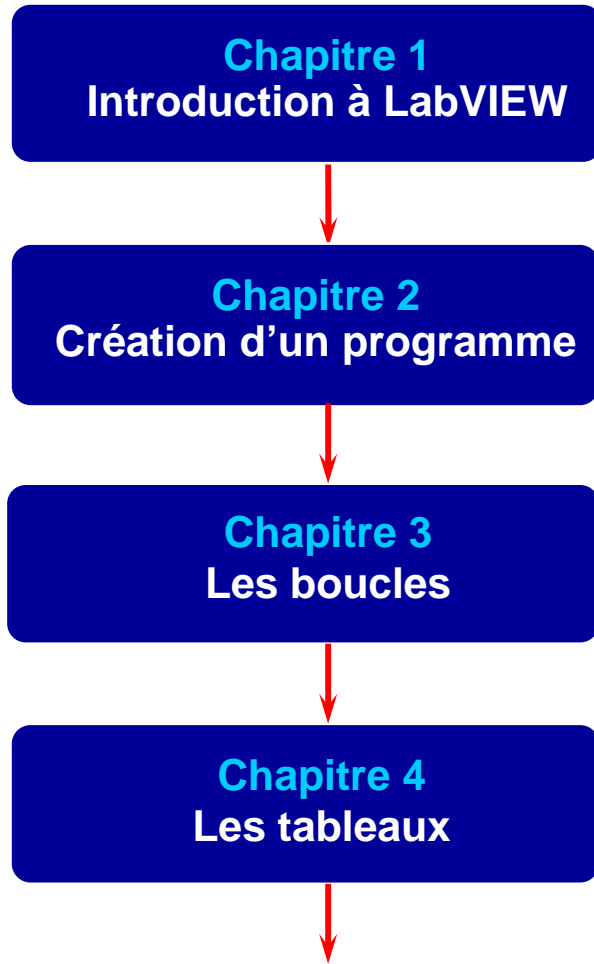


# Introduction à LabVIEW



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

# Intro LabVIEW - 1<sup>ère</sup> partie



# National Instruments



30 Years of National Instruments Excellence  
20 Years of LabVIEW Innovation  
Join in Our Celebrations During 2006 and  
Discover National Instruments History,  
Culture, and Vision.

<http://www.ni.com>

National Instruments Switzerland

Sonnenbergstrasse 53

CH-5408 Ennetbaden

Tél :056 200 51 51

Fax:056 200 51 55



National Instruments Suisse

Fondation : 1990

Directeur : Michael Dams

Fondé de pouvoir : Christian Moser

Employés : 20

Siège : Ennetbaden/Zürich

National Instruments Corporation

Fondation : 1976 à Austin, Texas/USA

Président/CEO : Dr James Truchard

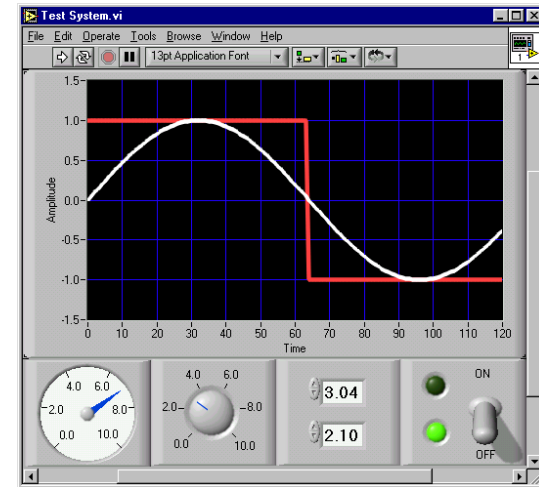
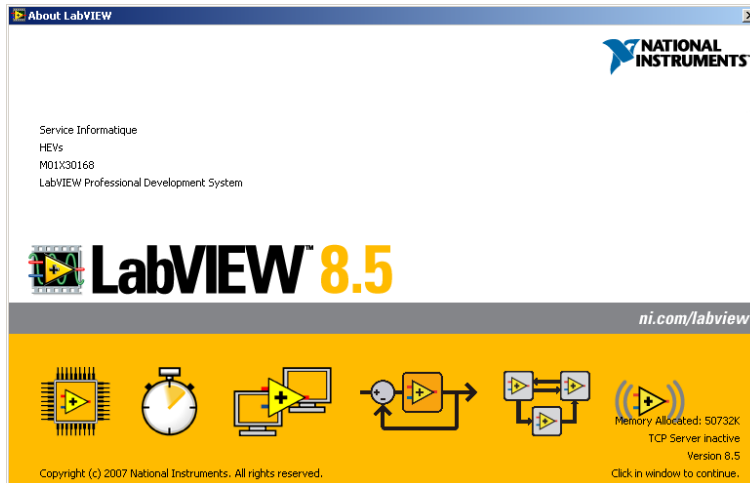
Employés : environ 4600

Siège social : Siège principal / recherche à Austin.

Production : Debrecen (Hongrie)

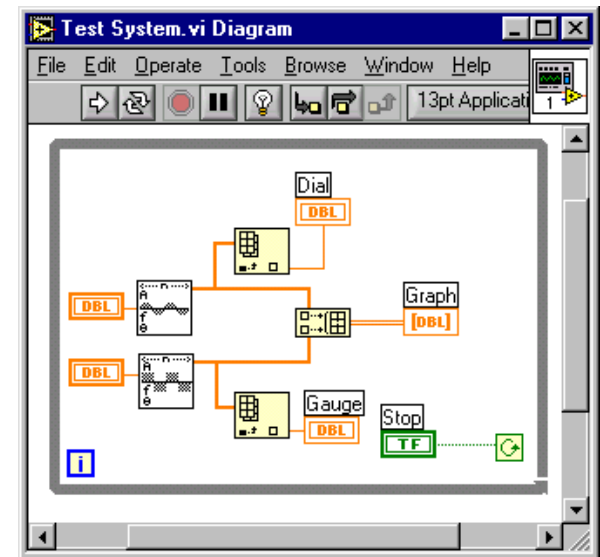
Filiales : National Instruments a plus de 40 filiales dans le monde entier.

# Programmation graphique



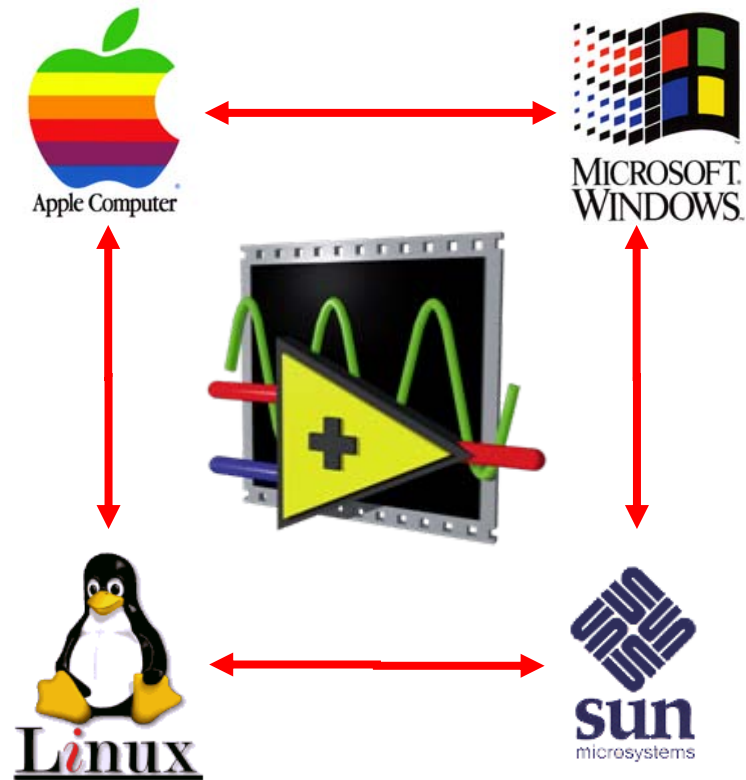
## Facile à programmer

- Temps de développement réduit
- Interface utilisateur graphique
- Code source sous forme graphique
- Programme facilement modulable
- Application Builder pour créer des exécutables

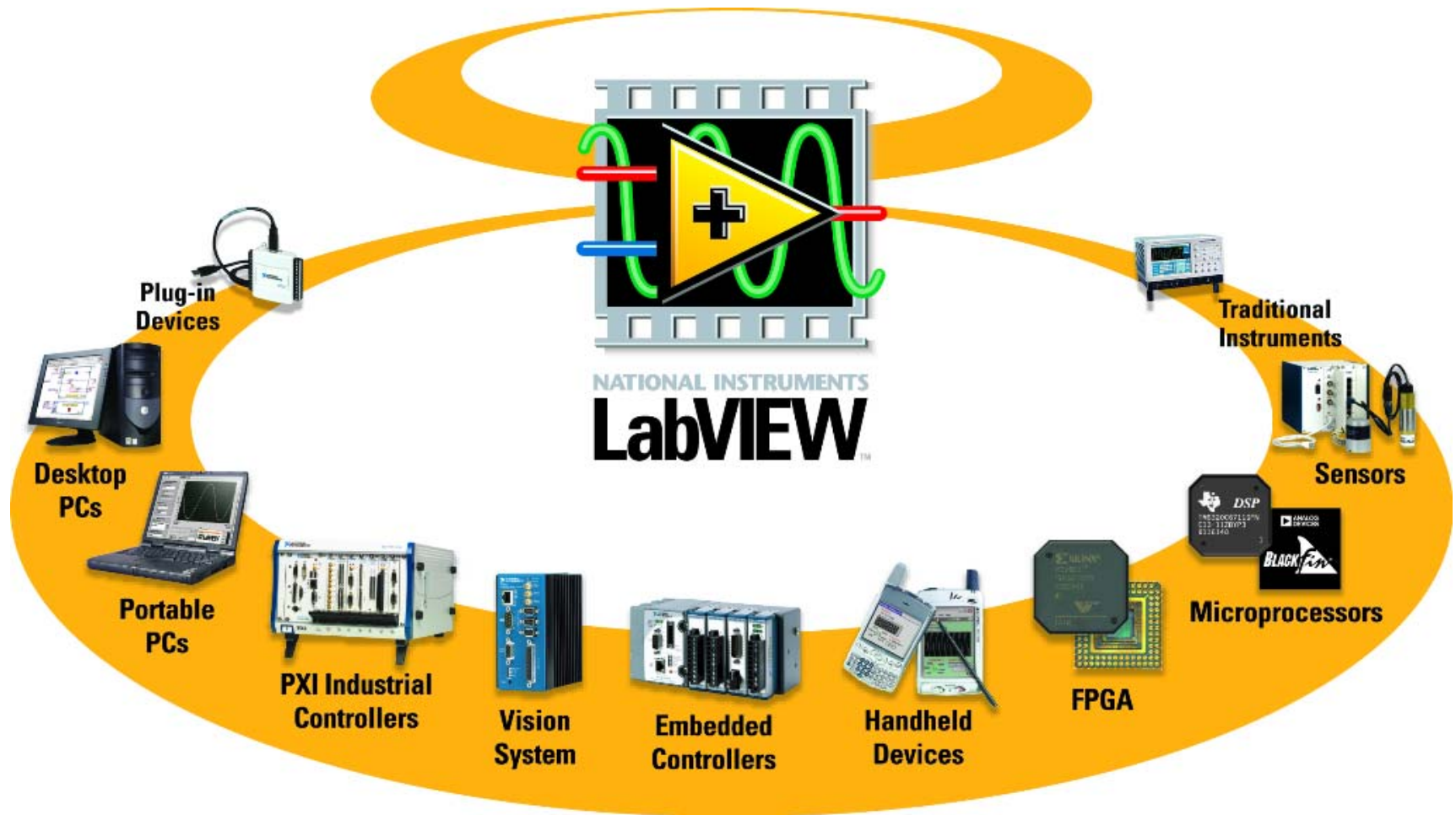


# Compatibilité Multi-Plateformes

- Programmation indépendante de la plateforme utilisée !
- Migration possible des applications entre plateformes



# Cibles matérielles LabVIEW





# Chapitre 1

## Introduction à LabVIEW

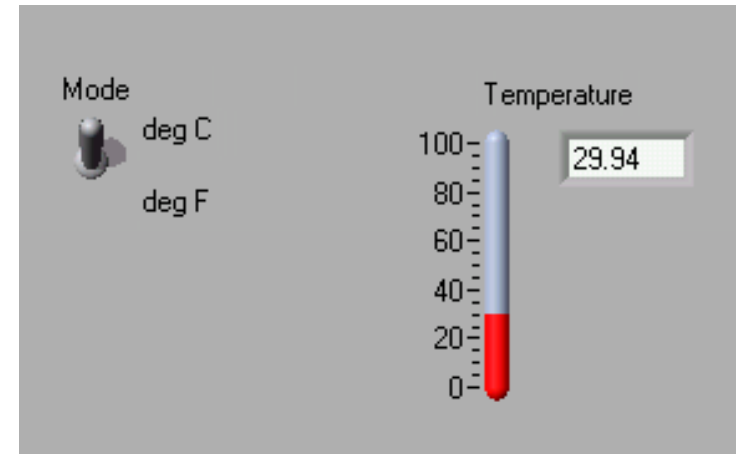
### Thèmes :

- A. Ce qu'est un 'Instrument Virtuel' (VI)
- B. L'environnement LabVIEW
- C. Les options d'aide LabVIEW

# Instruments Virtuels (VIs)

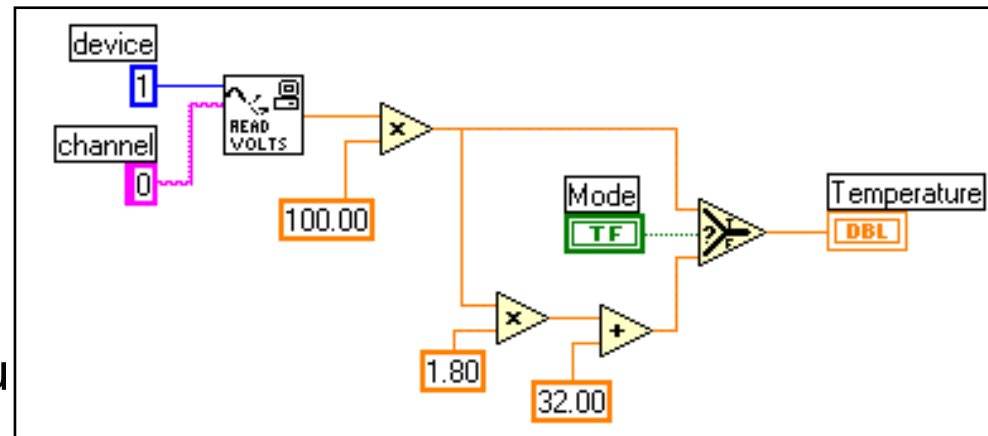
## ■ Front Panel (interface utilisateur)

- Controls = Entrées
- Indicators = Sorties



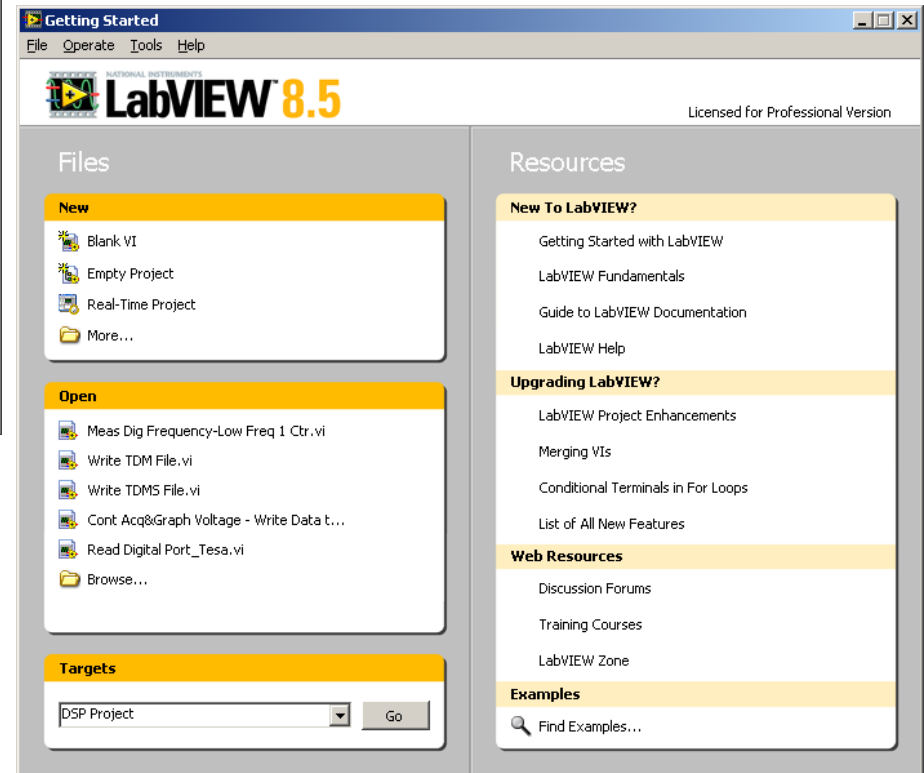
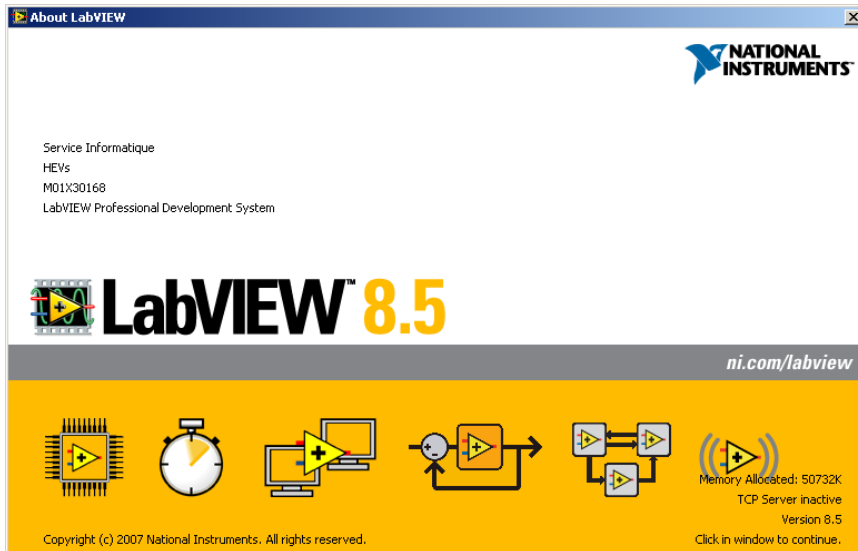
## ■ Block Diagram (écran de programmation)

- Programme correspondant au panel
- Composants "câblés" ensemble

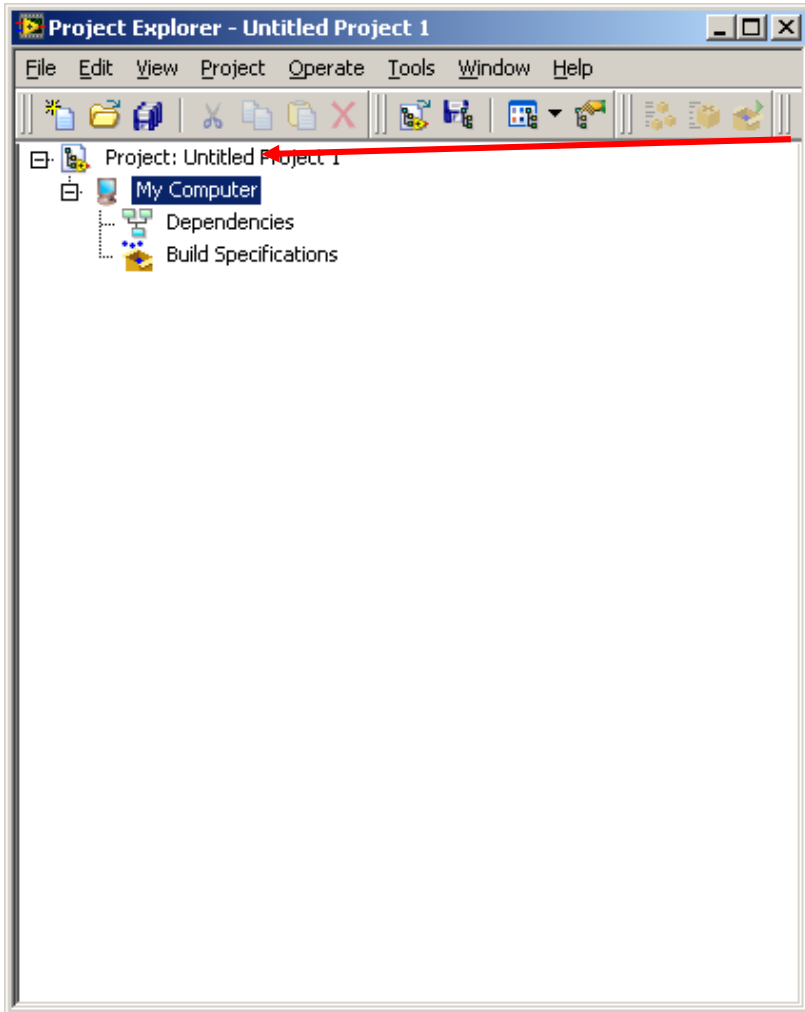




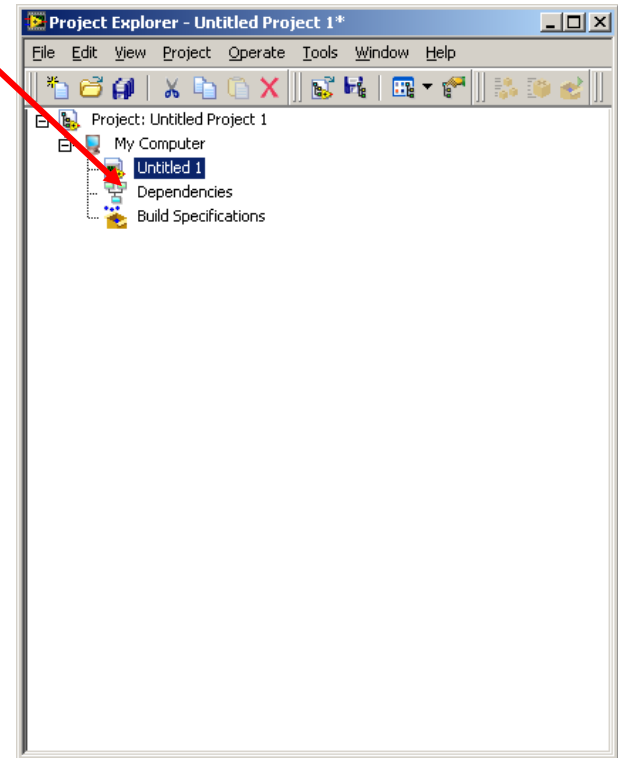
# Démarrage LabVIEW



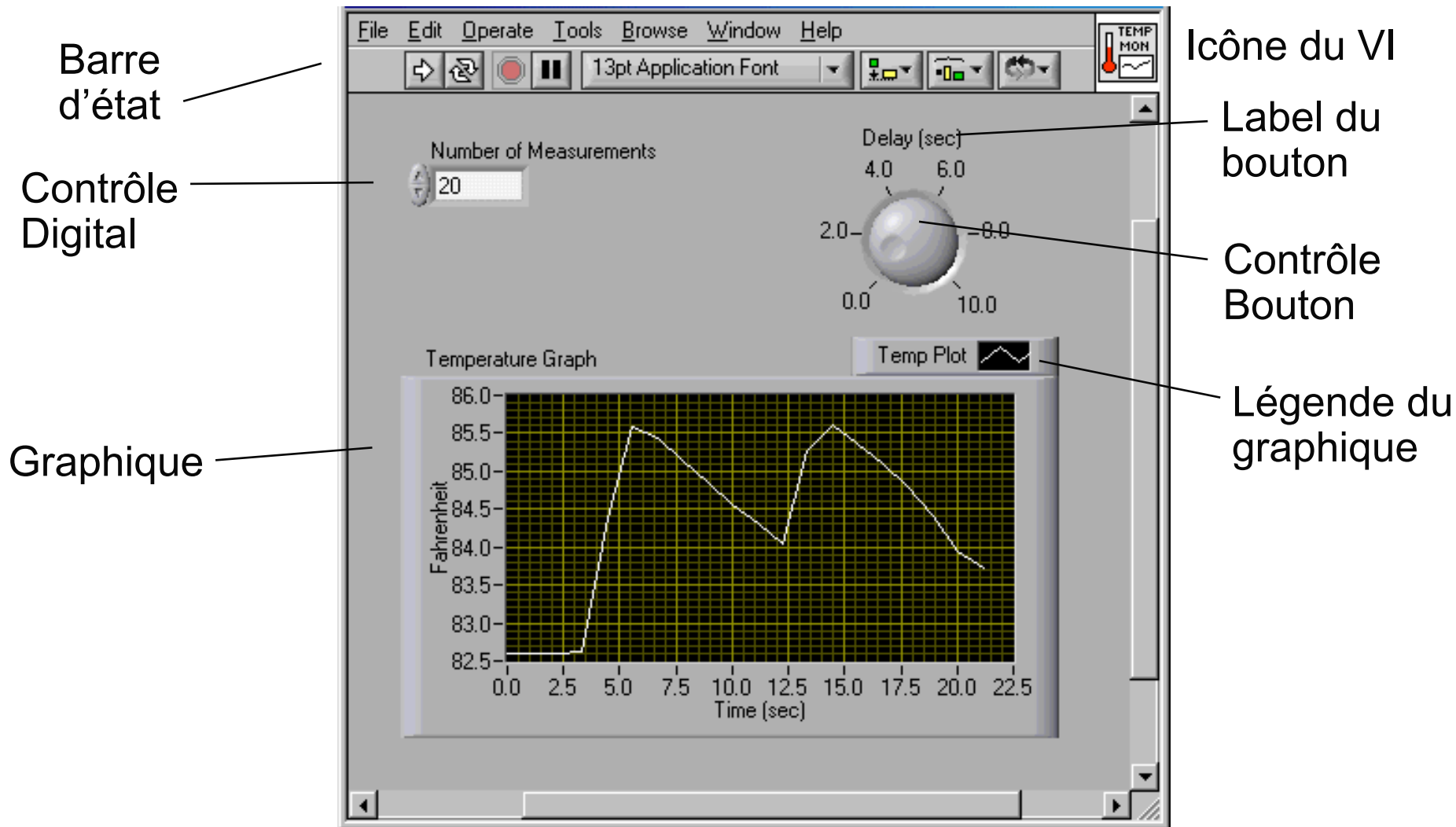
# Project LabVIEW (\*.LVproj)



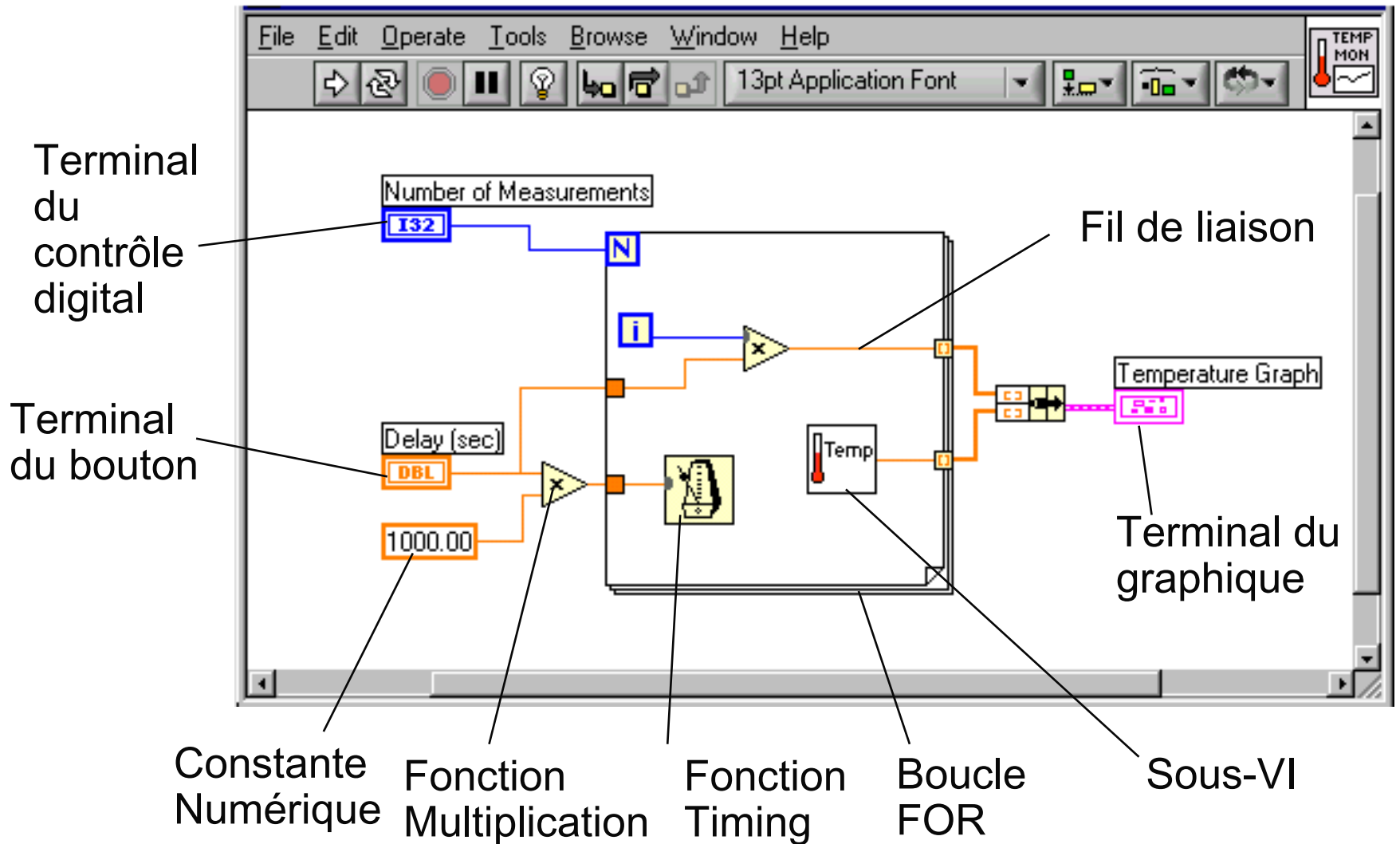
Right Click on my computer and  
NEW VI





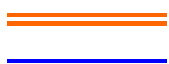








# Interface utilisateur (Front Panel)



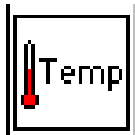
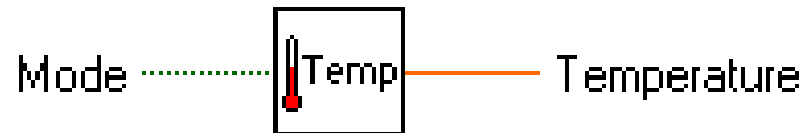
# Ecran de programmation (Diagram)



# Les types de données

	Scalaire	Tableau 1D	Tableau 2D	
Numérique(Numeric)				Orange (real) Bleu ( integer)
Booléen (Boolean)				Vert
Chaîne de caractères (String)				Rose
Enregistrement (cluster)				Brun
Nom de fichier (File Path)				Vert foncé

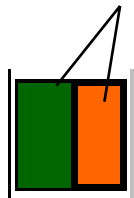
# Icône/Connecteur



icon

- L' icône représente un VI dans un autre 'block diagrams'

terminals



connector

- Le connecteur permet de passer et de recevoir des données d'un "sous-VI" au moyen de terminaux.

# Barre d'état



1. Bouton Run
2. Bouton Run en continu
3. Bouton STOP
4. Bouton Pause/Continue

Menu déroulant polices de caractères

Menu déroulant pour l'alignement

Menu déroulant pour la distribution

Menu pour ordre d'apparence



Boutons additionnels dans le Diagram

1. Execution Highlighting button
2. Step Into button
3. Step Over button
4. Step Out button



Bouton Enter

# Palette d'outils



- Outils d'édition et de débogage
- Palette flottante

- Sélection automatique des outils
- Outils de manipulation
- Outils de positionnement, redimensionnement
- Outils d'écriture
- Outils de câblage
- Outils 'pop-up' menu
- Outils de défilement
- Outils 'Breakpoint'
- Outils sonde de débogage
- Outils de copie de la couleur
- Outils de coloriage



# Palette de Contrôle et de Fonctions

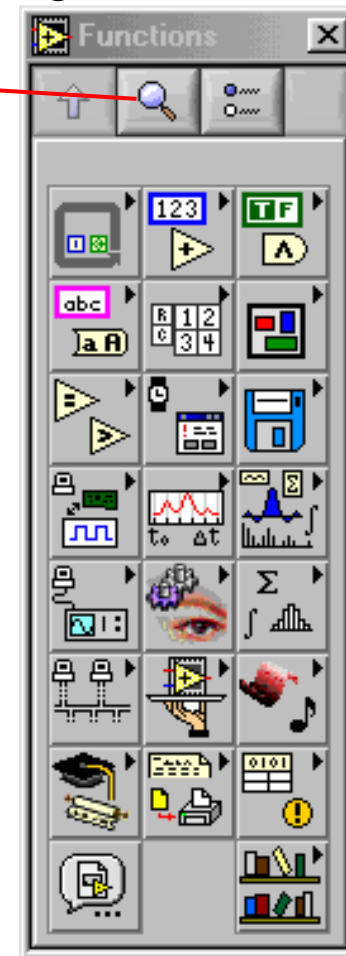
Palette de contrôle  
(Panel Window)

Remonter  
D'un niveau



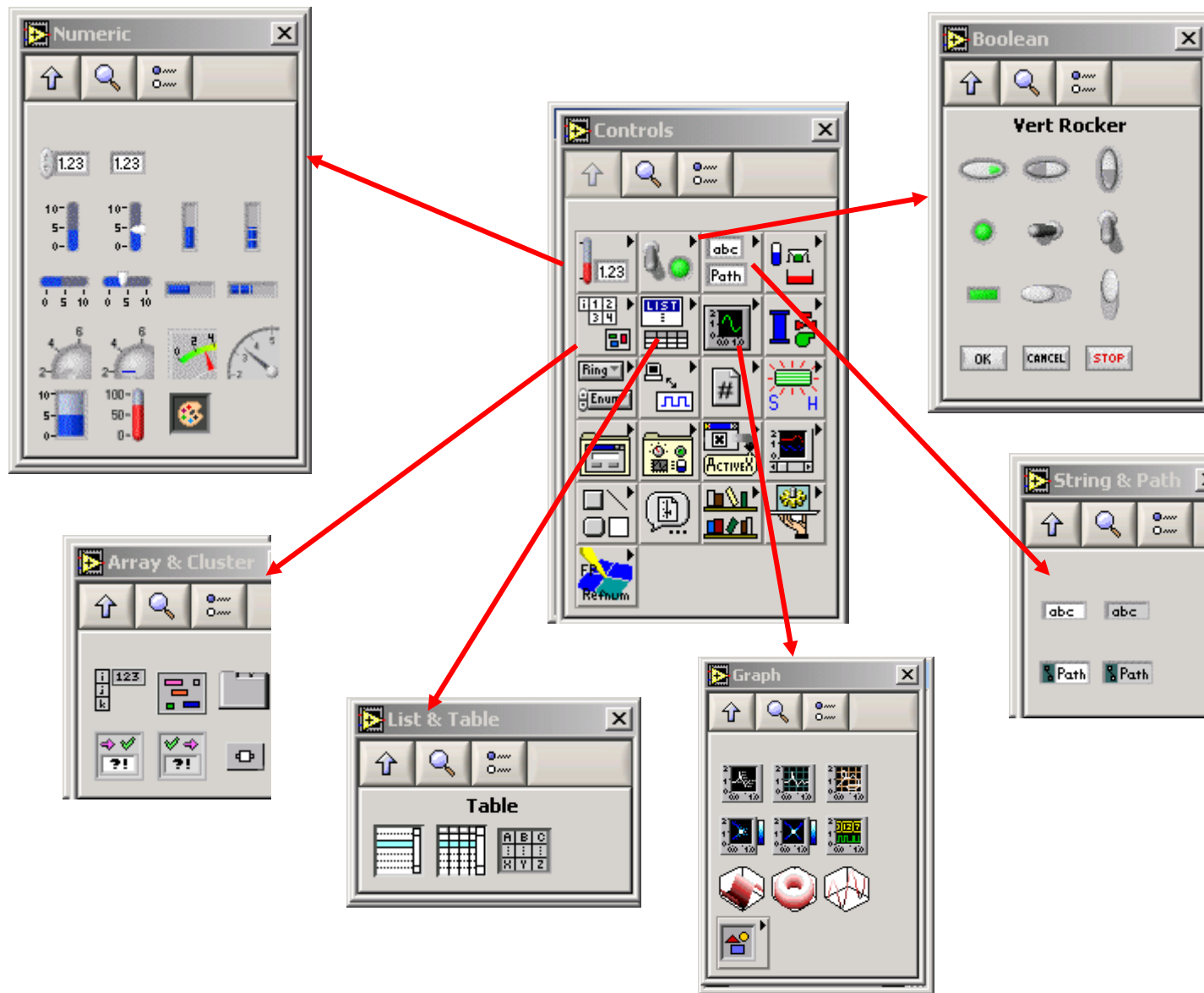
Palette de Fonctions  
(Diagram Window)

Rechercher

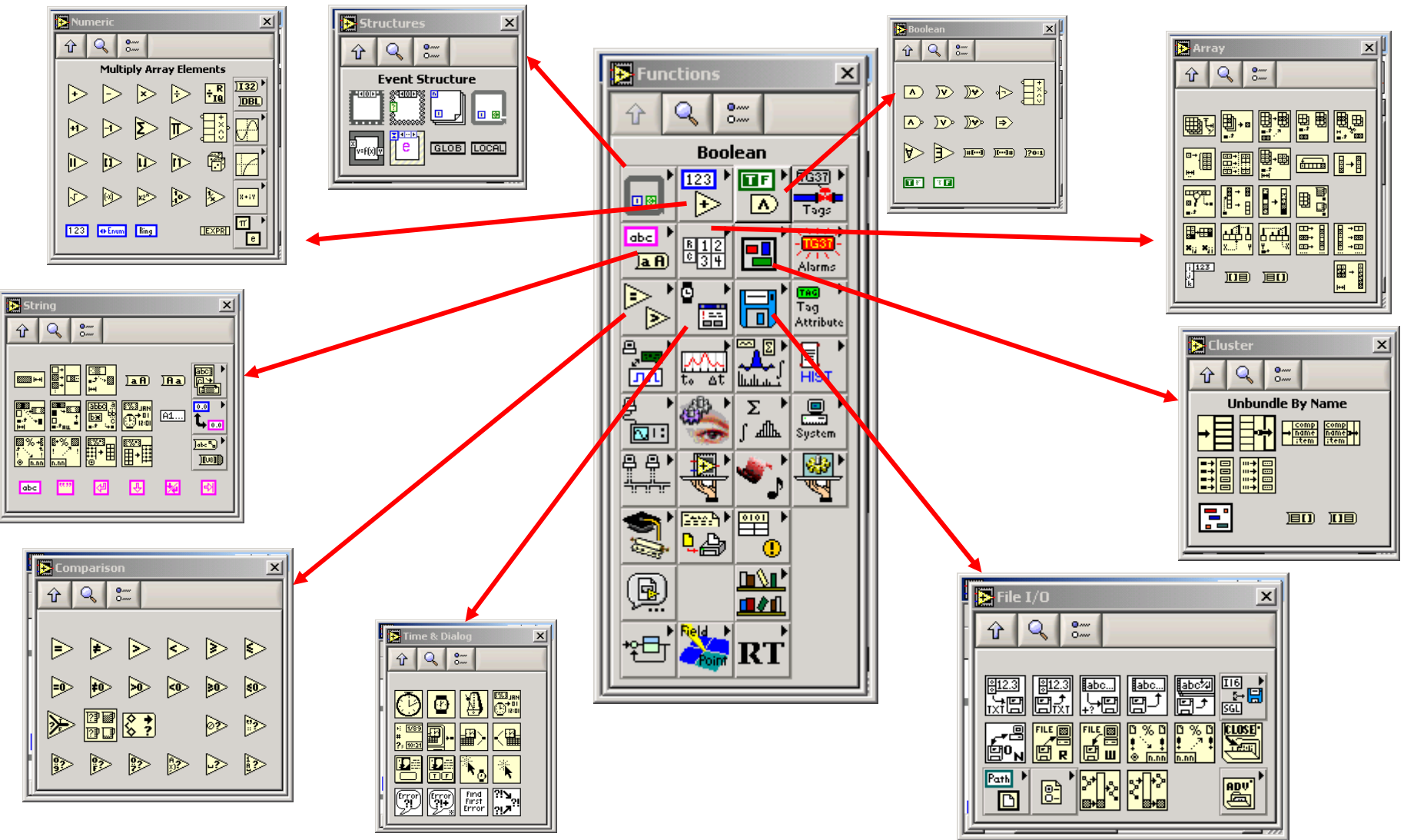


- Palettes graphiques, flottantes
- Les sous-palettes peuvent devenir aussi flottantes

# Palette de Contrôle en détail

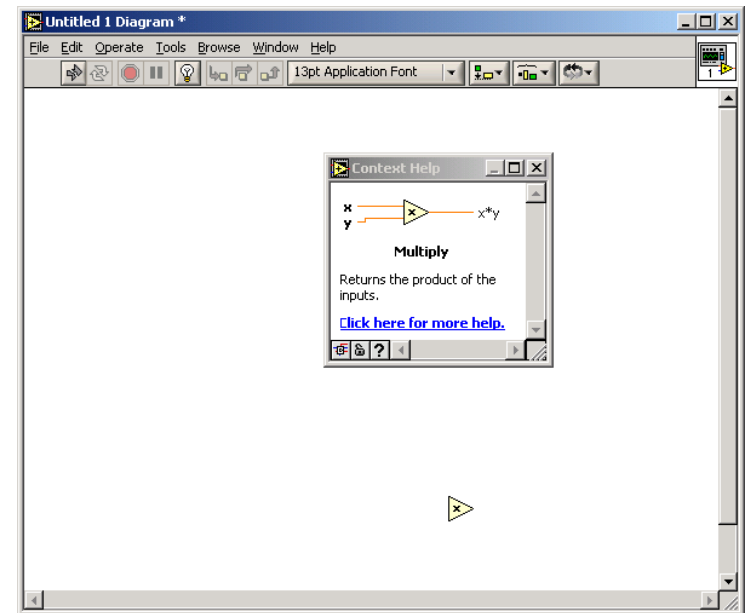
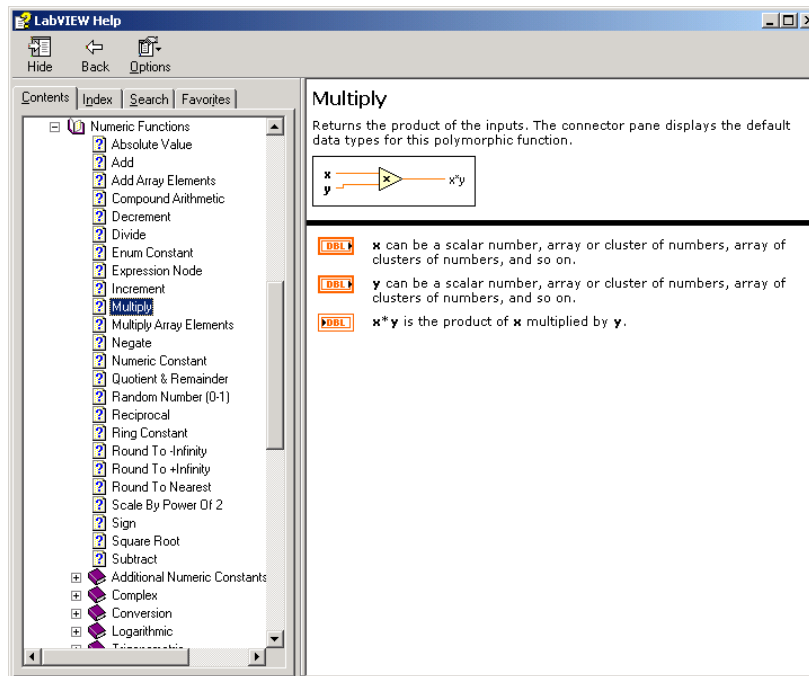


# Palette de fonctions en détail



# Help !

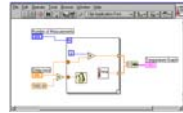
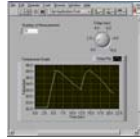
- Afficher le menu d'aide contextuel (Help menu) <ctrl+h>
  - Affiche d'aide Simple/Détaillé
  - Blocage de l'aide sur un élément
  - Aide online (manuels de référence)



# Résumé du chapitre 1

- Les instruments virtuels (Vis) sont composés de 3 parties principales :

- le front panel,
- le block diagram
- l'icône/connecteur



- Le front panel est l'interface utilisateur d'un programme LabVIEW.
- Le block diagram est le code exécutable.
- Click de droite et menu contextuel pour modifier les propriétés de chaque objet.
- Palettes flottantes :
  - Palette des outils
  - Palette des Contrôles (seulement dans le front panel)
  - Palette de fonctions (seulement dans le Diagram)
- Possibilité d'avoir de l'aide contextuelle en ligne...(ctrl+h)



# Exemple

Création du premier petit programme qui va additionner deux nombres et afficher le résultat.

The image shows a graphical programming environment with two windows. The left window, titled "Diagramme de Sans titre 1 \*", displays a flowchart with two input blocks labeled "IN-1" and "IN-2", each containing the number "123". These inputs are connected to a central addition block (+), which is then connected to an output block labeled "OUT" containing the number "123". The right window, titled "Face-avant de Sans titre 1 \*", shows a similar setup but with the input values changed to "4" and "1", and the output value changed to "5". A "Fonctions" (Functions) panel is open on the right, showing a search bar and a list of functions under the "Programmation" > "Numérique" category. The functions include basic arithmetic operators (+, -, \*, /, %), mathematical functions (sqrt, log, exp, etc.), and other numerical operations. The "Fonctions" panel also includes a search bar and a list of functions under the "Programmation" > "Numérique" category.

# Chapitre 2

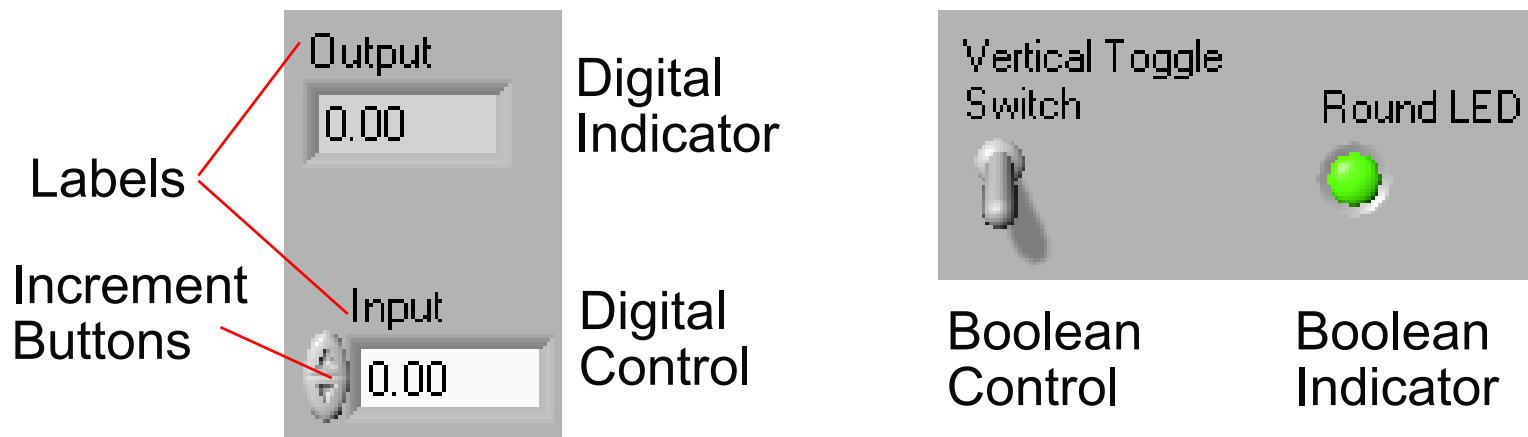
## Création/édition et debugging d'un VI

### Thèmes:

- A. Comment créer des VIs
- B. Comment éditer des VIs
- C. Comment déboguer des VIs

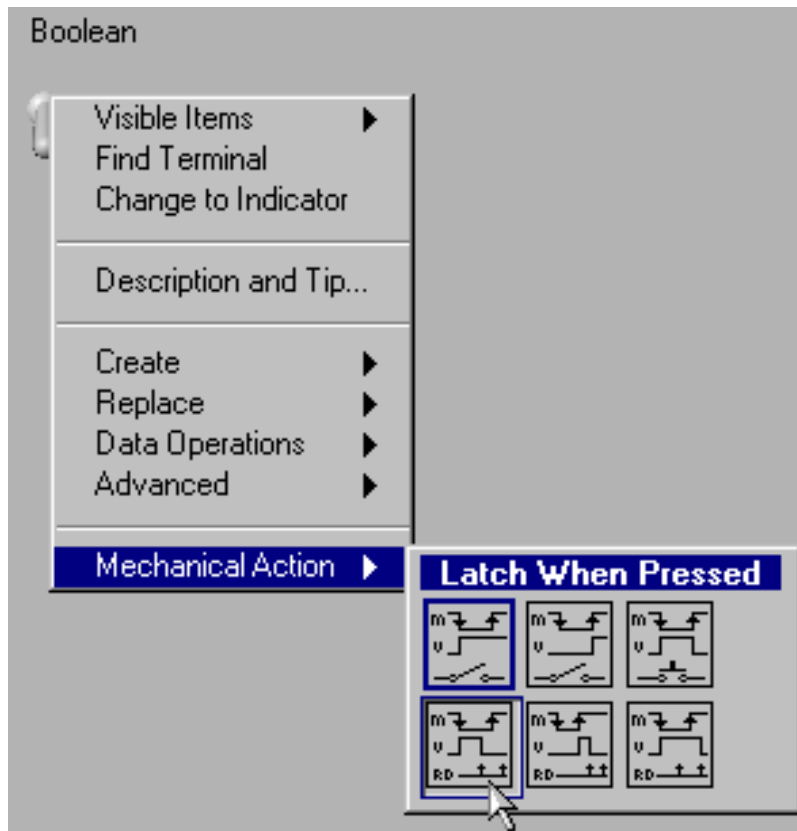
# Creation de l'interface utilisateur (Front Panel) d'un VI

- Contrôles et indicateurs numériques
- Contrôles et indicateurs booléens
- Configuration des contrôles et indicateurs
  - Utilisation des menus raccourcis
  - Les différents menus des Objets





# Action mécanique des Booléens



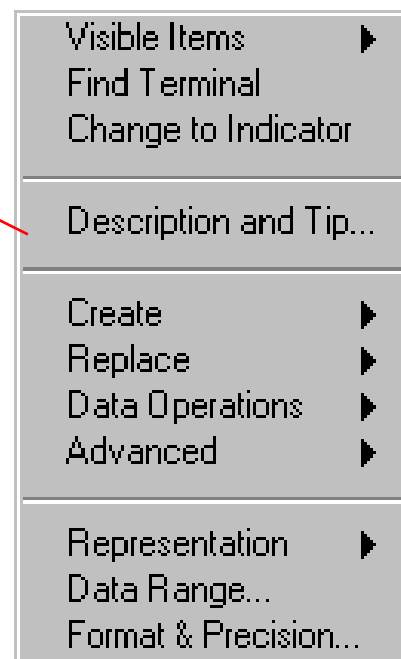
- Switch action: Le contrôle reste dans son état jusqu'au prochain changement à la main.
- Latch action: Le contrôle revient à son état par défaut dès lecture dans le diagram

# Acces aux menus contextuels

Click de droite sur le label pour afficher son menu contextuel



Click de droite sur l'affichage digital pour afficher son menu contextuel

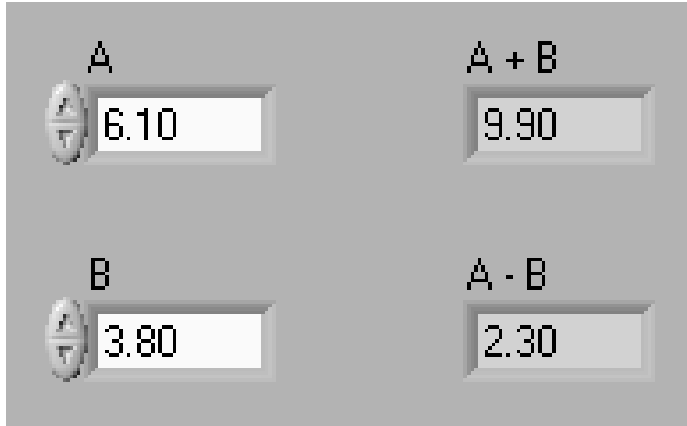


**Windows & UNIX: touche-soiuris de droite**

**MacOS: Control-Click**

# Creation du Block Diagram d'un VI

Interface utilisateur (Control Panel)

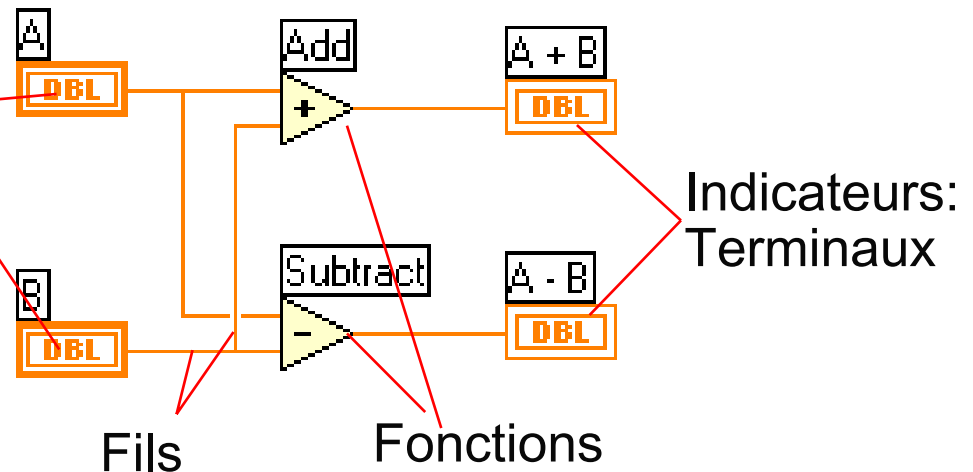


Code (Diagram Window)

Contrôles :  
Terminaux

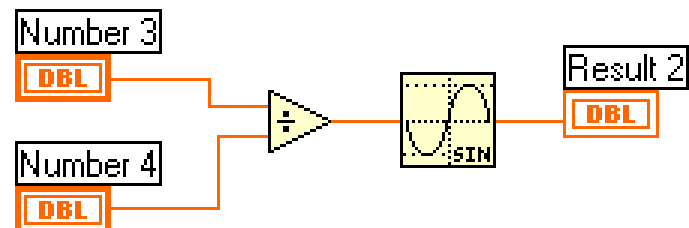
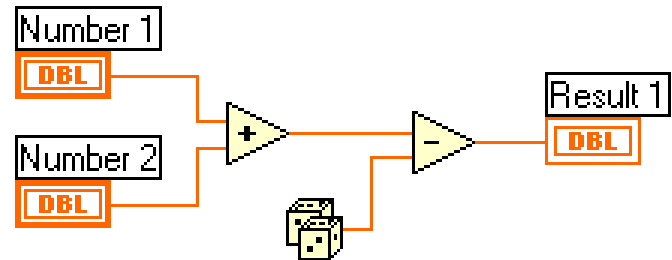
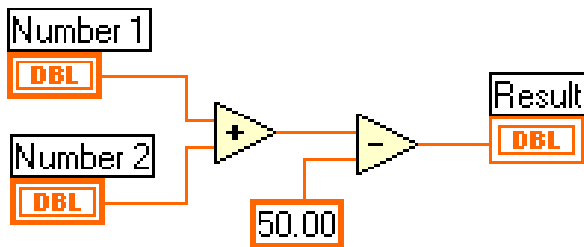


Aspect du Terminal pour la  
Fonction Add et la fonction  
Subtract (3 noeuds).

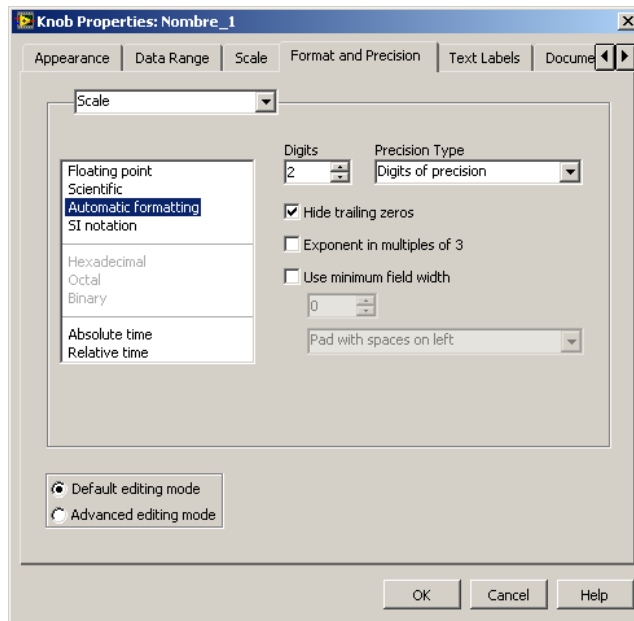


# Programmation selon le flux de données

- Les Block diagram ne s'exécutent **PAS** nécessairement de gauche à droite !
- Une fonction s'exécute uniquement lorsque elle a reçu **TOUTES** ses entrées !
- Une fonction fournit **TOUTES** ses sorties lors de son exécution !

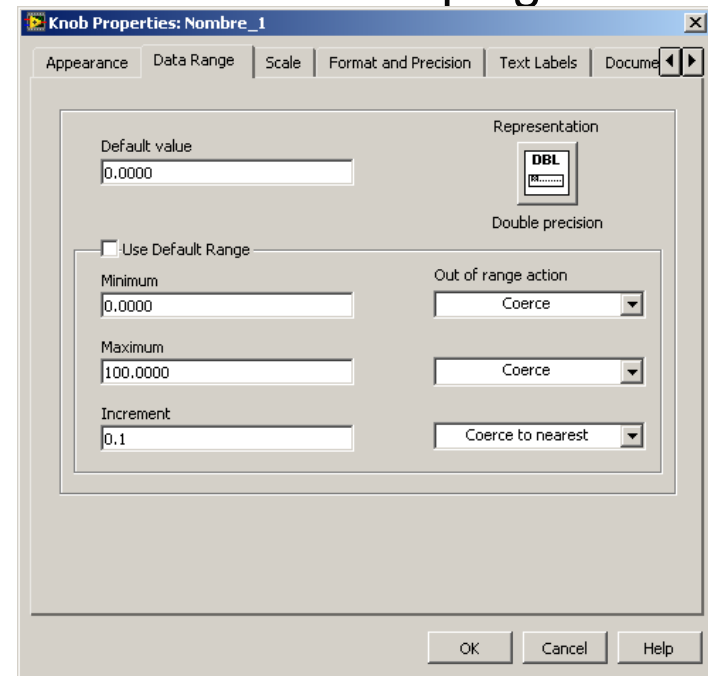


# Modifications des Contrôles & Indicateurs numériques



- Définition des digits de précision

## • Définition de la plage

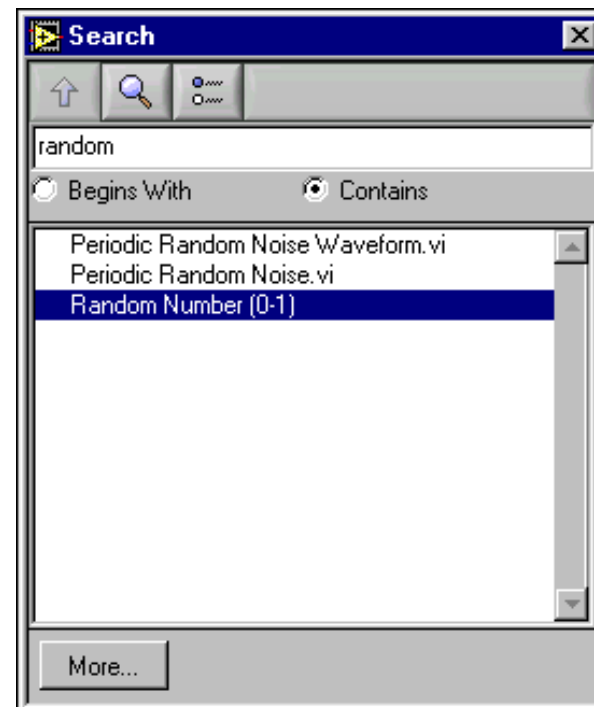


# Recherche de Contrôles, VI, Fonctions



- Utilisation des boutons se trouvant au sommet de la palette des Contrôles et des fonctions pour naviguer et chercher des contrôles et des fonctions spécifiques.

Appuyez sur le bouton search pour faire une recherche de texte dans la palette.



# Techniques de débogage

- Trouver les erreurs



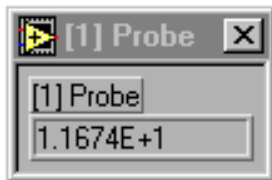
Click sur la flèche cassée Run et une fenêtre Apparaît montrant les erreurs du programme

- Mode debugging



Click sur l'ampoule, le programme tourne au ralenti et le déroulement du programme est animé. Les valeurs des données sont écrites sur les fils

- Sondes



Click de droite sur les fils pour poser une sonde qui permet de visualiser online les données passant dans le fil



Il est aussi possible de poser une sonde depuis la palette des outils (symbole jaune)

# Techniques de débogage

- Points d'arrêts



Sélection de l'outil point d'arrêt depuis la palette des outils et placez-le où le programme doit s'arrêter.

- Opérations pas-à-pas



Chaque click permet d'avancer d'un pas dans le Programme, y compris dans les sous-programmes



Chaque click permet d'avancer d'un pas dans le Programme, sans passer dans les sous-programmes



Permet de ressortir d'un sous-programme



# Résumé du chapitre 2

- Placement des contrôles (inputs) et des indicateurs (outputs) dans le control panel
- Utilisation des outils pour manipuler les objets du control panel. Utilisation de la flèche pour sélectionner, bouger, et redimensionner. Utilisation de la bobine pour connecter les objets dans le diagram.

- Les contrôles ont un terminal à bord plus épais que les indicateurs.

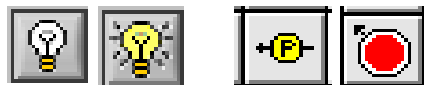


- Tous les objets LabVIEW ont un menu contextuel
- Le programme est réalisé par câblage des objets et fonctions selon le flux de données

- Bouton Run avec la flèche brisée = VI non-exécutable



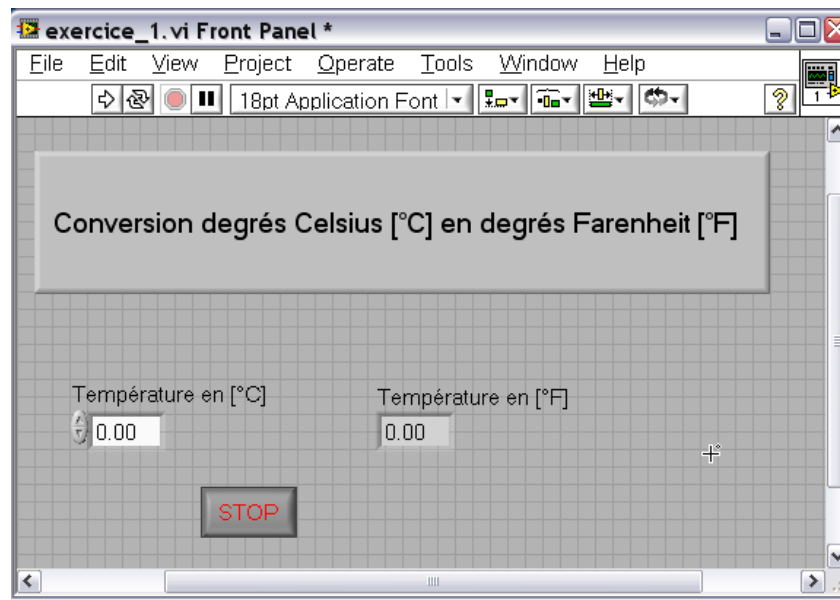
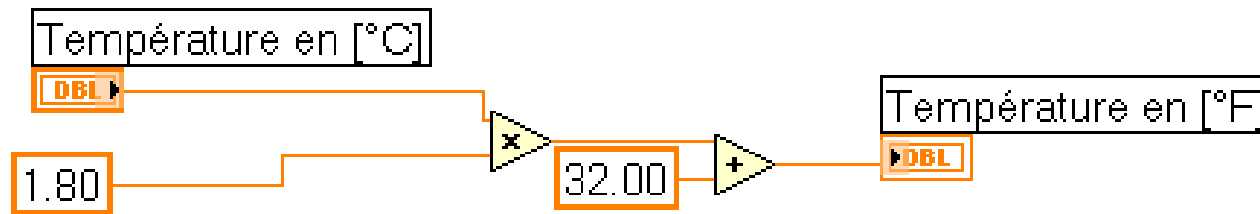
- Quelques outils et options de débogage



# Exercice

1. Réalisez un programme qui permet de convertir une température de degrés Celsius [°C] en degrés Farenheit [°F] et inversement.

$$\text{Deg } [^{\circ}\text{F}] = \text{Deg } [^{\circ}\text{C}] * 1.8 + 32$$

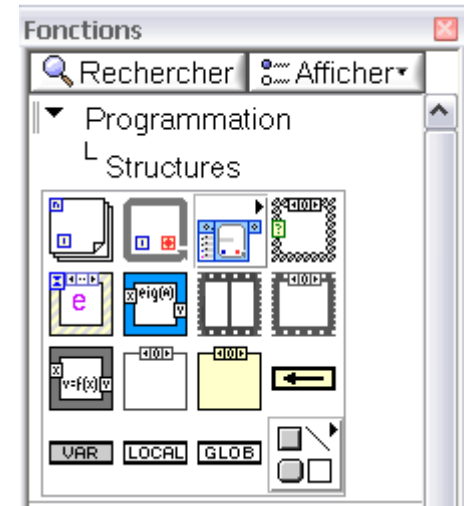


# Chapitre 3

## Conditions et boucles

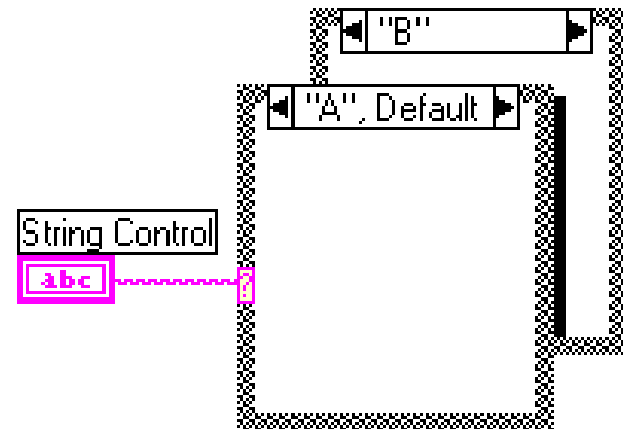
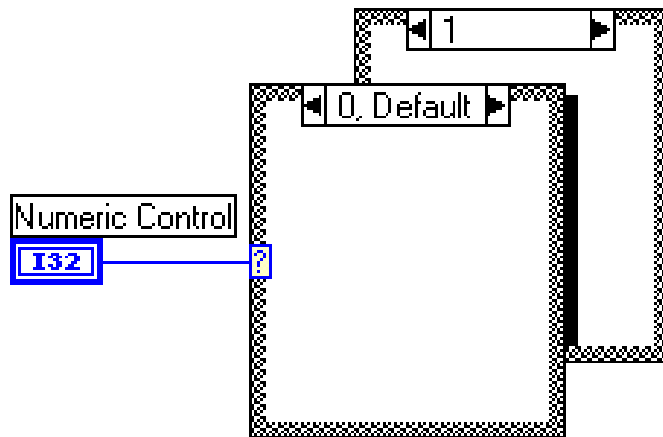
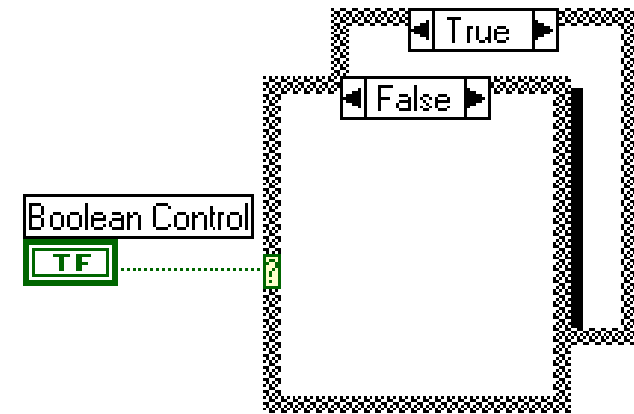
Thèmes :

- A. “Case”
- B. La boucle “While”
- C. Utilisation des registres à décalage
- C. La boucle “For”
- D. Timing pour l’exécution d’une boucle



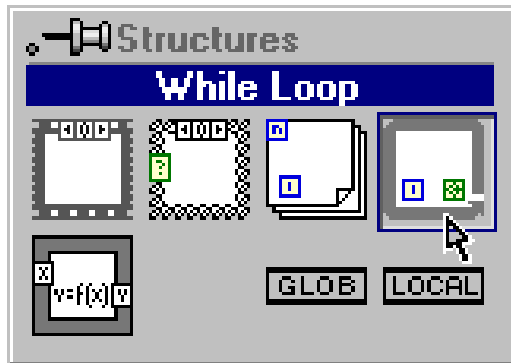
# Case

- 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

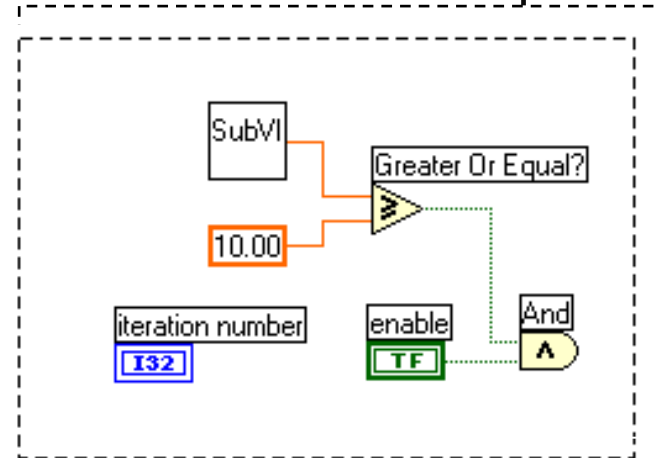


# La boucle While

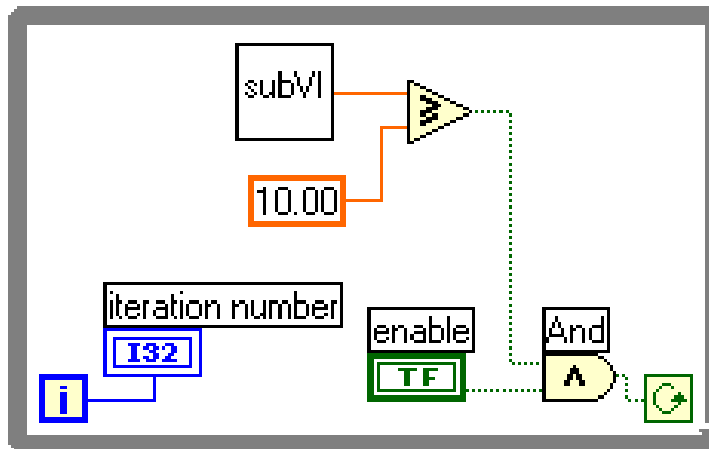
## 1. Sélection boucle While



## 2. Sélection du code à répéter



## 3. Placement du code dans la boucle While



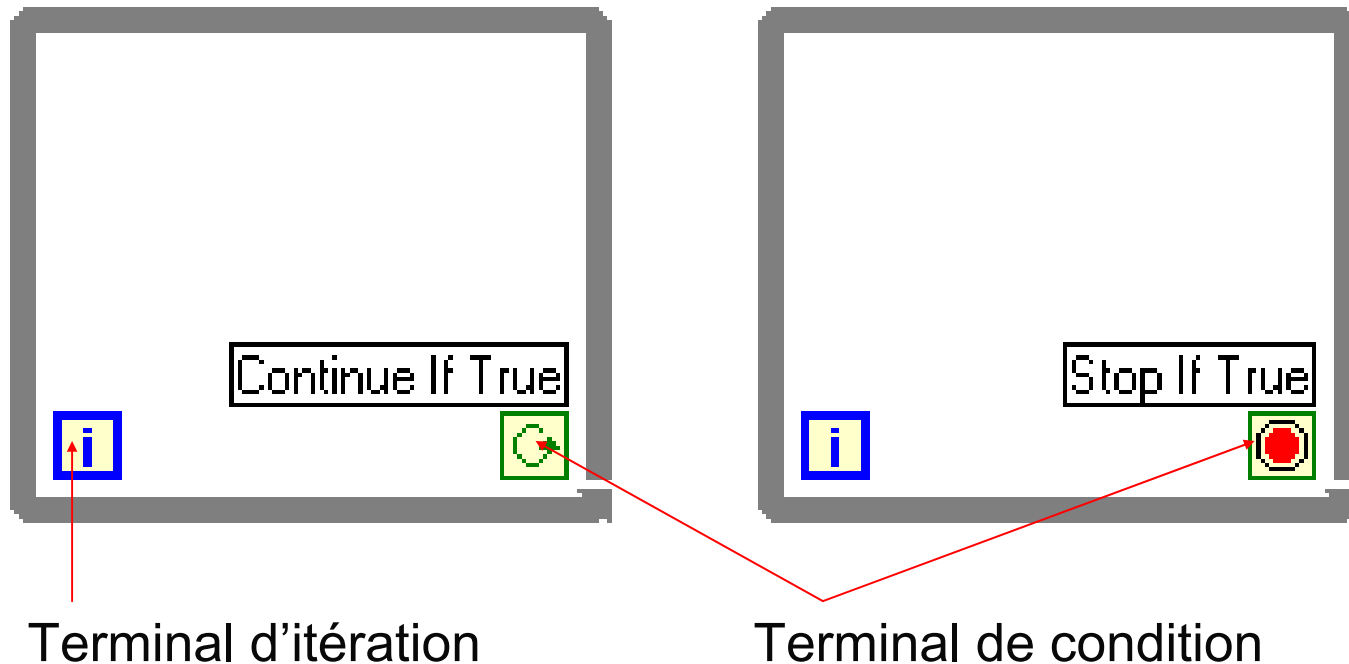
Do

(Execute diagram  
inside Loop)

While Condition is TRUE

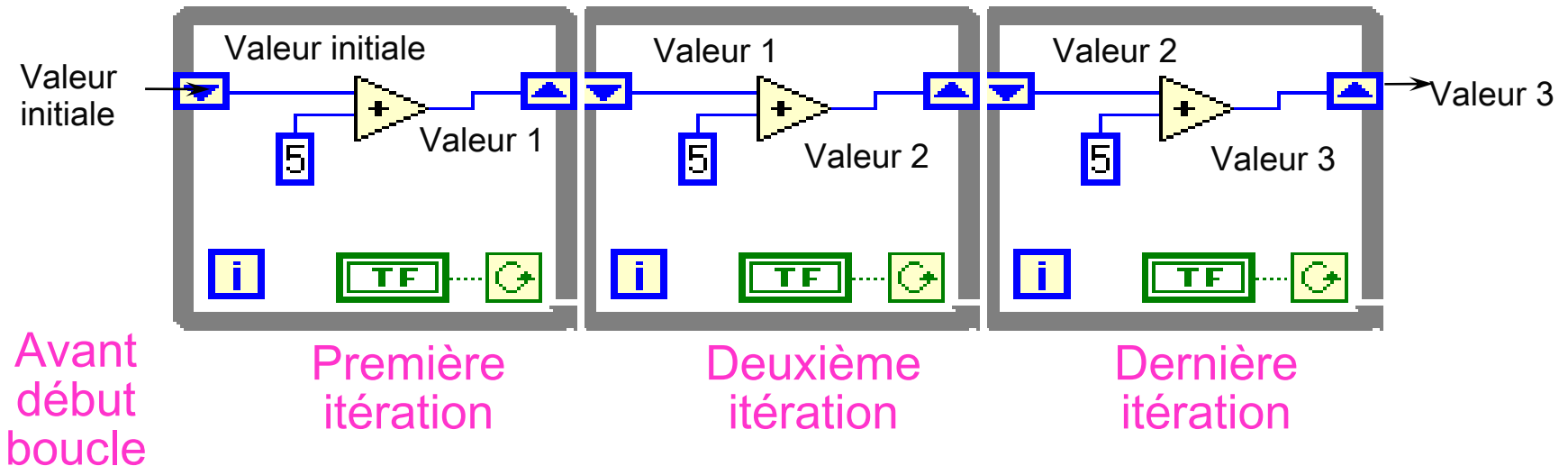
# Choix de la condition de boucle

- Click de droite sur le terminal de condition de la boucle pour définir quand elle doit s'interrompre



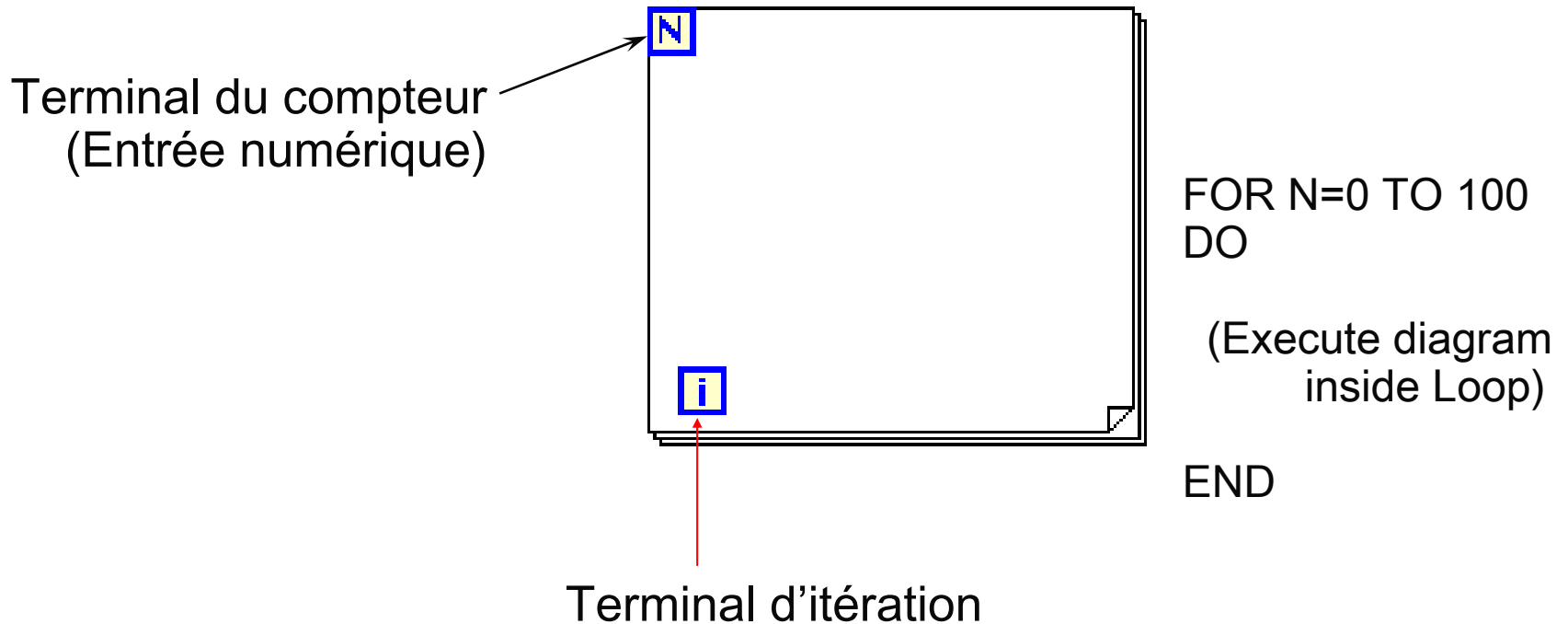
# Registres à décalage

- Disponibles sur les bord droit et gauche des boucles
- Click de droite sur le bord de la boucle et sélectionner Add Shift Register
- Le terminal de droit stocke la valeur après un passage dans la boucle
- Le terminal de gauche met à disposition la valeur stockée avant au début de l'itération suivante



# La boucle For

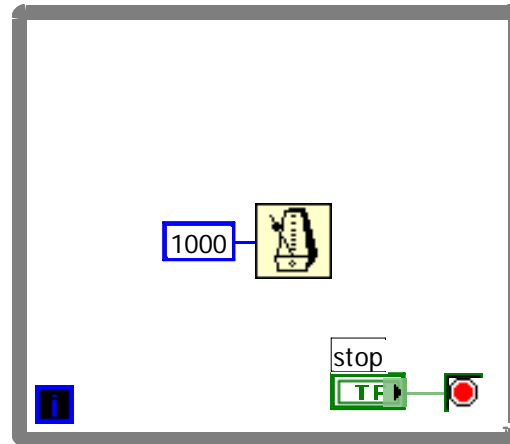
- Exécute le diagramme qui se trouve à l'intérieur un nombre prédéterminé de fois.
- Possibilité de créer des registres à décalage





# Timing d'exécution d'une boucle

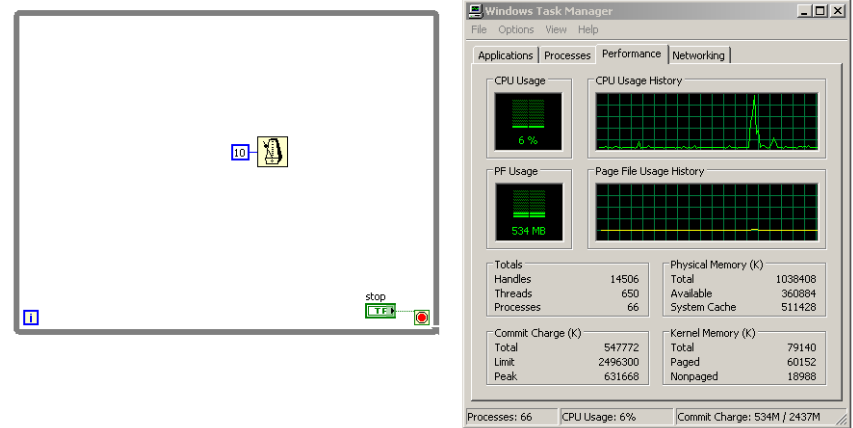
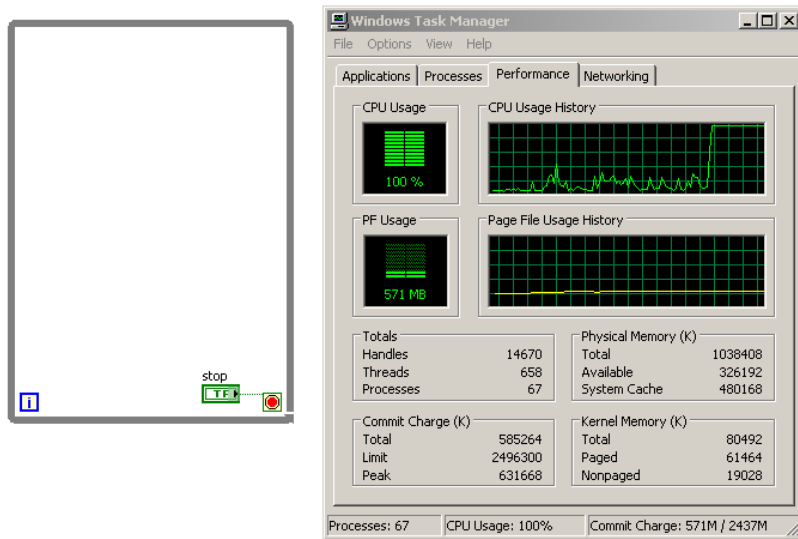
- On peut fixer le temps d'exécution de boucle par une fonction *Wait* (attendre).
- Attention, si le temps est plus petit que le temps nécessaire à l'exécution du programme dans la boucle, c'est le multiple suivant du temps qui sera pris.



- Cette boucle va se répéter chaque seconde

# Timing d'exécution d'une boucle

- Il faut toujours insérer un timing pour cadencer une boucle afin de préserver l'utilisation du processeur du PC

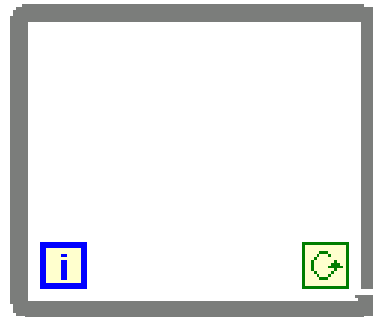


☹ Sans Cadencement de boucle

☺ Avec cadencement de boucle (10 ms)

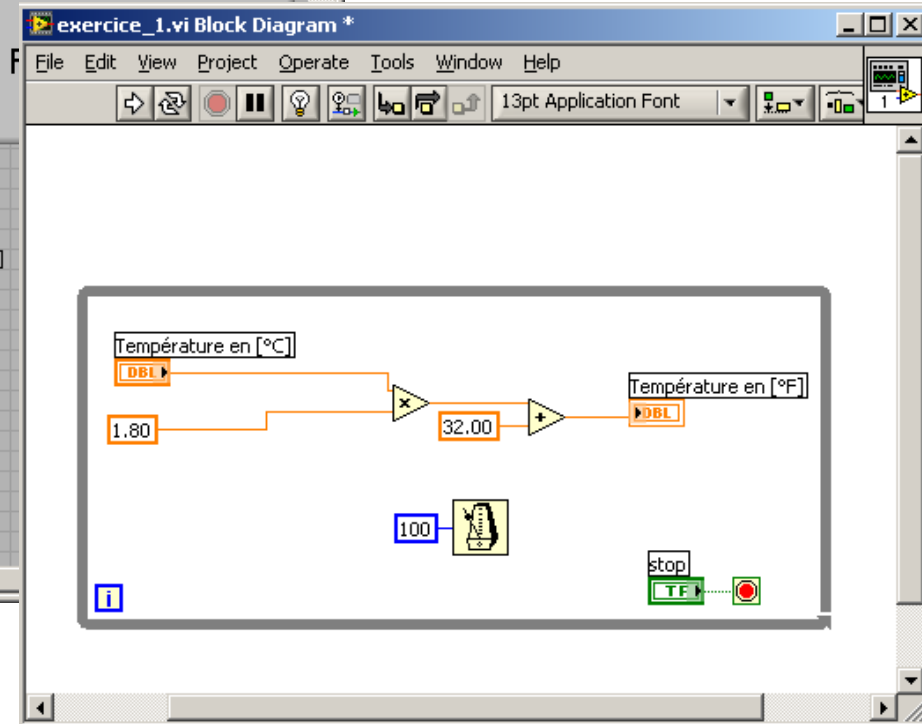
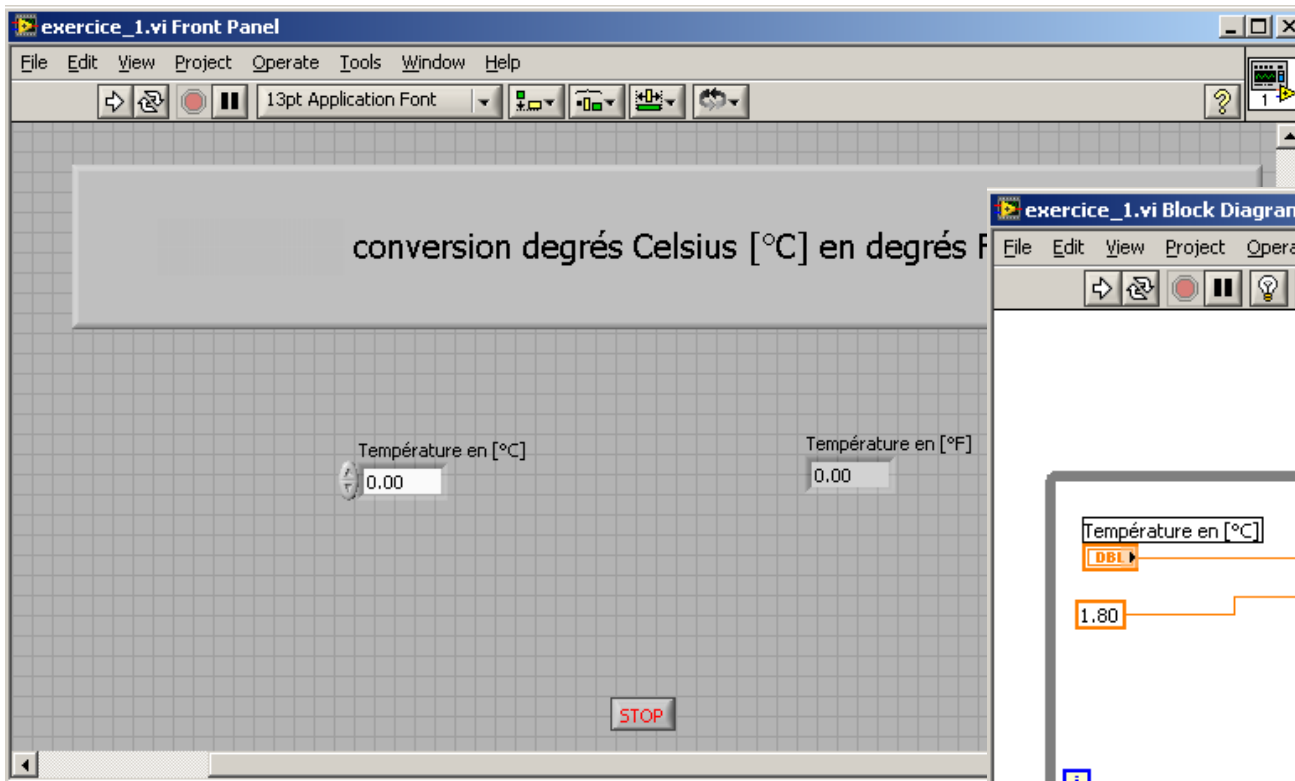
# Résumé du chapitre 3

- Deux types de boucles dans LabVIEW : Boucle While et boucle FOR
- La boucle While : Boucle 'Tant que' : s'exécute tant que la condition de fin n'est pas activée
- La boucle FOR : s'exécute un certain nombre de fois défini au départ
- Le diagramme devant être répété se place à l'intérieur de la boucle
- Utilisation d'un timer pour cadencer l'exécution de la boucle

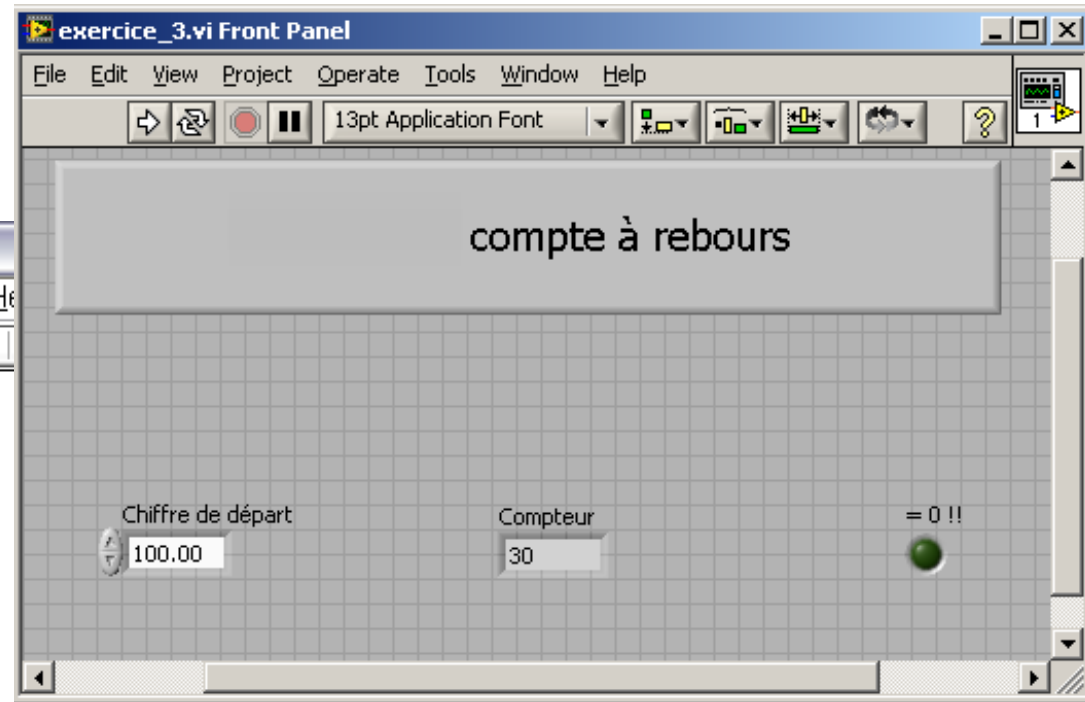
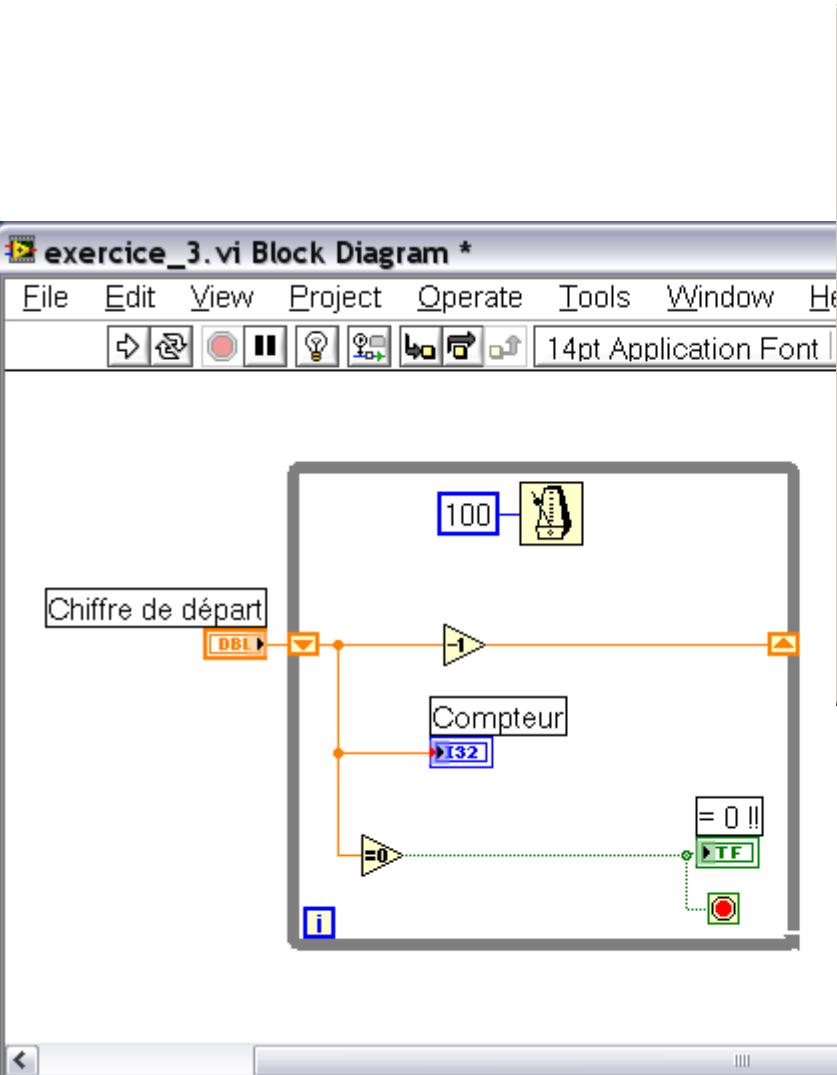


# Exercices

- «Encercler» d'une boucle *while* le programme qui permet de convertir une température de degrés Celsius [°C] en degrés Farenheit [°F], afin de convertir les valeurs sans redémarrer le programme chaque fois.



- Réaliser un programme sous forme de boucle qui permet de décompter de 100 à 0 chaque 100 ms et qui allume une lampe quand le compte à rebours est atteint (valeur = 0).





# 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
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

0 1 2 3 4 5 6

Tableau 2D

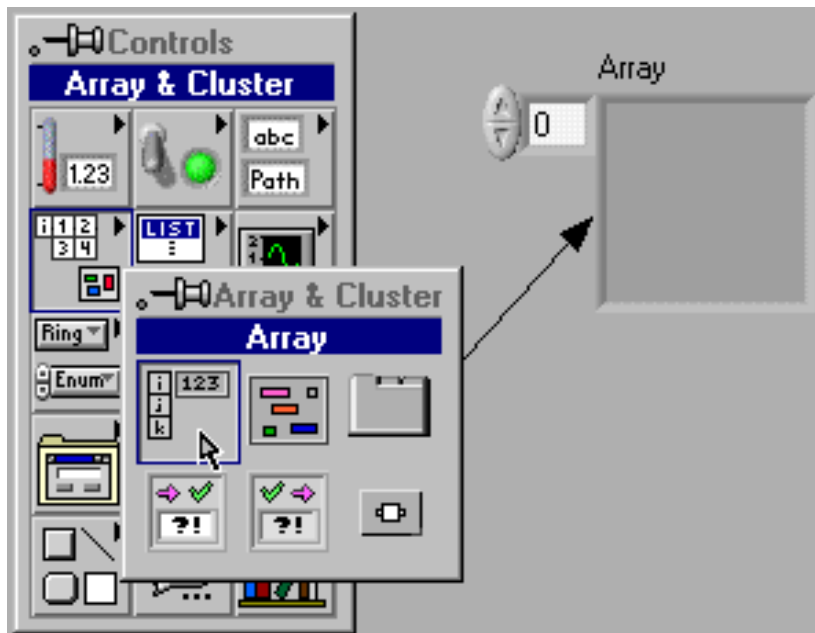
0							
1							
2							
3							
4							

5 lignes par 7 colonnes  
Tableau de 35 éléments

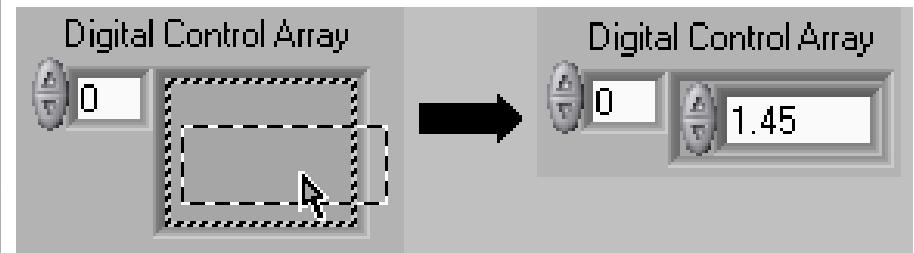


# Les contrôles et indicateurs de tableaux

1. Selection du contrôle Tableau (Array)



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

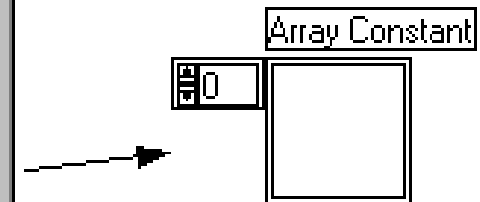
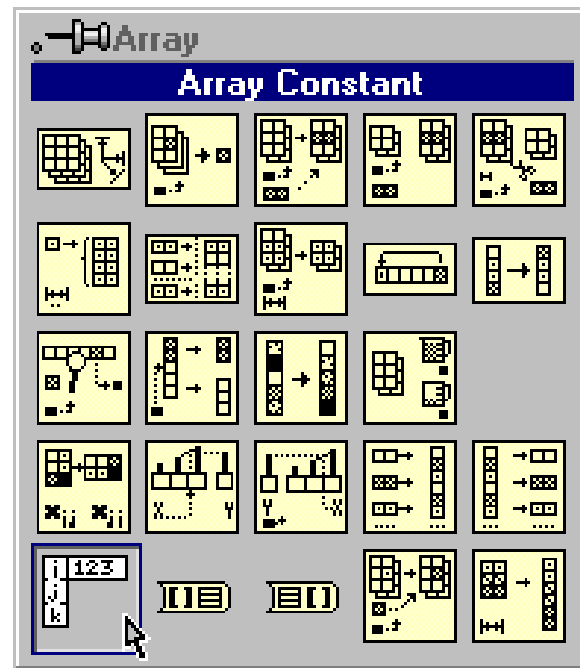


Redimensionner le contrôle pour ajouter une dimension

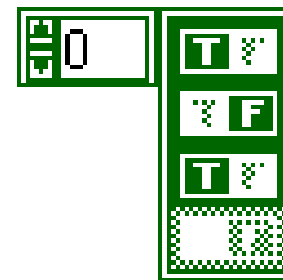


# Creation de constantes tableaux

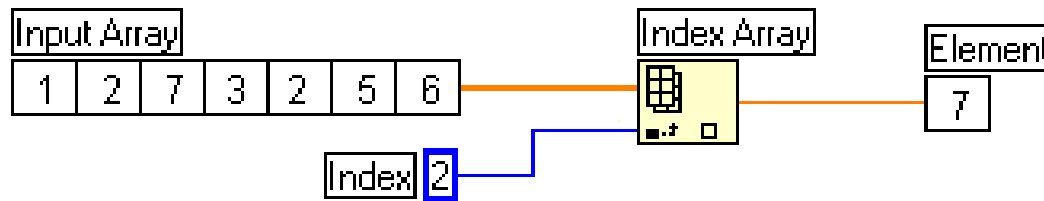
1. Choisir une constante de type tableau dans la palette de fonctions Array



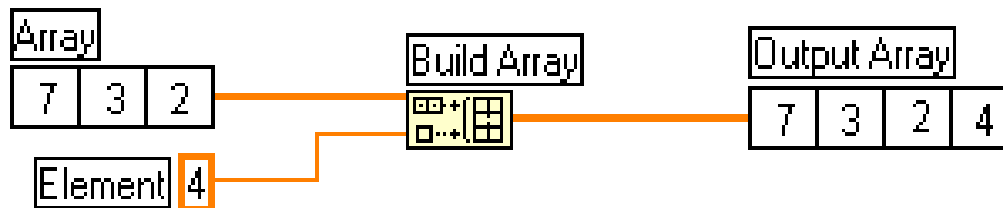
2. Placer une constante dans la constante tableau pour définir le type (ici, constante Booléene)



# Accès aux éléments du tableau



Extraction d'un élément

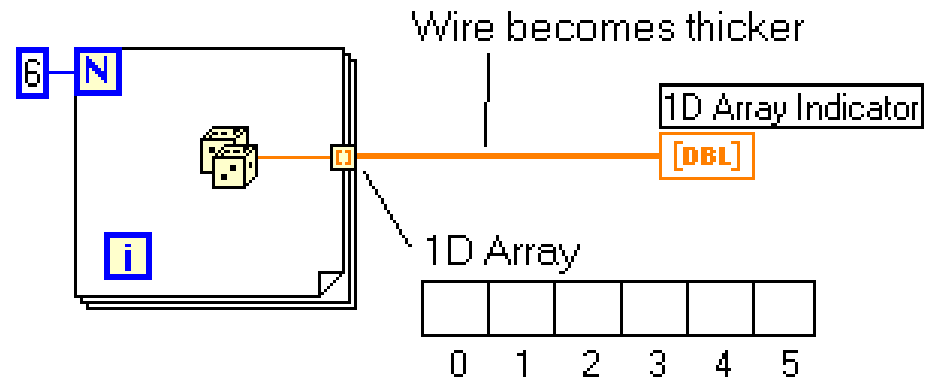


Ajouter un élément

# Création et utilisation de tableaux

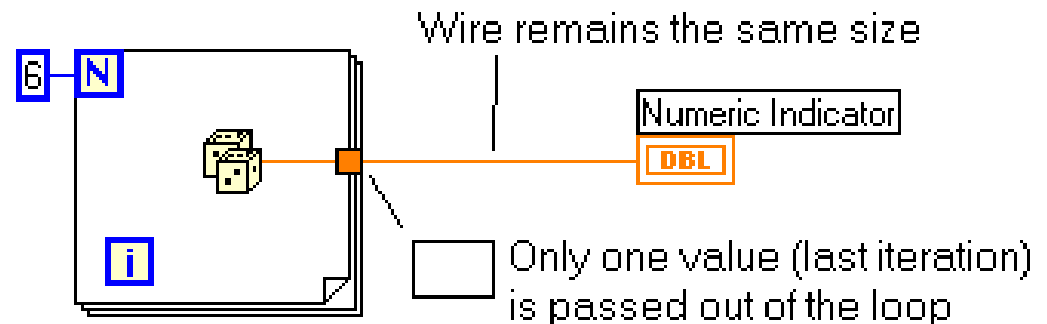
- *Sur une boucle, l'option 'Auto-Indexing' permet de construire automatiquement un tableau.*

## Auto-Indexing Enabled (Default of For Loops)

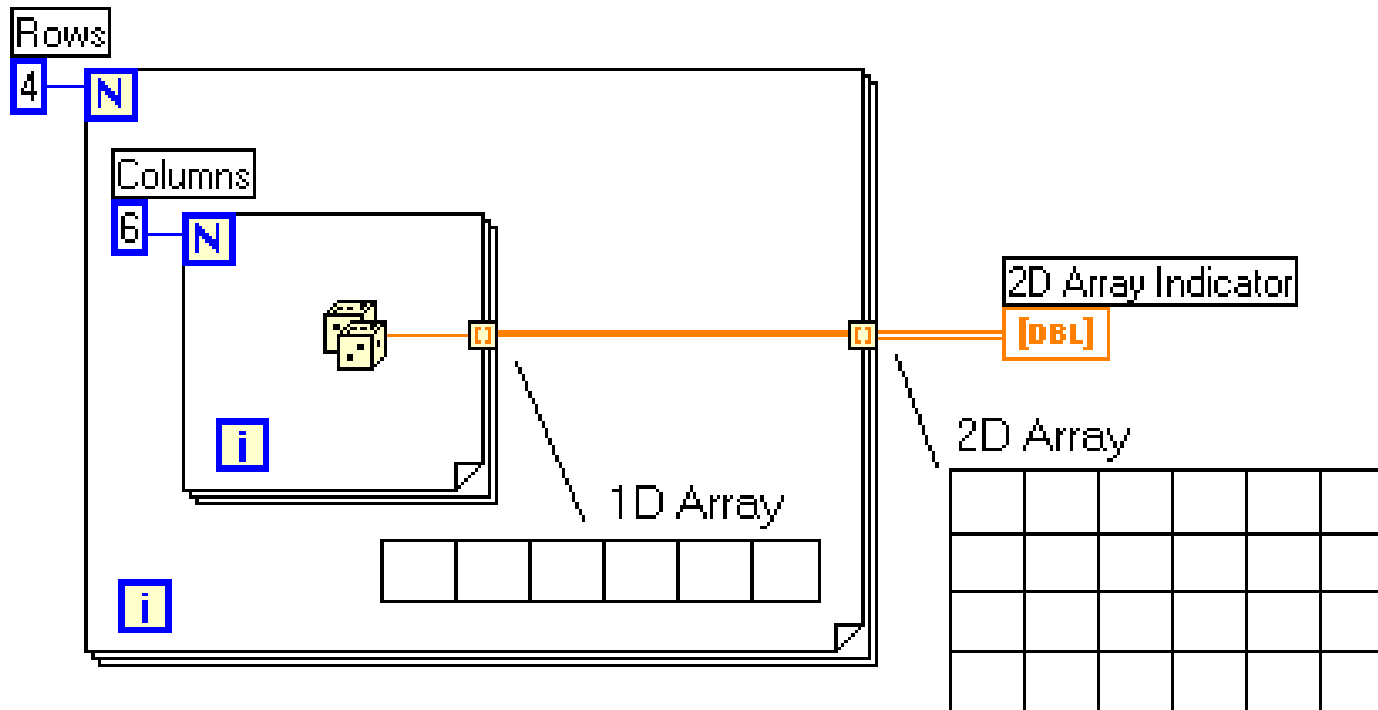


- Boucle For: auto indexing par default; Boucle While pas par défaut

## Auto-Indexing Disabled (Default of While Loops)



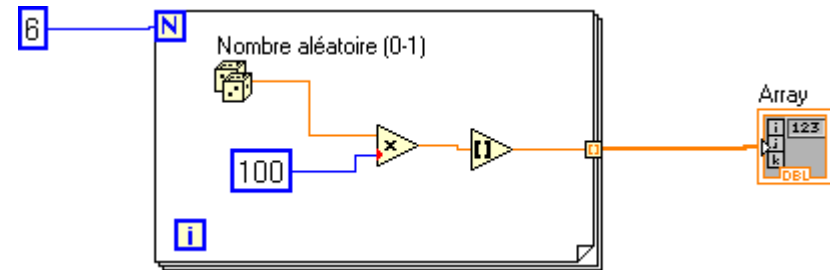
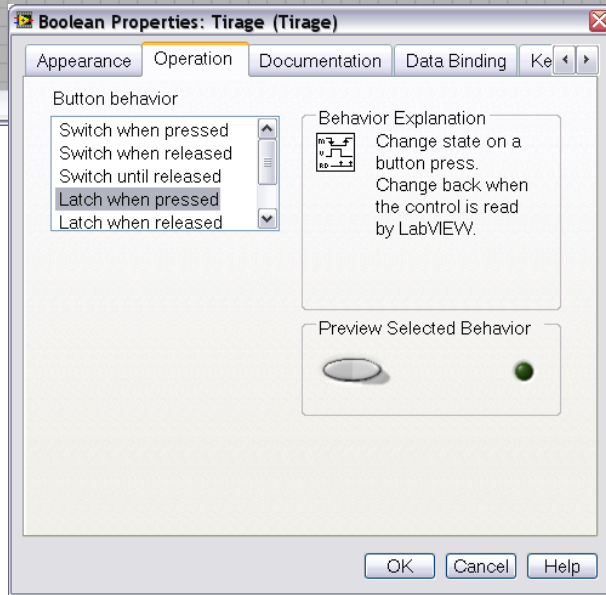
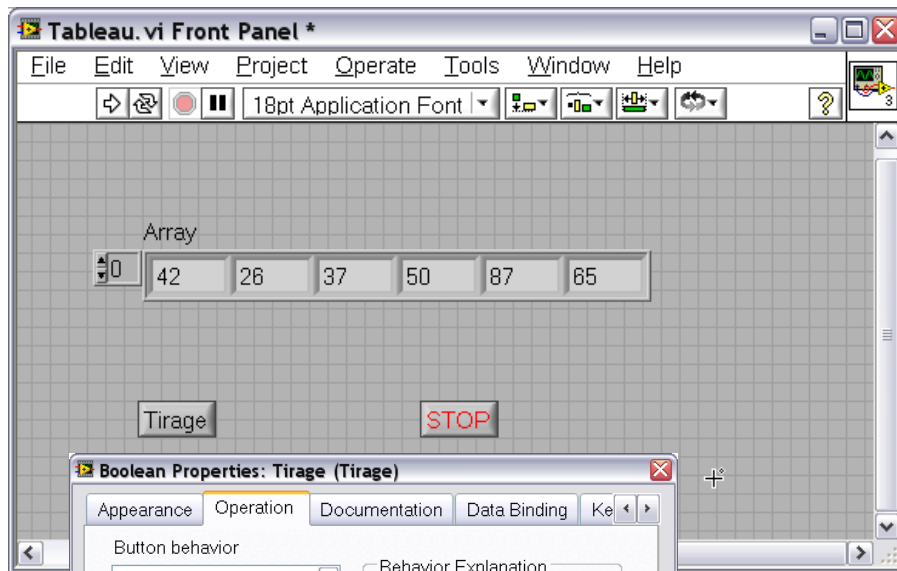
# Création de tableaux 2D



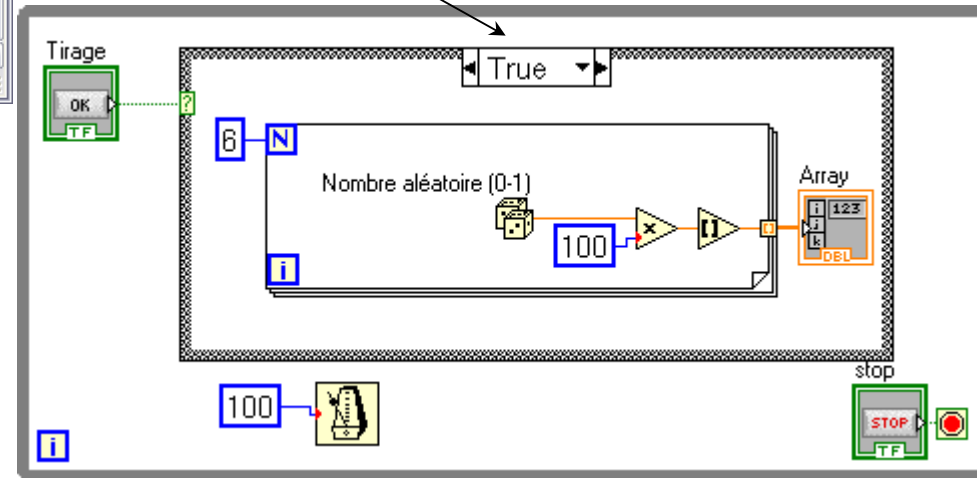
- La boucle interne crée les colonnes
- La boucle externe stocke les colonnes dans les lignes

# Exercice

- Créer un programme de loterie qui permet à chaque pression sur un bouton de générer 6 chiffres aléatoires entre 0 et 100 et d'afficher ces 6 chiffres à l'écran.



Case



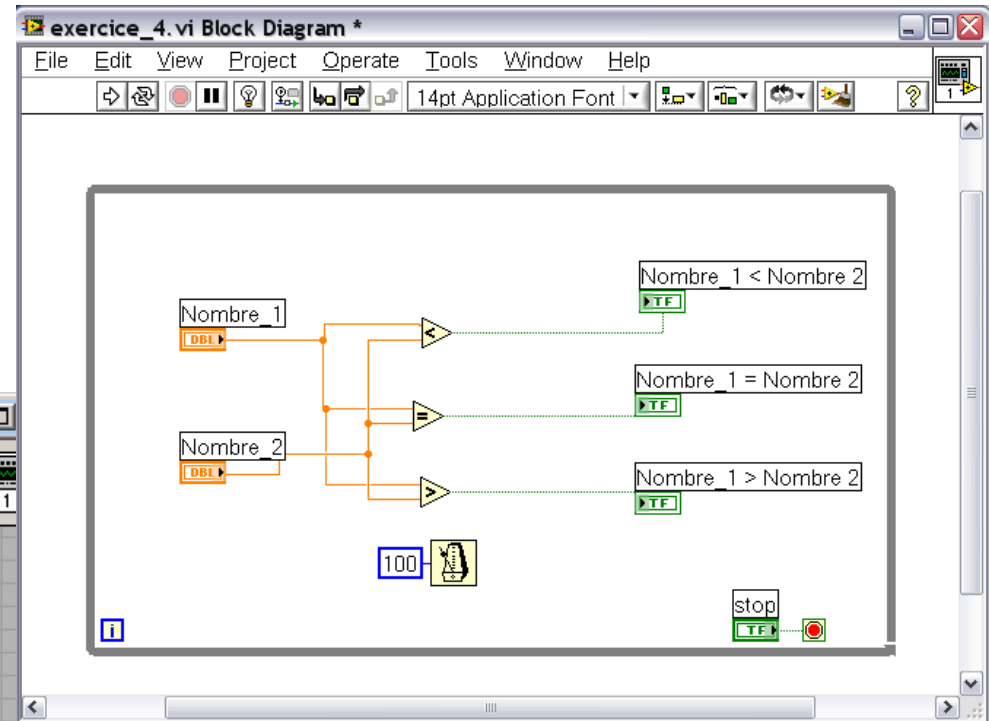
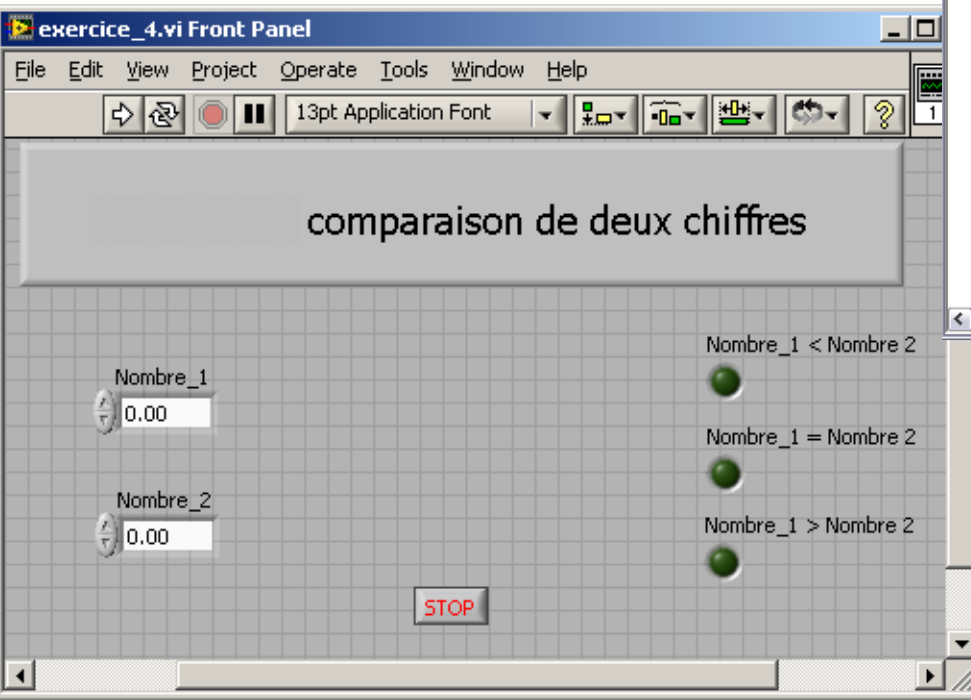
# Travail personnel

1. Réalisez un programme en boucle qui permet de comparer deux chiffres et qui affiche à l'utilisateur avec trois lampes si:

nombre 1 < nombre 2

nombre 1 = nombre 2

nombre 1 > nombre 2





2. Exécutez le tutoriel du chapitre 1 (pages 1-1 à 1-21) de *l'Initiation à LabVIEW* (estimé prendre 40 minutes).