

Chapitre I : Introduction à LabVIEW

Introduction

Le logiciel **LabVIEW (Laboratory Virtual Instrument Engineering Workbench)** est une Plate-forme Expérimentale d'Instruments Virtuels pour Laboratoire. C'est un environnement de programmation et de développement d'application disponible sur plusieurs systèmes d'exploitation commercialisé par la société "**National Instruments**" et possédant un langage de programmation essentiellement graphique qui s'appelle le langage G, aussi des bibliothèques de fonctions et de sous-programmes, ainsi que des outils de développement.

Le logiciel LabVIEW est dédié au contrôle, à l'acquisition, l'analyse et la présentation de données, au développement de systèmes de mesure, de test, d'instrumentation et de contrôle-commande sophistiqués. Utilisable avec une grande variété de matériels, il est multiplateformes (voir figure I.1), multi-cible et intègre un grand nombre de bibliothèques en natif. De plus en plus de constructeurs fournissent une bibliothèque LabVIEW avec leurs matériels, c'est un logiciel facile à programmer grâce au langage G, on n'écrit pas de lignes de programme dans un langage textuel comme Pascal ou C, Basic ou Fortran. On manipule des objets graphiques. Ces objets graphiques représentent à la fois les variables du programme, ainsi que des fonctions qui vont réaliser des actions portant sur ces variables. Donc, LabVIEW se distingue des autres logiciels sur au moins un point important. En effet, la majorité d'entre eux s'articulent autour de langages à base de texte dont la programmation consiste à empiler des lignes de code, tandis que LabVIEW utilise le langage G de programmation graphique pour créer un programme sous forme de diagramme, ce programme est indépendant de la plateforme utilisée, puis la migration d'applications est possible entre les plateformes.

La programmation en LabVIEW consiste simplement à concevoir le traitement de l'information, organiser et relier les variables avec les fonctions au moyen de fils. LabVIEW est dédié à la programmation conçue pour le pilotage d'instruments électronique! Avec LabVIEW on construit graphiquement des modules logiciels appelés des «VI» («VI» sigle de «Visual Instruments») au lieu d'écrire du code dans un langage informatique textuel. Son principe de programmation est basé sur l'assemblage graphique de modules logiciels appelés «Instruments Visuels («VI»)). Le rôle d'un VI est d'acquérir des données issues par exemple de fichiers, du clavier ou encore de cartes électroniques d'Entrée/Sorties », de les analyser, et de les présenter au travers d'interfaces hommes-machines graphiques (encore appelées «face avant» par analogie avec la face avant permettant de piloter un appareil électronique). Dans LabVIEW, ce qu'on appelle la «face avant» est donc l'interface utilisateur permettant d'exploiter, de piloter, le programme.

Préparé par D.YD

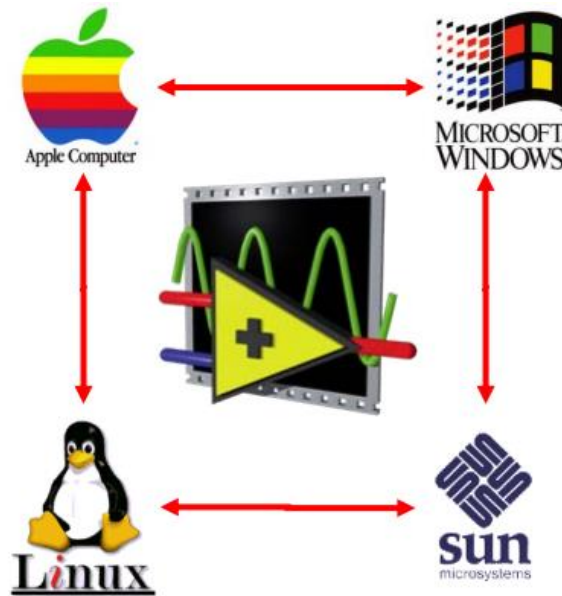


Figure I.1 : Compatibilité multiplateformes de LabVIEW.

I.1 Ouverture d'un programme LabVIEW

Pour ouvrir LabVIEW à partir du logiciel, il suffit d'utiliser le menu « démarrer » de la barre de menu du bas de l'écran, puis de sélectionner « LabVIEW 8.0 » dans « Tous les programmes ». On voit alors apparaître la fenêtre de la figure(I.2). Il suffit ensuite de sélectionner Blank VI pour créer un VI vide.

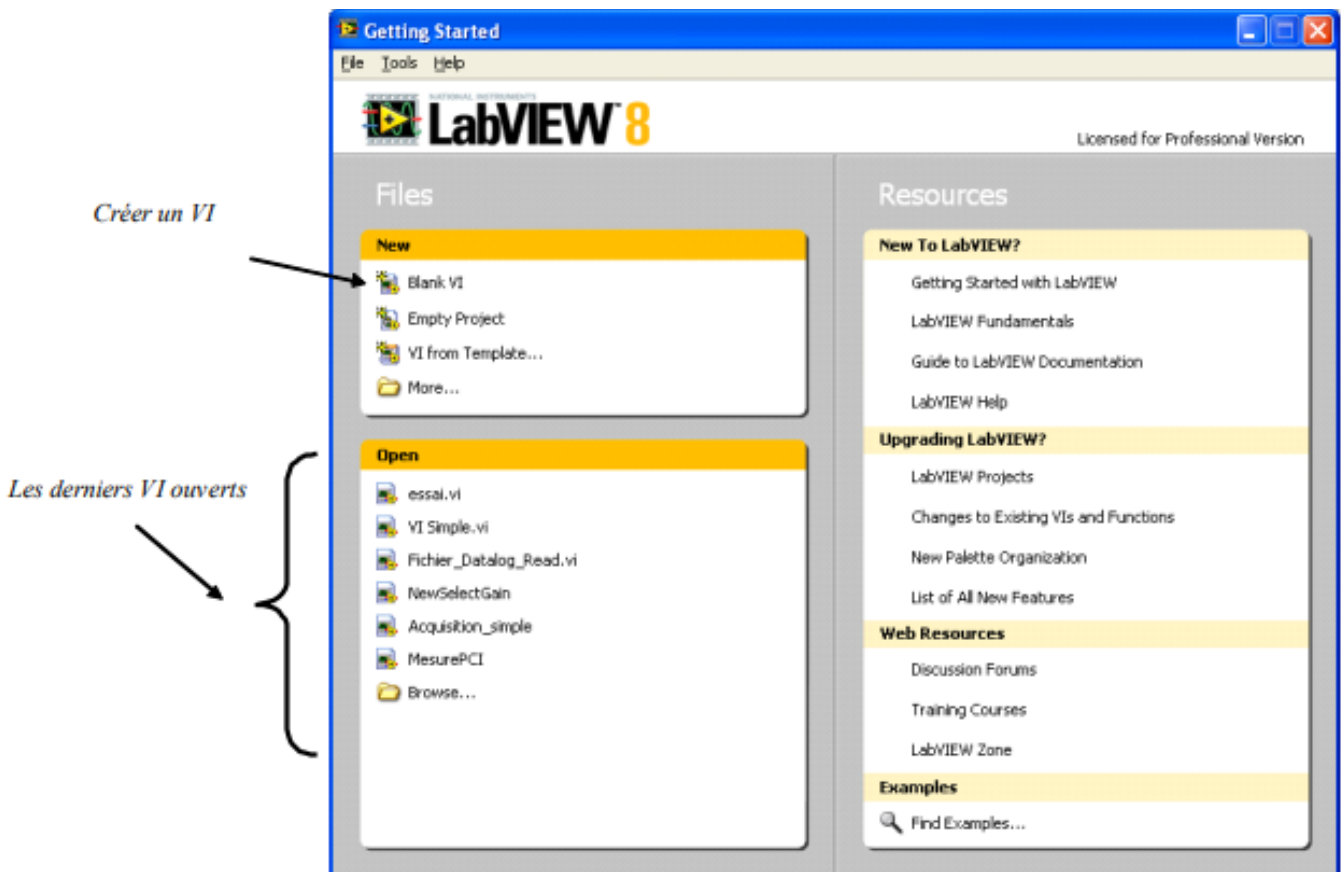


Figure I.2 : Fenêtre apparaissant à l'ouverture de LabVIEW.

I.2 Instrument Virtuel (VI) de LabVIEW

Le logiciel LabVIEW, grâce à son environnement à multiples fenêtres, permet de réaliser et d'exécuter rapidement des programmes simples, comparables aux fonctions d'un langage conventionnel, mais pouvant prendre à l'écran l'apparence d'un appareil de mesure, d'où leur appellation Instrument Virtuel (Virtual Instrument en anglais, ou « VI »). Un programme VI sur LabVIEW comprend 2 fenêtres distinctes, qui sont **Face avant** nommée aussi **Panneau avant** (en anglais **front panel**) et **Diagramme** (en anglais **block diagram**). La fenêtre **Face avant** servant d'interface avec l'utilisateur et la fenêtre **Diagramme** contenant le programme source en langage graphique G, ce langage est assimilable aux langages orientés objet, tel le C++, offrant des classes de données ayant des attributs spécifiques, ainsi que des opérateurs et des fonctions polymorphes agissant sur les données. Tout programme exécutable construit en langage G comporte une interface utilisateur et un programme graphique situé dans les deux fenêtres distinctes **Face avant** et **Diagramme**, L'environnement LabVIEW offre aussi trois autres fenêtres distinctes, des palettes indépendantes d'outils et d'objets permettant d'éditer les deux fenêtres du programme et de tester son fonctionnement.

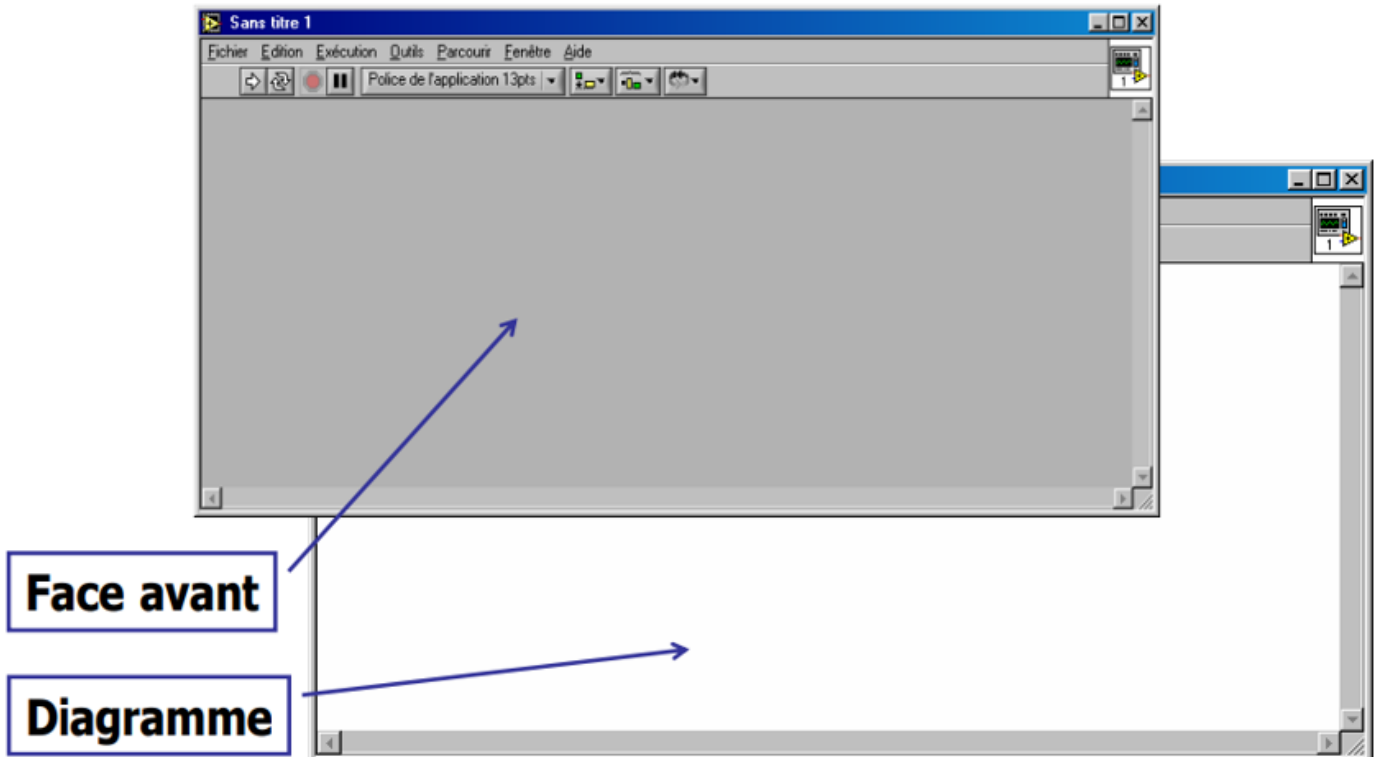


Figure I.3 : Face avant et Diagramme les deux fenêtres de VI.

Pour placer les deux fenêtres cote à cote cliquer (**Ctrl+T**), puis pour faire l'interaction entre la fenêtre **Face avant** et la fenêtre **Diagramme** cliquer (**Ctrl+E**).

I.2.1 Fenêtre Face avant

Cette fenêtre représente l'Interface Homme Machine, où apparaissent des objets sous forme de commandes d'entrée ou contrôleurs (Controls) ou d'indicateurs de sortie (Indicators), constitue l'interface interactive du programme. Elle Contient les contrôles et les afficheurs de données comme l'interrupteurs, variateurs, potentiomètres avec curseurs, boîtes de dialogue LEDs, compteurs à aiguilles, vumètres, boîtes de message, graphes. La fenêtre **Face avant** vide apparaissant par défaut lors de la création d'un programme est indiquée sur la figure ci-dessous.



Figure I.4 : Fenêtre **Face avant** vide.

Un programme écrit en langage G sera utilisable de façon interactive dès que son **Face avant** sera visible. Par un **Clic droit** sur la fenêtre **Face avant** on obtient la palette de Fonctions. L'interaction entre **Face avant** et **Diagramme** par (Ctrl+E).

I.2.2 Fenêtre Diagramme

Cette fenêtre contient le code source graphique représentant le programme écrit en langage G, ainsi que les Terminaux de commandes, nœuds (sous-programme, fonctions, structures, interfaces code), constantes, Terminaux d'affichage, des connecteurs, des fils. Le **Diagramme** vide apparaissant par défaut lors de la création d'un programme est indiqué sur la figure ci-dessous.



Figure I.5 : Fenêtre **Diagramme** vide.

On passe du **Face avant** au **Diagramme** à l'aide du choix Show Block Diagram du menu Window de la barre de menus. Réciproquement, on retourne au **Face avant** à partir du **Diagramme** à l'aide du choix Show Front Panel du menu Window de la barre de menus.

Préparé par D.YD

Par un **Clic droit** sur la fenêtre **Diagramme** on obtient la palette de Fonctions. En passant du **Diagramme** vers **Face avant** par (**Ctrl+E**). En utilisant (**Ctrl+U**) pour réorganiser le **Diagramme** .

I.2.3 Caractères communs au Face avant et au Diagramme

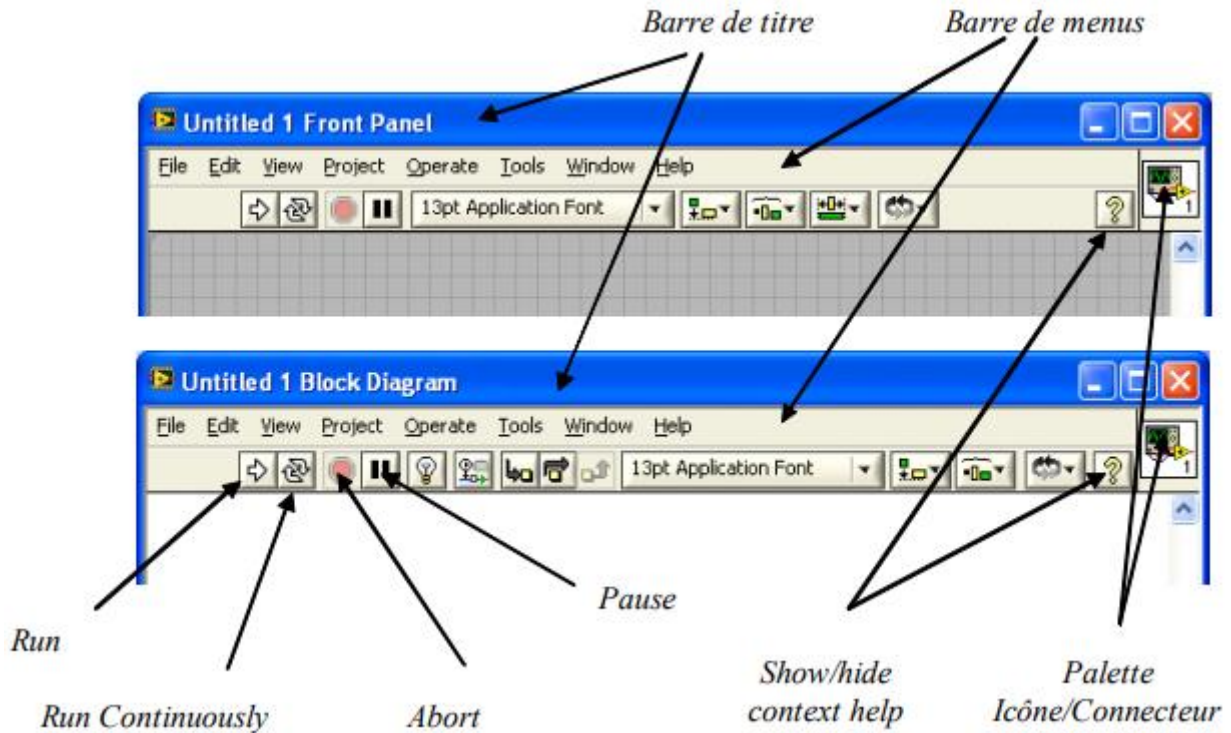


Figure I.6 : Caractères communs aux deux fenêtres.

- 1) **Barre de titre** : Située en haut de chaque fenêtre, la barre de titre (Figure I.6) affiche le nom du programme. Ce nom est *Untitled i* (où *i* est un nombre > 0) par défaut à l'ouverture d'un nouveau programme, et le reste tant que le programme n'a pas été sauvegardé ;
- 2) **Barre de menus** : Située sous la barre de titre (Figure I.6), elle présente une série de menus déroulant communs aux deux fenêtres. Ces menus à structure hiérarchisée proposent des choix pouvant entraîner une action immédiate, ou aboutir à une fenêtre de dialogue (...). Les principales actions déroulant de ces menus sont indiquées ci-dessous :
 - a) Menus hiérarchisés communs aux fenêtres du diagramme et du Face avant :
 - b) File Pour manipuler les fichiers (créer, ouvrir, sauver, imprimer) ;
 - c) Edit Pour éditer la fenêtre active (copier, coller) ;
 - d) View Pour voir la hiérarchie des VI en mémoire et rappeler les palettes Tools ou Controls ;
 - e) Project Nous ne nous en servons pas cette année Operate Pour choisir le mode de fonctionnement ;
 - f) Tools Nous ne nous en servons pas cette année ;

Préparé par D.YD

- g) Window Pour changer la fenêtre active (« panneau avant »/diagramme) ;
 - h) Help Pour afficher la fenêtre d'aide de LabVIEW et accéder à l'aide en ligne (Online Reference ...).
- 3) **Palette de boutons de sélection** : Située juste au-dessous de la barre de menus, elle offre de nombreux boutons de sélection, dont au moins quatre boutons de commande communs aux deux fenêtres.
- 4) **Boutons de commande communs aux fenêtres du diagramme et du « panneau avant »** :
Exécution (Run) Pour exécuter le programme Exécution Répétée (Run Continuously) Pour exécuter le programme continûment Arrêt (Abort Execution) Pour mettre fin au programme dans la phase de mise au point, l'algorithme achevé devant prévoir une fin de programme « normale » Pause (Pause) Pour faire une pause à l'appel d'un sous-programme afin d'observer ses entrées avant son exécution ou pour accéder au mode d'exécution pas à pas Aide (Help) Pour activer ou désactiver l'aide de LabVIEW sur chaque objet du programme
- 5) **Menus surgissants** : La zone située sous la palette des boutons de sélection est spécifique au diagramme ou au « panneau avant ». Lorsqu'on clique dans cette zone avec le bouton droit de la souris, un menu surgissant (Pop-up Menu) ou une palette d'objet apparaît à l'endroit pointé. Si l'on a cliqué dans une zone vide, une palette apparaît (Controls ou Functions suivant la fenêtre concernée), permettant d'introduire des objets dans la fenêtre. Si l'on a cliqué sur un objet situé dans la fenêtre, le menu surgissant sur l'objet est spécifique à l'objet et propose plusieurs choix pour modifier, initialiser, ... l'objet en question. L'emploi des menus surgissant sur les objets est très utile car il permet d'accéder aux paramètres de l'objet.
- 6) **Palette icône/connecteur (Icon/Connector Panel)** : Cette palette est située dans la partie supérieure droite des deux fenêtres de programme de LabVIEW (Figure I.6). Elle comprend deux éléments superposés : l'icône et le connecteur. L'icône (Icon), visible par défaut, symbolise le nom du programme tel qu'il apparaîtra sur les diagrammes lorsqu'il sera utilisé en tant que sous-programme. Le connecteur (Connector), accessible par le choix Show Connector du menu surgissant sur l'icône du « panneau avant », permet le passage des arguments du sous-programme vers le programme appelant. Il est conseillé d'écrire le nom du VI (ou un nom permettant sans ambiguïté de retrouver le VI) dans le carré réservé à l'icône. Ceci se fait à l'aide de l'éditeur d'icône accessible en mode d'édition, par le choix Edit Icon du menu surgissant de la palette icône/connecteur du « panneau avant ».

I.2.4 Outils de développement spécifiques au diagramme

La palette de boutons du diagramme offre cinq outils de développement spécifiques à cette fenêtre (voir Figure ci-dessous).

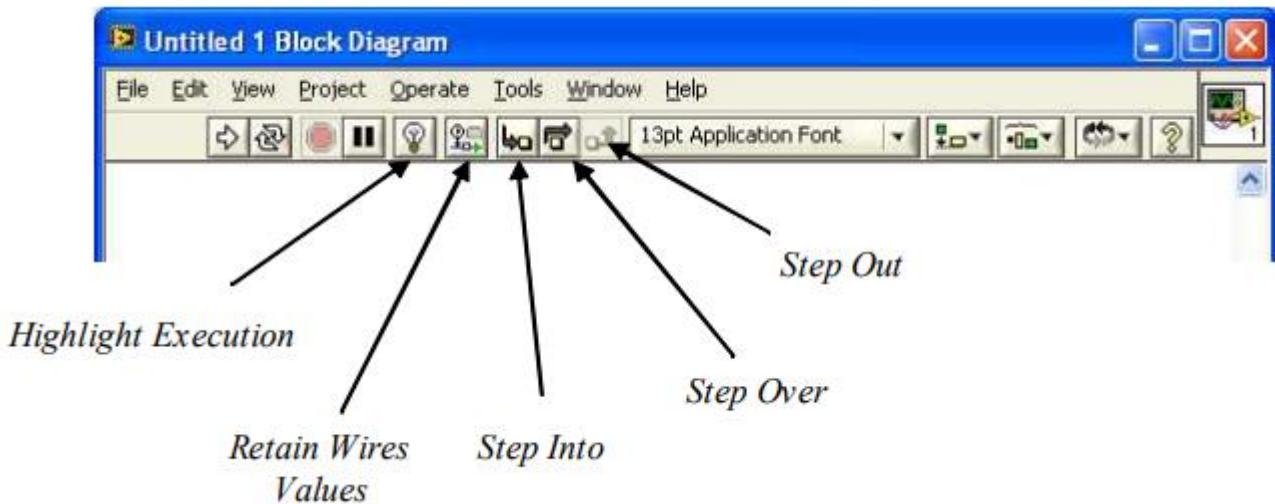


Figure I.7 : Outils spécifiques au diagramme.

Commandes spécifiques au diagramme :

- 1) **Exécution ralentie animée (Highlight Execution)** : Utile pour observer le déroulement du programme à l'aide de bulles simulant le passage des données sur les fils, mais en ralentissant considérablement l'exécution ;
- 2) **Etat de la mémoire (Retain Wires Values)** : Sert à visualiser les valeurs sur les fils reliant les VI en cours d'exécution ;
- 3) **Entrée en mode exécution pas à pas (Step Into)** : Sert à poursuivre, après une pause, l'exécution du programme pas à pas par pressions successives de ce bouton ;
- 4) **Saut de nœuds en exécution pas à pas (Step Over)** : Permet d'exécuter certains nœuds en mode normal et revenir au mode pas à pas à la sortie du nœud ;
- 5) **Sortie du mode pas à pas (Step Out)** : Permet de sortir du mode pas à pas.

Ces options sont très pratiques si on parvient à les maîtriser. Pour en savoir plus, voir l'aide en ligne sur le sujet.

I.2.5 Les palettes

Sous l'environnement LabVIEW, le programmeur dispose d'une palette d'outils et de deux palettes d'objets, apparaissant par défaut en fonction du contexte, selon qu'on travaille sur le « Face avant » ou sur le diagramme.

I.2.5.1 La palette d'outils

Une palette d'outils (Tools) utilisables aussi bien pour le diagramme que pour le « panneau avant » apparaît par défaut lorsqu'on ouvre un VI et reste néanmoins toujours accessible par le choix Tools Palette du menu View si on l'a supprimée. Elle permet de définir divers modes de fonctionnement du curseur :

Préparé par D.YD

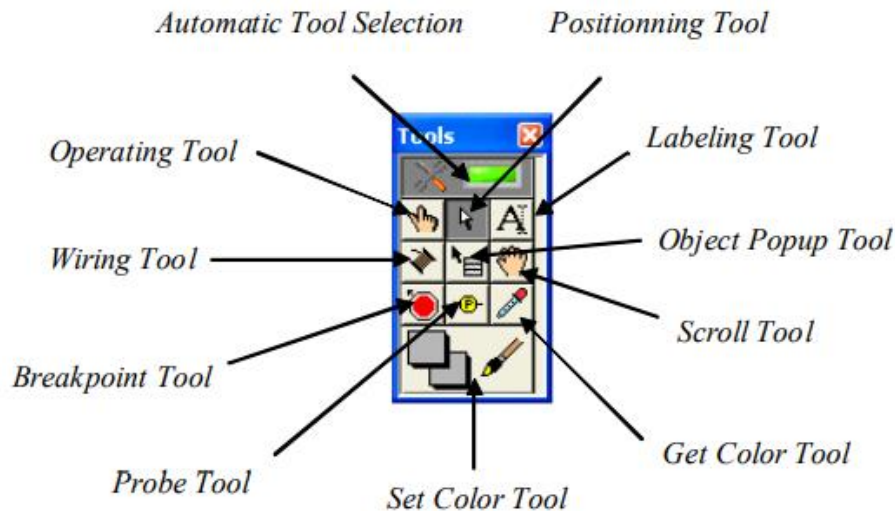


Figure 1.8 : Palette d'outils.

1) **Outils de développement :**

- a) **Breakpoint Tool** : Pour insérer des points d'arrêt ;
- b) **Probe Tool** : Pour insérer des sondes sur les fils permettant de connaître les valeurs transmises ;

2) **Outils d'édition :**

- a) **Operating Tool** : Sert à actionner les objets de la fenêtre ;
- b) **Positionning Tool** : Sert à sélectionner, déplacer, modifier les objets ;
- c) **Labeling Tool** : Utilisé pour entrer du texte ;
- d) **Wiring Tool** : Véhicule le flot de données entre les objets Object ;
- e) **Popup Tool** : Fait surgir un menu sur un objet ;
- f) **Scroll Tool** : Déplace la zone visualisée ;
- g) **Get Color Tool** : Sert à copier la couleur d'un objet ;
- h) **Set Color Tool** : Sert à colorier les objets et l'arrière-plan des fenêtres ;
- i) **Automatic Tool** : Selection Laisse à LabVIEW la possibilité de sélectionner lui-même les outils de la palette à utiliser selon le lieu où se trouve le curseur (Positionning Tool, Wiring Tool, ..).

I.2.5.2 La palette d'objets du « Face avant » (Controls Palette)

Cette palette apparaît par défaut lorsque la fenêtre de la « Face avant » est active et reste accessible, si on l'a fermée, par le choix Controls Palette du menu View ou par le bouton de droite de la souris sur la « Face avant ».

Préparé par D.YD

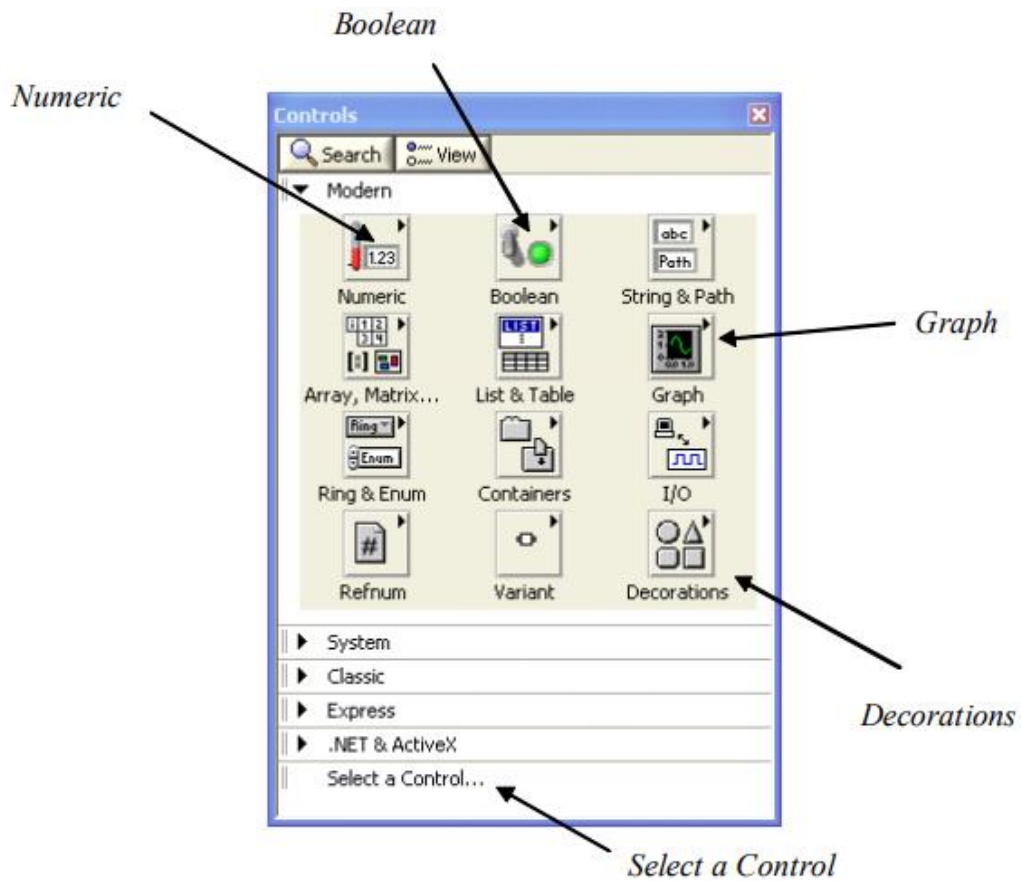


Figure I.9 : Palette d'objets du panneau avant (Controls Palette) apparaissant par défaut.

Les principaux choix offerts par cette palette fournissent des objets représentant les entrées-sorties du programme. Ils correspondent à des types prédéfinis du langage (Numeric, Boolean, String & Path, List & Tables, Array & Matrix, Graph, Refnum, ...), à des symboles de décoration divers (Decorations) ou à des types définis par l'utilisateur (Select a Control ...).

En fait, sur cette palette, les objets sont représentés plusieurs fois. La forme la plus synthétique de la Controls Palette est représentée sur la Figure (I.10). Schématiquement, on peut dire que les quatre choix « Modern », « System », « Classic » et « Express » représentent quatre styles graphiques différents, mais rien de plus.

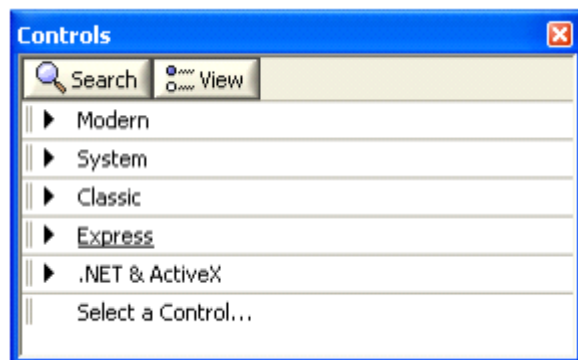


Figure I.10 : Forme la plus synthétique de la Controls Palette.

Les objets LabVIEW peuvent être pris dans l'un des quatre, seule leur apparence physique sur le « panneau avant » sera différente. Leurs fonctions seront identiques. Ceci est souligné sur la Figure (I.11) qui représente quatre objets identiques (des contrôleurs numériques) d'apparence différente, mais leur fonction intrinsèque reste la même.

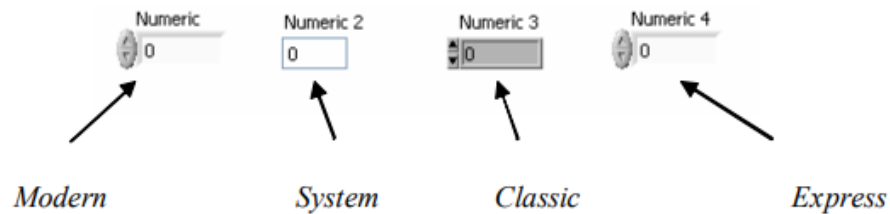


Figure I.11 : Les quatre types de contrôleur numérique disponibles.

I.2.5.3 La palette d'objets du diagramme (Functions Palette)

Cette palette apparaît par défaut lorsque la fenêtre du diagramme est active et reste accessible, si on l'a fermée, par le choix Fonctions Palette du menu View ou par le bouton de droite de la souris sur le diagramme.

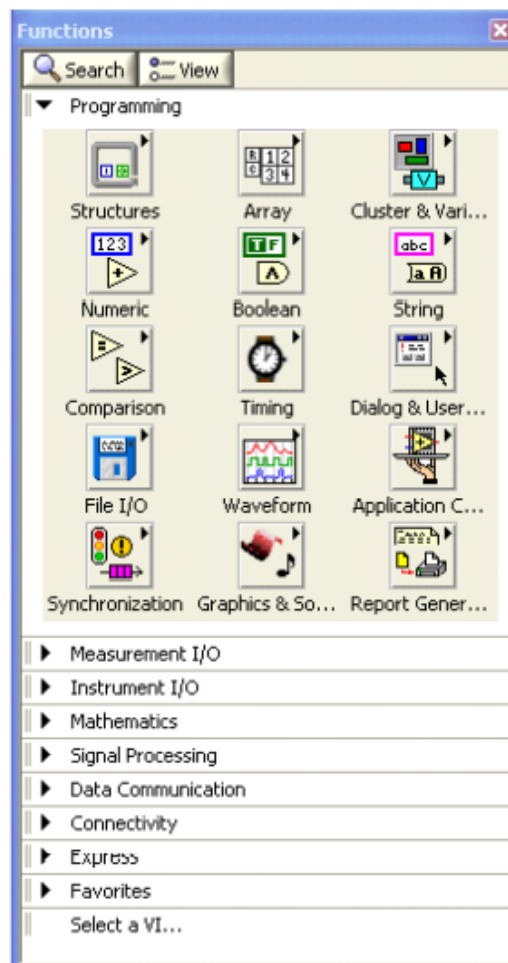


Figure I.12 : Palette d'objets du diagramme (Functions Palette) apparaissant par défaut.

Les principaux choix offerts par cette palette correspondent à des opérateurs ou des fonctions prédéfinies du langage. Sans panneau avant ni diagramme, les fonctions fournissent du code machine en ligne et sont pour la plupart polymorphes, c'est à dire qu'elles s'adaptent aux types (réels, entiers, tableaux de réels, tableaux d'entiers, ...) et représentations des données (décimales ou hexadécimales par exemple). La forme la plus synthétique de la Functions Palette est représentée sur la Figure (I.13). Cette palette regroupe toutes les fonctions qui peuvent être effectuées par LabVIEW. Elles seront détaillées par la suite.

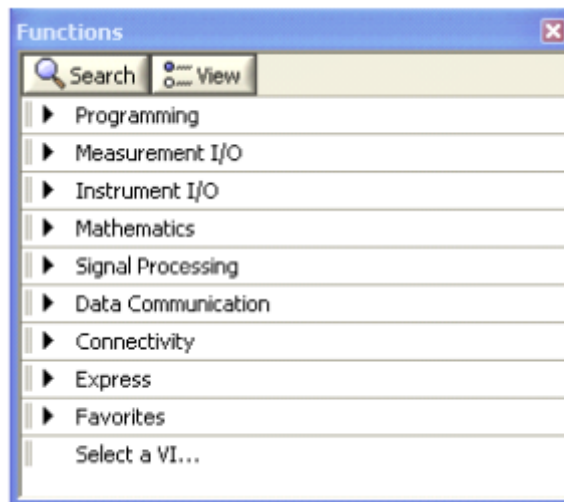


Figure I.13 : La forme la plus synthétique de la Functions Palette.

I.3 Exécution de programme VIconstruit sur LabVIEW

- 1) **Programme existant hors d'une bibliothèque :** Pour ouvrir un VI qui a été sauvé en dehors d'une bibliothèque, il suffit de l'ouvrir à l'aide de la souris, comme n'importe quel objet Windows.
- 2) **Programme existant au sein d'une bibliothèque :** Pour ouvrir un VI qui a été sauvegardé dans une bibliothèque LabVIEW (par exemple Demo.llb), il suffit d'ouvrir la bibliothèque à l'aide de la souris, comme n'importe quel dossier Windows. Il aura alors l'apparence de la figure (I.14). Les VI sont traités par Windows comme n'importe quel fichier. On peut les copier d'une bibliothèque vers une autre, les supprimer, etc. à l'aide des facilités standards de Windows. La seule restriction est qu'on ne peut mettre dans les bibliothèques LabVIEW que des objets reconnus par LabVIEW (les VI).

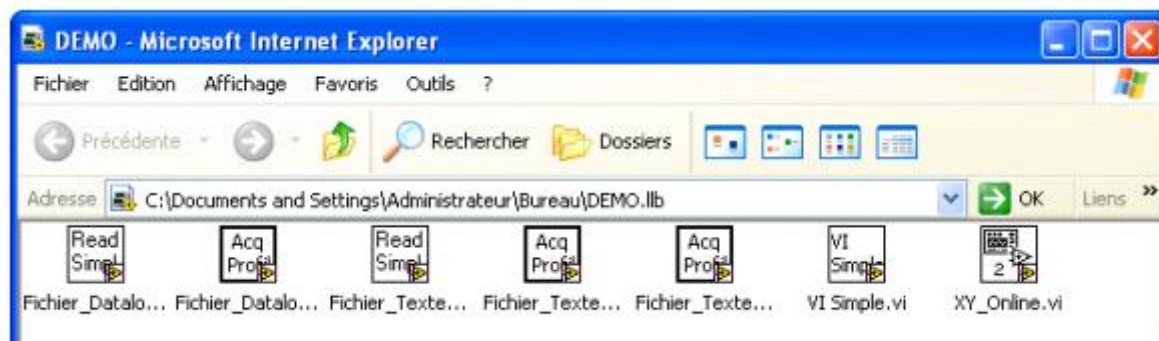


Figure I.14 : Programme sur la bibliothèque LabVIEW.

I.4 Sauvegarde des programmes VI sous LabVIEW

LabVIEW permet de sauvegarder un programme sous deux formes :

- 1) Sous forme d'un fichier indépendant. Le nom du fichier doit alors être conforme au système d'exploitation, sans limitation sur le nombre maximum de caractères (dans les limites du raisonnable). On pourra par exemple prendre « Nom de **VI.vi** » ;
- 2) Sous forme de constituant d'un fichier bibliothèque. **Ni** la longueur du nom du fichier **ni** celle de la bibliothèque ne sont limitées.

Pour des raisons de portabilité, de gestion par LabVIEW et d'économie d'espace disque, il faut privilégier les sauvegardes sous la deuxième forme. Les bibliothèques apparaissent sous Windows comme des dossiers d'extension « .llb » par exemple « **Demo.llb** ». La sauvegarde d'un programme est réalisée via les options **Save** ou **Save as** du menu **File** de la barre des menus. Si le chemin n'est pas correct, il convient de faire les choix convenables dans l'arborescence des répertoires et fichiers, à l'aide des menus déroulants de la fenêtre, choix du volume, des répertoires, ... etc.

❖ **Remarque :** Lorsque la sauvegarde doit être faite dans un fichier bibliothèque qui n'existe pas encore, on doit d'abord créer la bibliothèque par le choix Nouvelle bibliothèque de VIs de la fenêtre qui apparaît lorsqu'on sélectionne **Save** ou **Save as**. L'utilisateur doit taper le nom de la nouvelle bibliothèque (MaBiBli.llb) dans une fenêtre spécifique. LabVIEW présente une nouvelle fenêtre Name le VI avec le nom du nouveau fichier bibliothèque, afin de procéder à la sauvegarde du programme dans ce fichier.