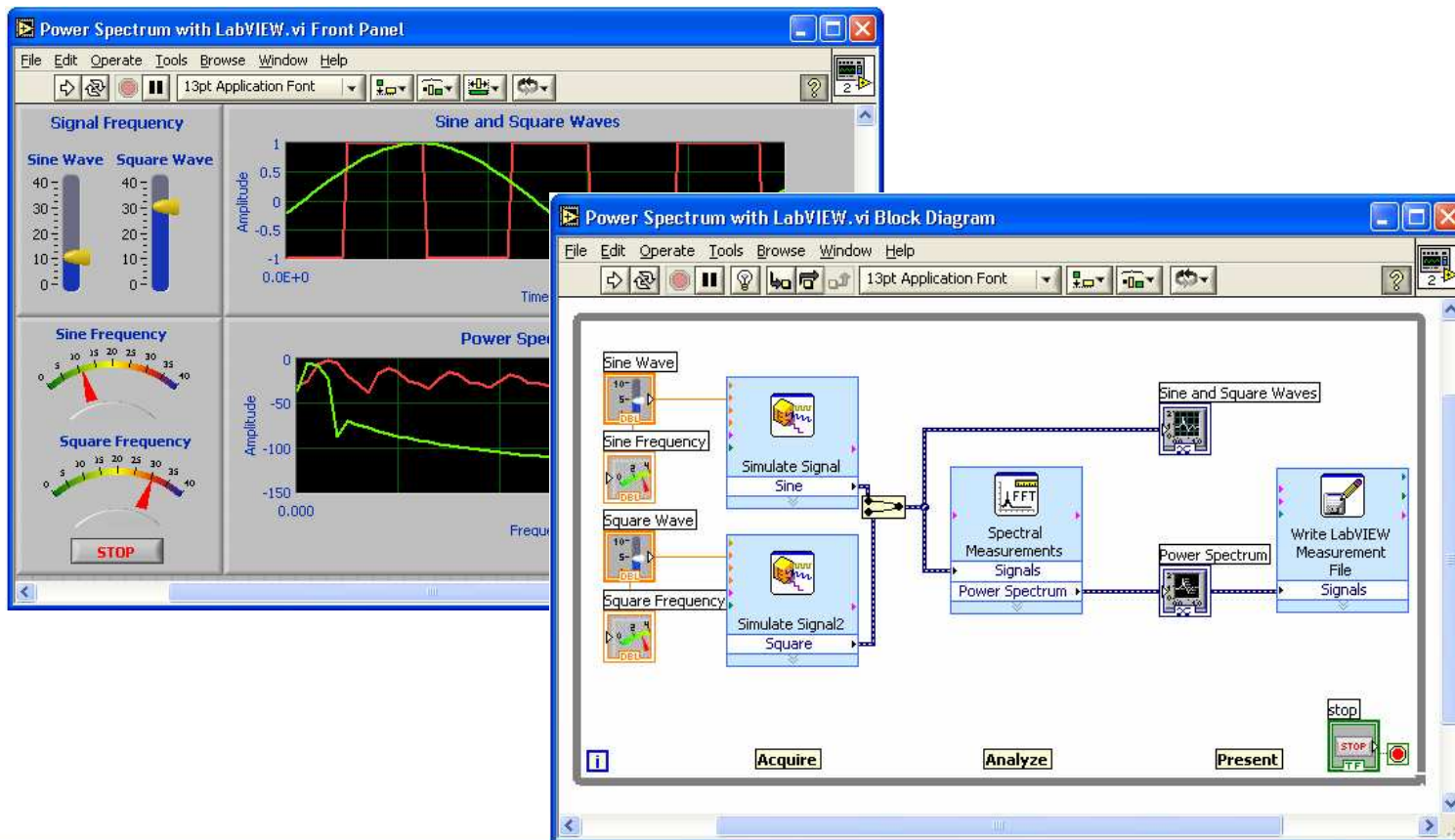


Instrumentation Virtuelle avec LabVIEW



Powered by :

IUT 1 GRENOBLE

Pierre Armand DEGRYSE
- David FREY



But du cours

- Introduire LabVIEW et ses fonctions de base
- Développer des architectures de programmation basiques.
- Travailler sur les boucles et structures condition
- Travailler avec des Tableaux et savoir utiliser des chaînes de caractères
- Savoir échanger des données avec d'autres systèmes

Powered by :

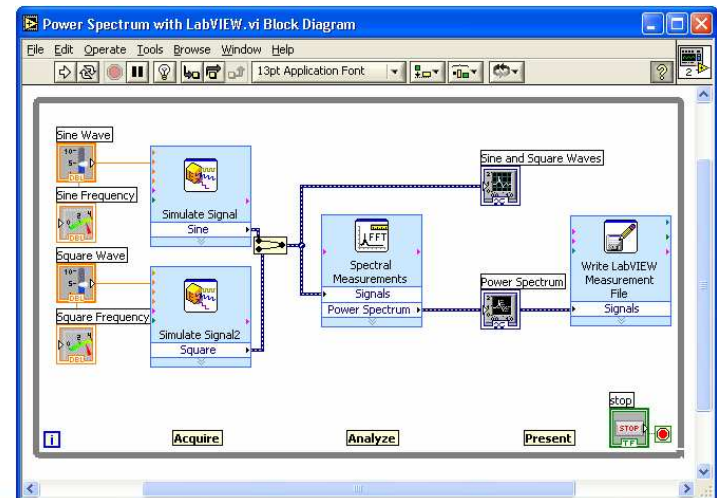
Les programmes LabVIEW sont appelés Virtual Instruments (VIs) (Instruments virtuels)

Front Panel (Face avant)

- Contrôles = Entrées
- Indicateurs = Sorties
- > Interface utilisateur

Block Diagram (Face arrière)

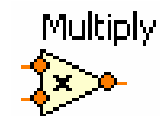
- Programme d'accompagnement de la Face avant.
- Les composants sont connectés ensemble via des fils.



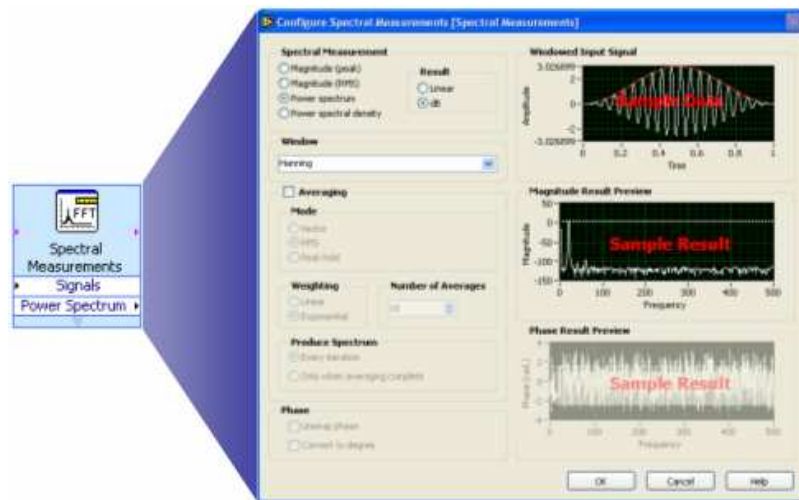
Powered by :

Express VIs, VIs et Fonctions

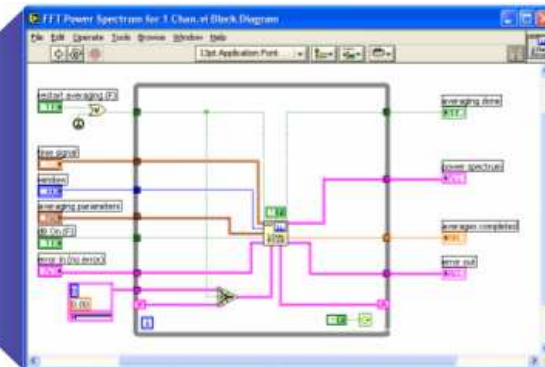
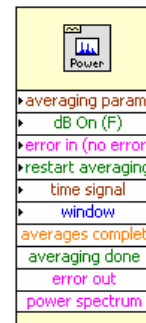
- Express VIs: VIs interactifs avec page de dialogue
- Standard VIs: VIs modulaires personnalisés
- Fonctions: Elements de base de Labview.



Fonction



Express VI

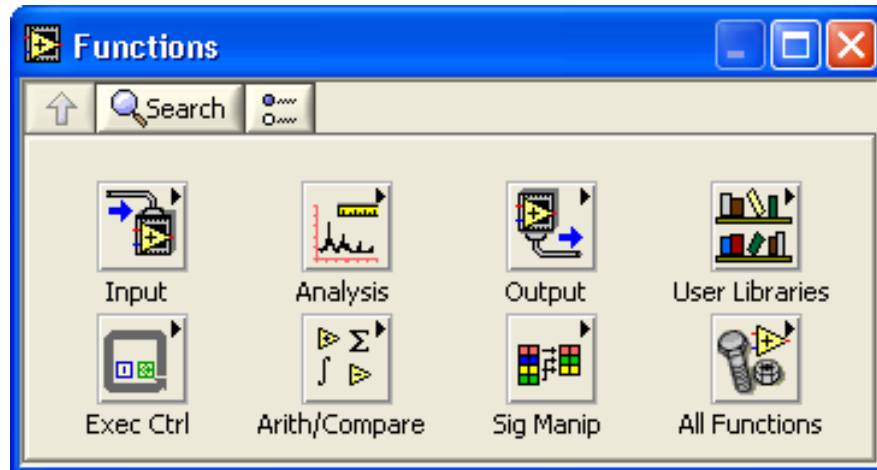
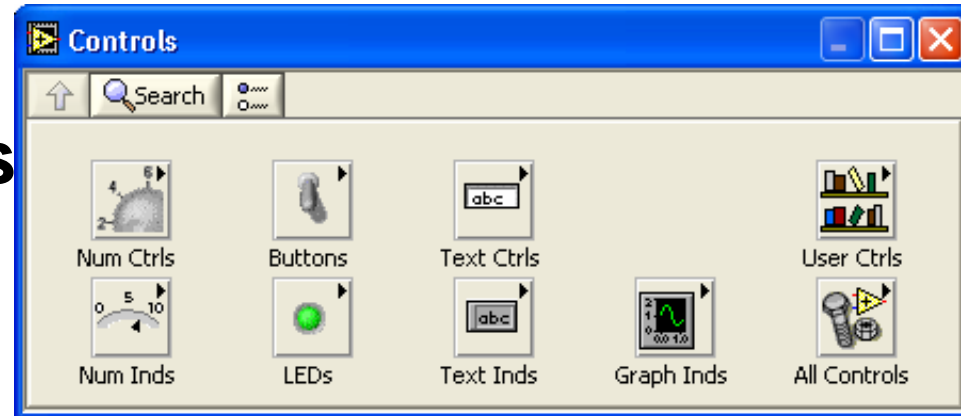


Standard VI

Powered by :

Palettes de control et de fonctions

**Palette de controls
(Face avant)**



**Palette de fonctions
(Face arrière)**

Powered by :

Palette rassemblant toutes les fonctions

Boucles de programmation

Gestion des chaînes de caractères

Gestion des tableaux

Fonction de test

Acquisition de données

Communication avec des instruments (Série, GPIB)

Communication réseau (TCP, Mail,...)

Gestion des nombres

Gestion des booléens

Gestion des clusters

Temporisations

E/S sur fichiers

Fonctions de traitement du signal

Images et son

Bibliothèque des Vis créés par l'utilisateur

Powered by :

Palette outils



- Palette flottante
- Utilisée pour créer et modifier des objets de la face avant et de la face arrière



Sélection automatique des outils



Outil de sélection



Outil de défilement



Outil de positionnement



Point d'arrêt



Outil d'écriture



Sonde



Outil de câblage



Outil copiage de couleur



Shortcut Menu Tool



Outil de coloration

Powered by :

Barre d'outils de la face avant



  **Lancement du programme**

  **Lancement continu**

 **Arrêt du programme**

 **Pause/Remise en route**

 **Paramètres des textes**

 **Alignement des objets**

 **Distribution des objets**

 **Réorganisation**

 **Redimensionnement des objets de face avant**

Boutons supplémentaires du diagramme

  **Montre le flux de données**

 **Début de l'exécution pas à pas**

 **Continuer l'exécution pas à pas**

 **Quitter l'exécution pas à pas**

Powered by :

Techniques de débogueur

- Trouver les erreurs



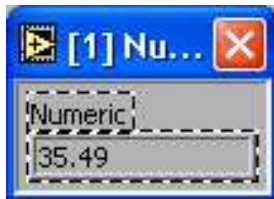
Cliquer sur le bouton Run brisé
Une fenêtre montrant les erreurs apparaît

- Déroulement de exécution



Le Click sur le bouton va animer le flot de données. Les valeurs sont affichées sur les lignes.

- Test



En cliquant droit sur les lignes on ajoute une sonde qui indiquera la valeur qui passe à travers le segment.



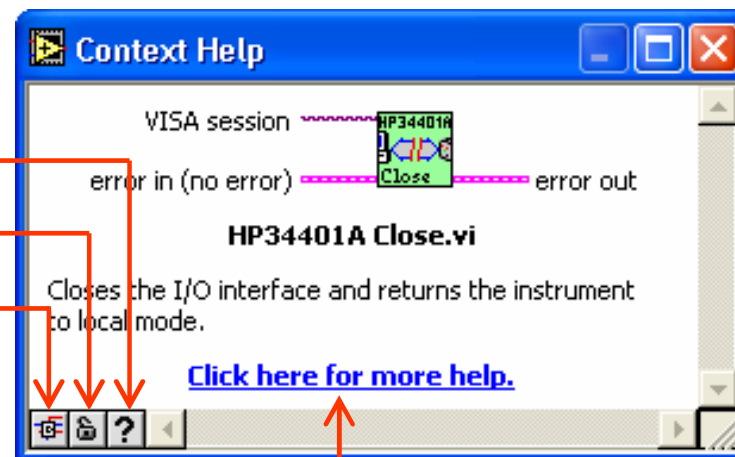
On peut aussi utiliser la sonde de la palette d'outils et cliquer sur la ligne.

Powered by :

Options de l'aide

Aide contextuelle

- Aide en ligne
- Verrouillage de l'aide
- Aide simple ou complète
- Ctrl + H

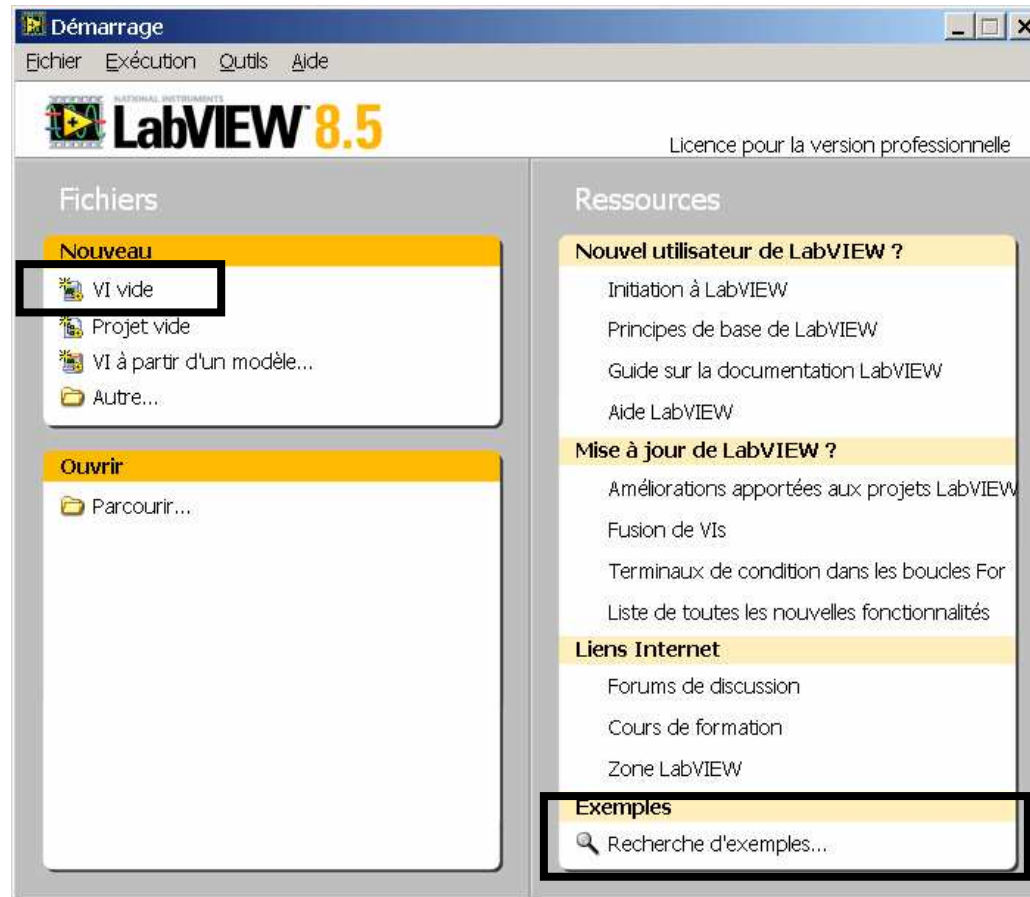


Manuel en ligne

- Accès à tous les menus
- Cliquez sur les icônes pour avoir accès aux aides

Powered by :

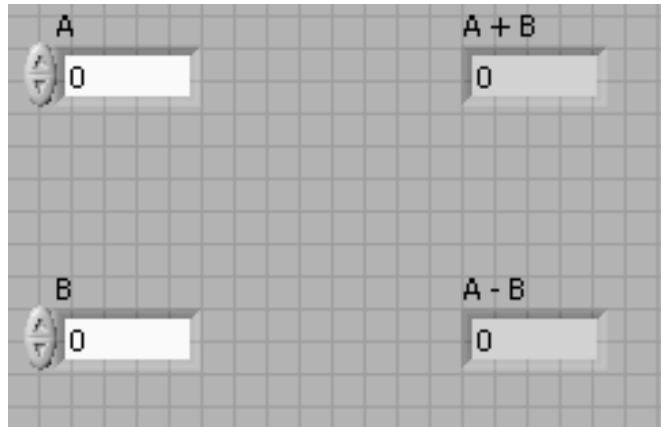
Créer un nouveau VI



Powered by :

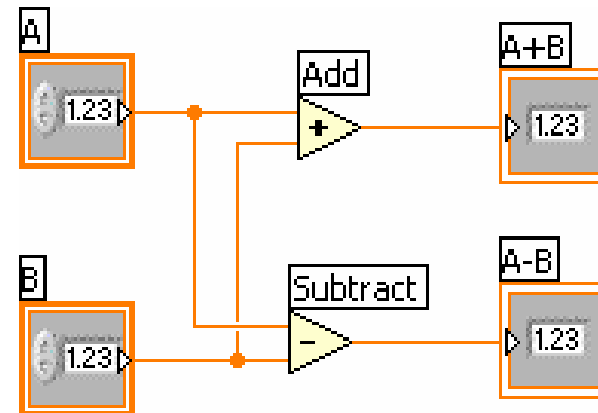
Exercice :

Face avant



Face arrière

Terminaux de control



Terminaux de sortie

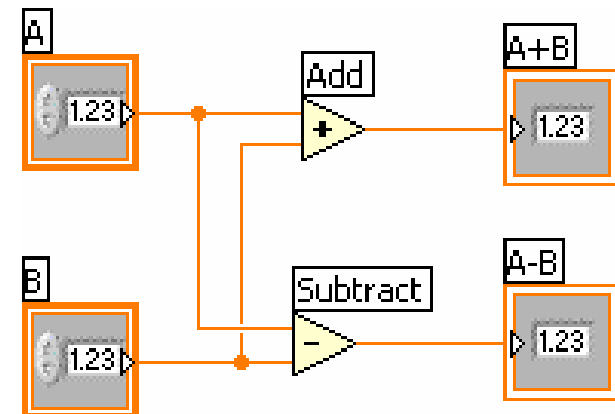
Powered by :

Programmation “Flux de données”

- L'exécution des blocs s'effectue en fonction du **flot des données** et NON de gauche à droite.
- Les noeuds s'exécutent quand les données sont présentes à tous les terminaux d'entrée.
- Les noeuds fournissent les données à tous les terminaux connectés en sortie quand le calcul est fait.

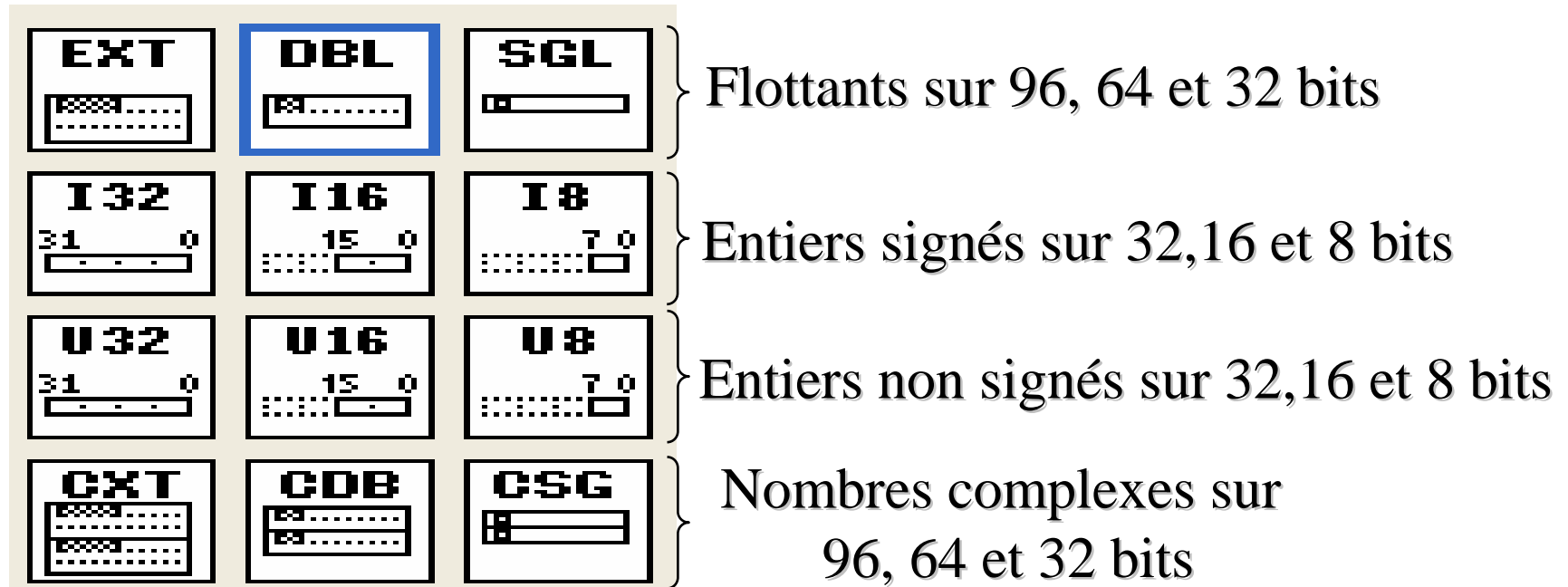


Le Click sur le bouton va animer le flot de données. Les valeurs sont affichées sur les lignes.



Powered by :

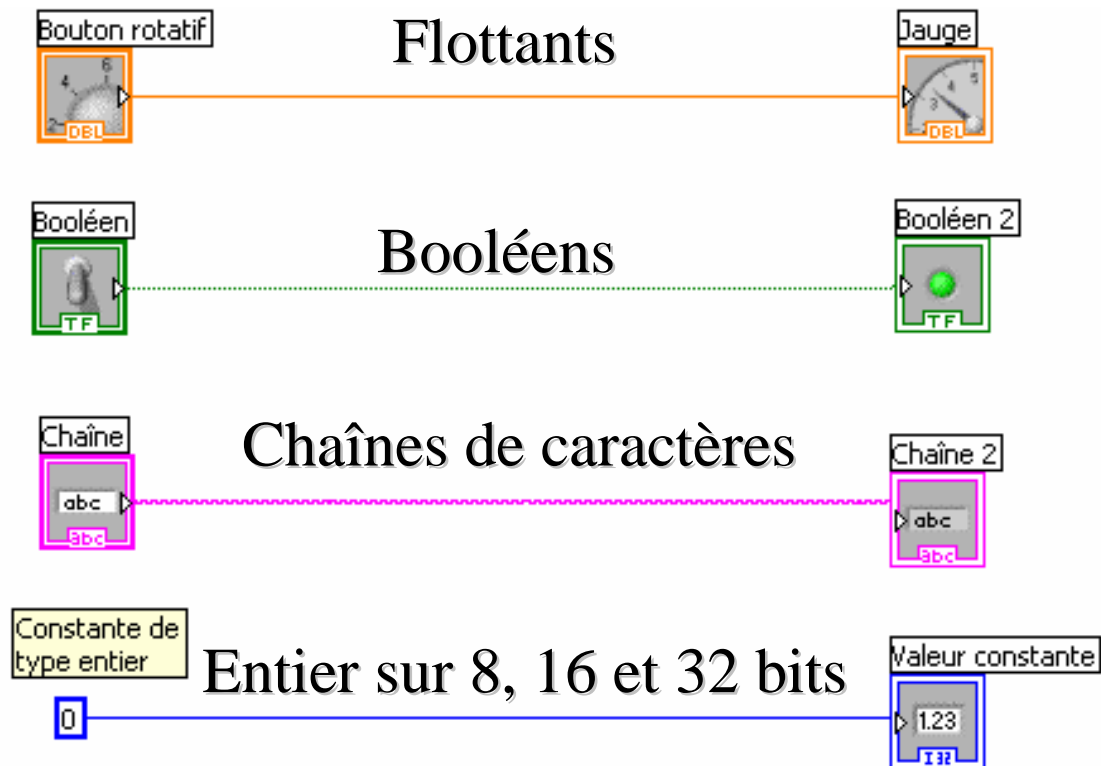
Les différentes variables dans Labview



On peut passer d'une variable à l'autre à l'aide de fonctions prédéfinies.

Powered by :

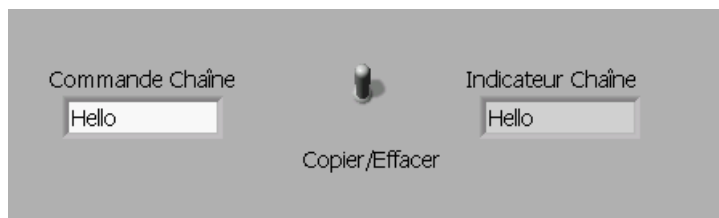
Les différentes variables dans Labview



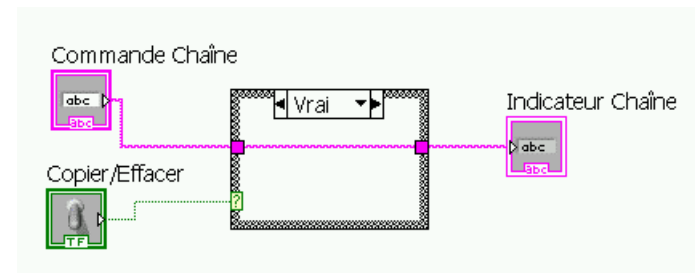
La structure de choix condition

- Exercice : programmer la recopie d'une chaîne de caractère si un interrupteur est activé, sinon effacer le texte.

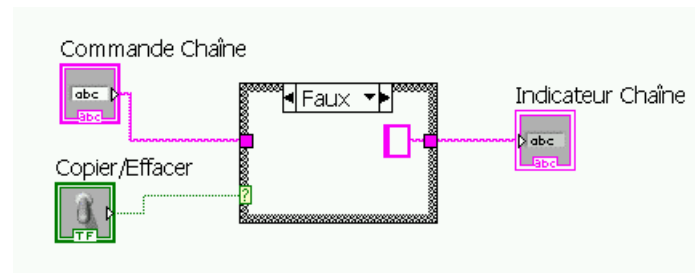
Face avant :



Face arrière : copier



Face arrière : effacer



Powered by :

Les tableaux

Labview permet de créer des tableaux de tous types de variables.

- Entiers
- Flottants
- Chaines de caractères
- Booléens
- ...

Powered by :

Les tableaux

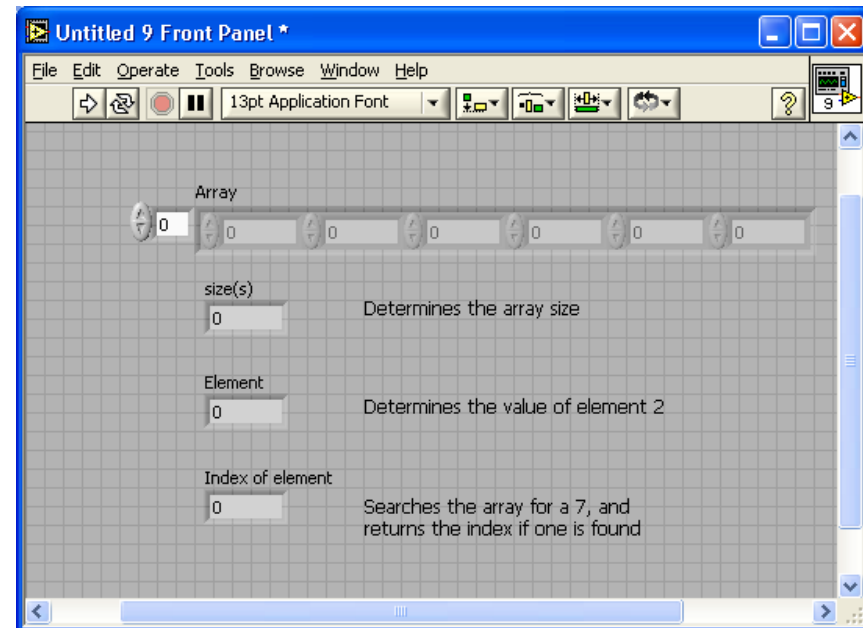
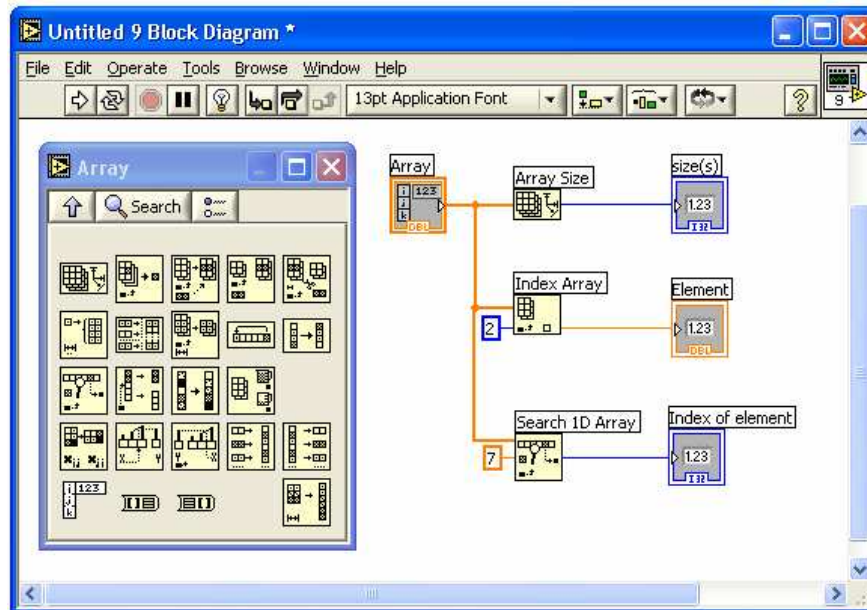
Il n'est pas nécessaire de déclarer les tableaux comme en 'C' par exemple.

On peut les créer pendant le programme et leur ajouter/enlever des éléments et donc faire varier leur taille sans problèmes.

Powered by :

Fonctions de tableau – Base

Fonctions >> Toutes les fonctions>>Tableau



Powered by :

Les tableaux (quelques fonctions)

Indexer un élément

Remplacer une partie

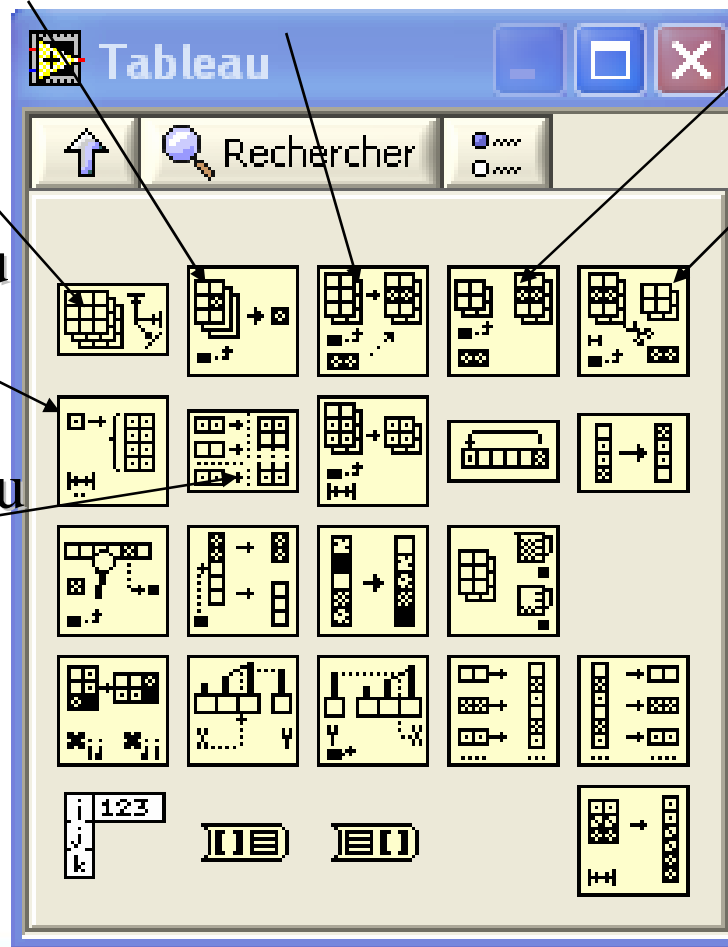
Insérer des éléments

Taille d'un tableau

Extraire des éléments

Initialiser un tableau

Construire un tableau



Etc.

Powered by :

Structure de boucles

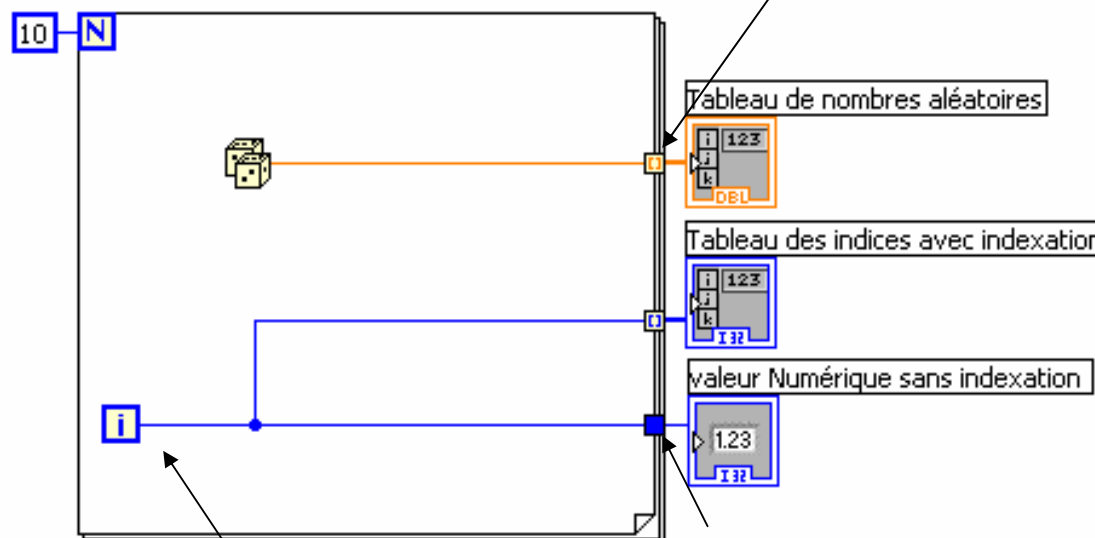
- Boucle For / Boucle While
- Auto-indexation
- Registres à décalage

Powered by :

Boucle For

Nombre de boucles à effectuer

Terminal de sortie auto indexé



Terminal de sortie non auto indexé

Terminal indiquant le numéro de l'itération (0 à N-1)

Powered by :

Boucle For

Auto indexation

Les boucles dans Labview ne renvoient les données que lorsqu'elles se sont entièrement exécutées.

Pb. Comment faire si on veut conserver à chaque boucle la valeur calculée?

Il faut auto indexer les données en sortie.
Labview crée un tableau est il va mettre à chaque boucle la donnée dans une autre case.

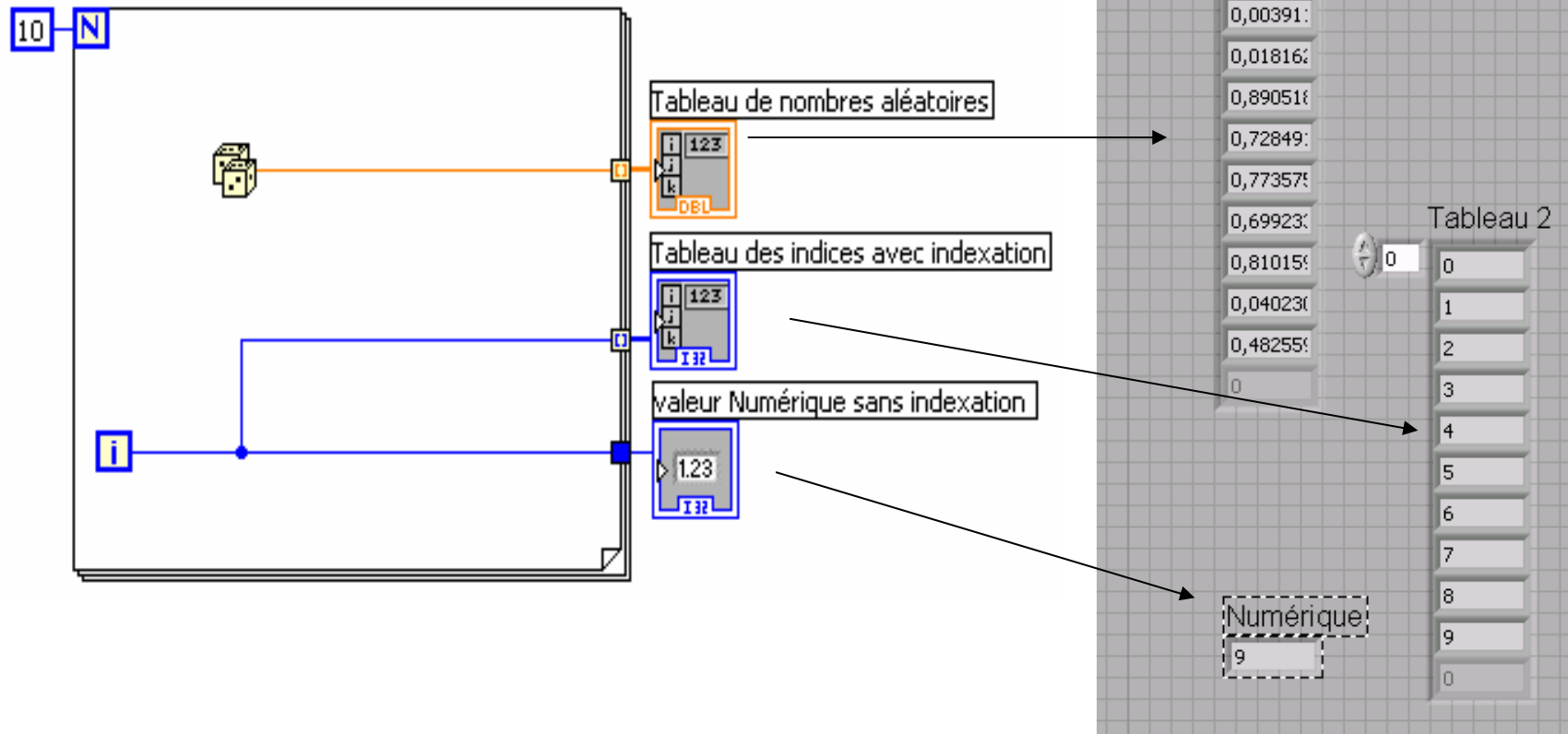
Pour cela, il faut faire un clique droit de la souris sur le terminal est choisir :
Activer l'indexation.

Powered by :

Boucle For

Auto indexation

Exemple : Génération aléatoire de 10 nombres



Powered by :

Boucle For

Auto indexation

Conclusion :

- Auto indexation activée : on récupère toutes les valeurs dans un tableau
- Auto indexation désactivée : on ne récupère que la dernière valeur

L'auto indexation fonctionne aussi en sortie pour les boucles While.

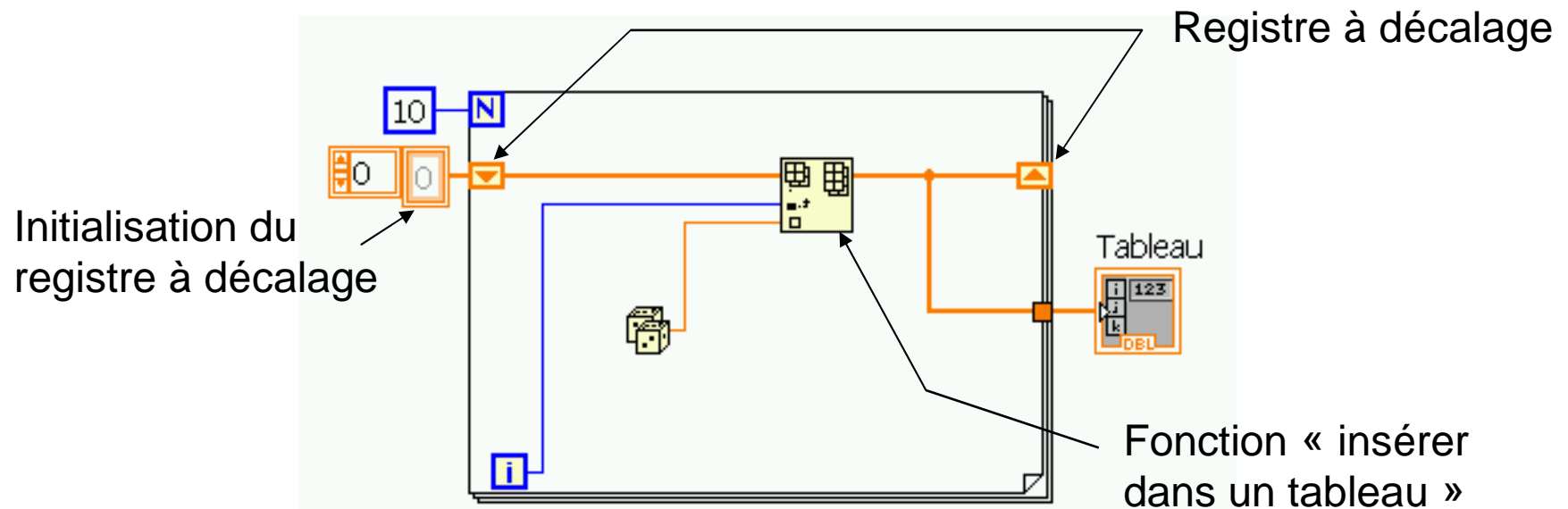
Si on utilise l'auto indexation quand on veut mettre un tableau en entrée d'une boucle For, il n'est pas nécessaire de mettre le nombre d'itérations à effectuer. Par défaut, Labview prendra la taille du tableau.

Powered by :

Registres à décalage

Comment faire pour utiliser une variable dans deux boucles successives d'une même boucle ?

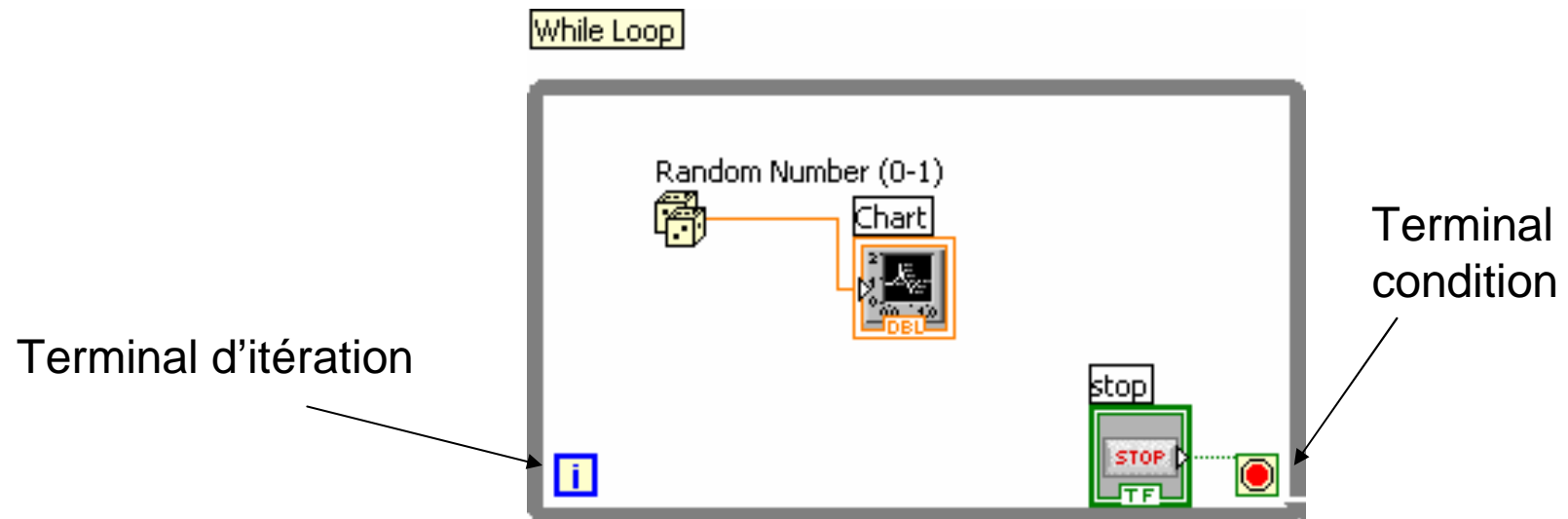
Nous souhaitons par exemple ne pas utiliser l'auto-indexation pour créer le tableau de nombre aléatoires et donc construire de manière itérative ce tableau.



Powered by :

Boucle While

- Possède un Terminal Iteration
- Fonctionne au moins une fois
- Fonctionne en fonction de l'état du Terminal condition



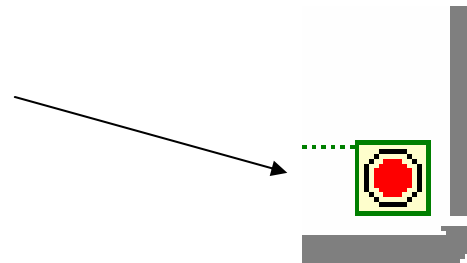
Powered by :

Boucle While

Conditions d'arrêt

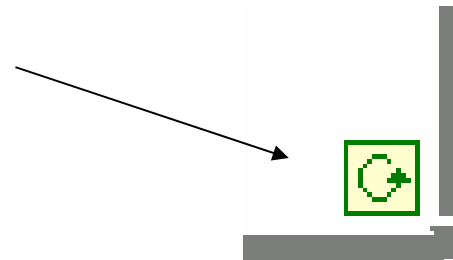
Arrêter sur condition vrai

While (toto = 0)...



Continuer sur condition vrai

While (toto = 1)...

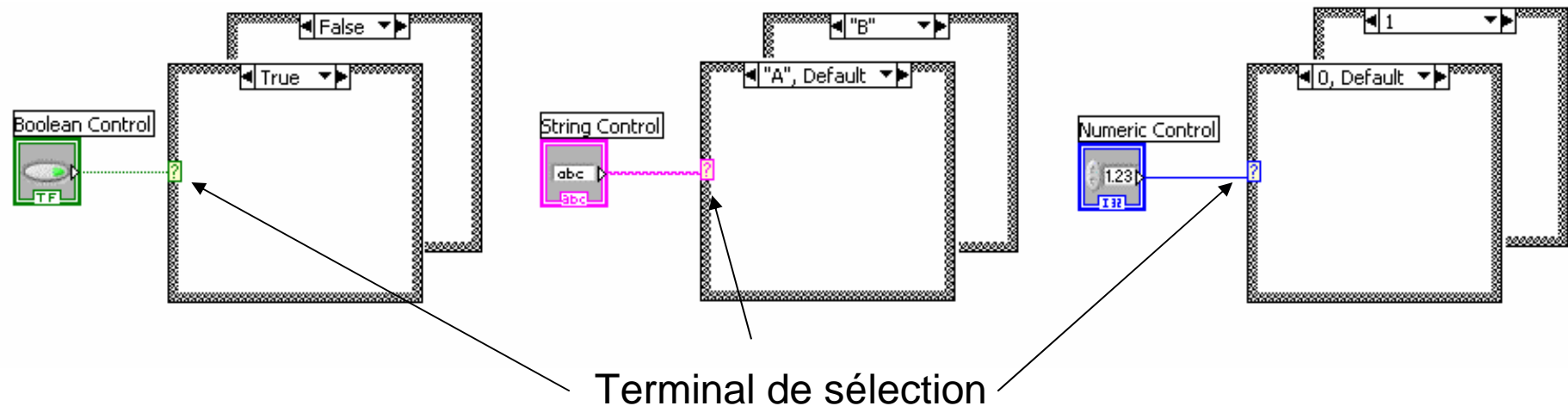


On choisit l'un ou l'autre en faisant un clique droit sur l'icône.

Powered by :

Structure Case

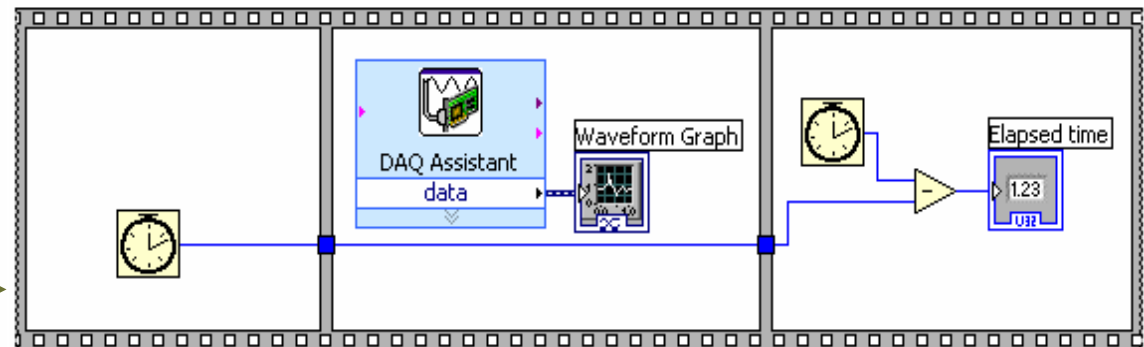
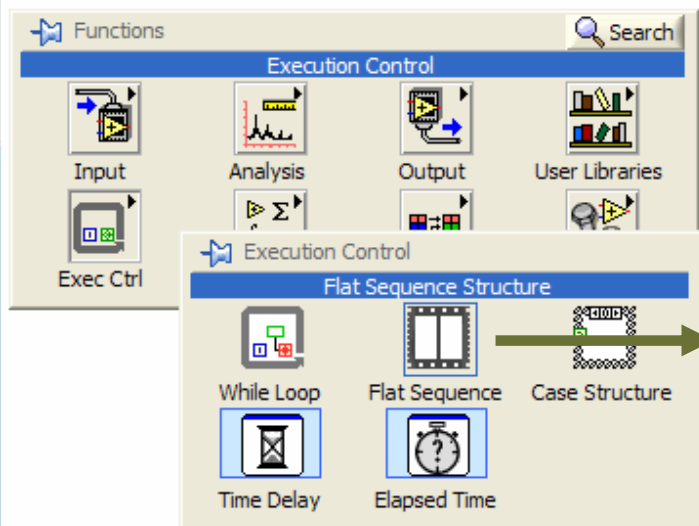
- Dans la palette Structure de la palette Fonction
- Entourer les noms ou déplacer les à l'intérieur du Case
- S'empilent comme des cartes. Seule celle du dessus est visible.
- Fonction switch ... Case en 'C'.



Powered by :

Structures Sequence

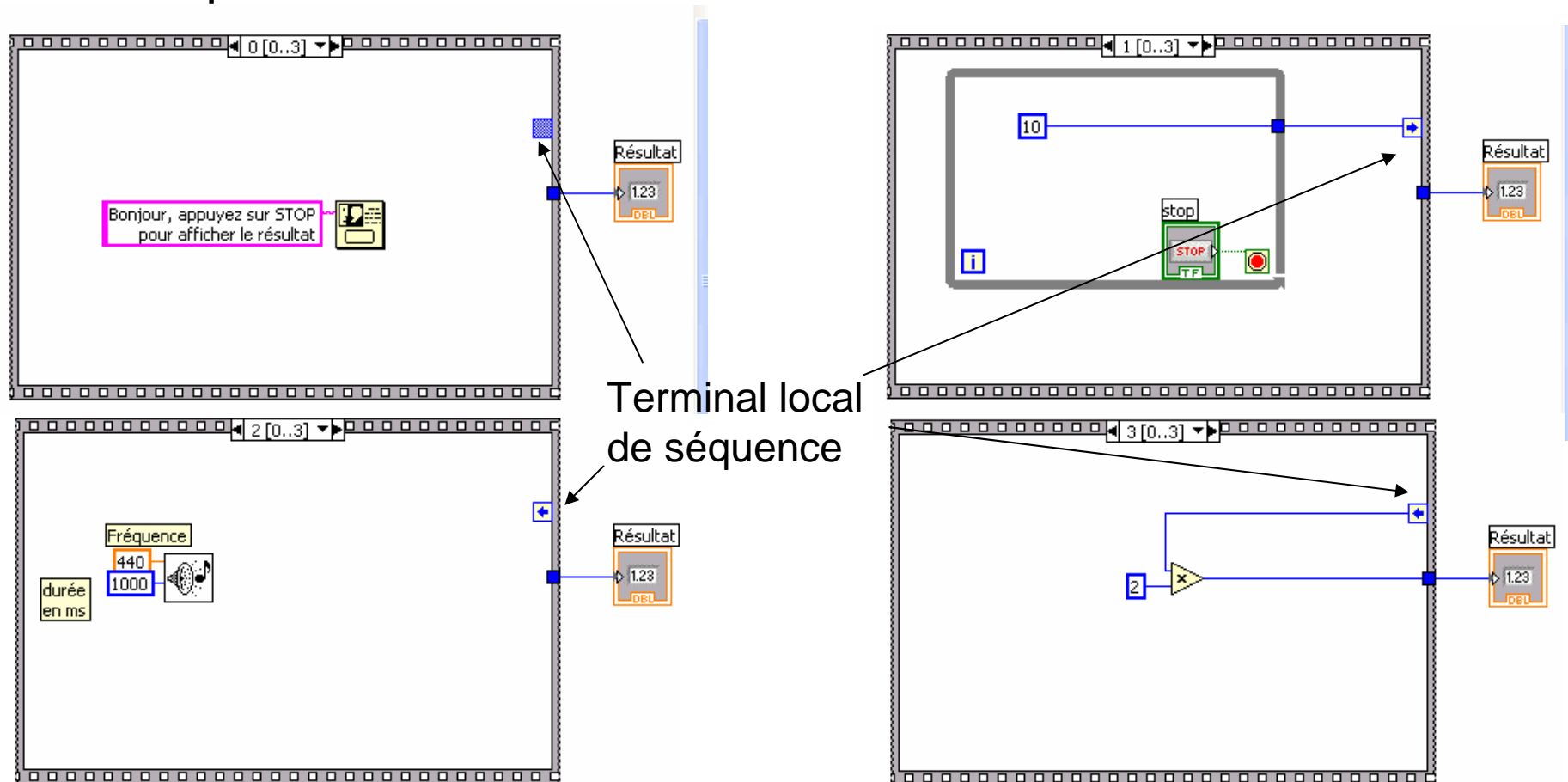
- Dans la palette **Structure** de la palette Fonctions
- Executes les diagrammes de façon séquentielle
- Click droit pour ajouter de nouvelles fenêtres
- Il existe deux types de séquences : empilées et déroulées



Powered by :

Séquence empilée

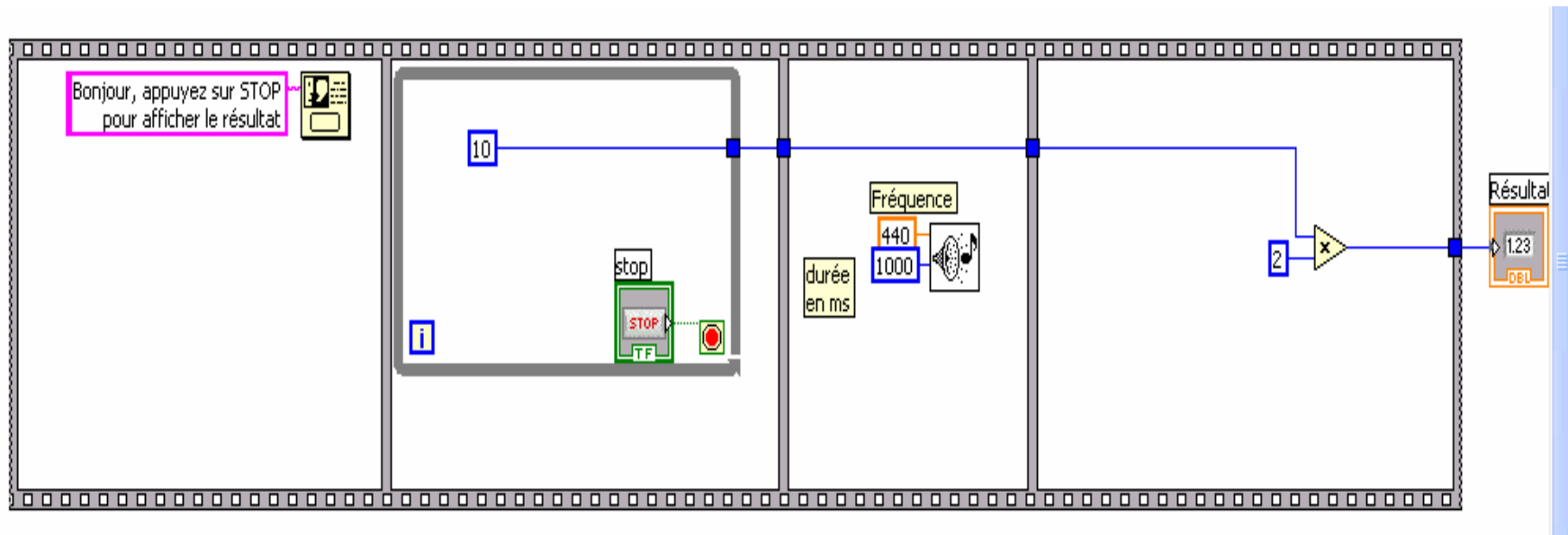
- Exemple



Powered by :

Séquence déroulée

- Les différents plans sont côte à côté
- Affichage complet, mais souvent moins clair



Powered by :