

Objectifs :

- Présenter labVIEW et ses fonctionnalités
- Comprendre les composants d'un Instrument Virtuel (appelé VI)
- Travailler avec les tableaux, les graphiques, les clusters et les structures....
- Développer différentes architectures de programmation



Le langage de programmation graphique Labview développé par la société National Instruments est un outil de programmation à caractère universel particulièrement bien adapté à l'instrumentation. Couplé à des cartes d'E/S, il permet le contrôle d'un processus physique : commande d'instruments, acquisition de données, traitement des signaux, présentation des résultats,

Son intérêt réside dans le fait qu'il ne nécessite pas de connaissances approfondies de l'informatique (pas de lignes de texte, pas de problème de syntaxe !) et que sa mise en œuvre est très intuitive.

- LabView est un langage de programmation graphique.
- LabView est basé sur la notion de flot de données pour ordonnancer les opérations.
- LabView intègre l'acquisition, l'analyse, le traitement et la présentation de données.
- LabView supporte les standards réseaux, RS-232/422, USB, IEEE 488 (GPIB) et VXI/PXI, ainsi que les cartes d'acquisition de données.

I. Présentation des instruments virtuels :

Les programmes Labview portent la dénomination d'instruments virtuels ou VIs, car leur apparence et leur fonctionnement imite celui d'instruments réels tels que les oscilloscopes, les multimètres..... Chaque VI utilise des fonctions qui manipulent les entrées de l'interface utilisateur ou d'autres sources et qui affichent ces informations ou les déplacent vers d'autres fichiers ou ordinateurs.

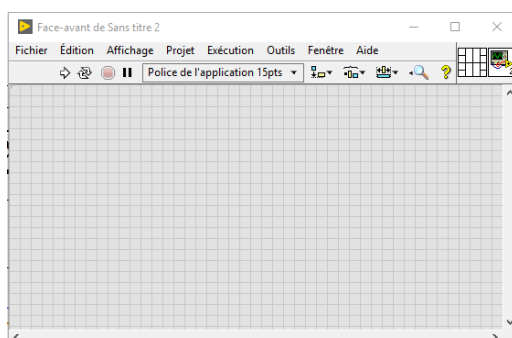
Un VI contient les trois composants suivants :

- **La face-avant:** Elle est construite à l'aide d'outils et d'objets et sert d'interface utilisateur. Elle réceptionne les données entrées par l'utilisateur et affiche celles fournies en sortie par le programme.
- **Le diagramme:** Il contient le programme de l'application construit à l'aide de représentations graphiques de fonctions. Elles contrôlent les objets de la face-avant qui apparaissent sous forme de terminaux.
- **Icône et cadre connecteur:** Il identifie le VI pour l'utiliser dans un autre VI. Un VI dans un autre VI est appelé un sous-VI.

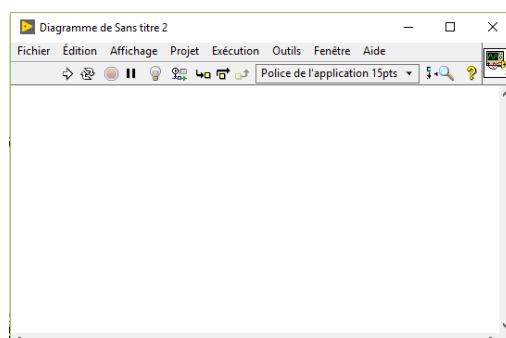
II. Les fenêtres de travail :

Le travail s'effectue dans deux fenêtres différentes : la face avant (ou interface utilisateur) et le diagramme. La commande clavier **ctrl T** permet de les faire apparaître simultanément. Elles s'activent d'un simple clic et dans chacune on trouve des menus déroulants (avec des fonctions proches de celles des applications fonctionnant sous Windows et d'autres spécifiques à Labview) et une barre d'outils.

Face avant



Diagramme



A. Face avant :

On note l'utilité des boutons principaux de la barre d'outils :



- Exécute le programme (si l'icône est signifie que le diagramme est mal câblé, en lançant l'exécution une liste d'erreurs est éditée).
- Indique que le VI est en cours d'exécution.
- Relance continuellement le VI après chaque fin d'exécution (équivalent à déposer le VI dans une boucle infinie). Les boutons stop ou pause arrêtent l'exécution.
- Suspend l'exécution du VI, l'icône devient rouge pour indiquer que le Vi est en pause, Appuyer de nouveau sur le bouton pour continuer l'exécution.
- 13pt Application Font Permet de modifier les caractéristiques du texte.
- Permet d'aligner des objets, d'en égaliser l'espacement et la taille.
- Cache ou affiche l'aide contextuelle.

B. Le diagramme :

La barre d'outils suivante est présente sur le diagramme, elle donne accès aux outils de mise au point et de présentation.



- Exécute le programme avec visualisation du flux de données (très utile pour rechercher l'origine d'erreurs).
- Entre dans une structure ou dans un sous VI, lors de l'exécution pas à pas. Chaque structure ou sous VI se met à clignoter lorsqu'il est prêt à être exécuté.
- Sauter l'exécution détaillée d'une structure ou un VI lors de l'exécution pas à pas.
- Termine l'exécution détaillée d'une boucle ou d'un sous VI.

C. Les palettes :

Elles permettent de créer et d'éditer les fenêtres de travail et sont disponibles dans le menu affichage.

- La palette d'outils,
- La palette de commandes,
- La palette de fonctions,

1. La palette d'outils :

Elle active aussi bien dans la face avant que dans le diagramme. Elle contient les outils nécessaires pour éditer et mettre au point les objets du diagramme et de la face avant. On note l'utilité des boutons principaux de cette palette.



Outil doigt : outil de commande qui permet notamment d'actionner les commandes de la face avant ou de faire varier leur valeur.



Outil flèche : outil de positionnement, de sélection et de redimensionnement des objets utilisés dans les 2 fenêtres.



Outil bobine : outil de câblage qui permet de connecter les icônes entre elles afin de constituer le diagramme.





Active la sélection automatique des outils. Dans ce mode, LabVIEW choisit l'outil adapté en fonction de l'emplacement du curseur (pas toujours très pratique). On peut lui préférer le raccourci clavier « touche **espace** », qui commute les deux outils des plus utilisés (sélection et doigt dans la face avant, sélection et bobine dans le diagramme) ou la « touche **Tab** » qui propose successivement les quatre outils les plus courants.



L'outil « édition de texte » permet de changer les étiquettes, d'éditer des objets de type caractère et de placer du texte libre dans une fenêtre.



Accède au menu contextuel par un clic à gauche (utilité douteuse !).



Pour se déplacer dans une fenêtre sans les ascenseurs.



L'outil « point d'arrêts » définit l'emplacement sur le diagramme où le programme passera en pause pour permettre le débogage.



L'outil « sonde » visualise la valeur de connections particulières.



L'outil « pipette » mesure la couleur d'un point.



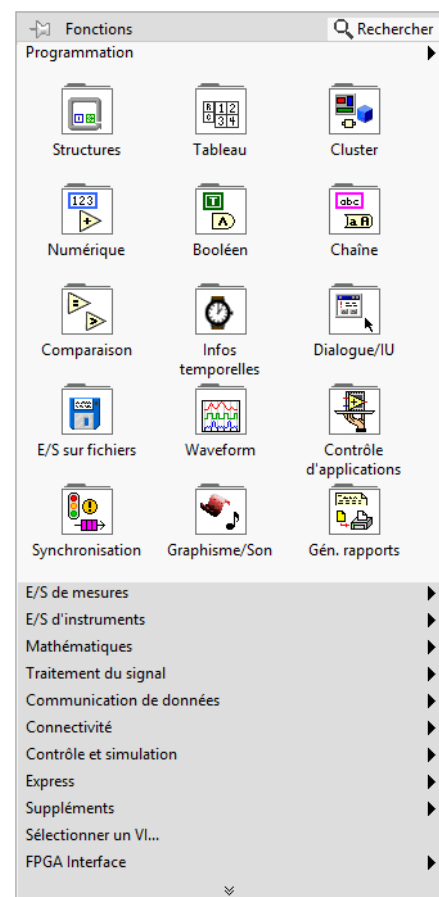
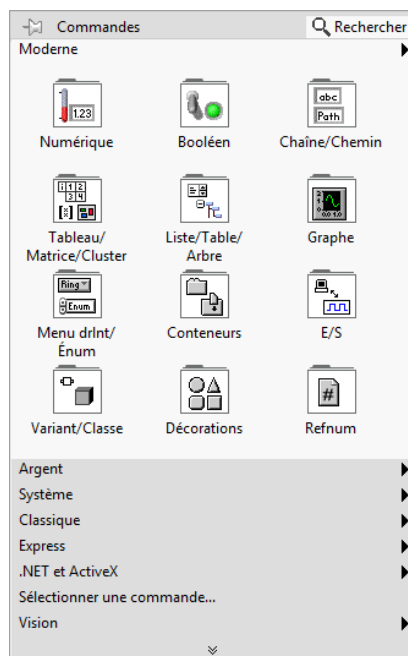
Le pinceau colorie un objet.

2. La palette de fonctions :

Elle active uniquement dans la face avant. Elle contient les commandes et les indicateurs qui servent à réaliser l'interface utilisateur. Les commandes et indicateurs sont situés dans des sous-palettes selon leur type.

3. La palette de commandes :

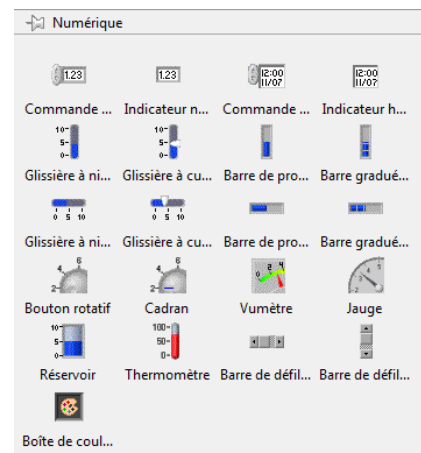
Elle active uniquement dans le diagramme. Elle contient les fonctions (et les sous Vis) utilisées pour construire le diagramme. Elles sont situées dans des sous-palettes selon leur type.



III. Les données et les flux de données :

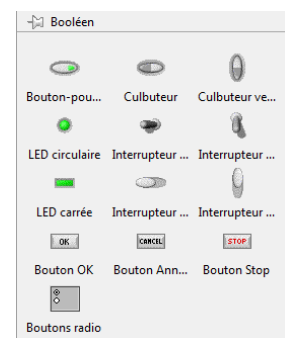
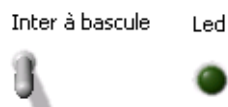
A. Contrôles et indicateurs numériques

Pour saisir ou modifier la valeur d'un contrôle numérique, on peut utiliser les flèches d'incrément ou entrer une valeur avec l'outil texte.



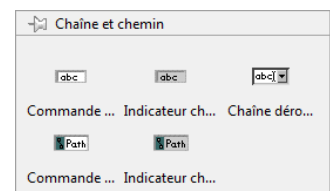
B. Contrôles et indicateurs booléens

A utiliser pour saisir et afficher des valeurs binaires. Les objets booléens les plus communs sont les interrupteurs à bascules et les diodes LED.



A. Contrôles et indicateurs Texte

A utiliser pour saisir et afficher les chaînes de caractères.



B. Les fils de données :

Les fils transfèrent les données dans le diagramme, ils sont ainsi analogues aux variables dans les langages textuels. Chaque fil provient d'une seule source mais peut être réuni à beaucoup de VIs ou de fonctions destinées à les traiter.

Type de la donnée	Scalaire	Tableau 1D	Tableau 2D	Couleur
Numérique				Orange (réel) Bleu (entier)
Booléen				Vert
Chaîne				Rose

IV. Les structures de programmation :

Labview est un langage de programmation, qui possède les structures (palette fonctions, sous palettes structures) de contrôle qui permettent d'exprimer tous les algorithmes. En pointant avec le curseur le bord de chacune des structures et un clic droit de la souris, on peut faire apparaître un menu surgissant permettant de la compléter.

- **Structure séquence** : qui permet de construire un ensemble de sous-diagrammes numérotés, qui seront exécutés séquentiellement dans l'ordre croissant des numéros.
 - **Structure condition** : qui permet de construire un ensemble de sous-diagrammes dont l'un sera exécuté selon la valeur du paramètre de choix (booléen ou numérique entier).
 - **Boucle FOR** : qui permet de contrôler des opérations répétitives. Elle exécute un sous-diagramme un nombre de fois prédéfini par la valeur de N avec un indice i qui varie par pas de 1 de la valeur initiale 0 jusqu'à la valeur finale $N-1$. Des registres à décalages permettent de transmettre des valeurs d'une itération à la suivante.
1. **Boucle WHILE**: qui répète le sous-diagramme situé à l'intérieur de la boucle jusqu'à ce que le terminal conditionnel d'entrée reçoive une valeur booléenne particulière. La valeur booléenne dépend de la condition de continuation définie. On retrouve également l'indice i et des registres à décalage.

