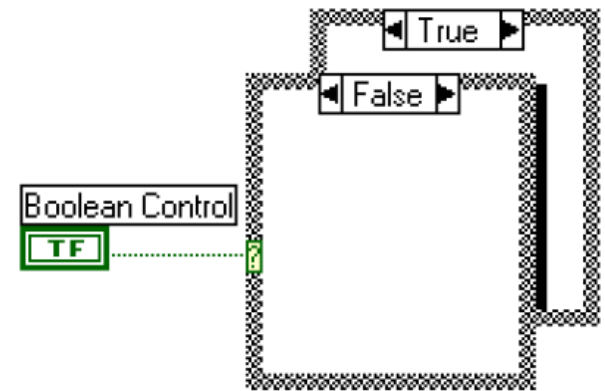
Séquences

Thèmes :

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

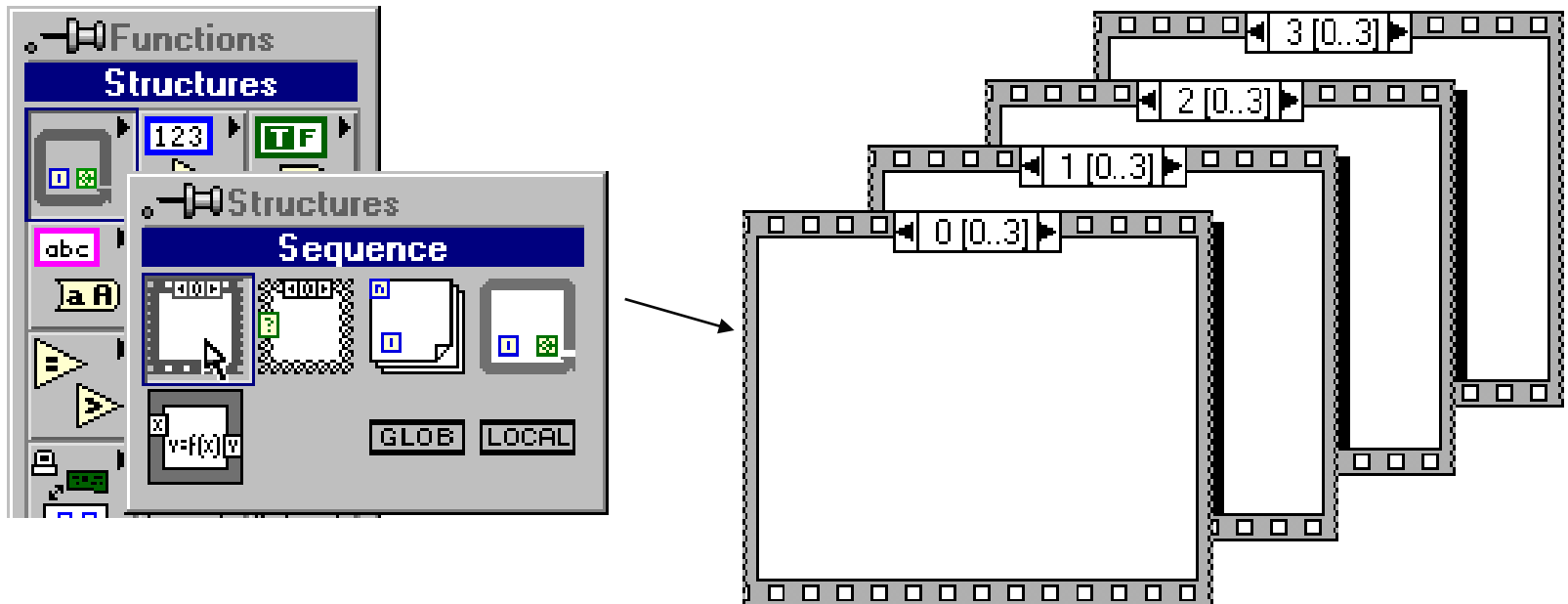
Case

- **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



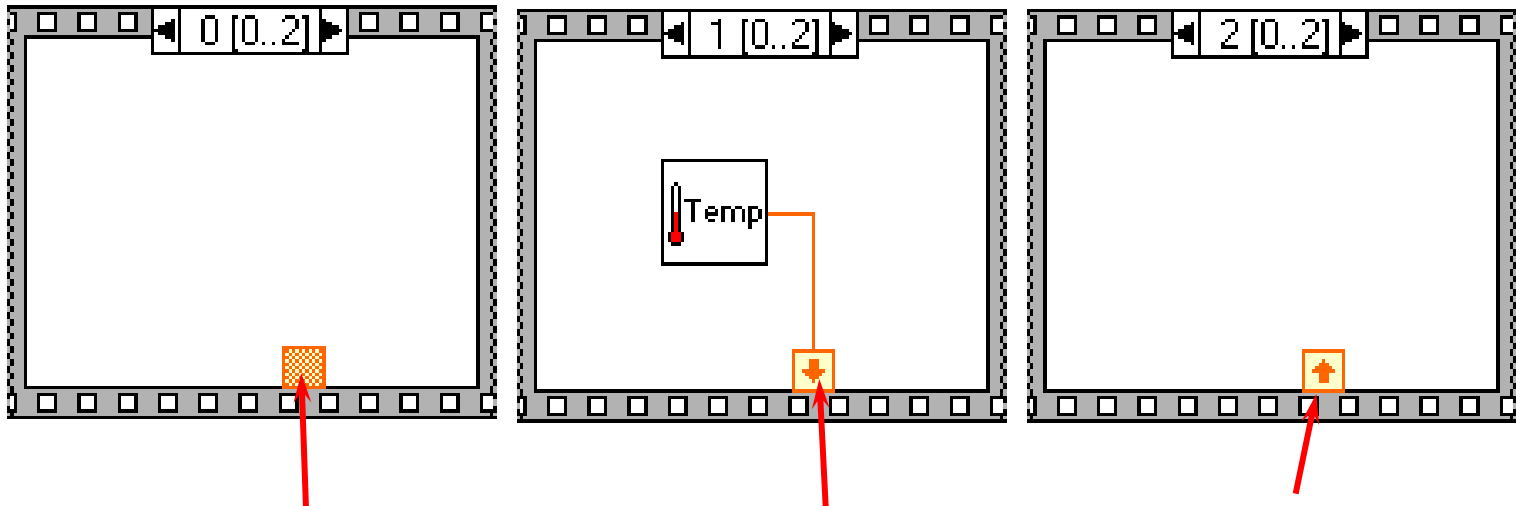
Séquences

- 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.



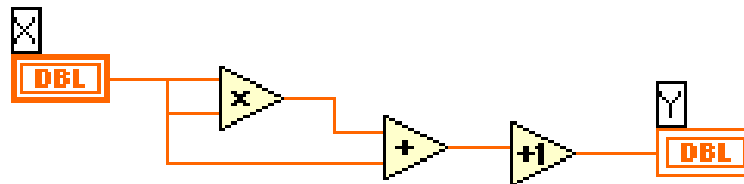
Pas de data
À disposition

Séquence locale
Crée dans le
Frame 1

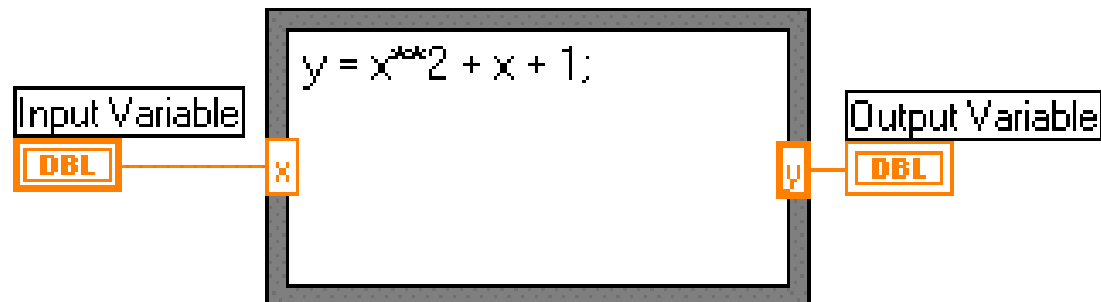
Data
Disponible depuis
Le Frame 1

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 semicolon (;).
- L'aide contextuelle montre les fonctions disponibles.
- Exemple: au lieu de:



- On peut avoir:

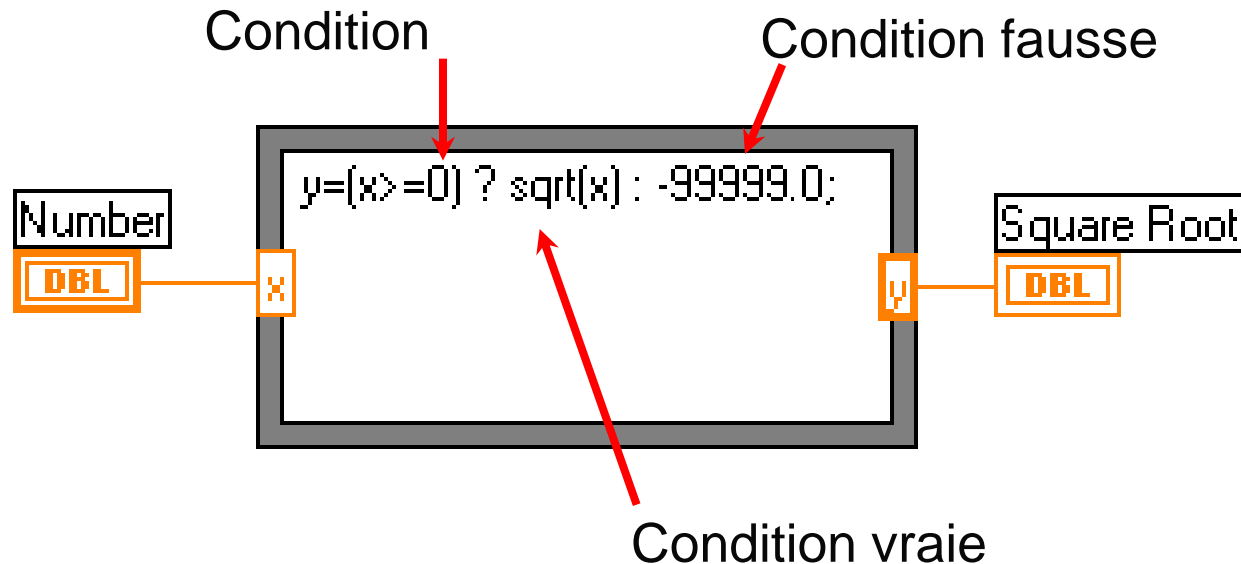


Exécuter des conditions dans un Formula Node

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

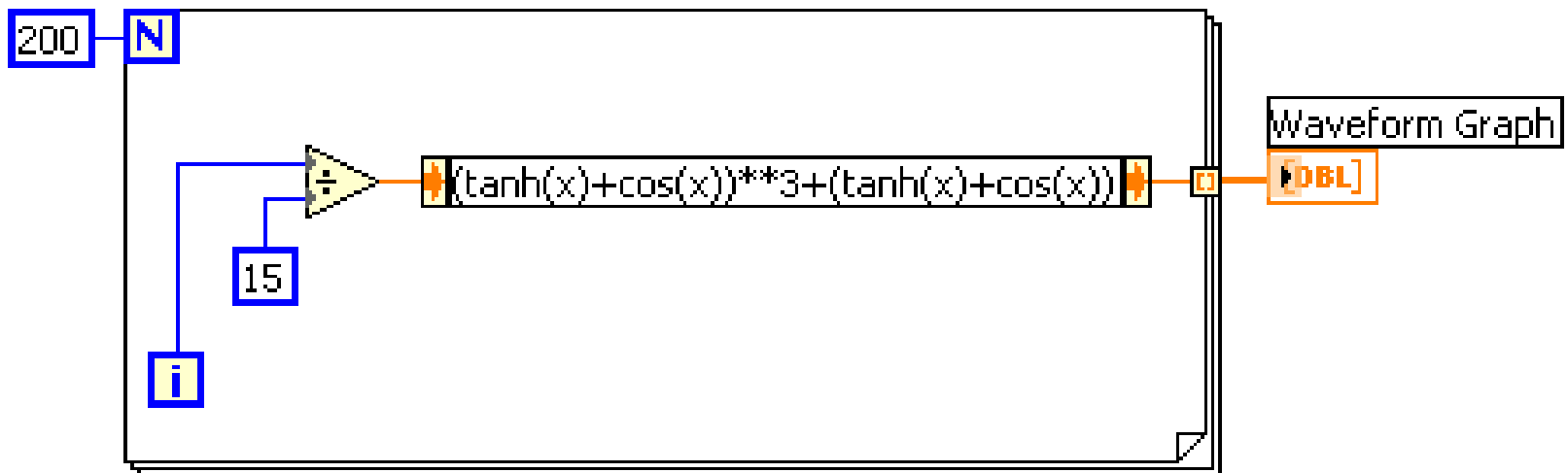
```
if (x >= 0) then
    y = sqrt(x)
else
    y = -99999.0
end if
```

- On peut faire:



Expression Nodes

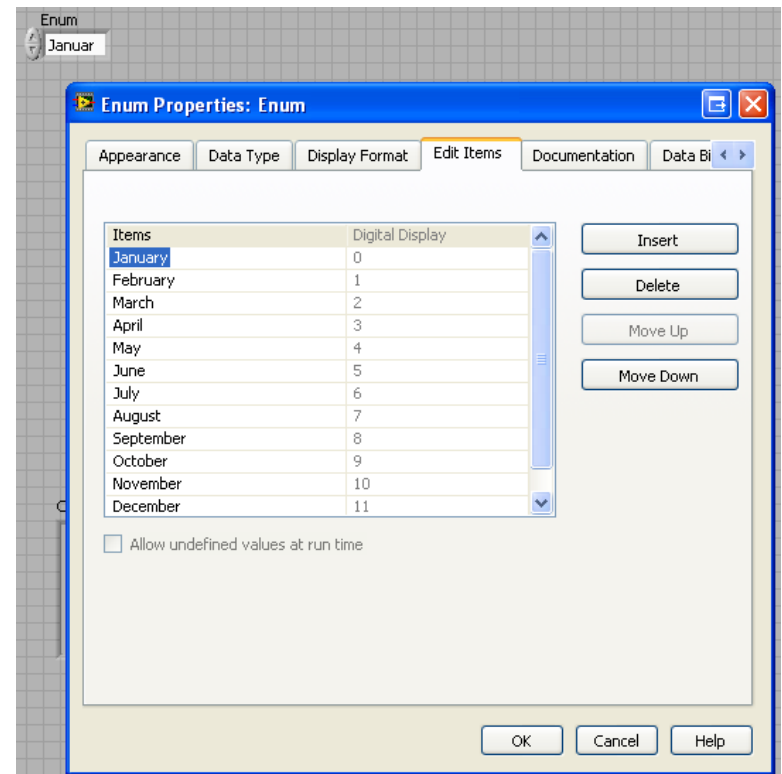
- Utilisation d' Expression Nodes pour résoudre des équations simples mono-variables





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.



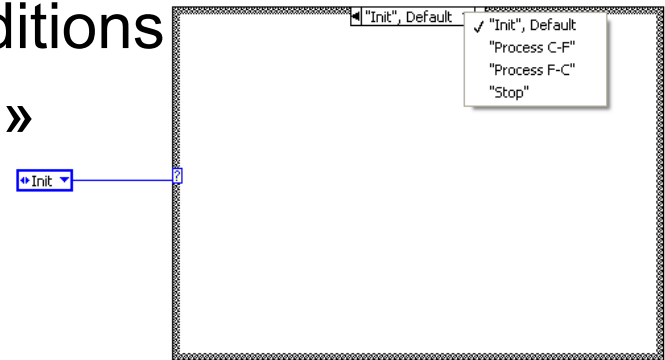
Résumé du chapitre

- 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.



Travail optionnel

- Créer un VI permettant de convertir les °C en °F et inversement grâce à une machine d'état.
- La machine d'état doit avoir 4 conditions
 - En attente de sélection = « Init »
 - °C en °F = « Process C-F »
 - °F en °C = « Process F-C »
 - Arrêt du programme = « Stop »
- Il n'y a pas de boutons pour arrêter le programme ! C'est l'état Stop qui arrête le VI.
- Il ne doit y avoir dans le front panel qu'un enum, un numérique control et un numérique indicateur



Travail optionnel

Selection du type de conversion choisie

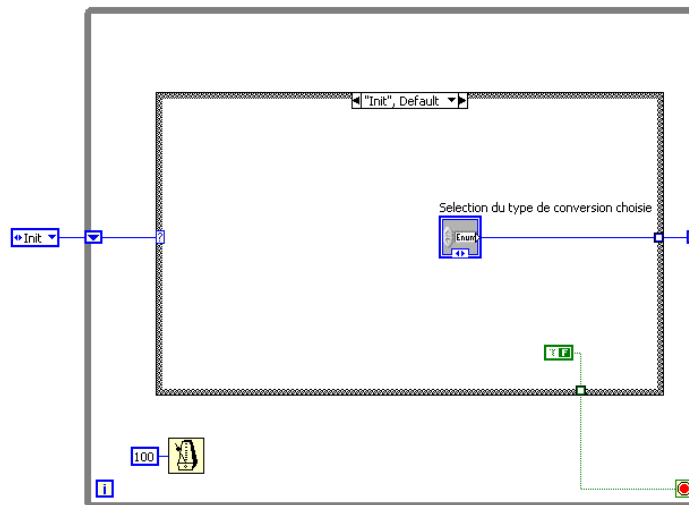
Init

Value to convert

0

Result

0



Chapitre 9

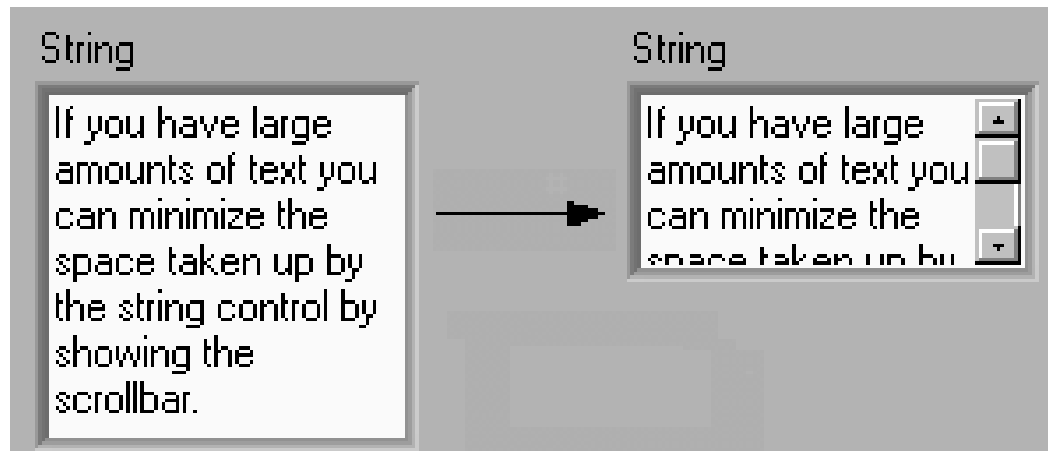
Chaînes de caractères et opérations sur fichiers

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



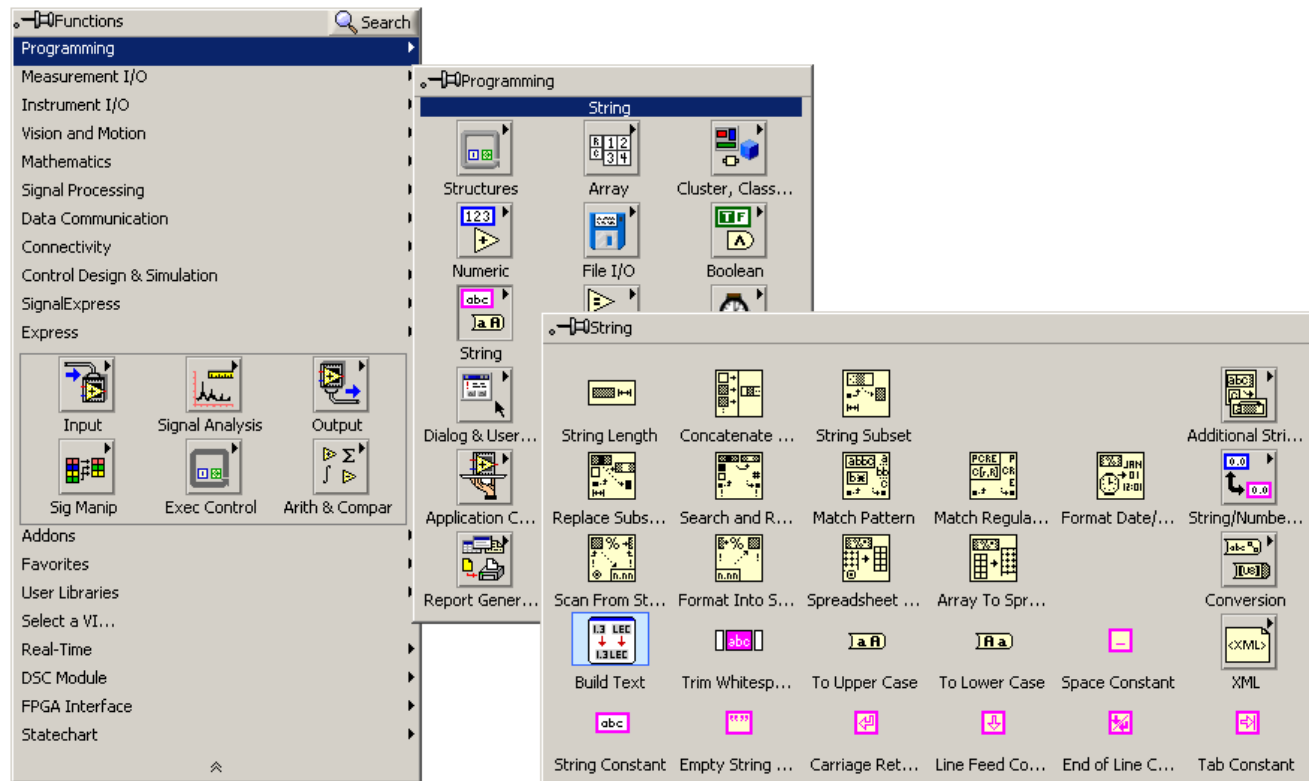
- Affichage 'Password'



- Affichage Hexadécimal



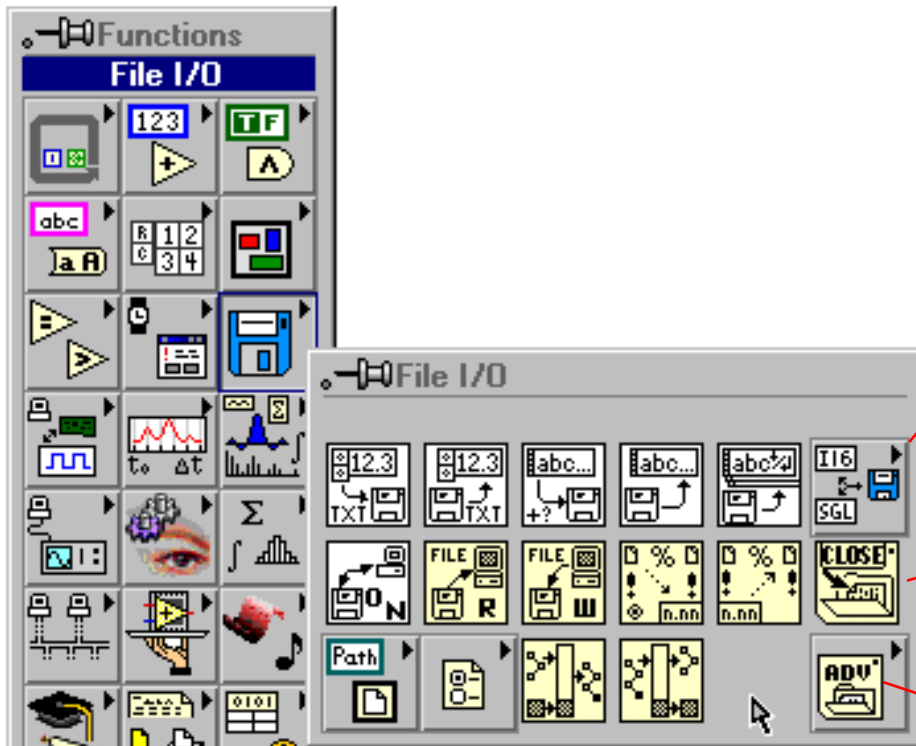
Opérations sur chaînes de caractères



Opérations sur fichiers

Trois niveaux de hiérarchie :

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

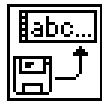
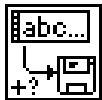
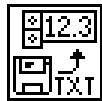


High Level File VIs

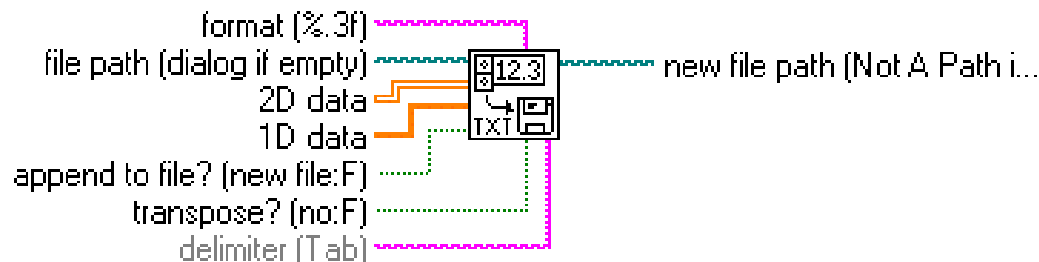
Intermediate File VIs and Functions

Advanced File Functions

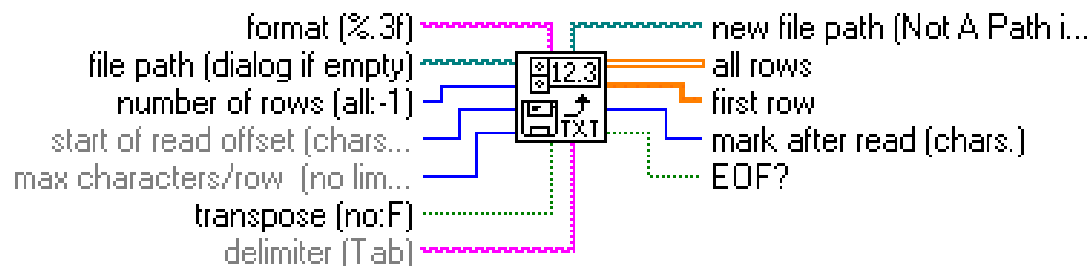
Vis de haut-niveau pour opérations sur fichiers



- 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



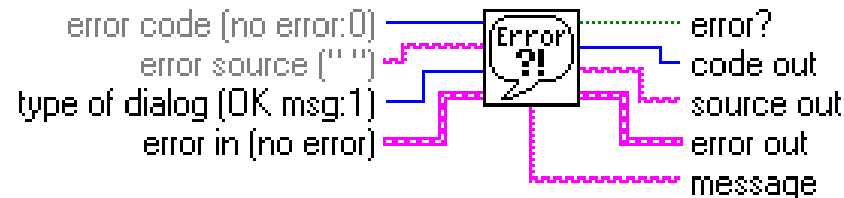
- **Write File** – écrire les données dans un fichier



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

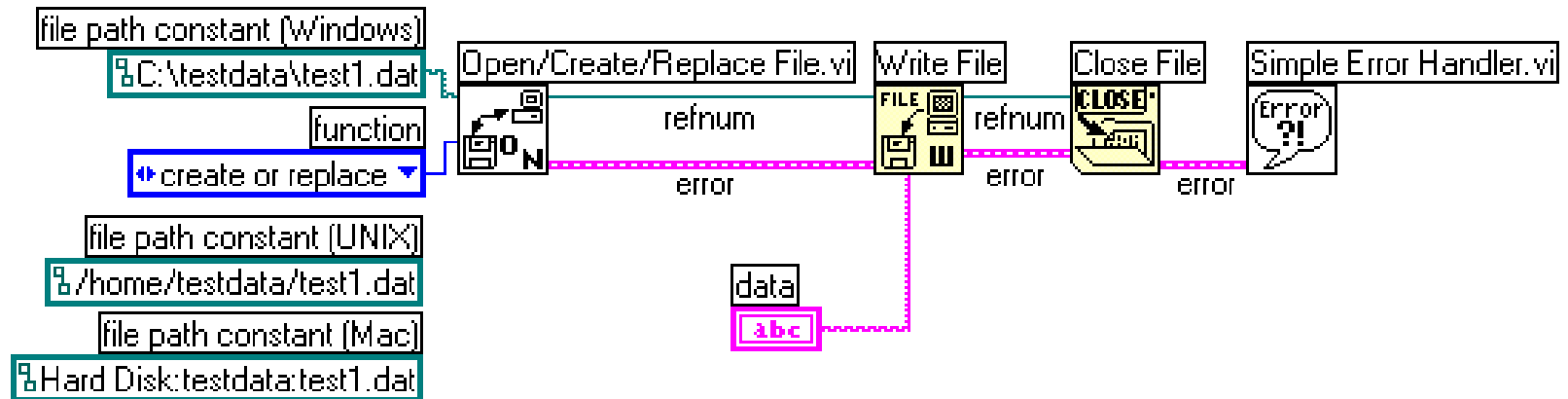
- Gestion des erreurs dans les fonctions fichier:
se trouve dans la sous-palette *'Time & Dialog'*

– Affiche un message d'erreur le cas échéant



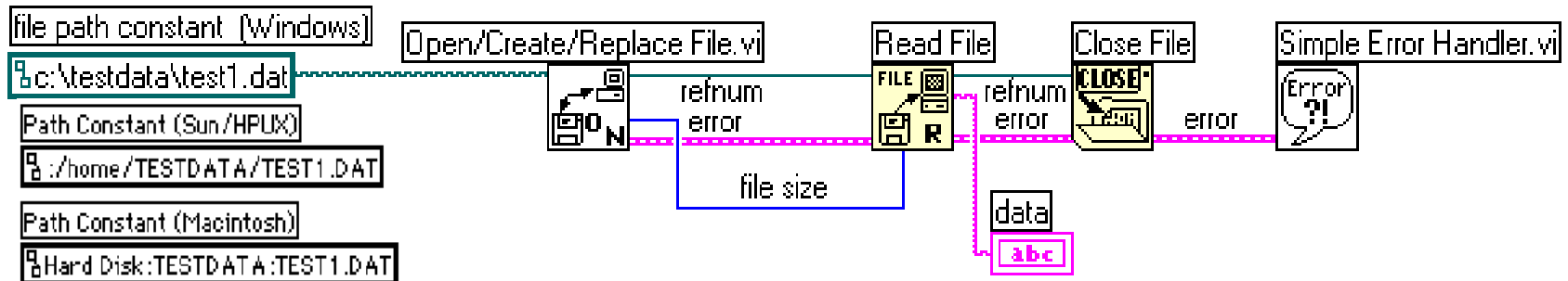
Simple Error Handler.vi

Sauver des données dans un fichier



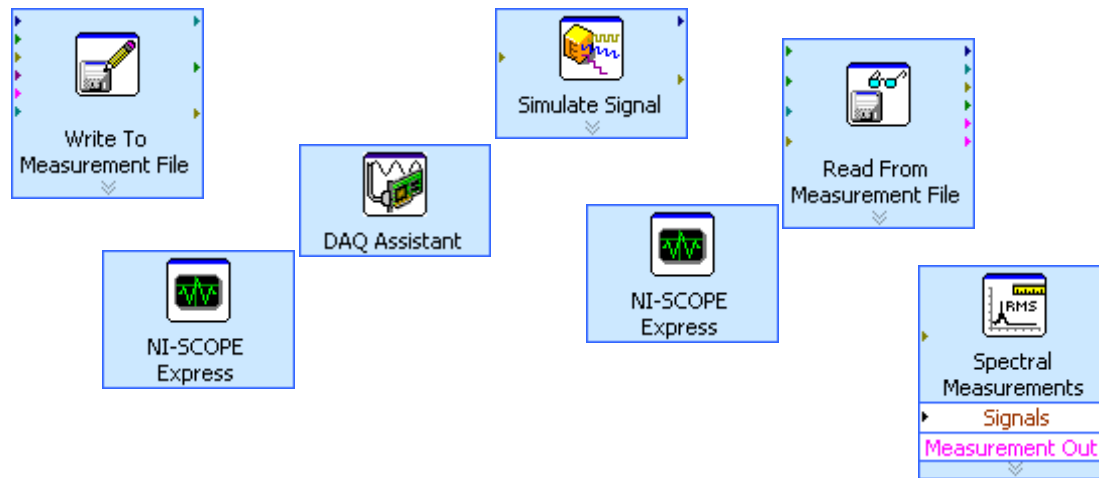
- **Open/Create/Replace:** ouvre le fichier existant TEST1.DAT et génère un cluster refnum et un cluster error pour les blocs suivants
- **Write File :** écrit les datas dans le fichier
- **Close File :** ferme l'accès au fichier
- **Simple Error Handler :** affiche un message en cas d'erreur

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" block with a "Sine" waveform, a "Write To Measurement File" block, a "stop" button, and a numeric control set to "100". A red arrow points from the "Write To Measurement File" block to the "Configure Write To Measurement File" dialog box.

The dialog box, titled "Configure Write To Measurement File [Write To Measurement File]", contains 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)

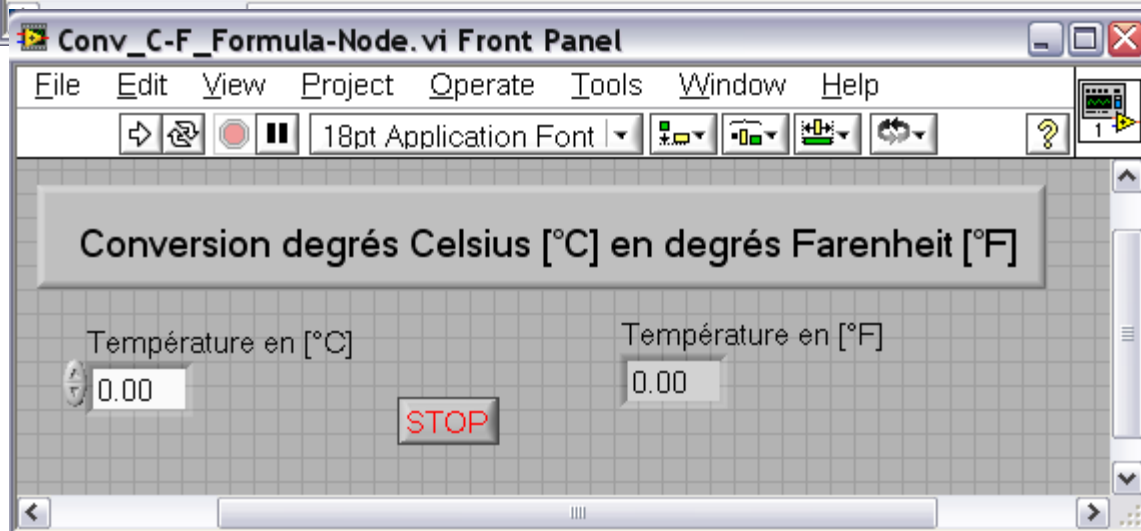
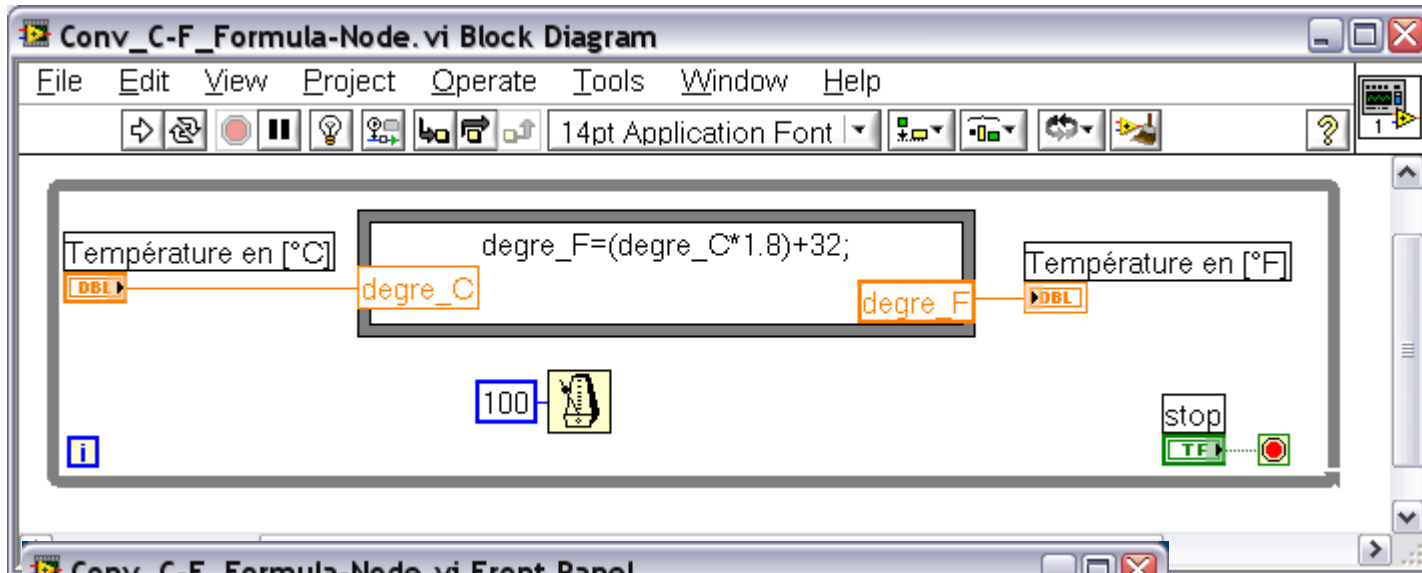
Below the dialog box, a Microsoft Excel spreadsheet is shown with a data table and a graph of a sine wave. The data table is as follows:

Time (s)	Amplitude (V)
0.00	0.000000
0.01	0.002991
0.02	0.012033
0.03	0.107301
0.04	0.24889
0.05	0.309117
0.06	0.309125
0.07	0.425779
0.08	0.487754
0.09	0.506207
0.10	0.507956
0.11	0.503244
0.12	0.494647
0.13	0.472869
0.14	0.437761
0.15	0.391017
0.16	0.334205
0.17	0.267307
0.18	0.190407
0.19	0.102976
0.20	0.009597
0.21	0.006983
0.22	0.002087
0.23	0.002115
0.24	0.006027
0.25	0.01
0.26	0.006027
0.27	0.002115
0.28	0.002087
0.29	0.006983
0.30	0.019597
0.31	0.032976
0.32	0.046407
0.33	0.0597307
0.34	0.0728307
0.35	0.084426
0.36	0.094617
0.37	0.103205
0.38	0.109189
0.39	0.112426
0.40	0.112795
0.41	0.110207
0.42	0.104679
0.43	0.096279
0.44	0.084075
0.45	0.068117
0.46	0.048301
0.47	0.024691
0.48	0.007301
0.49	0.002301



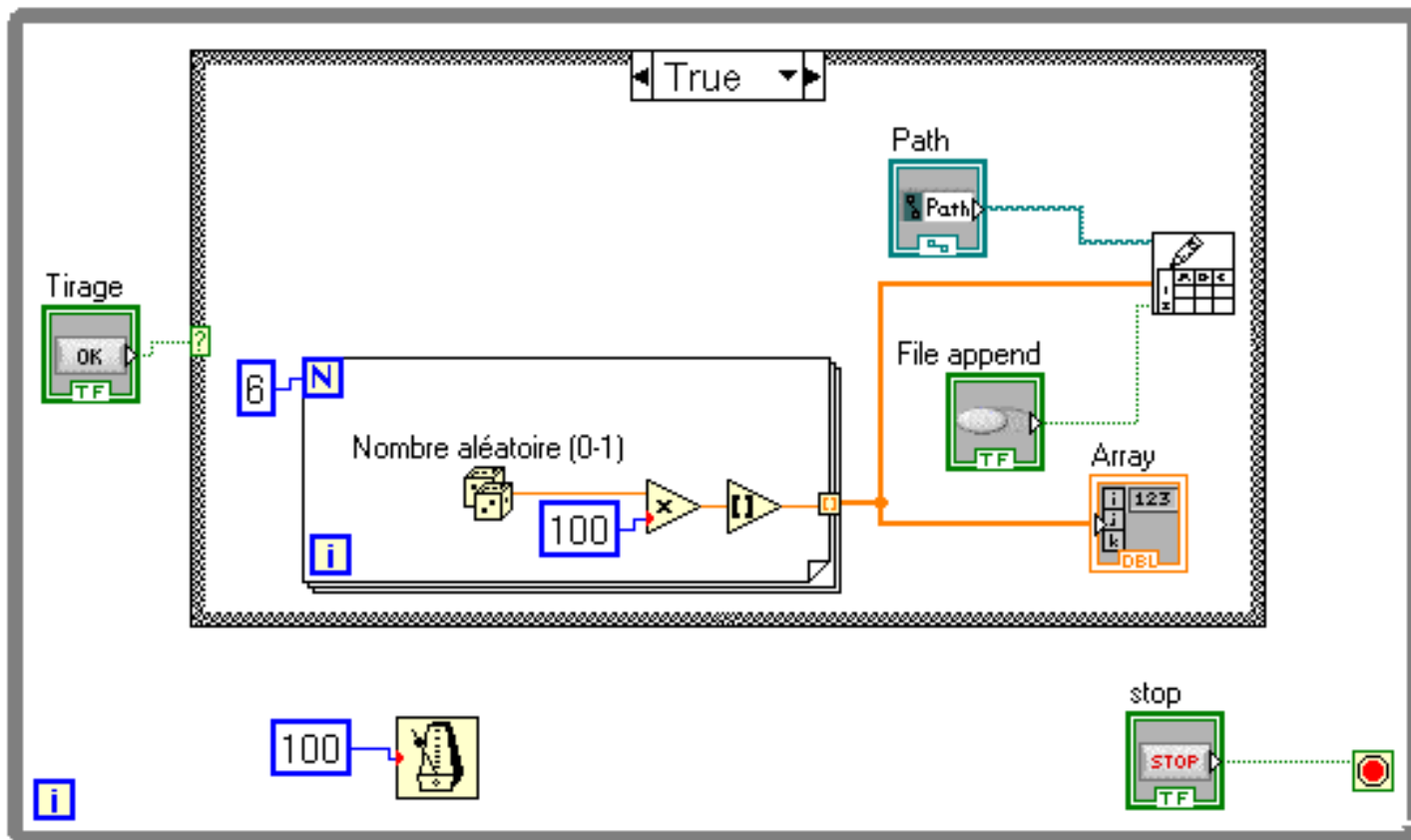
Exercice

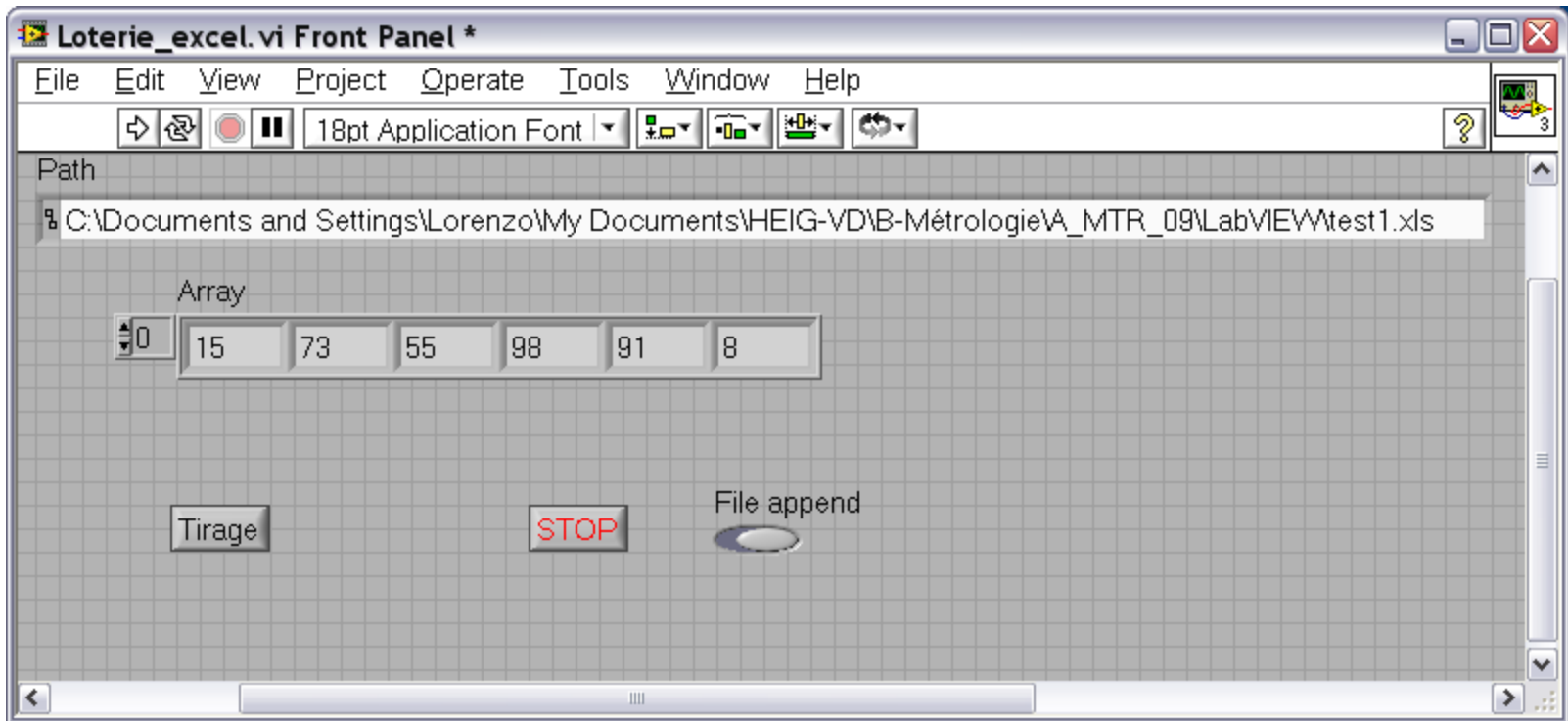
- Reprenez le programme de conversion de température de degrés Celsius en degrés Farenheit, mais réalisez-le à l'aide d'un Formula Node.



Exercice

- Reprendre le programme de loterie (slide 53 de 1^{ère} 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).