

**Introduction**

**à**

**LabVIEW**

Edition 2003-2004

R. Decourt

P. Dordor



<b>Chapitre 1</b>	<b><i>Introduction à LabVIEW</i></b>	<b>7</b>
<b>A.</b>	<b>LabVIEW</b>	<b>7</b>
<b>B.</b>	<b>Les Instruments Virtuels</b>	<b>7</b>
<b>C.</b>	<b>Environnement LabVIEW</b>	<b>9</b>
	Les fenêtres de la face avant et du diagramme	9
	Barre d'outils de la face avant	10
	Barre d'outils du diagramme	10
	Menus contextuels	11
	Menus	11
	Palettes	11
	Palette d'outils	11
	Palette de commandes	12
	Palette de Fonctions	13
	Chargement de VIs	13
	Enregistrement des VIs	14
<b>D.</b>	<b>Aide et Manuels</b>	<b>15</b>
	Aide contextuelle	15
	Aide LabVIEW	16
<b>Chapitre 2</b>	<b><i>Création, Modification, mise au point d'un VI</i></b>	<b>17</b>
<b>A.</b>	<b>Création d'un VI</b>	<b>17</b>
	Face Avant	17
	Contrôles et indicateurs numériques	17
	Contrôles et indicateurs booléens	17
	Configuration des contrôles et des indicateurs	17
	Diagramme	18
	Nœuds	18
	Terminaux	18
	Fils	18
	Câblage automatique des données	19
	Programmation par flots de données	19
	Recherche de contrôles, de VIs et de fonctions	20
<b>B.</b>	<b>Techniques de modification</b>	<b>22</b>
	Création d'objets	22
	Sélection d'objets	22
	Déplacement d'objets	22
	Effacement des objets	22
	Annuler /Rétablir	22
	Duplication des objets	22
	Nommer les objets	23
	Sélection ou suppression de fils	23
	Etirement des fils	23
	Fils cassés	23
	Modification de la police, du style ou de la taille du texte	24
	Modification de la taille des objets	24
	Alignement et distribution des objets	24
	Ordonner, Regrouper, Bloquer des objets	24
	Changer la couleur des objets	24
<b>C.</b>	<b>Techniques de mise au point</b>	<b>25</b>
	Recherche des erreurs	25
	Visualisation du flot de données	25
	Pas à pas	25
	Probes	26
	Points d'arrêt	26
<b>D.</b>	<b>Résumé et astuces</b>	<b>26</b>

Résumé	26
Conseils et astuces	26
Raccourcis	26
Modifications	27
Câblage	27
<b>Chapitre 3 Créer un sous VI</b>	<b>29</b>
<b>A. Sous VI</b>	<b>29</b>
<b>B. Icône et connecteur</b>	<b>29</b>
Création de l'icône	29
Définir le connecteur	30
Sélectionner ou modifier le motif du connecteur	31
Affectation de terminaux aux commandes et aux indicateurs	31
<b>C. Utilisation des sous VIs</b>	<b>33</b>
Ouverture édition de sous VIs	33
<b>D. Création d'un sous VI à partir d'une sélection.</b>	<b>34</b>
<b>Chapitre 4 Boucles et graphiques</b>	<b>35</b>
<b>A. Boucles de répétition conditionnelles (While)</b>	<b>35</b>
<b>B. Les graphes déroulants</b>	<b>36</b>
Câblage des graphes déroulants.	36
Action mécanique des interrupteurs	38
Gamme des données.	40
<b>C. Registres à décalage</b>	<b>41</b>
Initialisation des registres	42
<b>D. Boucles For</b>	<b>43</b>
<b>Chapitre 5 Tableau, Graphes et clusters</b>	<b>45</b>
<b>A. Les Tableaux</b>	<b>45</b>
Créer des tableaux sur la face avant.	45
Tableaux à deux dimensions	45
Tableaux de constantes	46
<b>B. Auto Indexation</b>	<b>46</b>
Boucles et tableaux 2D	46
<b>C. Fonctions sur tableaux</b>	<b>47</b>
Polymorphisme et tableaux	48
<b>D. Graphes fonction du temps et graphes XY</b>	<b>49</b>
Les Graphes	49
En mono courbe	49
Multi courbes	49
Les graphes XY	50
En mono courbe	50
Multi courbes	50
Graphe multi courbes	51
<b>E. Clusters</b>	<b>53</b>
Créer des clusters sur la face avant	53
Constantes de type Cluster	53
Ordonner les Cluster	53
<b>F. Fonctions sur Cluster</b>	<b>54</b>
Assembler & Assembler par nom	54
Désassemblage des clusters	55

<b>Chapitre 6</b>	<b><i>Structures de choix, séquences et nœuds de calcul</i></b>	<b>57</b>
<b>A.</b>	<b>Structure de choix</b>	<b>57</b>
	Tunnels d'entrée/sortie	57
	Exemples	57
	Structure de choix sur booléen	57
	Structure de choix sur entier	58
	Structure de choix sur chaîne	58
	Sélection des choix	58
<b>B.</b>	<b>Les Séquences</b>	<b>60</b>
	Variable locale de séquence	60
	Comment éviter l'utilisation des séquences	61
<b>C.</b>	<b>Boîtes de calcul et nœuds d'expression</b>	<b>61</b>
	Nœuds d'expression	61
	Boîtes de calcul	62
<b>Chapitre 7</b>	<b><i>Chaînes et Entrées/Sortie fichier</i></b>	<b>65</b>
<b>A.</b>	<b>Chaînes</b>	<b>65</b>
	Créer des commandes et des indicateurs de type chaîne	65
	Tables	66
<b>B.</b>	<b>Fonctions sur chaîne</b>	<b>66</b>
	Manipulation de chaînes	66
	Chaînes et Nombres	66
<b>C.</b>	<b>Entrées/Sorties sur fichier</b>	<b>68</b>
	Fonctions de haut niveau	68
	Fonctions de bas niveau	68
	Fonctions de base	68
	Gestion des erreurs	69
	Utilisation de VIs de haut niveau	70
<b>Chapitre 8</b>	<b><i>Acquisition de données</i></b>	<b>73</b>
<b>A.</b>	<b>Aperçu</b>	<b>73</b>
	Configuration matérielle	73
	Examen de la configuration	74
	Tester les E/S de la carte.	74
<b>B.</b>	<b>VIs d'acquisition</b>	<b>74</b>
	VIs simples	75
	VIs intermédiaires	75
	VIs utilitaires	75
	VIs avancés	75
<b>C.</b>	<b>Entrées analogiques</b>	<b>75</b>
	Nom de voies DAQ	75
<b>D.</b>	<b>Entrées analogiques type Waveform</b>	<b>77</b>
	Donnée type Waveform	77
<b>E.</b>	<b>Sorties analogiques</b>	<b>78</b>
	Génération de Waveform	78
<b>F.</b>	<b>Les compteurs et entrées/sorties analogiques</b>	<b>79</b>
<b>Chapitre 9</b>	<b><i>Contrôle d'instruments</i></b>	<b>81</b>
<b>A.</b>	<b>Généralités</b>	<b>81</b>
<b>B.</b>	<b>Configuration et Communication GPIB</b>	<b>81</b>
	Architecture du logiciel	81

Configuration du logiciel	82
<b>C. Communication avec les instruments</b>	<b>84</b>
Caractéristiques propres d'un appareil	84
Etapes essentielles d'une communication PC <=> périphérique	85
Sources d'erreurs classiques	86
<b>D. Bibliothèques de contrôle d'appareils dans LabVIEW</b>	<b>87</b>
Vi's spécifiques à l'interface GPIB ou RS 232C	87
Visa	87
Utilisation des VISA	88
<b>E. Driver d'instrument</b>	<b>89</b>
Exemple d'application simple	90
<b>F. Communications et configuration série</b>	<b>93</b>
Paramétrages possibles	93
Connexion matérielle	93
Visa et liaison série	94

# Chapitre 1 Introduction à LabVIEW

---

Ce chapitre présente les concepts de base de LabVIEW.

## Objectifs:

- A. Ce qu'est LabVIEW.
- B. Ce qu'est un instrument virtuel (VI).
- C. L'environnement LabVIEW.
- D. L'aide en ligne et les manuels.

## A. LabVIEW

---

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un langage de programmation dédié au contrôle d'instruments et l'analyse de données. Contrairement à la nature séquentielle des langages textuels, LabVIEW est basé sur un environnement de programmation graphique utilisant la notion de flot de données pour ordonnancer les opérations.

LabVIEW intègre l'acquisition, l'analyse, le traitement et la présentation de données.

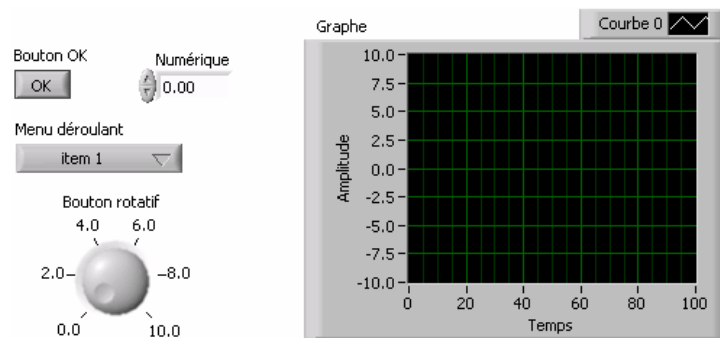
- Pour l'acquisition de données et le contrôle d'instruments, LabVIEW supporte les standards RS-232/422, IEEE 488 (GPIB) et VXI, ainsi que les cartes d'acquisition de données.
- Pour l'analyse et le traitement des données, la bibliothèque d'analyse étendue contient les fonctions pour la génération et le traitement de signaux, les filtres, les fenêtres, les statistiques, la régression, l'algèbre linéaire et l'arithmétique matricielle.
- LabVIEW intègre un grand nombre d'éléments de présentation tels les graphes déroulants, des graphes XY, des abaques de Smith, jauges, cadrans à aiguille...

## B. Les Instruments Virtuels

---

Les programmes LabVIEW s'appellent des *Instruments Virtuels* (VIs). Ces VIs ont trois parties principales : la *Face Avant*, le *Diagramme* et l'*Icône/Connecteur*.

La face avant d'un VI est avant tout une combinaison de *commandes* et *d'indicateurs*. Les commandes sont les entrées des VIs, elles fournissent les données au diagramme. Les indicateurs sont les sorties des VIs et affichent les données générées par le diagramme. Vous pouvez utiliser plusieurs types de commandes et d'indicateurs tels que les commandes et les indicateurs numériques, à curseur, booléens, chaîne de caractères, les tables et les graphes. Un exemple de face avant d'un VI est présenté ci-dessous :



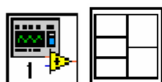
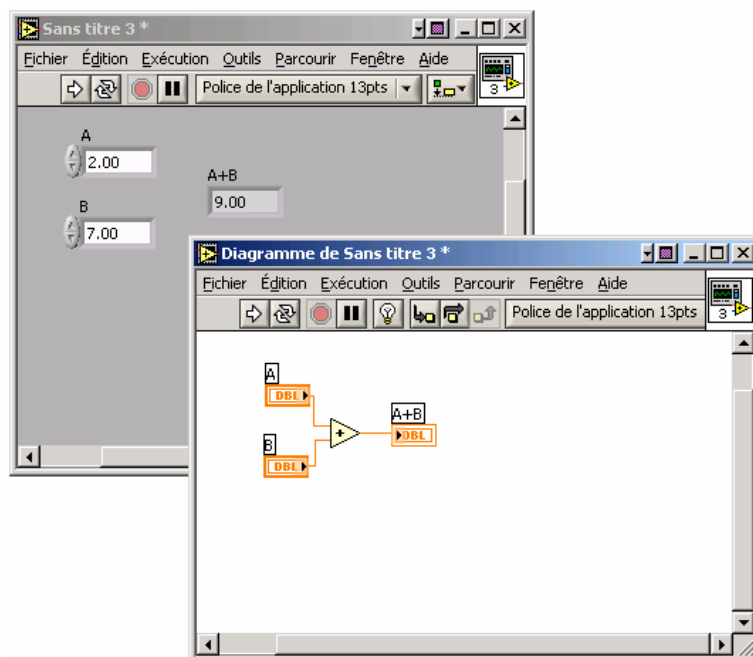
Vous construisez la face avant en plaçant des éléments graphiques accessibles dans une palette. Vous disposez de boutons, d'indicateurs numériques et de chaînes, de graphes, de LEDs, de listes déroulantes, de menus...



Chaque objet déposé sur la face avant génère dans le diagramme un symbole appelé *Terminal*. Ce terminal contient la *valeur* de l'objet graphique correspondant. Le symbole représente le type de la donnée (par ex. DBL pour double), le sens du flot, (maigre avec une flèche à droite s'il s'agit d'un indicateur, gras avec une flèche à gauche s'il s'agit d'une commande)

Le diagramme contient les terminaux, les sous VIs, les fonctions, les constantes, les structures ainsi que les fils qui relient les différents objets pour leur transmettre les données.

L'exemple suivant montre une face avant et le diagramme correspondant :

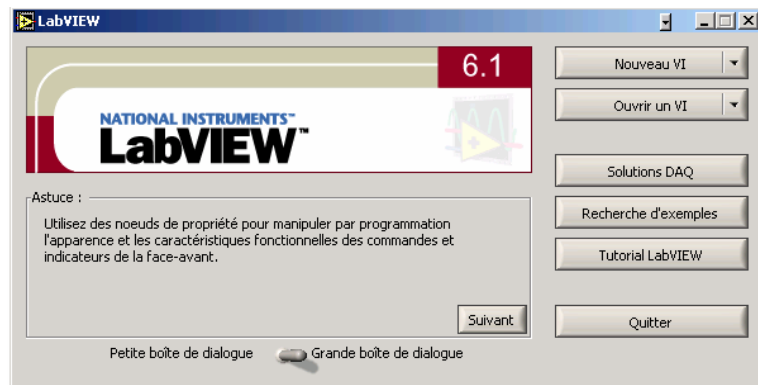


Après avoir construit la face avant et le diagramme, vous pouvez créer son icône et son connecteur. Si ce VI est utilisé dans un autre VI, il devient un *Sous VI*, il correspond à une routine dans un langage classique, le connecteur représente alors les paramètres entrant et sortant de la routine. L'icône identifiera le VI comme le ferait le nom de la routine. La hiérarchisation des applications facilite grandement la réutilisation du code et le débogage.



## C. Environnement LabVIEW

Au lancement de l'application, la boîte de dialogue suivante apparaît.

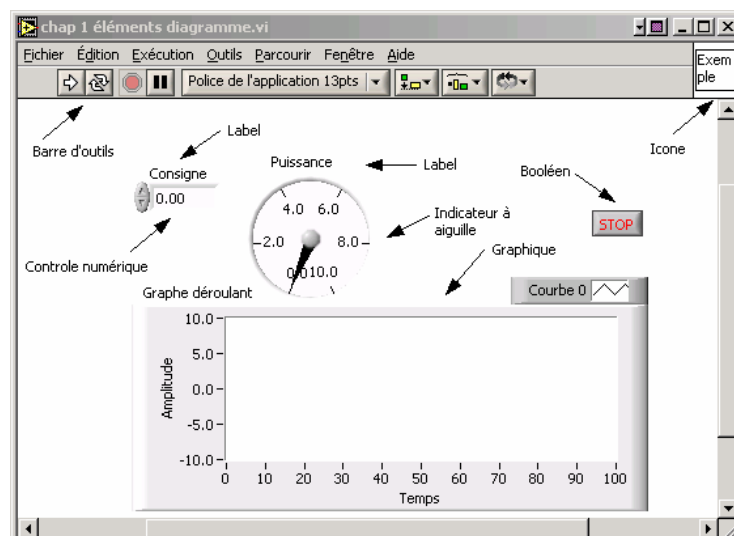


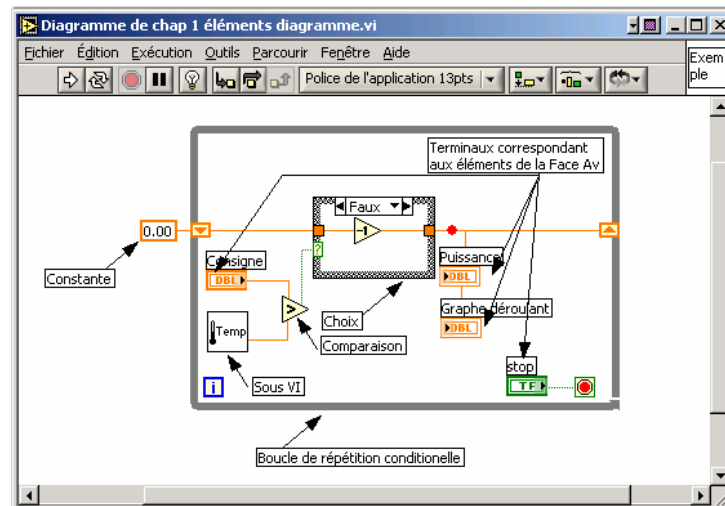
Elle permet les actions suivantes:

- **Nouveau VI** permet d'ouvrir une face avant et un diagramme vierge. Le fait de cliquer sur la flèche de la liste déroulante permet de créer d'autres types d'objets LabVIEW (des contrôles personnalisés, des variables globales, des VIs polymorphes...).
- **Ouvrir VI** permet d'ouvrir un VI existant, la flèche de la liste déroulante donne accès aux VIs récemment ouverts.
- **Solutions DAQ** charge un utilitaire destiné à simplifier le développement d'applications dédiées à l'acquisition de données par cartes d'entrées sorties.
- **Exemples** ouvre une boîte de dialogue permettant de trouver des exemples classés par thème.
- **Tutorial LabVIEW** ouvre le guide d'apprentissage.
- **Quitter** ferme LabVIEW.

### Les fenêtres de la face avant et du diagramme

Lorsque l'on ouvre un nouveau VI, une fenêtre de face avant et une fenêtre de diagramme apparaissent. L'illustration suivante présente les principaux éléments de ces deux fenêtres.





### Barre d'outils de la face avant

La barre d'outils suivante est présente sur la face avant, elle donne accès aux outils d'exécution et de présentation.



Lance l'exécution du VI.



Indique que le VI est en cours d'exécution et qu'il s'agit d'un VI de niveau supérieur (il n'a pas été appelé par un autre VI).



Indique que le VI est en cours d'exécution et qu'il s'agit d'un sous VI (il a été appelé par un autre VI)



Le bouton d'exécution apparaît brisé : le VI n'est pas exécutable (il contient des erreurs).



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.



Arrête l'exécution du VI.



**Note** Le bouton stop ne doit servir d'arrêt que lors des phases de mise au point, il faut prévoir une structure de programmation capable de finir correctement l'exécution du VI !



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.



Permet de choisir la fonte, la graisse, la couleur...d'un champ contenant du texte.



Permet d'aligner des objets.



Permet d'égaleriser l'espace entre objets.



Change le plan d'un objet, pour permettre des superpositions.

### Barre d'outils du diagramme

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





Fait apparaître le flot de données sur les fils et sur les connexions des VI.



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.



Saute l'exécution détaillée d'une structure ou un VI lors de l'exécution pas à pas.



Termine l'exécution détaillée d'un boucle ou d'un sous VI.



Dénote un problème d'exécution potentiel, mais n'empêchant pas le programme d'être exécuté. Cette option n'est pas naturellement active, on y accède par **Outils»Options»Débogage**.

## Menus contextuels

La plus part des objets présents dans les fenêtres de face avant et de diagramme possèdent des menus contextuels accessibles par un clic droit de souris.

## Menus

Une partie des menus contient des fonctions classiques d'enregistrement, d'édition, de changement de fenêtres... . D'autres sont spécifiques à LabVIEW.



**Note** : les menus sont parfois inhibés lorsque le VI est en cours d'exécution.

- **Fichier** donne accès à l'ouverture, l'enregistrement ou l'impression de VIs.
- **Edition** permet les copier/coller, la recherche, la création de menus...
- **Exécution** exécute, arrête, accède aux options d'exécution d'un VI.
- **Outils** facilite par le biais d'outils spécifiques, la communication avec des instruments, l'édition de bibliothèques, la comparaison de VIs, la configuration du serveur WEB...
- **Parcourir** facilite la navigation dans un VI et dans sa hiérarchie.
- **Fenêtre** accède aux fenêtres de l'application et aux palettes d'outils.
- **Aide** affiche l'aide, donne accès aux exemples...

## Palettes

LabVIEW possède trois palettes flottantes respectivement nommées: **Outils**, **Commandes**, et **Fonctions**.

## Palette d'outils

La palette d'outils existe sur le diagramme et sur la face avant. Elle permet de modifier des valeurs, des couleurs, mais aussi de câbler les entrées et les sorties des icônes entre elles, de poser des points d'arrêt, des sondes...

On y accède par Fenêtre»Afficher la palette d'outils.




**Note** : Il est possible de faire apparaître temporairement la palette d'outils en appuyant sur la touche maj. et en faisant simultanément un clic droit.




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 « doigt » change la valeur d'une commande, sélectionne un texte. Le curseur adopte la  forme lorsqu'il est placé dans une zone contenant des caractères.



L'outil « flèche » sélectionne, déplace, redimensionne les objets. Il adopte l'une des formes suivantes lorsqu'il est sur l'angle d'un objet  redimensionnable.



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.



L'outil « bobine » sert au câblage des VI's.



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.

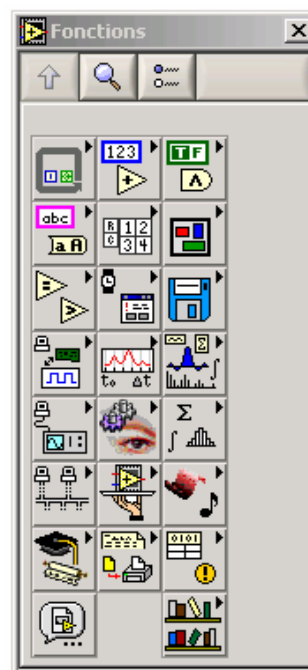
## Palette de commandes

La palette de commandes est disponible dans la fenêtre de face avant. Elle apparaît fugitivement lors d'un clic droit dans un endroit vierge de la fenêtre, ou par **Fenêtre»Afficher la palette de commandes**. Elle contient tous les éléments graphiques disponibles pour créer l'interface utilisateur. Ceux-ci sont hiérarchisés par type de données ou par grandes familles d'objets. Vous apercevez ci-dessous la palette de commandes.



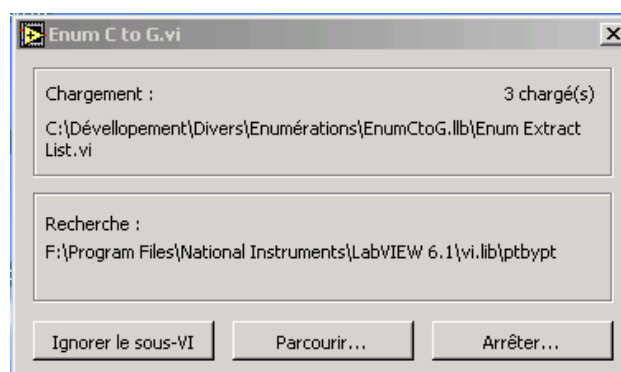
## Palette de Fonctions

Elle est accessible dans le diagramme par les mêmes méthodes que celle de commandes. Elle contient l'ensemble des fonctions de LabVIEW.



## Chargement de VIs

Lors du chargement la boîte de dialogue suivante apparaît (parfois très furtivement).



Le champ **Chargement** du panneau ci-dessus affiche le VI en cours de chargement et comptabilise les VIs en mémoire.

Si LabVIEW ne trouve pas un sous VI, il le cherche dans les chemins spécifiés dans le menu **Outils»Options»Item Chemin** du menu déroulant.

## Enregistrement des VIs

Utilisez les commandes **Enregistrer**, **Enregistrer sous**, **Enregistrer tout** ou **Enregistrer avec options** du menu **Fichier** pour enregistrer vos VIs. Il est possible de créer des bibliothèques d'extension .llb regroupant plusieurs VI. Cette fonctionnalité date de l'époque des noms limités à 8 caractères par MSDOS, National Instruments recommande actuellement d'enregistrer les VIs individuellement dans une architecture répertoire/sous répertoire.

LabVIEW utilise des boîtes de dialogues propres à NI pour les accès fichiers (pour une compatibilité avec d'autres plateformes), cette option peut être désactivée dans le menu **Outils»Options»Item Divers** du menu déroulant.

## Exercice 1-1 : Simulation de réponse en fréquence

Objectif: Ouvrir et exécuter un VI.

1. Démarrez l'application LabVIEW.
2. Cliquez sur le bouton **Recherche d'exemples**. Une boîte de dialogue affiche l'ensemble des exemples disponibles.
3. Dans l'onglet **Parcourir**, chargez **Frequency Response VI** dans l'arborescence **Apps, Freqresp.llb**.

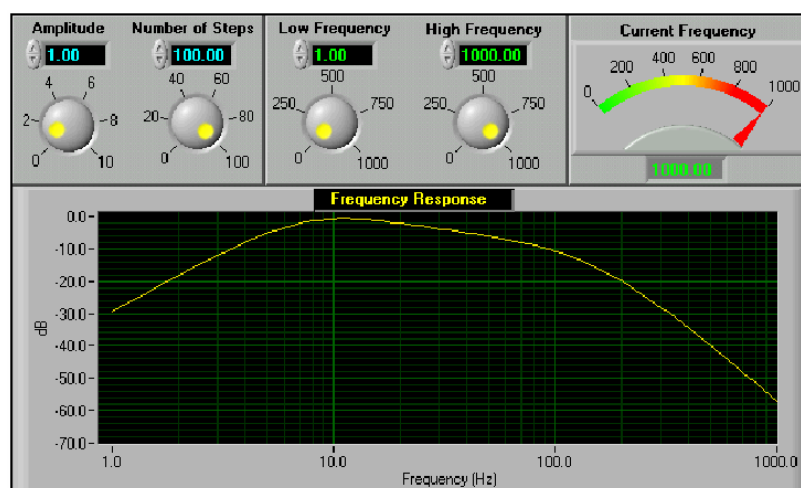


**Note** Vous accédez aussi aux exemples par le menu **Fichier»Ouvrir** puis en se déplaçant vers `labview\examples\apps\freqresp.llb\Frequency Response.vi`.

## Face Avant



4. Exécutez le VI en cliquant sur le bouton **Run**. Ce VI simule un test de réponse en fréquence d'un filtre, et affiche la courbe de réponse sur l'écran.



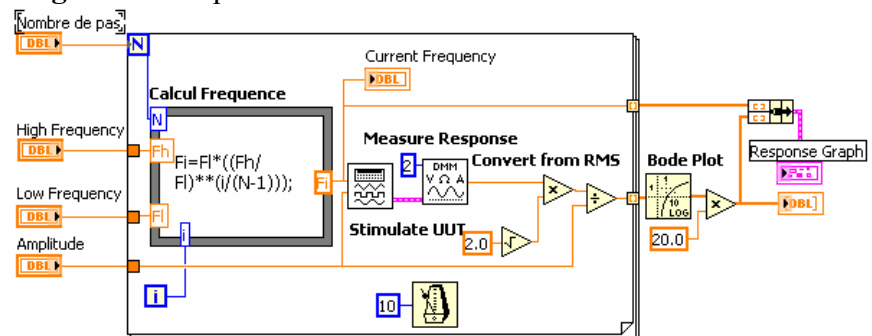
5. Utilisez l'outil Doigt pour changer la valeur des commandes, soit en tournant les boutons, (en cliquant sur la marque d'index et en faisant tourner), soit en cliquant sur les flèches à gauche des commandes, ou en entrant une valeur numérique dans l'afficheur numérique.



6. Lorsque vous entrez une valeur numérique, celle-ci n'est prise en compte qu'après validation par le bouton **Entrer**, ou par la touche **Entr** du pavé *numérique* (pas la touche **Entrée** de la zone alpha qui introduit un retour chariot s'il s'agit d'une zone texte).
7. Relancez le VI de nouveau pour constater les changements effectués.

## Diagramme

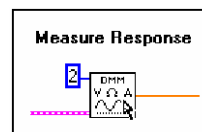
8. Visualisez la diagramme par le biais du menu **Fenêtre» Diagramme** ou par le raccourci <Ctrl-E>



Vous y voyez la plupart des éléments disponibles sur un diagramme, des fonctions, des sous VIs, des structures, des constantes... Ces notions seront détaillées dans ce cours.



9. En double-cliquant avec l'outil de sélection sur le sous VI suivant. Vous ouvrez la face avant du sous VI nommé Demo Fluke 8840A



La face avant est conçue pour ressembler l'appareil réel c'est pourquoi les applications LabVIEW sont nommées instruments virtuels. La hiérarchisation des applications en sous VIs permet d'améliorer la réutilisation du code.

10. Fermez le sous-VI Demo Fluke 8840A, et gardez le VI Frequency Response en mémoire.

## Fin de l'exercice 1-1

## D. Aide et Manuels

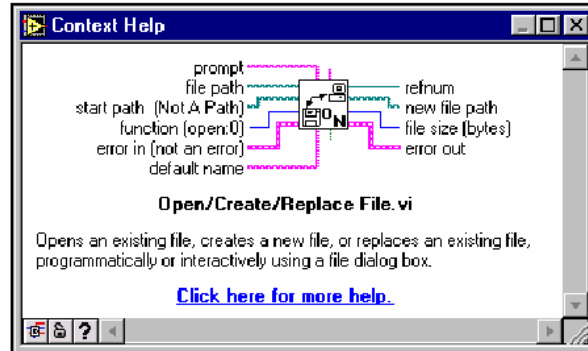
LabVIEW possède une aide classique, et une aide contextuelle.

### Aide contextuelle

Elle s'affiche soit par le menu **Aide»Aide contextuelle** soit par le raccourci <Ctrl-H>.

La fenêtre d'aide affiche continûment une information sur l'objet situé sous le curseur. Ce peut être une information sur la nature d'une liaison, le type de données à fournir à un indicateur, le fonctionnement d'un sous VI ou d'une fonction.

Ci-dessous un exemple de contenu d'une aide contextuelle.



Ce bouton affiche ou non les connections optionnelles d'un VI.

Ce bouton fige le contenu de la fenêtre (le déplacement de la souris ne met plus à jours la fenêtre).

Affiche l'aide classique (plus détaillée), associée à l'item correspondant au moment de l'appui.

## Aide LabVIEW

L'aide de LabVIEW détaille l'ensemble des fonctionnalités de l'application. National Instruments fournit également des tutoriaux et des manuels au format PDF. Le site [www.ni.com](http://www.ni.com) contient de nombreux exemples, des notes d'application, des liens...



# Chapitre 2 Création, Modification, mise au point d'un VI

Ou les bases de la création d'un VI.

## Objectifs:

- A. Comment créer des VIs
- B. Méthodes de modification
- C. Techniques de mise au point

## A. Création d'un VI

Les VI comportent trois parties – la face avant, le diagramme ainsi que l'icône et son connecteur. Se référer au chap. 3, Création d'un sous VI, pour plus d'information concernant l'icône et le connecteur associé.

### Face Avant

La face avant comporte des indicateurs et des contrôles qui sont les entrées et les sorties du VI.

Les boutons poussoirs ou rotatifs sont par défaut des contrôles.

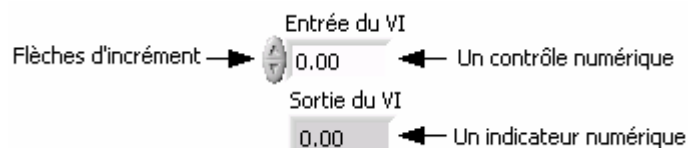
Les graphiques, voyants, vumètres sont par défaut des indicateurs.

Les contrôles simulent les entrées des instruments virtuels et fournissent les données au diagramme. Les indicateurs simulent la réponse des instruments et affichent les données acquises ou engendrées par les VIs.

Utilisez la palette de commandes pour placer contrôles et indicateurs sur la face avant.

### Contrôles et indicateurs numériques

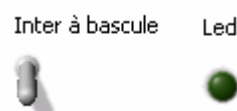
Les deux objets les plus couramment utilisés sont les contrôles et les indicateurs numériques, représentés ci dessous



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.

### 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.

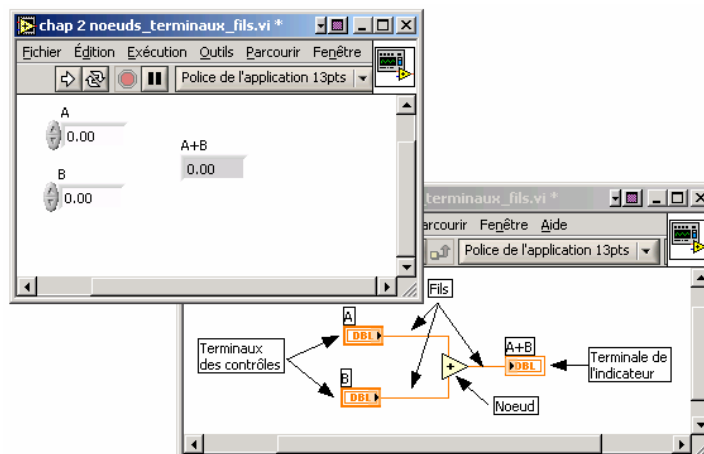


### Configuration des contrôles et des indicateurs

Pour configurer les contrôles et indicateurs, utilisez leur menu contextuel, accessible par un clic droit sur l'objet concerné.

## Diagramme

Le diagramme est composé de nœuds, terminaux, connexions comme le montre la figure suivante :



## Nœuds

Les nœuds sont des objets sur le diagramme. Ils possèdent des entrées et/ou des sorties et effectuent des tâches spécifiques lorsqu'un VI fonctionne. Ils sont équivalents à des fonctions dans les langages textuels.

## Terminaux



Les objets situés dans la face avant apparaissent comme des terminaux dans le diagramme. Les terminaux reflètent le type de données du contrôle ou de l'indicateur. Ainsi le terminal situé à gauche représente un contrôle numérique, défini comme un réel en double précision.

Les terminaux sont des ports de communication entre la face avant et le diagramme. Ils sont équivalents aux paramètres et aux constantes dans les langages textuels. Les données rentrées dans les contrôles de la face avant (A et B Fig. 2-1) entrent dans le diagramme par les terminaux. La fonction addition produit une nouvelle donnée qui arrive sur le terminal de l'indicateur et s'affiche dans la face avant.



Les connecteurs des opérateurs peuvent être visualisés en sélectionnant l'option **Terminals visibles** du menu contextuel.

## Fils

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

## Câblage automatique des données

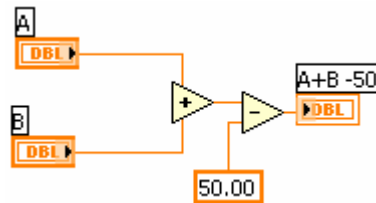
LabVIEW peut connecter automatiquement les objets lors de leur placement dans le diagramme. Lorsqu'un objet est approché près des autres, LabVIEW visualise provisoirement les connexions possibles. Lors du lâché, LabVIEW établit les connexions proposées.

Le mode de câblage automatique s'enclenche lorsqu'un objet est déplacé par l'outil de déplacement (flèche), la barre d'espace étant maintenue enfoncée. Les paramètres de câblage automatiquement sont définis dans le menu **Outils>>Options**.

## Programmation par flots de données

Utilisant le principe du contrôle de flot de données, LabVIEW n'exécute un nœud que lorsque l'ensemble des données arrivant sur ses entrées est présent. Après exécution, le nœud transmet les données sur ses sorties, les passant ainsi au nœud suivant.

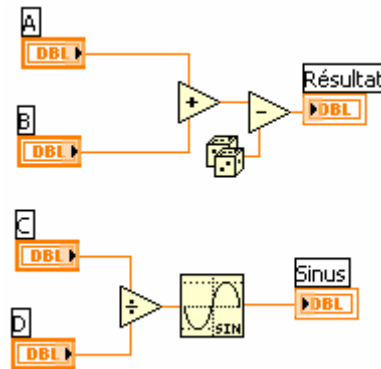
Considérons le diagramme suivant :



Il additionne deux nombres et soustrait 50 au résultat de l'addition. Dans ce cas, le diagramme s'exécute de la gauche vers la droite, non pas parce que les objets sont placés dans cet ordre, mais parce que l'une des entrées de la fonction « Soustraire » reste invalide tant que la fonction « Ajouter » n'a pas fini son exécution. Retenir qu'un nœud (fonction) s'exécute seulement quand les données sont disponibles à tous ses terminaux d'entrée et il ne fournit les données à ses terminaux de sortie qu'une fois l'exécution achevée.

Dans l'exemple ci-dessous quel segment de code devrait s'exécuter en premier : la fonction « Ajouter », la fonction « Nombre Aléatoire » ou la fonction « Division » ?

Impossible savoir car les entrées des fonctions Ajouter et Division sont validées au même moment et la fonction Nombre Aléatoire n'a pas d'entrée. Dans cette situation si un segment de code doit s'exécuter avant un autre et qu'il n'y a pas de dépendance entre les fonctions, il est nécessaire d'utiliser la structure Séquence pour donner un ordre d'exécution. (Cf. chapitre 6).



## Recherche de contrôles, de VIs et de fonctions

Les boutons de la palette de fonctions permettent de naviguer et de rechercher les contrôles, les VIs et les fonctions.



- Remonter—Remonte d'un cran dans la hiérarchie de la palette.
- Options—Ouvre l'option **Parcourir les fonctions** (Browser), à partir de laquelle vous pouvez reconfigurer les palettes.
- Rechercher —Change la palette en mode recherche. Celle-ci permet d'effectuer des recherches textuelles pour localiser les contrôles, les VIs, les fonctions dans les palettes.

## Exercice 2-1 : Conversion C to F VI

Objectif: Créer un VI.



Face avant

Créer un VI qui convertit un nombre, exprimé en degrés Celsius, en degrés Fahrenheit.

Dans les dessins illustrant le câblage, la flèche à la pointe de la souris indique où cliquer. Le nombre à l'extrémité de la flèche précise le nombre de clics à effectuer.

1. Choisir **Fichier»Nouveau** pour ouvrir une nouvelle face avant.



2. Sélectionner **Fenêtre»Mosaïque horizontale** pour afficher la face avant et le diagramme côte à côte
3. Créer un contrôle numérique pour saisir la température exprimée en degrés Celsius.
  - a. Sélectionner **Commande numérique** dans la palette **Numérique**. Si la palette « Commandes » n'est pas visible, effectuer un clic à droite dans une zone libre de la face avant pour l'afficher.
  - b. Ecrire deg C dans l'étiquette puis cliquer en dehors de celle-ci ou valider le raccourci Entrée, situé à gauche dans la barre d'outils ou bien frapper la touche **Entrée** du pavé numérique.



sinon LabVIEW utilise une valeur par défaut qu'il est possible de modifier en utilisant l'outil d'écriture.

4. Créer un indicateur numérique et l'utiliser pour spécifier la température exprimée en degré Fahrenheit
  - a. Sélectionner et placer un indicateur numérique.
  - b. Ecrire `deg F` dans la zone d'étiquette.



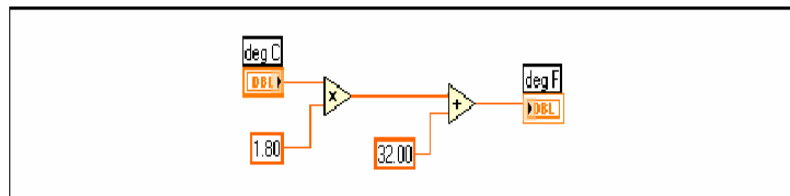
LabVIEW crée le contrôle et l'indicateur ainsi que les terminaux sur le diagramme. Le terminal est caractéristique du contrôle ou de l'indicateur associé. Ainsi, le terminal ci-contre représente un contrôle de type réel double précision.



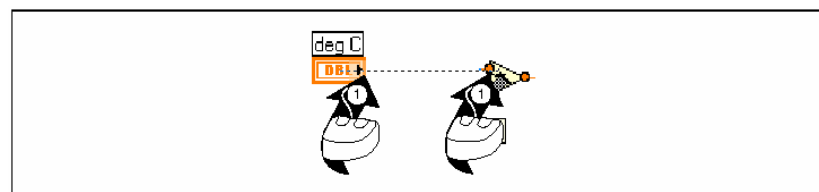
**Note :** Les terminaux des contrôles ont des listels plus épais que ceux des indicateurs.

## Diagramme

5. Afficher le diagramme (**Fenêtre»Afficher le diagramme**)



6. Choisir les fonctions Multiplication et Addition dans la palette **fonctions numériques**. Si celle-ci n'est pas visible, cliquer dans le diagramme dans une zone libre
7. Placer deux constantes numériques dans le diagramme. Lors de son placement initial, celle-ci est en surbrillance indiquant qu'une valeur peut y être directement saisie.
8. Saisir `1.8` dans l'une des constantes et `32.0` dans l'autre.
9. Utiliser l'outil de câblage pour réaliser le câblage du diagramme.
  - a. Pour relier un terminal à l'autre, cliquer à l'aide de la bobine sur le premier terminal, déplacer la jusqu'au second, cliquer dessus comme montré ci-dessous.



- b. Ancrer le fil à l'aide d'un clic de souris puis le courber à angles droits. Un appui sur la barre d'espace commute la direction proposée pour le câblage.
  - c. Pour identifier proprement les terminaux des nœuds, effectuer un clic droit sur les fonctions Multiplier puis Additionner. Sélectionner **Eléments visibles»Terminaux** pour afficher le connecteur. La même opération affiche à nouveau les icônes.
  - d. Quand la bobine passe sur un terminal celui clignote, indiquant qu'un clic effectuerait la connexion. De plus un message précisant le nom du terminal apparaît.

- e. Pour supprimer un fil en cours de création, appuyer sur <Echap>, cliquer à droite ou à gauche sur la source du fil.
- 10. Afficher la face avant
- 11. Enregistrer le VI car il servira plus tard.

a. Choisir **Fichier»Enregistrer**, Aller en c:\exercices\LV Basics I.



**Note** Enregistrer tous les VIs de ce cours dans le répertoire c:\exercices\LV Basics I.

b. Enregistrer sous le nom Convert C to F.vi.

- 12. Entrer un nombre dans le contrôle numérique et lancer le VI.



a. A l'aide de l'outil contrôle ou de l'outil d'écriture saisir un nouveau nombre. Essayer plusieurs nombres et relancer le VI.

## Fin de l'exercice 2-1

## B. Techniques de modification

Diverses méthodes permettent d'insérer des objets et de modifier les faces avant et les diagrammes

### Création d'objets

La création d'objets est possible à partir de la palette de contrôle mais également en effectuant un clic à droite sur le terminal d'un nœud puis en sélectionnant l'option **Créer**.

La suppression des contrôles et des indicateurs n'est possible que depuis la face avant. Les terminaux correspondants disparaissent alors automatiquement du diagramme.

### Sélection d'objets

Utiliser la flèche de position pour sélectionner un objet. Une fois sélectionné l'objet est entouré d'un pointillé. Pour sélectionner plusieurs objets, enfoncer la touche <majuscule> puis cliquer sur les objets concernés. On peut également cliquer dans une zone dépourvue d'objets, puis déplacer la souris jusqu'à entourer les objets à sélectionner.

### Déplacement d'objets

Pour déplacer un objet, le sélectionner puis le tirer jusqu'à la position choisie ou le déplacer à l'aide des flèches.

Le déplacement peut être limité à une seule direction (horizontale ou verticale) en appuyant sur la touche <majuscule> pendant la translation.

### Effacement des objets

Une fois sélectionné l'objet à supprimer, appuyer sur la touche <Suppr>.

### Annuler /Rétablir

En cas d'erreur d'édition d'un VI, on peut **Annuler** ou **Rétablir** les étapes précédentes en choisissant l'option correspondante dans le menu **Edition**. Le nombre d'actions pouvant ainsi être annulées ou revalidées est défini dans le menu **Outils»Options»Diagramme**.

### Duplication des objets

Sélectionner l'objet à dupliquer, enfoncer la touche <Ctrl>, puis tirer l'objet ainsi dupliqué vers sa position finale.

**Copier/Coller** permet d'obtenir le même résultat.

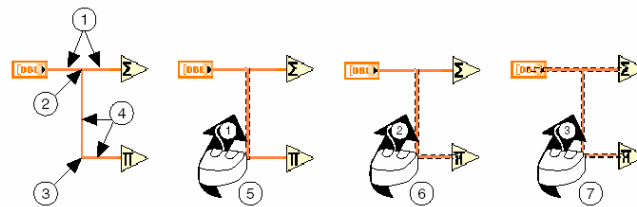
## Nommer les objets

Utiliser les étiquettes pour identifier les objets sur la face avant et le diagramme. Associées aux objets, celles-ci se déplacent avec eux, leur position vis à vis de l'objet est cependant modifiable.

Il est possible d'écrire dans des zones de texte, rattachées à aucun objet, pour insérer des commentaires. Pour créer une zone de texte, utiliser l'outil texte. En fin de message, cliquer dans le bouton <Entrée> de la boîte d'outil ou <Entrée> du pavé numérique ou <Majuscule>+ **Entrée**. La touche <Entrée> ajoute un saut de ligne dans la boîte de texte.

## Sélection ou suppression de fils

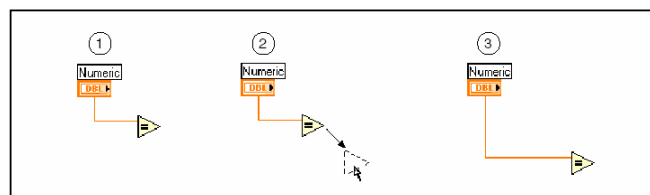
Un segment de fil est un tronçon horizontal ou vertical. Un coude est la jonction de 2 segments. Une branche est la zone de fil entre 2 jonctions ou 2 terminaux. Pour sélectionner un segment cliquer une fois dessus, pour une branche cliquer 2 fois, pour un fil entier 3 fois.



① Segment	④ Branche	⑥ 2 clics sélection d'une branche
② Jonction	⑤ 1 clic sélection d'un segment	⑦ 3 clics sélection de tous les fils
③ Changement de direction		

## Etirement des fils

Des objets câblés peuvent se déplacer avec leurs fils. Pour cela utiliser la flèche de positionnement comme indiqué ci-dessous.



## Fils cassés

Un fil cassé apparaît en pointillé. Cela arrive si l'on tente de relier 2 objets véhiculant des données incompatibles.



En plaçant la bobine de fil sur un fil cassé, un message apparaît décrivant l'origine du conflit. Pour supprimer un fil, effectuer un triple clic dessus et frapper la touche <Suppr>. Pour supprimer tous les fils cassés utiliser

le raccourci <CTRL>+B



**Attention** : un fil apparaît parfois en pointillé tant que le diagramme n'est pas complètement câblé.

### Modification de la police, du style ou de la taille du texte

Pour changer l'aspect d'un texte, utiliser le menu <**Police de l'application**> situé dans la barre d'outils.

### Modification de la taille des objets



La taille de la majorité des objets peut être changée. Après avoir pointé sur un objet avec la flèche de sélection, des poignées apparaissent dans les coins des objets de forme rectangulaire ou autour des objets circulaires. Lors de la modification de taille, la police reste la même. Etirer les objets jusqu'à obtenir la taille recherchée. En appuyant sur la touche <Majuscule> pendant l'étirement, les proportions de l'objet sont conservées.

### Alignement et distribution des objets



Pour aligner des objets sur un axe : sélectionner les puis utiliser le menu <**Aligner**> les objets dans la barre de menus. Pour les distribuer de façon homogène, utiliser le menu <**Distribuer**> de la barre de menus.

### Ordonner, Regrouper, Bloquer des objets



Si des objets se chevauchent, il est possible d'imposer qui est devant qui est derrière. Sélectionner un objet et imposer sa position en utilisant <**Réorganiser**> de la barre d'outils.

Il est possible de grouper et/ou verrouiller des objets depuis le menu <**Réorganiser**>. Les objets groupés gardent leurs dispositions relatives. Les objets verrouillés conservent leur emplacement sur la face avant et ne peuvent pas être supprimés.

### Changer la couleur des objets

On peut modifier la couleur de beaucoup d'objets, sauf ceux dont la couleur est spécifique : fils, représentation des données.

Pour les autres, choisir l'outil couleur et effectuer un clic droit sur l'objet (face avant ou diagramme). La couleur par défaut peut être modifiée en choisissant les menus **Outils>>Option>> Couleurs sur la barre de menus**.

De la même façon, il est possible de rendre les objets transparents.

## Exercice 2-2 : Edition d'un VI

Objectif: utiliser les outils d'édition.

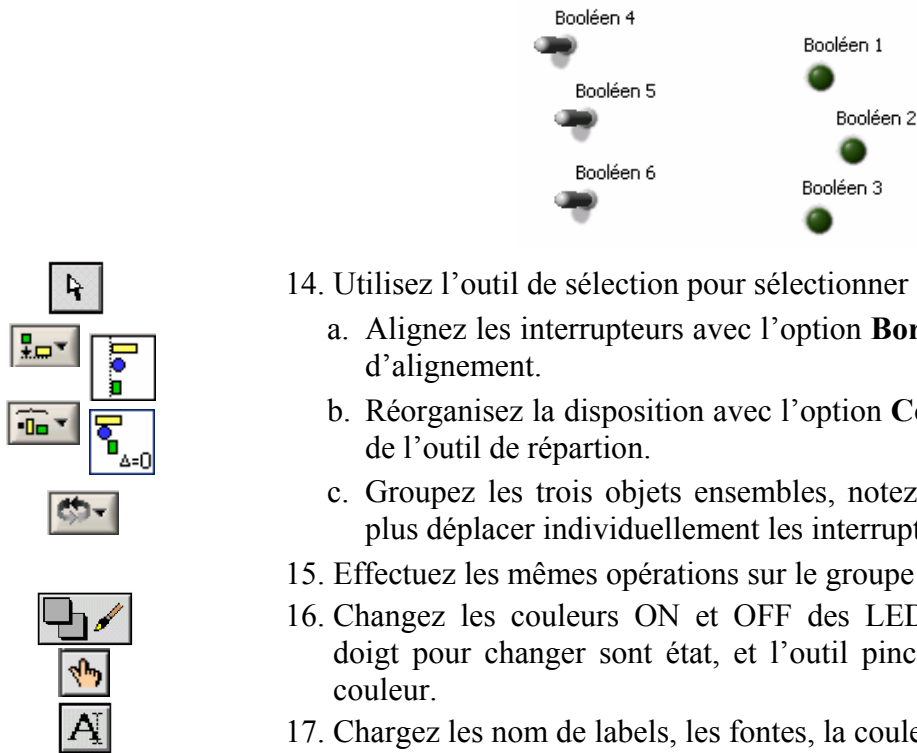


**Note** On peut toujours revenir en arrière par le menu **Annuler/Rétablir**.

### Face avant

13. Déposez sur la face avant trois interrupteurs à bascules et trois LED sans vous soucier de leurs labels ni de leurs positions. Vous pouvez utiliser le clonage avec la touche <CTRL>.





14. Utilisez l'outil de sélection pour sélectionner les trois interrupteurs.
  - a. Alignez les interrupteurs avec l'option **Bords gauches** de l'outil d'alignement.
  - b. Réorganisez la disposition avec l'option **Compression verticale** de l'outil de répartition.
  - c. Groupez les trois objets ensemble, notez que vous ne pouvez plus déplacer individuellement les interrupteurs.
15. Effectuez les mêmes opérations sur le groupe de LEDs.
16. Changez les couleurs ON et OFF des LEDs en utilisant l'outil doigt pour changer son état, et l'outil pinceau pour modifier sa couleur.
17. Chargez les nom de labels, les fontes, la couleur du texte.....

### Diagramme

18. Câblez chaque indicateur à chaque contrôle, écrire des commentaires en texte libre, essayez les fonctionnalités de câblage automatique présentées plus haut.

### Fin de l'exercice 2-2

## C. Techniques de mise au point



Si le bouton **Lancer** un VI semble cassé, comme montré ci contre, n'est pas exécutable. Si en fin de câblage la flèche est toujours cassée le VI ne fonctionnera pas

### Recherche des erreurs

Cliquer sur le bouton **Lancer** ou choisir le menu **Fenêtre»Visualiser les erreurs**. Un double clic sur la description de l'erreur met en surbrillance l'objet qui contient l'erreur.

### Visualisation du flot de données



Une visualisation du flot de données est obtenue en cliquant sur le menu **Animer l'exécution**. Cette option utilisée conjointement avec le mode pas à pas permet de visualiser les données transmises d'un nœud à l'autre.



**Note :** Cette visualisation réduit fortement la vitesse d'exécution VI.

### Pas à pas



Le mode pas à pas visualise intégralement le déroulement d'un VI ou sous VI. Ce mode affecte exclusivement un seul VI à la fois Pour rentrer dans ce mode cliquer sur le bouton **Commencer l'exécution en mode pas à pas**. Déplacer la souris sur **Commencer l'exécution**. Un message

annonçant la tâche à venir apparaît. On peut exécuter les sous VIs en pas à pas ou normalement.



Si le mode pas à pas est activé ainsi que la visualisation du flot, une flèche verte, en surimpression, apparaît sur les icônes des VIs en train de s'exécuter.

## Probes



Utiliser l'outil **Sonde** pour visualiser immédiatement la valeur qui chemine dans une connexion quand le VI fonctionne. En mode pas à pas ou après un point d'arrêt, il est possible de visualiser la valeur qui vient de transiter dans un fil. Les sondes disparaissent lorsque le VI est fermé.

## Points d'arrêt



Utiliser l'outil point d'arrêt, présenté ci contre, pour placer un point d'arrêt sur un VI, un nœud ou un fil du diagramme et arrêter l'exécution du diagramme en ce point.

Placé sur un fil, l'exécution stoppe après que la donnée y ait transité. Placé sur un diagramme, le VI stoppe son exécution une fois tous les nœuds exécutés. Quand un VI est mis en pause, LabVIEW l'entoure d'un listel rouge.

Positionner l'outil **Point d'arrêt** sur le point d'arrêt ou sur la structure concernée, pour l'enlever.

## D. Résumé et astuces

### Résumé

- Les contrôles et les indicateurs sont les entrées et sorties interactives du VI.
- La bordure d'un contrôle est plus épaisse que celle d'un indicateur.
- Pour transformer un contrôle en indicateur, choisir dans le menu contextuel **Changer en indicateur**.
- Utiliser l'outil doigt pour configurer les contrôles et les indicateurs
- Utiliser l'outil position (flèche) pour choisir, déplacer, modifier la taille des objets.
- Utiliser l'outil loupe pour effectuer des recherches dans les palettes de contrôles et de fonctions.
- Lorsque la flèche <Lancer> est cassée le VI ne s'exécute pas. Cliquer sur la flèche cassée, affiche les erreurs
- Utiliser la lampe d'animation, le pas à pas, les sondes et les points d'arrêt pour la mise au point.

### Conseils et astuces

La plus part des raccourcis combinent la touche CTRL avec une autre touche :

### Raccourcis

- <Ctrl-S> Sauve un VI.
- <Ctrl-R> Lance un VI.
- <Ctrl-E> Commute de la face avant au diagramme.

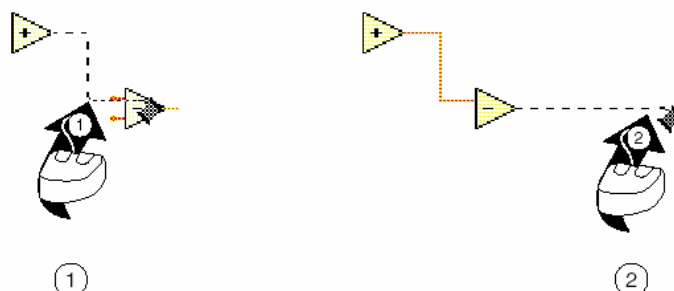
- <Ctrl-H> Affiche l'aide contextuelle.
- <Ctrl-B> Supprime les mauvais fils ( Bad wires).
- <Ctrl-F> Trouve les VIs, les textes, les objets chargés en mémoire ou dans une liste de VIs.
- Pour balayer les objets de la palette d'outils, presser la touche TAB.
- Pour passer de la flèche à la bobine, appuyer sur la barre d'espace.
- Pour incrémenter ou décrémenter rapidement des contrôles, appuyer simultanément sur la touche<**majuscules**> et sur les flèches du clavier numérique.

## Modifications

- Tirer et lâcher les contrôles et les indicateurs de la face avant vers le diagramme pour créer des constantes et vice-versa.
- Cloner un objet en appuyant sur la touche <Ctrl> et en le tirant jusqu'à sa position finale.
- Limiter le déplacement d'un objet en appuyant sur la touche <Maj.> lors de son déplacement.
- Maintenir la proportionnalité d'un objet lors de sa réduction en appuyant sur la touche <Maj.>.
- Pour retoucher symétriquement la taille d'un objet, appuyer sur la touche <Ctrl> en l'étirant.
- Pour remplacer plutôt qu'effacer, utiliser le menu <Remplacer> depuis le menu contextuel.
- Pour terminer la frappe d'une étiquette, taper < Maj.-Entrée> ou <Entrée> du pavé numérique.
- Pour cloner la couleur d'un objet utiliser l'outil **copier la couleur** (pipette de la palette outils) et repeindre la cible avec le pinceau.
- Pour revenir en arrière, choisir les menus Annuler/Rétablir.
- Pour créer de l'espace libre dans une zone du diagramme : enfoncer <CTRL> en même temps que l'outil flèche dessine un rectangle sur le diagramme.

## Câblage

- Pour savoir quels terminaux câbler, afficher l'aide contextuelle frapper <CTRL>+H. Les connexions obligatoires sont en gras, celles qui sont recommandées en normal, les autres en gris.
- Pour sélectionner tout ou partie d'un fil cliquer 1, 2 ou 3 fois.
- Pour couder un fil, l'ancrer par un clic de souris avant de changer de direction.



① Ancrer un fil en cliquant	② Ancrer et couper le fil par un double clic
-----------------------------	--

- Pour matérialiser la jonction des fils par des points, aller dans le menu **Outils>>Option>>Diagramme >>Points aux jonctions**.
- Pour bouger un objet d'un pixel utiliser les flèches du pave numérique. Pour le déplacer de plusieurs appuyer simultanément sur <Majuscules>.
- Pour supprimer un fil en cours de création appuyer sur <Echap>.

## Chapitre 3 Créer un sous VI

Ce chapitre décrit l'icône et le connecteur d'un VI puis comment l'utiliser en tant que sous VI dans un autre VI.

### Objectifs:

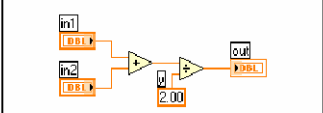
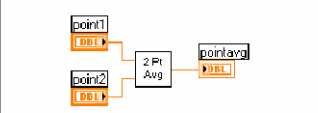
- A. Qu'est ce qu'un sous VI.
- B. Comment créer une icône et un connecteur.
- C. Comment utiliser un VI en tant que sous VI.
- D. Comment créer un sous VI à partir d'une partie d'un VI.

### A. Sous VI

Après avoir créé un VI, son icône et son connecteur, il est possible de l'utiliser dans un autre VI. Un VI inclus dans un autre VI est appelé sous VI. Il correspond à une fonction dans un langage textuel.

Le nœud d'appel du sous VI correspond à l'appel de la fonction. L'usage de sous VI augmente la lisibilité, la réutilisation de code et le débogage.

La comparaison entre le code ci dessous et le diagramme montre l'analogie entre les approches.

Function Code	Calling Program Code
<pre>function average (in1, in2, out) {   out = (in1 + in2) / 2.0; }</pre>	<pre>main {   average (point1, point2, pointavg) }</pre>
SubVI Block Diagram	Calling VI Block Diagram
	

### B. Icône et connecteur

Après avoir créé une face avant et son diagramme, il faut personnaliser l'icône associée et son connecteur.

#### Création de l'icône

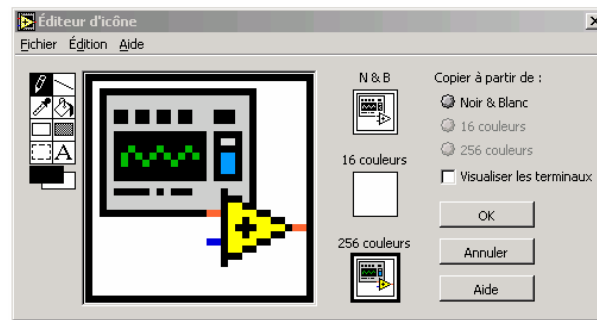
Chaque VI possède une icône dans le coin supérieur droit de la face avant et du diagramme. L'icône est la représentation graphique du VI. Si un VI est utilisé en tant que sous VI, il est identifié par son icône.



Éditer l'icône...

L'icône par défaut est représentée à gauche. Utilisez l'éditeur d'icône pour personnaliser l'icône en sélectionnant **Éditer l'icône** dans le menu contextuel de l'icône par défaut.

Utilisez les outils situés sur la gauche pour concevoir l'icône dans la zone d'édition. Une image de la taille réelle de l'icône apparaît dans l'un des cadres sur la droite de la zone d'édition.



Vous pouvez concevoir une icône différente pour les modes monochrome, 16 couleurs et 256 couleurs. Concevoir et sauvegarder chaque version d'icône séparément.

**Visualiser les terminaux** : affiche le modèle (l'implantation des broches) de terminaux du connecteur.

Les outils situés à gauche de la zone d'édition représentent les fonctions suivantes :



*Outil Crayon* Trace et efface pixel par pixel



*Outil Ligne* Trace des lignes droites. Appuyez sur <Maj>, puis faire glisser cet outil pour tracer des lignes horizontales, verticales ou diagonales



*Outil Pipette* Copie la couleur du premier plan d'un élément.



*Outil Pot de peinture* Remplit une zone avec la couleur du premier plan.



*Outil Rectangle* Trace une bordure rectangulaire de la couleur du premier plan. Double-cliquez sur cet outil pour encadrer l'icône dans la couleur du premier plan.



*Outil Rectangle plein* idem mais le rectangle est rempli avec la couleur d'arrière-plan. Double-cliquez : même fonctionnalité mais avec le fond rempli.



*Outil Marquise* Sélectionne une zone de l'icône à déplacer, à cloner, à effacer ou à modifier. Double-cliquez pour sélectionner toute la surface de l'icône.



*Outil Texte* Insère du texte dans le dessin de l'icône. Double-cliquez pour changer la police.



*Outil Plans* Affiche la couleur actuelle du premier et de l'arrière-plan. Cliquez sur chaque plan pour obtenir la palette de couleurs.

## Définir le connecteur



Pour utiliser un sous VI il faut définir un connecteur. Le connecteur est un ensemble de terminaux qui correspondent à tout ou partie des contrôles et indicateurs du VI à la façon des paramètres passés dans l'appel d'une fonction en langage textuel. Le connecteur définit ainsi les entrées/sorties du VI.

Pour visualiser les bornes du connecteur, effectuer un clic droit sur l'icône dans la face avant, choisir l'option **Montrer le connecteur**.


Chaque rectangle sur le connecteur représente une connexion. Le nombre de connexions par défaut est égal au nombre de contrôles et d'indicateurs de la face avant. Dans l'exemple ci-dessous, on distingue trois contrôles et un indicateur



### Sélectionner ou modifier le motif du connecteur

Pour sélectionner un autre connecteur, appeler le menu contextuel du connecteur et choisir **Modèles**. Il est souvent préférable de choisir une configuration avec un nombre de connexions plus important que nécessaire. Les connexions supplémentaires peuvent rester non câblées, cela permet d'effectuer des modifications sans bouleverser la hiérarchie du VI

Le nombre maximum de terminaux pour un sous VI est fixé à 28.

 **Note** Eviter de définir plus de 16 connexions sur le même VI. Trop de terminaux réduisent sa lisibilité du VI et sa facilité de mise en œuvre. Il est toujours possible de faire un cluster de plusieurs fils (mis dans une gaine). Dans ce cas on passe alors la gaine et non les fils individuellement. (Cf. Chapitre 5)


Pour modifier l'agencement des terminaux du connecteur, choisir l'une des commandes : **Basculement horizontal**, **Basculement vertical** ou **Rotation de 90 degrés**.

### Affectation de terminaux aux commandes et aux indicateurs

Après avoir choisi un modèle de connecteur, il faut le câbler. Placez de préférence les entrées à gauche et sorties à droite.

Pour faire correspondre un terminal à un contrôle (ou un indicateur) cliquer sur le terminal avec la bobine puis sur l'objet graphique correspondant. Le terminal prend la couleur du type d'objet auquel il correspond.

L'ordre dans lequel on clique sur les objets n'a pas d'importance.

 **Note** Bien que les liens soient établis en utilisant l'outil bobine, aucun trait ne relie le connecteur et les objets avec lesquels il communique.

### Exercice 3-1 : Conversion °C en degré F

Objectif: Créer une icône et un connecteur pour utiliser ce Vi en tant que sous VI

Suivre les étapes suivantes pour créer une icône et un connecteur pour le VI déjà existant qui transforme des °C en Fahrenheit

## Face avant

### 19. Ouvrir Convert C to F.vi



**Truc** L'appel du menu **Fichier»Fichiers récemment Ouverts** permet d'accéder aux derniers fichiers utilisés.

La face avant ci dessous apparaît.



20. Double cliquez sur l'icône située en haut, à droite de l'écran ou choisir **Editer l'icône**.

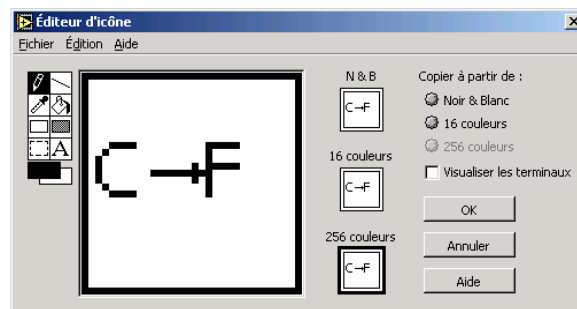


21. Double cliquez sur l'outil de sélection pour sélectionner l'icône par défaut et appuyer sur <Suppr> pour l'effacer.



22. Double cliquez sur l'outil rectangle pour redessiner le cadre.

23. Créez l'icône ci-dessous.



a. A l'aide de l'outil d'écriture ci-contre taper C→F. Un Double-clic sur l'outil écriture permet de changer de police de caractère.



b. Utilisez le cayan pour dessiner la flèche,



**Note** Pour dessiner des traits horizontaux ou verticaux, enfoncer la touche <Maj.> en même temps que le crayon est déplacé.

c. Utilisez l'outil de sélection et les flèches du pavé numérique pour déplacer les éléments de l'icône.

d. Choisissez l'icône Noir&Blanc (finale) et partir de la copie 256 couleurs, sauf si l'on dispose d'une imprimante-couleur.

e. Une fois l'icône terminée, cliquer su OK. L'icône apparaît en haut, à droite de l'écran (diagramme et face avant).



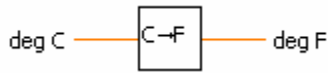
24. Cliquez à droite sur l'icône et choisir **Visualiser le connecteur** pour définir les terminaux. LabVIEW choisit une configuration basée sur le nombre de contrôles et d'indicateurs de la face avant. Celle-ci comporte 2 terminaux **degC** et **degF**, LabVIEW propose une configuration avec 2 connexions.

25. Assignez deux terminaux, l'un au contrôle, l'autre à l'indicateur.

a. Affichez l'aide contextuelle (Ctrl +H).

b. Cliquez sur le terminal de gauche, l'outil bobine apparaît automatiquement. La connexion se colore en noir.





- c. Cliquez sur **deg C**, le terminal devient orange et un pointillé entoure la connexion affectée.
- d. Cliquez dans une zone libre de la face avant. Le pointillé disparaît et la connexion se pare de la couleur de l'objet.
- e. Opérez de même avec l'indicateur Deg F.
- f. Déplacez le curseur sur le connecteur. La fenêtre d'aide montre 2 terminaux orange (réels) et leurs labels.

26. Sauvez le VI.

### Fin de l'exercice 3-1

## C. Utilisation des sous VIs



Une fois un VI créé, il est utilisable en tant que sous VI. Pour le placer dans le diagramme d'un autre VI, utilisez, dans la palette de fonctions, **Sélection d'un VI**. Si le VI est déjà ouvert vous pouvez, à l'aide de l'outil de sélection, cliquer dans son icône et la déposer dans le diagramme du VI appelant.

### Ouverture édition de sous VIs

Pour ouvrir la face avant d'un sous VI depuis un VI appelant, double cliquez avec l'outil sélection sur l'icône. Les changements que vous opérez n'affecteront que cette instance tant que le sous VI n'est pas enregistré. Une fois la modification enregistrée, elle affectera **toutes les instances dans tous les programmes** où ce VI est utilisé !

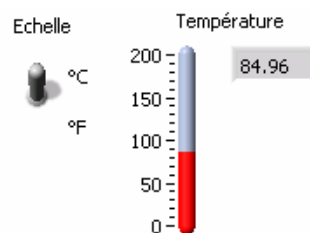
### Exercice 3-2 : Thermomètre VI

Objectif: Créer un VI utilisant un sous VI.

Vous disposez d'un capteur fournissant une tension proportionnelle à la température tel que  $T(^{\circ}\text{C}) = 100 \cdot V$ . On se propose de réaliser un thermomètre utilisant ce capteur et capable d'afficher la température en  $^{\circ}\text{C}$  et  $^{\circ}\text{F}$ .

### Face avant

1. Dans un nouveau VI créez la face avant :



- a. Le thermomètre est dans la palette **Numérique**.
- b. L'indicateur numérique est dans le menu contextuel **Eléments visibles»Indicateur numérique**.
- c. L'interrupteur à bascule est vrai en haut, on peut s'en assurer par le menu contextuel **Opération sur les données»Changer la valeur en...**

2. Il est possible de documenter le VI (cette description apparaîtra dans l'aide contextuelle) en sélectionnant dans le menu contextuel de l'icône **Propriété du VI** puis **Documentation** dans la liste déroulante. Entrez une brève description du VI dans le cadre.

### Diagramme

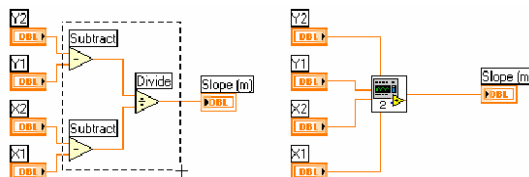


3. Vous utilisez les éléments suivants :
  - a. **Tutorial»Démonstration de lecture de tension** simule la prise de tension sur le capteur de température.
  - b. **Nomérique»Multiplier** pour le coefficient de proportionnalité de 100.0. La constante peut être placée directement en déroulant le menu contextuel de l'entrée y en en sélectionnant **Créer»Constante**.
  - c. **Sélection d'un VI**, pour placer votre VI Convert C to F.vi.
  - d. **Comparaison»Sélectionner** permet d'aiguiller l'une ou l'autre de ses deux entrées vers la sortie en fonction d'un booléen.
4. Enregistrez le VI sous le nom `Thermometer.vi`. Affichez la face avant et exécutez le code. Vous avez la possibilité d'exécuter en boucle le code sans structure particulière dans la phase de test en utilisant le bouton **Exécuter en continu** au lieu du bouton **Exécuter**.

### Fin de l'exercice 3-2

## D. Création d'un sous VI à partir d'une sélection.

Il est possible de créer un sous VI en sélectionnant une partie de diagramme puis en utilisant **Edition»Créer un sous VI**. LabVIEW crée automatiquement un sous VI contenant les contrôles et indicateurs nécessaires. D'un point de vue conceptuel il ne semble pas que cette fonctionnalité fournisse un code bien réfléchi !



### Résumé trucs et astuces

- Un VI appelé d'un autre VI est un sous VI, l'utilisation de sous VI rend le code plus fonctionnel et facile à déboguer.
- Un sous VI est identifié par son icône, les paramètres sont passés par son connecteur.
- Il est possible d'ajouter une description succincte des fonctionnalités de ses VI en sélectionnant propriétés dans le menu contextuel de l'icône.

## Chapitre 4 Boucles et graphes déroulants

Les structures de répétitions sont représentées graphiquement par un cadre, englobant un certain nombre de VIs et fonctions. Selon le type de cadre, il s'agira de répétitions conditionnelles, d'itération ou de choix.

Vous verrez dans ce chapitre l'utilisation des boucles While et For, et utilisez ces structures pour générer des graphiques.

### Vous apprendrez:

- A. Comment utiliser une boucle While.
- B. Comment afficher des données sur un graphe.
- C. Ce qu'est un registre à décalage, et son utilisation.
- D. Ce qu'est une boucle For.

### A. Boucles de répétition conditionnelle (While)



Identique à une structure de type Faire...Répéter tant que, la boucle While exécute le code contenu dans son cadre tant qu'une condition est Vraie ou Fausse. Cette boucle est située dans la palette **Fonctions»Structures**.

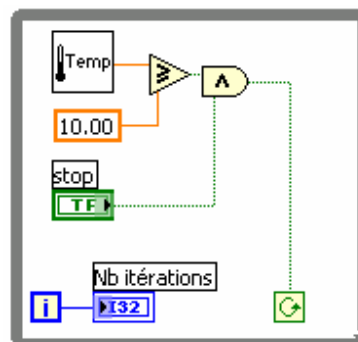
On utilise la souris pour définir la zone de code à répéter. Lors du relâchement du bouton de la souris, la zone sélectionnée se trouvant enfermée à l'intérieur de la structure sera répétée. Il est possible d'enrichir le code contenu dans la boucle en déposant d'autres VIs, fonctions ou structures.



La condition d'arrêt, présentée ci-contre, permet de contrôler l'exécution de la boucle. La condition par défaut est **Continuer si vrai**. D'un clic droit il est possible d'en inverser cette condition. L'icône prend alors l'allure présentée à gauche

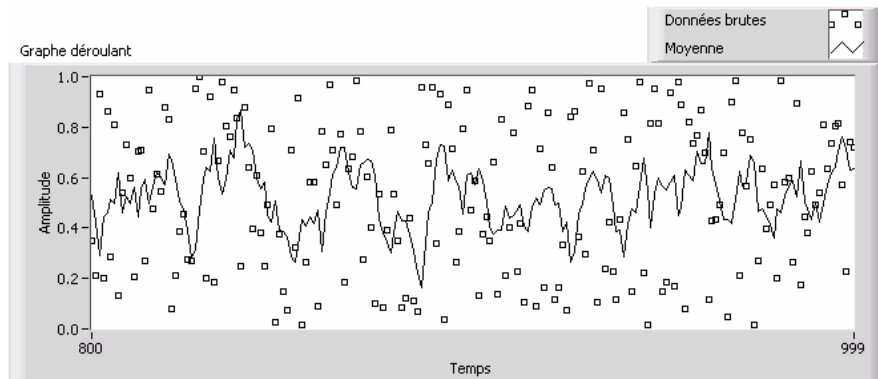
Le nombre d'itérations effectuées est contenu dans l'icône présentée à gauche. Cette variable est initialisée à 0 lors de la première itération.

L'exemple suivant présente une boucle dont l'exécution se poursuit tant que la sortie du sous-VI est inférieure à 10, ou que le bouton **stop** est faux (continuer tant que vrai  $\Leftrightarrow$  arrêter dès que faux).

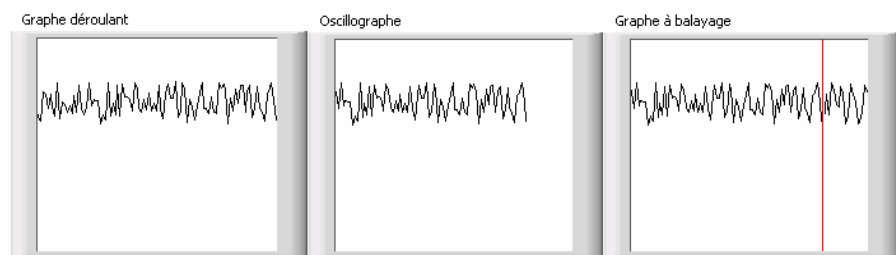


## B. Les graphes déroulants

Les graphes déroulants sont des indicateurs numériques destinés à afficher l'évolution d'une (ou plusieurs) variable(s) sur une (ou plusieurs) courbe(s). L'axe des X a donc pour échelle un numéro d'ordre d'arrivée du point. Les graphes déroulants sont situés dans la palette **Graphes**. L'exemple suivant vous présente un graphe multi courbes.



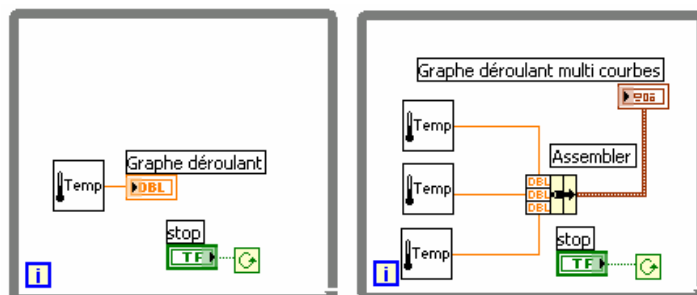
Ce type de graphe dispose de trois modes de rafraîchissement accessibles depuis le menu contextuel **Avancé»Mode de rafraîchissement**.



Le **graphe déroulant** est une fenêtre dans laquelle l'ensemble des données est décalé d'un point vers la gauche à chaque nouvelle arrivée. L'**oscillographe** efface complètement la fenêtre à chaque nouveau remplissage du buffer. Enfin le **graphe à balayage** efface les anciennes données à chaque nouvelle arrivée, une barre sépare les anciennes des nouvelles, façon écran radar.

### Câblage des graphes déroulants.

L'entrée d'un graphe déroulant est un scalaire. Le terminal d'entrée du graphe s'adapte automatiquement au type de données.



L'affichage de courbes multiples s'obtient en assemblant plusieurs courbes uniques à l'aide de la fonction **<Assembler>** située dans la palette **Fonctions»Cluster**.

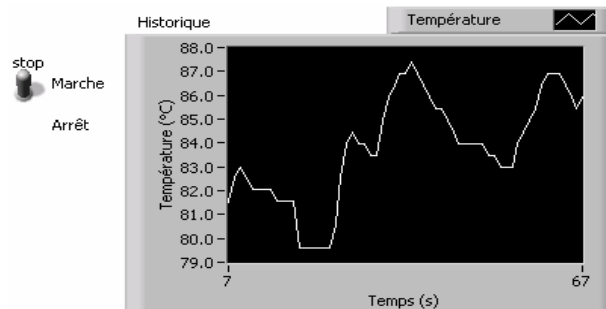
### Exercice 4-1 : Surveillance de température

Objectif : Utiliser une boucle While et un graphe déroulant pour faire une acquisition et afficher des données.

Réalisez les étapes suivantes pour simuler un suivi et une acquisition de température.

#### Face avant

1. Créez une nouvelle face avant et construisez cette interface.



- a. Prenez un interrupteur à bascule verticale, il servira à arrêter le programme. Il se situe dans la palette **Commandes** » **Booléen**. Positionnez le vers haut à l'aide de l'outil doigt, d'un clic droit. Imposez cette position comme position par défaut par défaut, depuis le menu contextuel : **Opération sur les données** » **Prendre la valeur actuelle par défaut**
- b. A l'aide de l'outil Texte, entrez `stop` comme label du bouton puis les commentaires `Marche` et `Arrêt`. La validation se fait en cliquant sur l'icône Entrée ou par la touche *Entr.* du pavé numérique
- c. Placez un graphe déroulant, modifiez les labels d'axe, de légende et de graphe en conséquence.
- d. Le VI est sensé mesurer la température de la pièce, changez (directement sur l'axe) du graphe les valeurs minimum et maximum de l'axe Y en 0 et 50

#### Diagramme

2. Passez à la fenêtre diagramme.
3. Incluez dans une boucle While les deux terminaux présents (Eventuellement les rapprocher avant). Pour cela :



- a. Sélectionnez la boucle While dans **Fonctions** » **Structures**.
- b. Cliquez et agrandissez le rectangle de sélection autour des deux terminaux.
- c. Utilisez l'outil de positionnement pour retoucher la boucle si nécessaire.



4. Positionnez le VI ci-contre que vous trouverez dans **Fonctions» Tutorial» Thermomètre Numérique**.
5. Connectez la sortie du VI au graphe déroulant. Pour obtenir une température en °C, il faut connecter une constante booléenne Vraie à l'entrée du VI, accessible par un clic droit sur son entrée. Choisissez l'option **Créer» Constante**.
6. Enregistrez ce VI sous le nom *Surveillance de température.vi* vous le réutiliserez plus tard.
7. Lancez l'exécution en cliquant sur la flèche de la barre d'outils ou en tapant <ctrl> r.  
Le programme exécute la section de code à l'intérieur de la boucle tant que la condition d'arrêt est VRAIE.
8. Basculez l'interrupteur en position Arrêt pour mettre fin au VI.
9. Vous pouvez modifier l'aspect des graduations, la mise à l'échelle, le format des points, la grille... par un clic droit sur le graphe, à vous de jouer...

### Action mécanique des interrupteurs

Lorsque vous appuyez sur l'interrupteur pour mettre fin au VI, l'interrupteur demeure en position Arrêt. Il est possible de changer le comportement mécanique en sélectionnant **Action mécanique** dans les propriétés de l'interrupteur.

**Actions non bufférisées** (si le contrôle n'est pas lu par le programme pendant l'action de l'utilisateur, cette dernière est ignorée)



- **Commutation à l'appui**—L'interrupteur bascule entre deux positions stables à chaque appui du bouton de souris (interrupteur bistable à bascule). Il peut être contrôlé au relâchement du bouton de la souris en choisissant **Commutation au relâchement**.



- **Commutation jusqu'au relâchement** —La valeur du contrôle reproduit fidèlement l'état d'un bouton monostable.

**Actions bufférisées** (l'action de l'utilisateur est enregistrée jusqu'au moment où le programme lit la valeur du contrôle correspondant, utile pour les boutons poussoirs.)



- **Armement à l'appui**—La commande change de valeur à l'appui sur le bouton de souris et retombe lors de la première lecture de cet état par le programme (mono coup). Cette commande est lue une et une seule fois à chaque action de l'utilisateur.



- **Armement au relâchement** —Idem mais au relâchement du bouton de souris.
  - **Armement jusqu'au relâchement** — La commande change de valeur à l'appui sur le bouton de souris et retombe lors de la première lecture après relâchement. La valeur est donc vraie tout le temps où le bouton est appuyé et peut être lue plusieurs fois.
10. Modifiez l'action mécanique de l'interrupteur pour qu'il revienne seul en position par défaut lorsqu'il est appuyé.
  11. Enregistrez et lancez l'exécution.

## Ajouter une temporisation

Lors que le VI s'exécute dans la boucle, il tourne aussi vite que possible, il est possible de ralentir l'exécution de la boucle à l'aide d'une temporisation. Nous utiliserons une temporisation de 0.5s.



- a. Placez la fonction **Attendre un Multiple de ms** située dans la palette **Fonctions» Temps & Dialogues**. Cette fonction bloque le programme jusqu'à ce que l'horloge interne atteigne un multiple entier de son entrée, en l'occurrence 500 ms.



**Note:** si le temps d'exécution des autres VIs de la boucle est  $< 500$  ms, cette fonction assure un timing parfait. De plus, la fonction met cette partie de code dans un état "idle" (inhibé) ce qui rend du temps processeur pour les autres parties du programme.

500

- b. Il est possible de placer très simplement une constante à l'entrée d'un VI donné, pour cela, cliquez à droite au niveau de la connections de l'entrée, et choisissez **Créer»Constante**

12. Enregistrez et lancez l'exécution.

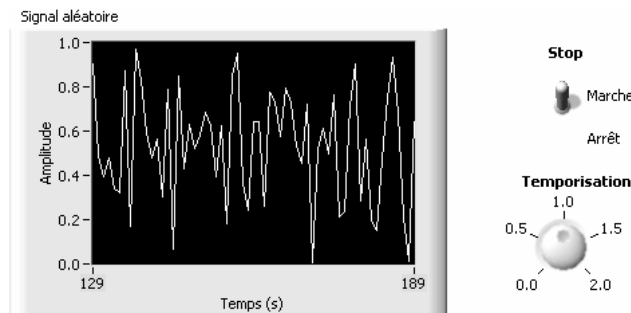
## Fin de l'exercice 4-1

## Exercice 4-2 : Signal aléatoire

Objectif: Ajouter une temporisation variable à un graphe déroulant.

Réalisez les étapes suivantes pour créer un VI qui génère un nombre aléatoire et l'affiche à l'écran.

1. Placez un bouton rotatif, un interrupteur et un graphe déroulant sur la face avant.



2. Configurez l'interrupteur pour qu'il revienne en position marche à chaque départ d'exécution, définissez les graduations du bouton, les labels...
3. Construisez le diagramme en suivant les recommandations suivantes :
  - a. Utiliser Générer un nombre aléatoire situé dans Fonctions» Numérique pour fabriquer le données.
  - b. Multiplier la valeur du bouton rotatif par 1000 pour obtenir des ms, unité utilisée par le timer.
4. Enregistrez votre VI et l'exécuter changez la temporisation durant l'exécution et vérifiez son effet.

## Fin de l'exercice 4-2

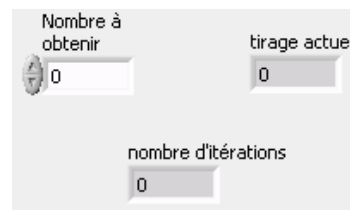
### Exercice 4-3 : Avec de la chance

Objectif: Renvoyer des données au travers d'une boucle, via un tunnel.

Réalisez les étapes suivantes pour créer un VI qui génère des nombres aléatoires entiers compris entre 0 et 100 000 tant qu'aucun ne correspond à un nombre spécifié par l'utilisateur. La boucle, une fois terminée, fournit le nombre d'itérations nécessaires.

### Face Avant

1. Créer la face avant suivante.



**Nombre à obtenir** est le nombre que vous voulez voir sortir au tirage.

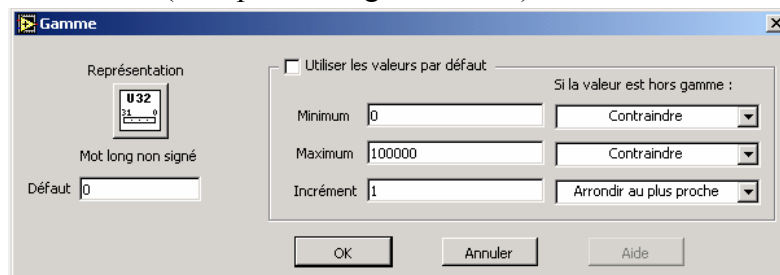
**Tirage actuel** est le nombre compris entre 0 et 100 000 que l'ordinateur vient de choisir.

**Nombre d'itérations** est le nombre de coups nécessaires pour obtenir le tirage.

### Gamme des données.

Utilisez la propriété **Gamme des données** des objets numériques pour limiter l'étendue des valeurs possibles.

2. Par un clic droit, ouvrez la propriété **Gamme des données** de la commande *Nombre à obtenir*. Sélectionnez les options suivantes. (U32 pour Unsigned 32 bits)



### Diagramme

3. Construisez le diagramme, en vous aidant des indications suivantes :



- a. « **Générer un nombre aléatoire** » fournit un nombre compris entre 0 et 1, il faut donc le multiplier pour obtenir un nombre entre 0 et 100 000.



- b. **Fonctions»Numérique»Arrondir à l'entier le plus proche** permet d'arrondir la valeur obtenue après multiplication.



- c. **Fonctions»Comparaison»Différent** retourne une valeur booléenne capable d'arrêter l'exécution de la boucle While.





- d. Ce terminal contient le numéro de l'itération courante (attention la première itération a le numéro 0 !). Il faut sortir cette valeur de la boucle en la connectant à un tunnel créé automatiquement en câblant cet élément à une entrée de fonction ou à un indicateur extérieur à la boucle.
  - e. **Fonctions»Numérique»Incrémenter** incrémente de 1 l'entrée. Pour palier la valeur 0 associée à l'itération 1...
4. Sauvegardez votre VI sous le nom *Avec de la chance.vi*
  5. Entrez un nombre dans la commande « **Générer un nombre aléatoire** » et lancer le VI.
  6. Vous pouvez observer le fonctionnement de la boucle grâce à l'outil **Animer l'exécution** qui fait apparaître le flot de données le long des fils, et les valeurs qui y circulent.
  7. Entrez un nombre supérieur à 100000 ou inférieur à 0 et vérifiez que lors du lancement, LabVIEW contraint l'entrée au mini ou maxi le plus proche.

### Fin de l'exercice 4-3

## C. Registres à décalage

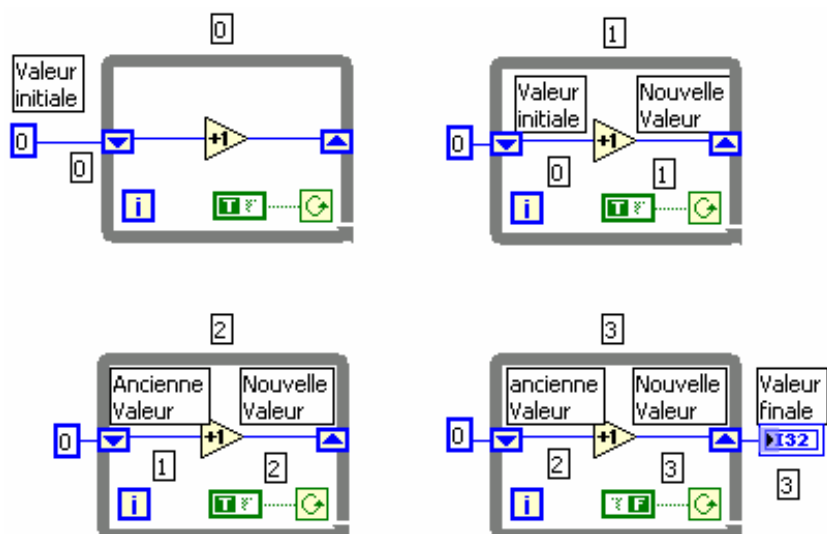
Les registres à décalages sont présents uniquement dans les boucles, et permettent à une itération I, de connaître la valeur d'une variable évaluée à l'itération I-1 (mais aussi I-2, I-3...).

On place les registres à décalage en cliquant à droite sur le coté d'une boucle et en choisissant **Ajouter un registre à décalage**.



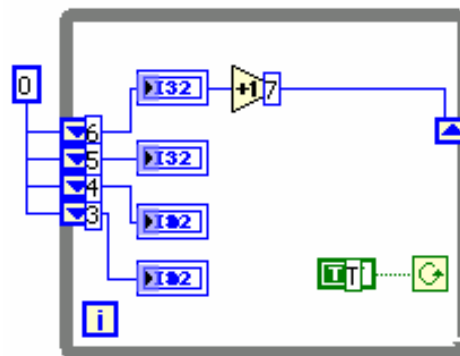
Les registres à décalage apparaissent sous la forme d'une paire de terminaux. Le registre de droite reçoit des données de l'intérieur de la boucle, elles sont transférées au registre de gauche à l'itération suivante. Le registre de gauche est une source de données pour les éléments de l'intérieur de la boucle.

L'illustration suivante montre le fonctionnement du registre à décalage



Au temps 0, la valeur initiale est transférée dans le registre à décalage. Au temps 1, premier tour de boucle, la valeur initiale 0 est transférée pour être incrémentée, la valeur 1 est stockée dans la partie droite du registre. Lorsque tout le code contenu dans la boucle a été exécuté, la valeur 1 est transférée de la partie droite à la partie gauche du registre à décalage. Au temps 2, deuxième tour de boucle, la valeur 1 est extraite du registre gauche puis incrémentée, le résultat 2 est stockée dans la partie droite du registre ... lorsque la boucle se termine, la dernière valeur peut sortir de la boucle par une connexion du registre extérieure à la boucle.

Il est possible de rajouter des éléments aux registres de gauche pour obtenir les valeurs à  $t-1$ ,  $t-2$ ,  $t-3$ .... comme l'illustre la figure ci après.



### Initialisation des registres

Pour initialiser un registre, nous avons vu qu'il suffisait de câbler une valeur, externe à la boucle, au registre de gauche.

Si cette valeur n'est pas câblée, au premier appel du VI, le registre est initialisé à 0, s'il s'agit d'un numérique et à FAUX, s'il s'agit d'un booléen. Aux appels suivants, les registres gardent la valeur qu'ils avaient au dernier appel.

### Exercice 4-4 : Moyennage de Température

Objectif: Utiliser un registre à décalage pour réaliser une moyenne glissante.

Reprenez le code de l'exercice 4-1 Surveillance de température.vi Modifiez le VI pour afficher, non pas la température instantanée, mais la moyenne des trois dernières prises de points.

### Face avant

1. Pas de modifications, sélectionnez **Fichier»Enregistrer sous** et renommez le fichier Surveillance de température moyenne.vi.

### Diagramme

2. Ajoutez un registre à décalage à la boucle.
3. Ajoutez un élément au registre de gauche (lors de la prise de température à l'instant  $t$ , vous aurez les valeurs aux instants  $t-1$  et  $t-2$  pour moyennner sur trois éléments.



4. Modifiez le diagramme pour visualiser la moyenne des trois derniers points sur le graphe. Tenez compte des remarques suivantes :
  - a. L'opérateur **Fonctions»Numérique»Opérateur arithmétique** permet d'appliquer le même opérateur à plusieurs entrées (simplification du diagramme).
  - b. Placez une temporisation de 500 ms pour modérer les ardeurs de l'exécution.
5. Enregistrez les modifications et exécutez le VI.  
A chaque itération, le VI thermomètre fournit une température, le VI additionne cette température aux deux dernières stockées dans le registre et divise le résultat par trois.
6. Placez la fonction **Fonctions»Cluster»Assembler** pour permettre l'affichage simultané de la température instantanée et de la température moyenne. L'aide contextuelle sur les graphes fournit les renseignements essentiels pour cette opération.
7. Enregistrez les modifications et exécutez le VI, changez le type d'affichage des points, des traits... pour chacune des courbes affichées.

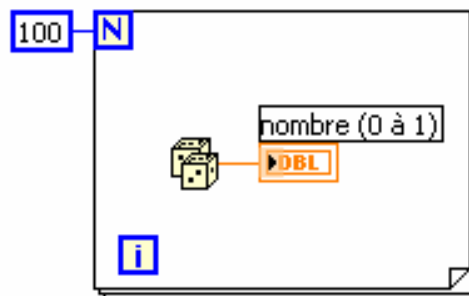
### Fin de l'exercice 4-4

## D. Boucles For



La boucle d'itération (boucle For) représentée à gauche exécute un nombre de fois déterminé la partie de diagramme qu'elle englobe. Ce type de boucle est accessible dans : **Fonctions»Structures»Boucles For**. Une fois lancée, il est impossible d'arrêter l'itération prématurément (pas d'équivalent du *break*). L'entrée N correspond au nombre d'itérations à faire. Le terminal i, au numéro de l'itération en cours.

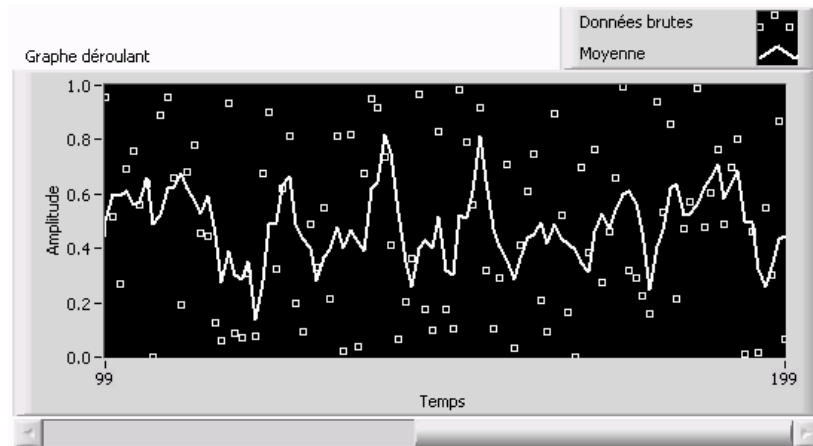
La boucle suivante tire 100 nombres aléatoires.



### Exercice 4-5: Moyenne de nombres aléatoires

Objectif: Créer un VI qui affiche deux courbes, un nombre aléatoire, et une moyenne glissante des 4 derniers points.

1. Construisez un VI en suivant les recommandations suivantes:
  - a. Utilisez une boucle For de 200 itérations.
  - b. Créez une face avant ressemblant à celle-ci :



- c. Utilisez un registre à décalage pour conserver les trois dernières valeurs.
  - d. •Vous pouvez utiliser une boîte de calcul (**Fonctions » Structures » Boîte de calcul**) pour réaliser le tirage, la somme et la division.
2. Enregistrez les modifications et exécuter le VI.

### Fin de l'exercice 1-1

### Exercice supplémentaire

Réaliser un VI qui mesure la température une fois par seconde et l'affiche sur un graphe déroulant. Deux commandes situées sur la face avant permettent de fixer des seuils d'alarmes haute et basse. Deux voyants s'allument lorsque ces seuils sont dépassés dans un sens ou dans l'autre. En plus de la température instantanée, le graphe affiche les valeurs minimale et maximale de la température.

## Chapitre 5 Tableau, Graphes et clusters

Cette partie décrit comment utiliser les tableaux, comment créer des graphiques (autres que déroulants) en fonction du temps ou d'une autre variable. Il montre également comment utiliser les clusters pour rassembler des données de natures différentes.

### Vous apprendrez:

- A. Les tableaux
- B. Comment créer les tableaux à partir de structures de répétitions.
- C. Fonctions sur tableaux
- D. Le polymorphisme
- E. Les graphes
- F. Les clusters
- G. Fonctions sur les clusters

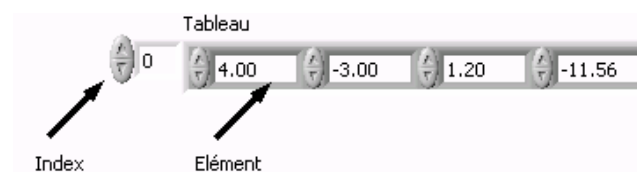
### A. Les Tableaux

Les tableaux sont des groupements ordonnés de données de même type. La dimension d'un tableau est le nombre d'indices définissant un élément. (1 dimension, le tableau contient 1 seule colonne, 2 dimensions : ligne/ colonne, 3 dimensions : ligne/ colonne/ profondeur...). Chaque élément est ainsi repéré de façon unique par ses coordonnées.

Les tableaux peuvent contenir n'importe quel type d'éléments (diodes, interrupteurs...), excepté d'autres tableaux.

#### Créer des tableaux sur la face avant.

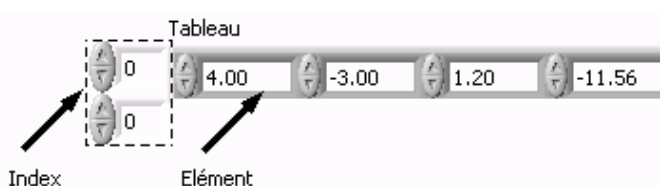
Pour créer un tableau sur la face avant, sélectionnez **Commande»Tableaux et Cluster»Tableaux**. Cette commande place un tableau vide sur la face avant. A présent déposez un indicateur ou une commande du type voulu, à l'intérieur du tableau.



Vous pouvez changer la taille de l'affichage des éléments, ainsi que le nombre d'éléments visibles. L'index indique le numéro du premier élément affiché.

#### Tableaux à deux dimensions

Les données des tableaux à deux dimensions sont affichées dans une grille lignes/colonnes. On augmente la dimension en redimensionnant l'affichage de l'index vers le bas.



Il devient alors possible d'étirer aussi l'affichage des éléments.



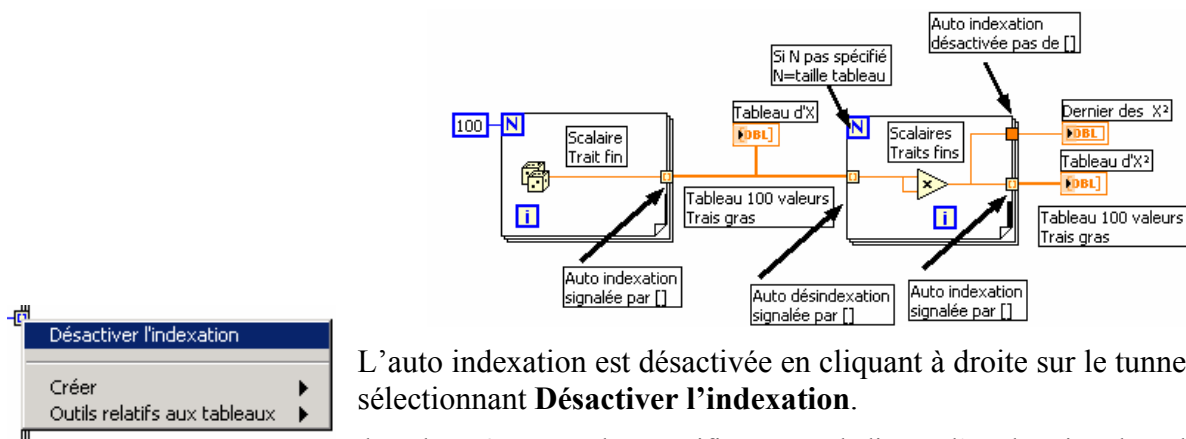
La méthode s'applique aux tableaux de toutes dimensions. Cependant, l'affichage reste 2D...

### Tableaux de constantes

Même méthode, mais dans le diagramme : **Fonctions** » **Tableau** » **Constante tableau** puis placer une constante du type voulu dans la structure vide.

## B. Auto Indexation

Lorsque l'on câble une connexion vers l'extérieur d'une boucle FOR, LabVIEW stocke **automatiquement** les valeurs successives dans un tableau, c'est **Auto Indexation**. De façon réciproque, un tableau connecté à l'entrée d'une boucle sera automatiquement désindexé (les éléments sont passés un à un), et la boucle sera exécutée autant de fois qu'il y a d'éléments dans le tableau. L'illustration suivante illustre ce mécanisme.

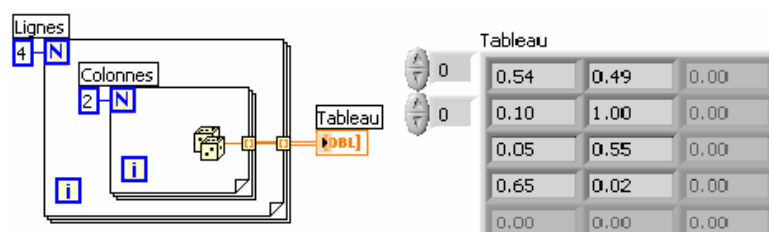


L'auto indexation est désactivée en cliquant à droite sur le tunnel et en sélectionnant **Désactiver l'indexation**.

**Note** Parce que les boucles FOR sont plus spécifiquement dédiées à l'exploration des tableaux, l'auto indexation y est automatiquement activée, en revanche dans les boucles While, elle ne l'est pas, il faut agir comme ci-dessus en sélectionnant **Activer l'indexation**.

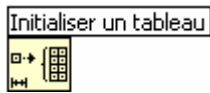
### Boucles et tableaux 2D

Pour construire un tableau 2D, il suffit de d'imbriquer deux boucles FOR. La boucle intérieure crée les colonnes, la boucle extérieure, les lignes. Notez, dans l'exemple ci-dessous, l'augmentation de l'épaisseur des fils de liaisons avec le nombre de dimensions du tableau.

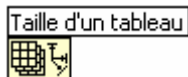


## C. Fonctions sur tableaux

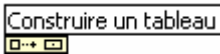
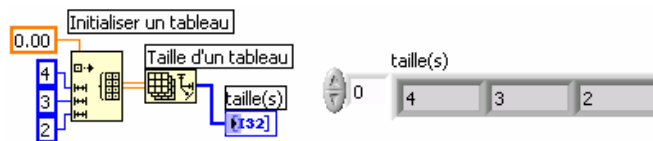
Les fonctions sur les tableaux sont accessibles par **Fonctions»Tableaux**. Parmi les fonctions les plus fréquemment utilisées, citons :



Crée un tableau à N dimensions dont tous les éléments ont la même valeur. Cette fonction (comme la plus par des fonctions opérant sur les tableaux), est redimensionnable pour s'adapter aux nombres de dimensions des tableaux. Cette fonction initialise donc le tableau.



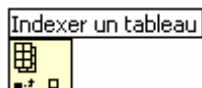
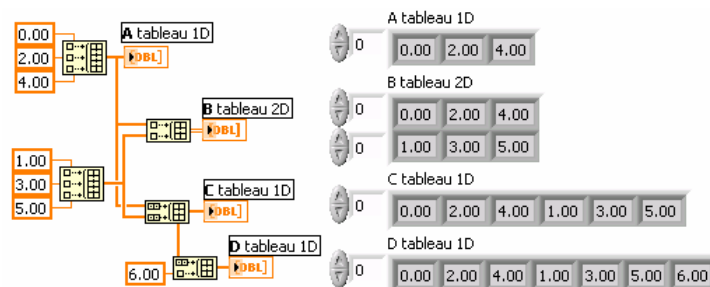
Renvoie la taille du tableau. Si le tableau est à N dimensions, la fonction renvoie un tableau contenant la taille de chaque dimension.



Permet de :

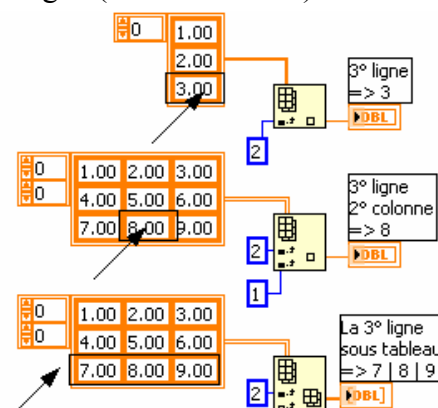
- placer des éléments dans un tableau. S'il s'agit de scalaires séparés, le résultat est un tableau 1D (tableau A ci-dessous) ; s'il s'agit de plusieurs tableaux 1D, le résultat obtenu est un tableau 2D (Tableau B)
- ajouter en début ou fin d'un tableau, un élément (Tableau D).
- concaténer des tableaux de mêmes dimensions, en un tableau unique de mêmes dimensions (Tableau C)

La concaténation est activée en cochant l'option **Concaténer les entrées** du menu contextuel.



Permet de :

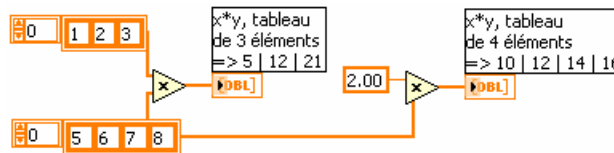
- extraire un élément du tableau d'entrée.
- extraire une ligne (ou une colonne) d'un tableau 2D



## Polymorphisme et tableaux

Les fonctions arithmétiques sont valides pour les calculs sur les tableaux. Toutes ces fonctions sont polymorphiques, c'est à dire qu'elles opèrent sur des variables de types différents.

Par exemple, la fonction multiplier, peut multiplier les éléments du tableau 1 avec les éléments du tableau 2 câblés à ses entrées. Si un tableau et un scalaire sont reliés aux entrées, l'ensemble des éléments du tableau est multiplié par le scalaire.



### Exercice 5-1: Génération d'une table trigonométrique

Objectif: Créer un tableau 2D contenant les angles et leurs fonctions trigonométriques sin, cos, et tg.

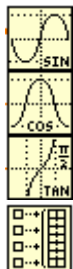
L'exercice suivant vous propose de réaliser une table trigonométrique donnant, pour des angles allant de 0 à 90° par pas de 10°, les sinus, cosinus et tangentes.

#### Face avant

1. Créez la face avant suivante :

	Angle	Sinus	Cosinus	Tang
0	0.00E+0	0.00E+0	1.00E+0	0.00E+0
0	10.00E+0	173.65E-3	984.81E-3	176.33E-3
	20.00E+0	342.02E-3	939.69E-3	363.97E-3
	30.00E+0	500.00E-3	866.03E-3	577.35E-3
	40.00E+0	642.79E-3	766.04E-3	839.10E-3
	50.00E+0	766.04E-3	642.79E-3	1.19E+0
	60.00E+0	866.03E-3	500.00E-3	1.73E+0
	70.00E+0	939.69E-3	342.02E-3	2.75E+0
	80.00E+0	984.81E-3	173.65E-3	5.67E+0
	90.00E+0	1.00E+0	61.23E-18	16.33E+15

#### Diagramme



2. Construisez le diagramme en tenant compte de ces éléments :
  - a. Les fonctions trigonométriques sont dans la palette **Numérique»Trigonométrique**. Les angles sont en radian.
  - b. Utilisez la fonction **Construire un tableau** pour assembler dans une ligne de tableau : angle, sin, cos et tg.
  - c. Une boucle For permettra de calculer les 10 ensembles de valeurs.

3. Enregistrez le VI sous le nom `trigo.vi`, exécutez le.

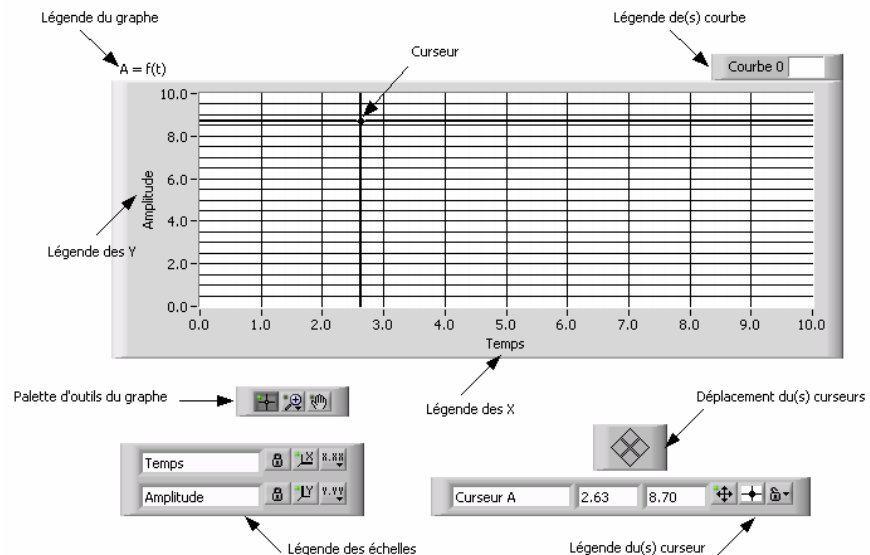
Il est possible de remplacer les icônes de calculs par une boîte de calcul. Tentez de remplacer l'ensemble des fonctions numériques par une seule boîte de calcul.(Cf Chap6 §C)

#### Fin de l'exercice 5-1



## D. Graphes fonction du temps et graphes XY

Les VIs utilisant des graphiques collectent généralement les données dans des tableaux, les traitent puis les affichent. L'illustration suivante montre les éléments d'un graphe



Les graphes sont accessibles par **Commande»Graphes**. Les graphes inscrivent les points en donnant une abscisse initiale et un incrément à chaque nouveau point, cela suppose donc des points régulièrement distribués sur l'axe des X (par ex. toutes les secondes, tous les °C...). Les graphes de type XY nécessitent un tableau de valeurs d'X, et peuvent donc afficher des points pris non régulièrement. Ces deux graphes sont capables d'afficher des courbes multiples.

### Les Graphes

Les graphes acceptent :

En mono courbe

**Graphes :**  
Câblez les données directement au graphe :

Tableau Y	Graphe résultant
1D	Courbe unique
WDT	Courbe unique
2D	Multicourbes

Le type WDT contient les infos temporelles. Les autres types ont par défaut la valeur 0 pour  $X_0$  et 1 pour  $\Delta x$ . Combinez les informations temporelles via la fonction assembler :

graphique

tableau Y

Consultez l'exemple : **Waveform graph.vi**

- Soit un tableau 1D, dans ce cas, l'axe des X est gradué, par défaut de 1 en 1 à partir de 0. On peut modifier ces valeurs à partir du menu contextuel **Echelle des X»Formatage...**
- Soit un cluster contenant la valeur de  $X_0$ , la valeur de  $\Delta x$ , et le tableau d'Y nécessaires à la construction de la forme d'ondes (waveform)

Multi courbes

- Soit un tableau 2D, chaque ligne correspond à une courbe. Dans ce cas, l'axe des X est gradué, par défaut de 1 en 1 à partir de 0. On peut modifier ces valeurs en prenant au clic droit **Echelle des X»Formatage...**
- Soit un cluster contenant la valeur de  $X_0$ , la valeur de  $\Delta x$ , et le tableau 2D d'Y comme précédemment. Il n'y a qu'un seul axe des X, et donc qu'une seule graduation pour les deux courbes (même  $X_0$  et  $\Delta x$ ).

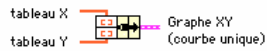
## Les graphes XY

Les graphes XY acceptent :

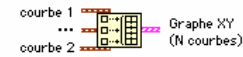
En mono courbe

**Graphes XY :**

Graphe XY à courbe unique :



Graphe XY multi-courbes :



Consultez l'exemple : **XY Graph.vi**

- Un couple de tableau d'X et d'Y assemblés dans un cluster par la fonction **Cluster»Assembler**.

Multi courbes

- Un tableau 1D, dont chaque colonne contient les données (portées en abscisse) d'une courbe.

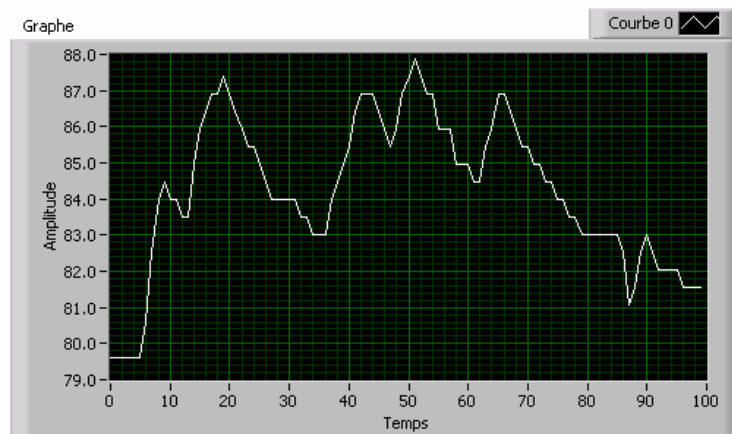
### Exercice 5-2 : Graphe

Objectif: faire une boucle pour créer un tableau 1D et afficher le résultat sur un graphe.

Reprendre l'exercice 4-1, le modifier en remplaçant le graphe déroulant par un graphe. La boucle While par un For le nombre d'itérations sera de 100.

### Face Avant

1. Vous devriez obtenir quelque chose comme :



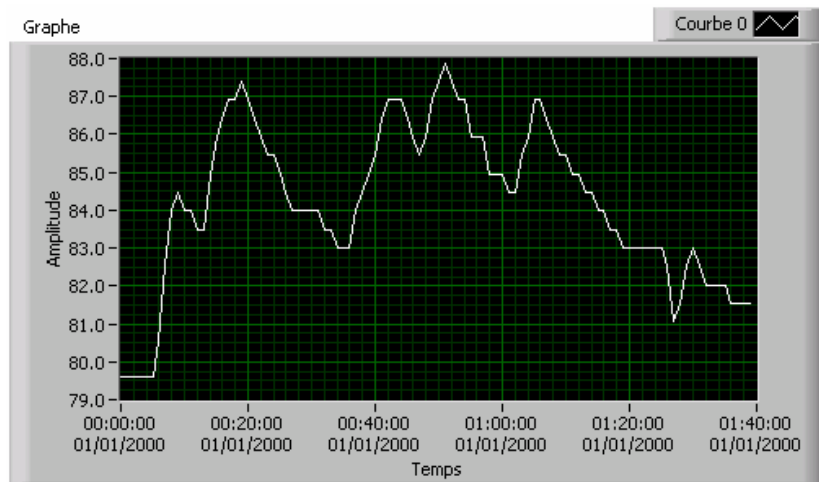
### Diagramme



2. Exécutez le code pour vérifier son fonctionnement.
3. Modifiez le code pour avoir un  $X_0 = 3029526000$  et  $\Delta x = 60$ , en utilisant la fonction **Cluster»Assembler**. Relancez l'exécution pour constater le changement sur l'échelle des X.

### Face Avant

4. Changez les propriétés de formatage de l'échelle des X, choisissez le format « Heure & Date, 24 heures, HH :MM :SS, J/M/A, Année sur 4 chiffres ». Vous devriez obtenir :



En fait, 3029526000 est le nombre de secondes séparant le 1/1/1904 à 1h du matin (origine des temps dans le PC) au 1/1/2000 à 0h. 60 est le nombre de secondes entre chaque point (6000 s soit environ 1h40).

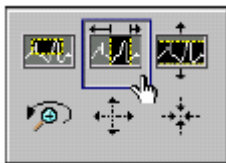
## Graphe multi courbes

### Diagramme



5. Ajoutez le VI:Digital Thermometer.vi dans la boucle For pour obtenir deux flots de données, réunissez les deux tableau 1D en un tableau 2D grâce à **Tableau»Construire un tableau** qui entrera dans la fonction **Assembler** en lieu et place du tableau 1D précédent.

### Face Avant



6. Faire apparaître la « **Palette du graphe** » depuis le menu contextuel **éléments visibles**

Pour zoomer une partie du graphique :

- a. Sélectionnez l'outil **Loupe** à l'aide de l'outil doigt. Un menu se déroule alors.

Les options sont :

<Zoom dans un rectangle>.<Zoom entre deux X>.<Zoom entre deux Y>.<Annuler le zoom>.<Agrandir à chaque clic>.<Rétrécir à chaque clic>

- b. Essayez ces différentes fonctionnalités.

L'outil déplacement permet de mouvoir la courbe dans sa fenêtre lorsqu'on l'a agrandie.

L'outil curseur est l'outil par défaut pour déplacer les curseurs, mais c'est pour plus tard...

7. Enregistrez le VI sous le nom `multigraphe.vi` et fermez le.

### Fin de l'exercice 5-2

### Exercice 5-3 : Analyse de l'évolution de Température

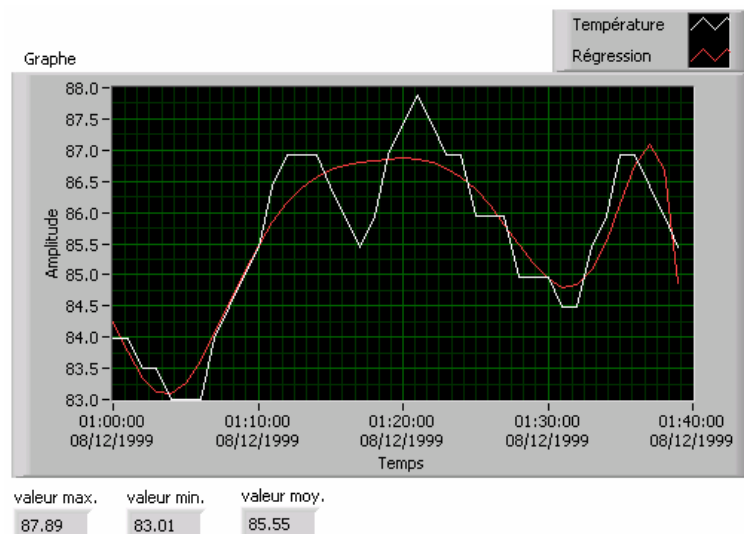
Objectif: Afficher et analyser l'évolution d'une température.

Reprenez l'exercice 5-2 précédent, supprimez les éléments ajoutés à l'étape 5, le nombre de points sera maintenant de 40, puis complétez le diagramme pour :

1. Afficher le minimum et le maximum de la température
2. La température moyenne
3. Ajouter au graphe une courbe représentant une régression polynomiale d'ordre 8 de l'évolution de la température.

Face avant

4. Vous devriez obtenir quelque chose comme :



Diagramme



5. Utilisez les éléments nouveaux suivants : (Pour les détails, utilisez l'aide en ligne)
  - a. **Tableau» Max & Min d'un tableau** pour rechercher les extrema du tableau de température.
  - b. **Mathématiques» Probabilité et Statistiques» Moyenne** pour calculer la moyenne des valeurs du tableau de température.
  - c. **Mathématiques» Ajustement de courbes» Ajustement polynomial général** pour calculer le tableau de valeurs ajustées. Le tableau d'X que réclame la fonction sera créé le plus simplement possible... L'ordre de régression est de 8. Vous procéderez comme précédemment pour ajouter la courbe régressée au graphe.
6. Enregistrez le VI sous le nom `analysetempérature.vi` et lancer l'exécution du code.

Fin de l'exercice 5-3

## E. Clusters

Les clusters sont des assemblages d'éléments disparates qui sont l'équivalent des structures en langage C. Ils permettent de rassembler dans une même connexion des données de types différents ayant une relation plus ou moins proche entre elles (préférable pour l'analyse !), par exemple Nom, Prénom, Age.

Il est important d'utiliser les clusters pour deux raisons :

- Regrouper les données dans des ensembles structurellement cohérents
- Diminuer le nombre des connexions des VIs, limité à 28.



**Notes :** Il est impossible de regrouper des contrôles et des indicateurs dans le même cluster. Il est possible d'assembler et de désassembler les éléments d'un cluster, soit comme en C, par leur nom, soit par leur ordre.

### Créer des clusters sur la face avant



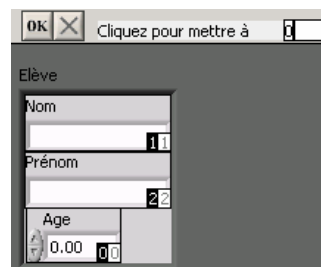
Pour créer un cluster sur la face avant, sélectionnez **Tableau & Cluster»Cluster** depuis le menu contextuel/ **Commandes**. Une fois le cadre posé, déposez à l'intérieur les différents éléments constitutifs. L'ordre de placement sera l'ordre des éléments dans le cluster leur position physique ne change rien à l'ordre, les étiquettes seront le nom des éléments du cluster. Tous les éléments sont déplaçables, et modifiables, la taille du cluster peut être ajustée à la main ou automatiquement. Cependant, un cluster étant un ensemble, cet ensemble est, soit une commande, soit un indicateur. Les éléments le constituant sont soit **tous** des commandes, soit **tous** des indicateurs.

### Constantes de type Cluster

Il est possible de créer sur le diagramme une constante de type cluster, en utilisant **Cluster»Cluster constante**, puis en y plaçant comme précédemment des constantes du type désiré.

### Ordonner les Cluster

Les éléments de cluster sont ordonnés, nous l'avons dit, dans leur ordre d'insertion. Si vous désirez changer l'ordre des éléments d'un cluster (par ex., vous ajoutez un élément omis qui ne doit pas être le dernier) il faut choisir, dans le menu contextuel, **Ordonner les commandes dans le cluster**. La barre d'outils prend alors l'aspect suivant :



Chaque élément voit son numéro d'ordre apparaître à droite de la commande, en sur fond blanc, l'ordre avant modification, sur fond noir l'ordre actuel. Il suffit de cliquer sur chaque commande dans l'ordre souhaité pour réordonner le cluster.

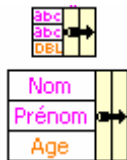
Attention, changer l'ordre d'un cluster peut rendre le programme non exécutable : si l'élément n'était une chaîne, et qu'une fois réordonné il s'agit d'un nombre, toutes les liaisons entre cet élément et les fonctions qui l'utilisent, seront brisées. De fait, nous recommandons vivement d'utiliser l'assemblage et le désassemblage par nom qui évite cet inconvénient, mais qui est cependant plus lourd à utiliser.

## F. Fonctions sur Cluster

Quatre fonctions principales permettent d'assembler ou de désassembler par ordre ou par nom les clusters, elles sont situées dans la palette d'outils **Cluster**



### Assembler & Assembler par nom

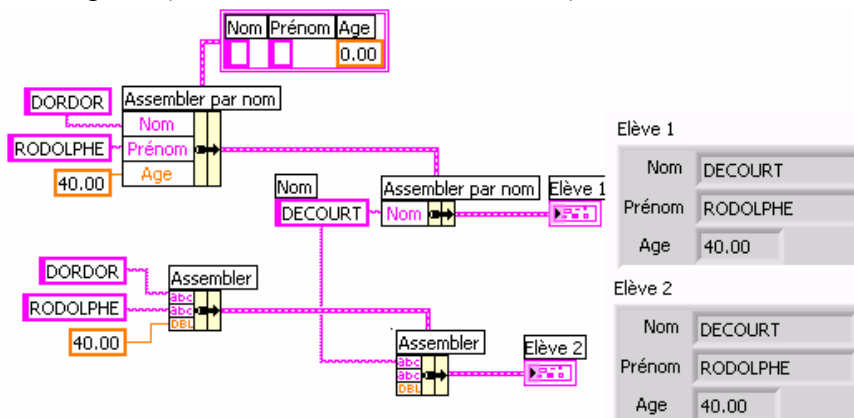


Les fonctions d'assemblages permettent de regrouper divers éléments dans un cluster ou de changer la valeur de certains éléments d'un cluster existant. Ces fonctions sont redimensionnables.

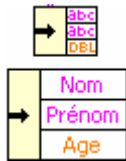
L'exemple ci-dessous montre l'utilisation de ces fonctions. Les fonctions de gauche créent un cluster à partir de constantes, celles de droite modifient un élément du cluster créé. Le problème est résolu simultanément avec un assemblage par nom et avec un assemblage par ordre.

Plusieurs remarques s'imposent :

- L'assemblage par nom est éminemment plus clair! Si le cluster n'existe pas déjà, la connexion **cluster d'entrée** de la fonction **cluster/assembler** doit être reliée à une constante, de type cluster. Celle ci contiendra les prototypes ou les valeurs initiales des éléments du cluster. Il est possible de sélectionner l'élément dont on veut changer la valeur : clic avec l'outil doigt sur l'entrée ou menu contextuel/sélectionner un élément.
- L'assemblage par ordre est plus compact, on ne câble que l'entrée qui doit changer de valeur, mais on ne sait pas à quoi elle correspond (il faut alors commenter le code).

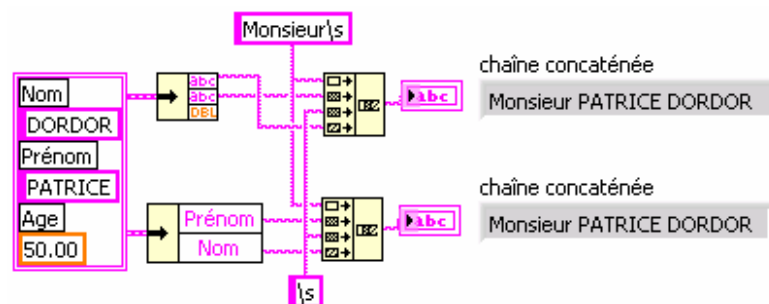


## Désassemblage des clusters



Les fonctions de désassemblage permettent d'extraire des éléments d'un cluster pour les utiliser individuellement. La fonction de désassemblage par ordre s'ajuste automatiquement au nombre total d'éléments du cluster. Le désassemblage par nom doit être dimensionné à la main en fonction du nombre d'éléments à extraire, l'outil doit permettre de sélectionner le nom des éléments.

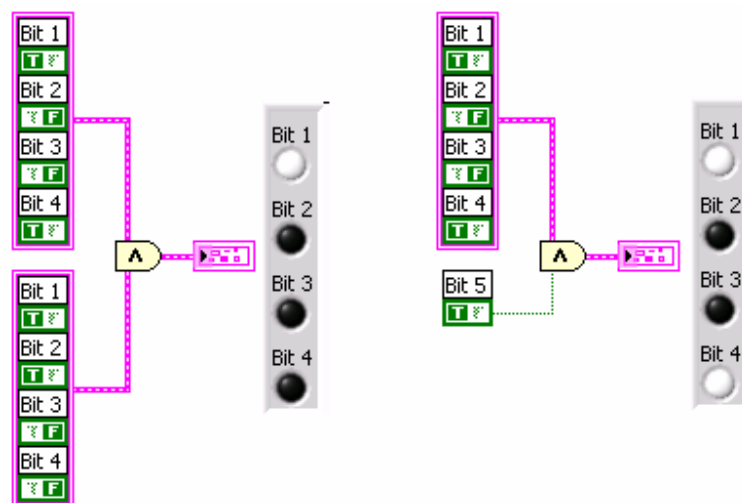
L'exemple suivant montre l'utilisation des fonctions de désassemblage, notez que le désassemblage par nom permet d'extraire le nombre souhaité d'éléments et dans l'ordre le plus pratique pour le câblage. Le caractère « \s » apparaissant dans les constantes chaînes est le caractère d'échappement correspondant à « espace ». Les fonctions les plus à droites sont des fonctions de concaténation de chaîne.



## Polymorphisme des clusters

Si les données d'un cluster sont de même type (chaîne, nombre, booléen), il est possible d'effectuer directement des opérations logiques sur des clusters.

L'exemple suivant montre le résultat obtenu lors d'un « ET » logique entre deux clusters de booléens puis entre un cluster de booléens, et une constante booléenne. Dans le premier cas le « ET » est réalisé élément de cluster à élément de cluster (par ordre de création et pas par ordre visualisé à l'écran), dans le second cas entre chaque élément du cluster et la constante (la LED noire indique une valeur fausse).



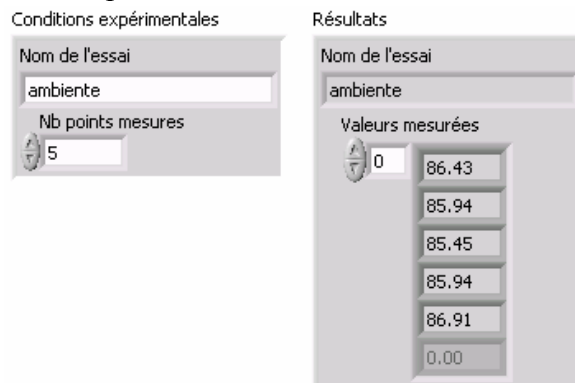
**Exercice 5-4 : Paramètre de mesure et résultats**

Objectif: Créer des clusters sur la face avant et utiliser les fonctions d'assemblage et de désassemblage.

Les conditions expérimentales d'un essai sont regroupées dans un cluster qui contient les éléments suivants : le nom de l'essai et le nombre de points de mesure. Les résultats seront regroupés dans un autre cluster contenant le nom de l'essai, et un tableau de valeurs d'essai. Ces valeurs seront produites comme précédemment par un boucle contenant le VI `Digital Thermometer.vi`

**Face avant**

1. La face avant peut ressembler à :

**Diagramme**

2. Le diagramme n'utilise pas de fonctions nouvelles particulières, il est conseillé d'utiliser les fonctions d'assemblage et de désassemblage par nom.
3. Enregistrer le VI sous le nom `Cluster manip.vi` puis l'exécuter.

**Fin de l'exercice 5-4**



## Chapitre 6 Structures de choix, séquences et nœuds de calcul

Cette partie décrit la dernière structure disponible sous LabVIEW, la structure de choix. Vous verrez également deux autres éléments disponibles dans la palette **Structure** qui s'apparentent plus à des outils commodes qu'à des structures, la séquence et le nœud de calcul.

### Vous apprendrez:

- A. Comment utiliser la structure de choix.
- B. Comment utiliser la structure de séquencement.
- C. Comment utiliser les nœuds de calculs.

### A. Structure de choix



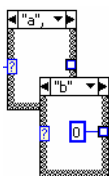
La structure de choix est le parallèle à la structure « if...then...else » utilisée dans les langages textuels.

Elle s'apparente à une boîte possédant un sélecteur de choix dans sa partie centrale supérieur (le « case » du C) et une entrée de sélection (le « Switch » du C) en forme de point d'interrogation. Les flèches de part et d'autre du sélecteur de choix permettent de faire défiler les divers cas possibles.

Par défaut, l'entrée de sélection est booléenne, il est possible d'y connecter des entiers, des énumérations (collection d'entiers et de labels disponibles dans **Commande»Menu déroulant & enum**) et des chaînes. Le sélecteur de choix s'adapte automatiquement au type d'entrée connectée.

Un <cas par défaut> doit être spécifié si l'ensemble des cas possibles correspondant à l'entrée de sélection n'est pas décrit (c'est le cas si l'entrée de sélection est reliée à un entier ou un chaîne). On spécifie le <cas par défaut> en sélectionnant **Prendre ce cas par défaut** dans le menu contextuel du sélecteur de cas. Ce menu permet aussi d'ajouter, d'enlever de dupliquer des cas.

### Tunnels d'entrée/sortie

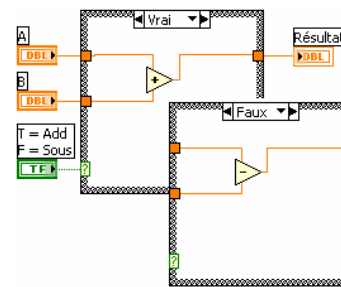


Il est possible de créer des tunnels d'entrée et de sortie d'une structure de choix. Cependant, les connections de sorties doivent être **évaluées** dans **tous les cas**, tant qu'un tunnel de sortie n'est pas connecté pour tous les cas, l'intérieur du cadre du tunnel reste blanc, et la flèche d'exécution est brisée.

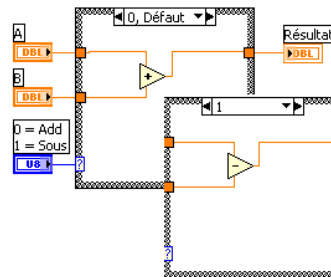
### Exemples

Les exemples suivants exposent les différents types de variables connectables à l'entrée de sélection. Chaque exemple est identique et opère soit une addition, soit une soustraction en fonction d'une variable de choix.

Structure de choix sur booléen

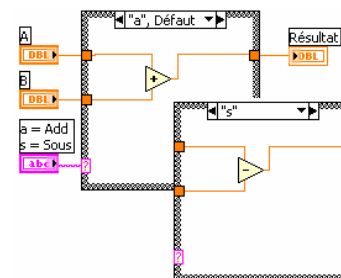


Structure de choix sur entier

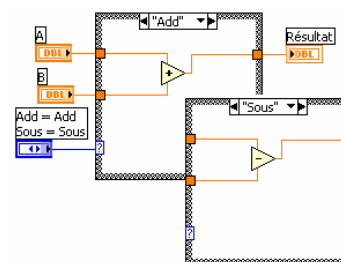


Note : apparition d'un cas par **Défaut**, pour définir les opérations à faire dans les cas non prévus.

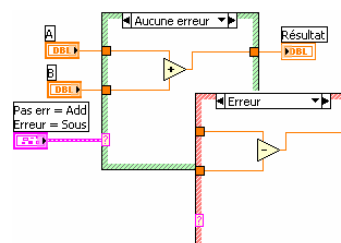
Structure de choix sur chaîne



Structure de choix sur énumération



Structure de choix sur erreur



## Sélection des choix

Pour entrer la valeur de sélection, frappez à l'aide de l'outil texte la valeur désirée dans la case de sélection. Le menu contextuel permet d'ajouter des cas avant ou après le cas actuel, de dupliquer le code d'un cas, d'enlever un cas. Pour chaque cas, il est possible d'entrer une valeur unique, une liste de valeurs distinctes ou une étendue de

valeurs continues.

Pour entrer une liste, il faut séparer les valeurs par des virgules. Pour entrer une étendue, il faut entrer la borne inférieure deux points et la borne supérieure (les bornes sont incluses) si une borne est omise, cela signifie que toutes les valeurs inférieures ou supérieures sont admises.

Par exemple :  $5, 7, 11..20, 30$  signifie que ce cas sera exécuté pour toutes valeurs inférieures à 5, pour la valeur 7, les valeurs comprises entre 11 et 20 et la valeur 30.

### Exercice 6-1 : Racine carrée

Objectif: Utiliser une structure de cas simple.

Il s'agit de réaliser un VI capable de calculer la racine carrée d'un nombre positif ou nul, et d'afficher un message d'erreur si l'utilisateur entre une valeur négative.

#### Face avant

1. Elle peut ressembler à :



#### Diagramme



2. Les éléments nouveaux sont hormis la structure de choix sont :
  - a. Les outils de comparaison sont disponibles dans la palette **Fonctions»Comparaison**.
  - b. Vous trouverez une boîte de dialogue à un bouton dans la palette de **Fonctions»Temps & Dialogue**.
  - c. La racine carrée est dans la palette de **Fonctions»Numérique**
3. Enregistrez le VI sous le nom `Racine.vi` et exécutez le.

### Fin de l'exercice 6-1

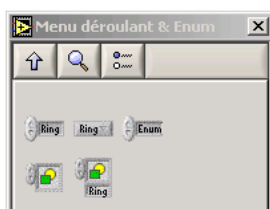
### Exercice 6-2 : Programmation d'un appareil IEEE488

Objectif: Utiliser une structure de choix et une énumération.

Le choix de la fonction d'un multimètre interfacé se fait par l'envoi de codes ASCII. Le VI est chargé de transformer une énumération de fonctions en chaînes de programmation selon la relation suivante :

Fonction	chaîne de programmation
Voltmètre	F0X
Ampèremètre	F1X
Ohmmètre	F2X

#### Face avant



1. Elle contient une commande de type énumération et un indicateur de type chaîne de caractère. Les énumérations sont des ensembles **finis** de valeurs associant un entier à un label

texte. Vous les trouverez dans la palette **Commandes»Menu déroulant & Enum.** Ceci permet d'offrir un choix de valeurs limitées à l'utilisateur.

2. La face avant peut ressembler à :



3. Les labels de l'énumération sont entrés à l'aide de l'outil texte, il est possible d'ajouter avant ou après la position actuelle des éléments, et d'en retirer dans le menu contextuel de l'énumération.

## Diagramme

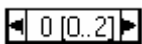
4. Le diagramme ne contient qu'une structure de choix, vous constaterez que le sélecteur de la structure connaît automatiquement les choix de l'énumération une fois l'entrée de sélection reliée.
5. Enregistrez le VI sous le nom `MotPrg.vi` et exécutez le.

## Fin de l'exercice 6-2

## B. Les Séquences



Une séquence est un artifice pour imposer l'ordre d'exécution de VI n'ayant pas de liaisons (s'il y avait des liaisons, l'ordre d'exécution serait imposé par le flux de données). Son cadre est une diapositive, les diapositives sont placées les unes derrière les autres ; l'exécution commence par le code contenu dans la première (n°0) et continue dans l'ordre 1, 2, 3...



Un sélecteur de séquences permet d'écrire le code dans la séquence choisie, il est possible d'enlever, d'ajouter, de déplacer des séquences par le biais du menu contextuel du sélecteur.

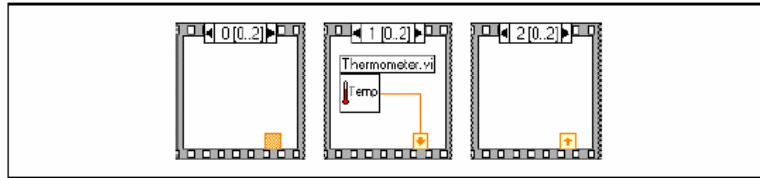
## Variable locale de séquence



Le passage de variable(s) d'une séquence à l'autre se fait par des variables locales de séquences, ajoutées par le menu contextuel.

L'exemple suivant en montre l'utilisation :

- a. Séquence 0, la variable locale n'a pas été affectée, elle est grisée, et inutilisable.
- b. Séquence 1, la variable locale est affectée, la flèche indique que cette valeur sort de la séquence
- c. Séquence 2 et suivantes, la variable locale peut être connectée à l'entrée d'un VI, la flèche indique que cette valeur entre dans la séquence.

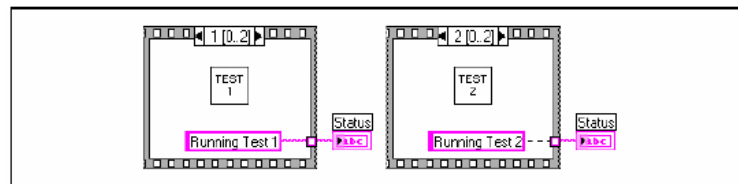


### Comment éviter l'utilisation des séquences

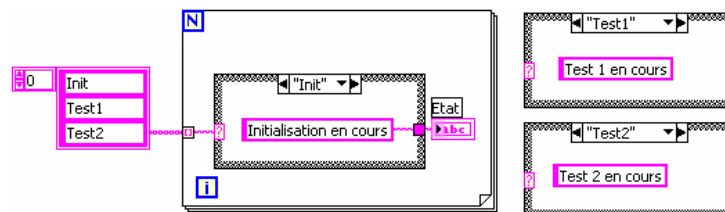
L'utilisation des séquences va à l'encontre du concept de flux de données de LabVIEW, et diminue grandement la lisibilité du code en superposant des pages de code dans le même cadre. Il est grandement préférable d'utiliser l'ordre d'exécution naturel en liant vos VI, ne serait-ce que par une connexion d'erreur.

Il est souvent intéressant d'utiliser le parallélisme inhérent à LabVIEW, mais des tâches asynchrones utilisant des ports d'entrée/sortie peuvent parfois entrer en conflit, il faut dans ce cas utiliser soit des séquences, soit, plus rationnellement des mécanismes d'autorisation (jetons, sémaphores).

Autre inconvénient, une donnée ne peut sortir que d'une seule séquence, puisque toutes les séquences doivent être exécutées avant de sortir de la structure. L'exemple suivant montre l'impossibilité de mettre à jour un simple indicateur.



Nous recommandons l'utilisation d'une structure de type case dans une boucle While ou For, comme dans l'exemple ci-dessous. Notez la possibilité de ne pas exécuter certains cas en modifiant la constante tableau, sans pour autant supprimer le code qui leurs est associé.

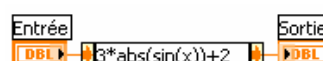


## C. Boîtes de calcul et nœuds d'expression

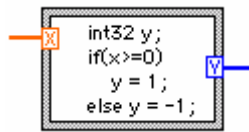
Les boîtes de calcul et les nœuds d'expression permettent d'effectuer des calculs complexes dans un mode texte. Ils rendent le diagramme plus lisible, lorsque les expressions sont complexes.

### Nœuds d'expression

Cet outil est très utile pour calculer une valeur fonction d'une seule variable, il se situe dans les **Fonctions/ Numériques**. On y entre une expression mathématique, les opérateurs et leurs priorités sont explicités dans l'aide.



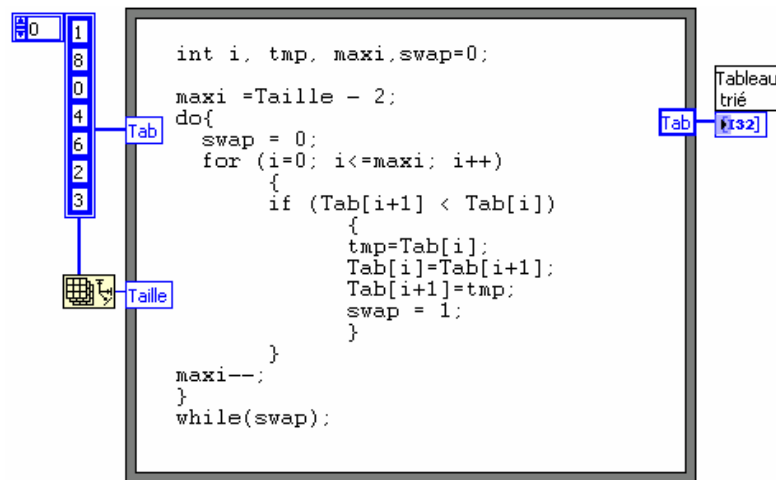
## Boîtes de calcul



Elles sont utilisées dans les opérations multi-variables, dès que les expressions sont un peu complexes ou lorsque l'on reprend un code de calcul écrit en C. Elles sont accessibles par la palette **Fonctions/Structure**.

Le menu contextuel permet de créer des entrées et des sorties qui seront nommées à l'aide de l'outil texte.

Les boîtes de calcul ont une syntaxe comparable à celle du C. On peut y effectuer de multiples opérations, y compris des structures de boucles ou de choix, consultez l'aide pour de plus amples informations. L'exemple suivant vous montre un tri à bulle, comme vous pouvez le constater, le code est exécutable en C.



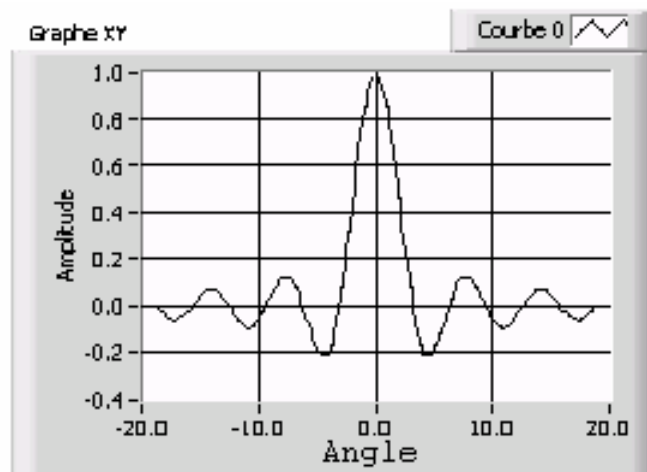
## Exercice 6-3 : Nœuds de calcul

Objectif: Utiliser des fonctions simples d'un nœud de calcul.

On se propose d'utiliser un nœud de calcul dans une boucle for de 100 pas, pour tracer la courbe  $f(x) = \sin(x)/x$  pour des valeurs d'angle comprises entre  $-6\pi$  et  $+6\pi$ .

### Face avant

1. La face avant est présentée ci-dessous.



**Diagramme**

2. Utilisez une boucle FOR à l'intérieur de laquelle est placée la boîte de calcul. La boîte possède trois terminaux, une entre, l'indice de boucle, deux sorties,  $\sin(x)/x$ , et  $x$ .
  - a. Attention aux points-virgules en fin de lignes
  - b. Vous pouvez déclarer des variables intermédiaires pour clarifier le calcul
3. Les boîtes de calcul distinguent minuscules et majuscules (syntaxe du C).

**Fin de l'exercice 6-3**





## Chapitre 7 Chaînes et Entrées/Sortie fichier

Cette partie décrit les principales fonctions de manipulation de chaînes et les outils de lecture et d'écriture de fichiers.

### Vous apprendrez:

- A. Comment créer des commandes et des indicateurs de type chaîne
- B. Comment utiliser quelques fonctions sur chaînes
- C. Comment faire des opérations d'E/S sur fichier
- D. Comment formater un fichier texte pour l'utiliser avec un tableur
- E. Comment utiliser les fonctions avancées d'E/S des fichiers.

### A. Chaînes

Les chaînes sont des suites de caractères ASCII, affichables ou non. Elles permettent de transférer facilement des données entre des matériels quelconques. Les fonctions principales dans cette optique sont :

- Créer et manipuler des chaînes.
- Convertir des nombres en chaînes et réciproquement.
- Enregistrer et lire des fichiers ASCII.

#### Créer des commandes et des indicateurs de type chaîne



Utiliser la palette **Chaîne&chemins** pour créer des commandes et des indicateurs de type chaîne. La taille du cadre d'affichage est modifiable à l'aide de l'outil flèche. Il est possible d'afficher une barre de défilement pour les textes longs en sélectionnant **Eléments visibles»Barre de défilement**.

Par défaut, le mode d'affichage interprète les codes ASCII et les transforme en caractères. Pour visualiser ou entrer des caractères non imprimables, vous disposez du mode **Affichage des codes** 'V' qui affiche les caractères d'échappement comme en C :

- a. \00 - \FF Valeur hexadécimale d'un caractère 8 bits; doit être en majuscules
- b. \b Retour arrière (ASCII BS, équivalent à \08)
- c. \f Saut de page (ASCII FF, équivalent à \0C)
- d. \n Retour à la ligne (ASCII LF, équivalent à \0A)
- e. \r Retour chariot (ASCII CR, équivalent à \0D)
- f. \t Tabulation (ASCII HT, équivalent à \09)
- g. \s Espace (équivalent à \20)
- h. \\ Barre oblique inverse (ASCII \, équivalent à \5C)

Ou bien de l'**Affichage Hexadécimal** qui affiche les valeurs hexadécimales des caractères.

L'**Affichage style mot de passe** affiche des étoiles à la place des caractères.

## Tables



Les tableaux de chaînes peuvent être créés comme les autres tableaux, cependant, un type spécial de commande existe dans la palette **Liste et Tables**, ces sont les tables. Une table est un tableau 2D de chaînes, mais il est possible d'y adjoindre des entêtes de lignes et de colonnes, des barres de défilement, de donner un style de caractères pour chaque case, de changer la largeur d'une colonne ou la hauteur d'une ligne, bref de faire comme dans un tableur.

	T (K)	P (MPa)	Commentaire
Essai 1	10.65	12.4	SUP
Essai 2	14.87	12.5	SUP
Essai 3	54.38	12.5	RAS
Essai 4	77.21	12.4	RAS
Essai 5	96.35	12.5	RAS
Essai 6	45.87	12.4	RAZ

## B. Fonctions sur chaîne



La palette **Fonctions»Chaîne** contient une grande diversité de fonctions sur chaînes, voici les principales :

### Manipulation de chaînes



**Longueur d'un chaîne**—renvoie le nombre total de caractères d'un chaîne (y compris les caractères non imprimables).



**Concaténer des chaînes**—permet de « coller » bout à bout des chaînes. Cette fonction est redimensionnable.



**Sous-ensemble d'une chaîne**—permet d'extraire une sous chaîne d'un chaîne de départ, en spécifiant le point de départ et le nombre de caractères de la sous chaîne.



**Rechercher une expression**—recherche un motif dans une chaîne et extrait la chaîne précédant le motif, le motif, la chaîne suivant le motif.



**En majuscule/En minuscule**—change la casse d'une chaîne.



**Constantes de type chaîne**— pour créer des constantes, ou obtenir des caractères non imprimables « chaîne vide », « retour chariot CR », « saut de ligne LF », « CR/LF », et « tabulation ».



Note : il existe d'autres fonctions de manipulation de chaînes, accessibles par **fonctions de chaînes supplémentaires** depuis la palette des chaînes.

### Chaînes et Nombres

Les fonctions **Balayer une chaîne** et **Formater une chaîne** (équivalentes aux fonctions Fmt et Scan du C) permettent d'éclater une chaîne en éléments séparés ou de composer une chaîne à partir d'éléments séparés.



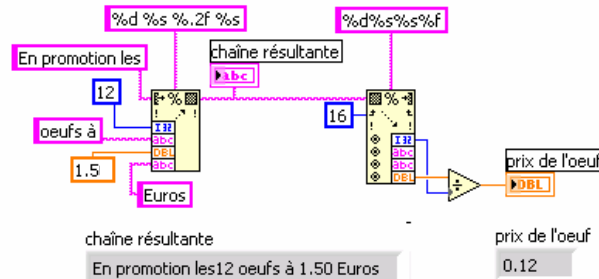
**Formater une chaîne** : convertit des arguments de tous types en une chaîne.



**Balayer une chaîne** : extrait d'une chaîne des variables de tout types.

Ces deux fonctions utilisent une chaîne de format pour spécifier les

types de données à convertir, leur longueur ... Consultez l'aide LabVIEW pour de plus amples renseignements. L'exemple ci-dessous illustre une utilisation de ces fonctions. Le nombre d'arguments de ces fonctions est modifiable en sélectionnant **Ajouter/Supprimer un paramètre** dans le menu contextuel.



Pour des conversions moins sophistiquées, vous disposez d'une palette d'outils dans **Chaîne»Conversion chaînes/nombres** qui ne nécessitent pas de chaînes de format.

### Exercice 7-1 : Extraction de données

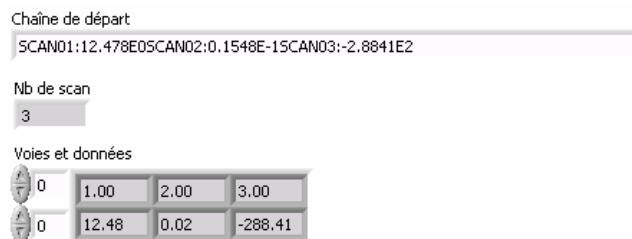
Objectif: Utiliser quelques fonctions de manipulation de chaînes.

Une centrale d'acquisition automatique renvoie des données au format suivant :

SCANxx:yy.yyyEySCANxx:yy.yyyEySCANxx:yy.yyyEy  
etc. où xx est un entier correspondant au numéro de la voie, et yy.yyyEy, un nombre en virgule flottante représentant la valeur lue sur la voie correspondante. On se propose de réaliser un VI capable d'extraire les couples Voie/Valeur dans un tableau 2D.

### Face Avant

1. Elle se présente comme suit :



### Diagramme



2. Le VI utilise une boucle WHILE et un registre à décalage. Il recherche la séquence « SCAN » pour extraire ensuite le numéro de la voie et la valeur. Les fonctions de manipulation de chaîne utilisées sont :
  - a. **Rechercher une expression**, cette fonction recherche une expression dans une chaîne de caractère, elle renvoie la sous chaîne avant l'expression, l'expression, la sous chaîne après l'expression et l'offset ou la correspondance est trouvée. Si aucune correspondance n'est trouvée, elle renvoie -1 dans l'offset.



b. **Balayer une chaîne** vous permettra de convertir la séquence `xx:yy.yyyEy` en deux nombres. ( Cf. la fin de l'aide en ligne de la fonction : exemples de balayage de chaînes)

3. Enregistrez votre VI sous le nom `Extraitvaleurs.vi`, entrez une chaîne au format souhaité et exécutez votre VI.

## Fin de l'exercice 7-1

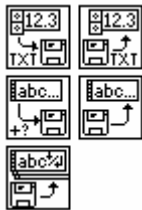
## C. Entrées/Sorties sur fichier



Les fonctions **E/S sur fichier** permettent de gérer toutes les opérations sur fichiers, notamment :

- Ouverture/Fermeture de fichiers
- Lecture/Ecriture
- Lecture/Ecriture au format tableur
- Déplacement, effacement, création de répertoires..
- Changement d'attributs
- Gestion de fichiers de configuration (.ini)

### Fonctions de haut niveau



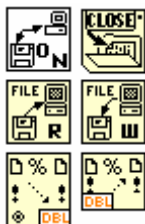
Les fonctions de haut niveau (icônes blanches en haut à gauche de la palette) permettent la lecture et écriture directe de fichiers textes, ou de fichiers tableurs (tableaux 2D avec séparateur). Elles intègrent l'ouverture/fermeture des fichiers, les boîtes de dialogues, la conversion des données... Ces fonctions sont à utiliser en priorité dans tous les cas simples.

### Fonctions de bas niveau



Ces fonction sont situées dans la sous palette **Fonction de fichiers avancée**, elle permettent l'ouverture/fermeture, l'accès séquentiel, la gestion des droits, la manipulation de répertoires, de noms...Ces fonctions ne seront pas étudiées dans ce cours.

### Fonctions de base



Une opération standard sur fichier opère ainsi :

1. Création ou ouverture d'un fichier.
2. Lecture ou écriture des données (éventuellement formatage).
3. Fermeture du fichier.

Les VI suivants permettent ces opérations :



**Ouvrir/créer/Remplacer un fichier**—renvoie un « Refnum » (un handler en C) qui identifie le canal de communication vers le fichier. Si le nom de fichier n'est pas spécifié dans les entrées du VI, une boîte de dialogue Windows standard s'ouvre.



**Lire un fichier**—extraît un nombre de caractères d'un fichier, identifié par son Refnum à partir d'un emplacement spécifié.



**Ecrire dans un fichier**—envoie une suite de caractères dans un fichier à l'emplacement spécifié. Ces deux fonctions travaillent sur des chaînes (ou des binaires après un changement de type)



## Gestion des erreurs

**Balayer à partir d'un fichier** et **formater dans un fichier**, permettent d'effectuer les opérations de lecture/écriture en même temps que la transformation des données.

**Fermer un fichier**—ferme le fichier spécifié par Refnum.

La gestion des fichiers est l'occasion de parler de la gestion des erreurs, car ce type d'opération en provoque souvent (média absent, disque plein, fichier en lecture seule...). La majeure partie des fonctions avancées de LabVIEW fait circuler une connexion d'erreur. Cette connexion est un cluster qui comporte :



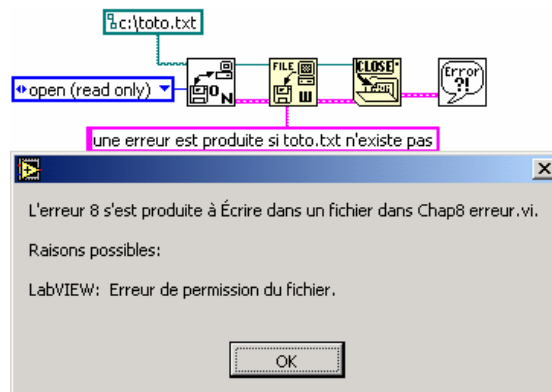
- Un **état** (booléen) indiquant si une erreur a eu lieu.
- Un **code** (int32) qui identifie le type d'erreur (codes d'erreurs complets dans l'aide), si code est différent de zéro et que l'état est faux, il s'agit d'un avertissement.
- Une **source** (string) qui indique le VI responsable de l'erreur.

Les VIs recevant une erreur n'effectuent aucune opération, il est donc important de prévoir au minimum l'affichage du cluster d'erreur, car on peut croire que l'ensemble des opérations s'est normalement déroulé alors qu'aucune n'a été effectuée.



Utilisez le **gestionnaire d'erreurs simples** de la palette **Temps&Dialogue** pour afficher un panneau signalant l'erreur.

L'exemple suivant illustre ce propos, `toto.txt` est ouvert en mode lecture seule. La fonction **Ecrire dans un fichier** génère donc une erreur. Notez que si l'on ne pose pas le gestionnaire d'erreur sur le diagramme rien ne permet de penser que `toto.txt` n'a pas été écrit.



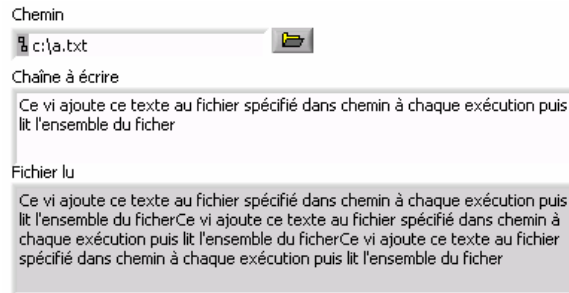
## Exercice 7-2 : Lecture/écriture d'un fichier

Objectif: Utiliser les fonctions de base pour faire des E/S sur fichier.

Créez un VI capable d'écrire une chaîne de caractères à la fin d'un fichier à chaque exécution et de lire l'intégralité du fichier avant de le fermer.

## Face Avant

1. Voici un exemple de la face avant, après la troisième exécution :



- a. Les commandes de type Chemin d'accès sont dans la palette **Chaînes&chemins**. L'icône à droite du champ ouvre une boîte de dialogue fichier de Windows.

## Diagramme



2. Le diagramme utilise les éléments suivants.
  - a. **Ouvrir/créer/Remplacer un fichier** permet de sélectionner le mode d'ouverture, choisissez **open or create**. Elle renvoie aussi le nombre de caractères contenus dans le fichier ; il vous sera utile lors de la lecture.
  - b. **Ecrire dans un fichier** permet d'écrire une chaîne de caractère à partir de la position spécifiée. Choisissez le « mode pos » adéquat.
  - c. **Lire un fichier** récupère un certain nombre d'octets à partir d'une position donnée.
  - d. **Longueur d'une chaîne** vous permettra de calculer le nombre de caractère à lire dans le fichier, connaissant sa taille à l'ouverture et le nombre de caractères ajoutés.
  - e. N'oubliez le **Gestionnaire d'erreurs** pour afficher d'éventuels problèmes d'accès.
3. Enregistrez votre VI sous le nom `demofichier.vi` et vérifiez son fonctionnement.

## Fin de l'exercice 7-2

### Utilisation de VIs de haut niveau

Les VIs de haut niveau effectuent l'ensemble des opérations **Ouverture/Conversion/Lecture&Ecriture/Fermeture** avec un seul VI. Il faut les utiliser dès qu'un fichier texte est accédé de façon occasionnelle car la procédure ouverture/fermeture systématiquement effectuée consomme inutilement des ressources systèmes lors d'accès fréquents.

Les VIs de haut niveau sont :



**Ecrire dans un fichier tableur**—Convertit un tableau 2D ou 1D de réels en une chaîne de caractères. Les données sont séparées par défaut en colonnes par des tabulations et en lignes par des retours chariot. Une chaîne de format peut changer le nombre de chiffres significatifs, la fonction peut transposer le tableau avant de l'enregistrer. Ce type de fichier est un format d'importation standard à tous les tableurs.



**Lire dans un fichier tableur** —Pour relire le fichier tableur, retourne un tableau 2D. Il est possible de ne lire qu'un certain nombre de lignes à partir d'une ligne donnée.



**Ecrire des caractères dans un fichier**—Ecrit une chaîne dans un nouveau fichier ou l'ajoute à un fichier existant.



**Lire des caractères dans un fichier** —Lit un certain nombre de caractères dans un fichier à partir d'une position donnée.



**Lire des lignes dans un fichier** —Identique, mais par ligne entière.



**VIs de fichiers binaires**—Lecture/Ecriture binaire d'entiers ou de réels simple précision (gain de place important).

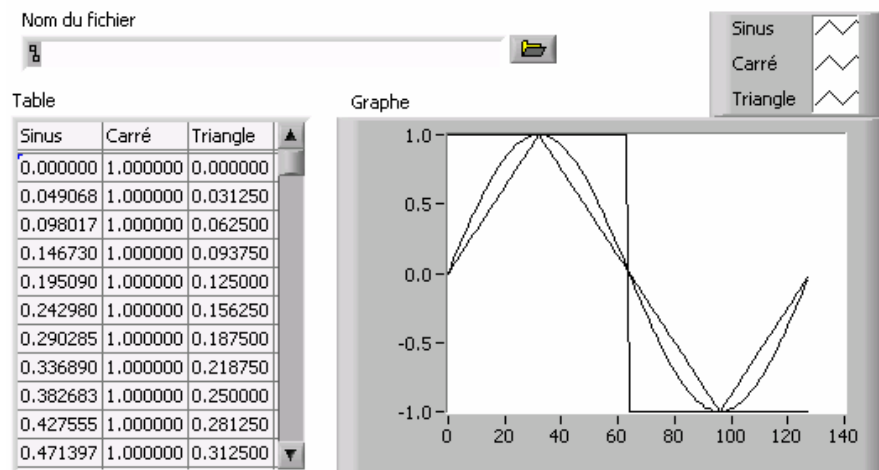
### Exercice 7-3 : Exemple d'enregistrement tableur

Objectif: Enregistrer un tableau 2D dans un fichier texte tabulé.

Dans cet exercice, vous créez trois formes d'ondes, Sinus, Carré et Triangle (128pts, Amplitude 1, fréquence  $7.81 \times 10^{-3}$  (1/128)). Ces signaux seront présentés sur un graphe, dans une table puis enregistrés au format tableur dans un fichier.

#### Face avant

- En voici une représentation, l'indicateur à gauche est une table (rappel : disponible dans la palette **Liste&tables** indicateur de type texte, choisir dans le menu contextuel éléments visibles/entête).



#### Diagramme



- Le diagramme ne contient pas de structure, les signaux seront générés par les fonctions d'analyse. Vous aurez besoin principalement des éléments suivants :
  - Signal sinus / Signal carré / Signal triangulaire.** Ces VIs génèrent des formes d'ondes dans un tableau, elles sont disponibles dans la palette **Analyse»Traitement du signal»Génération de signaux.**
  - Transposer un tableau 2D** permet de renverser lignes/colonnes en effet une fois les trois tableaux 1D (sinus, carré, triangle) convertis



en un tableau 2D, vous obtenez un tableau 3 lignes, 128 colonnes, et il est plus classique de présenter l’affichage en colonnes.

- Ecrire dans un fichier tableur vous permettra l’enregistrement du tableau 2D. Limiter, par une chaîne de format, l’enregistrement à 2 chiffres après la virgule.
3. Enregistrer le VI sous le nom `onde.vi` et exécuter le. Donnez le nom `fichier.txt` au fichier tableur, puis importer le dans Excel.

### Fin de l’exercice 7-3



## Chapitre 8 Acquisition de données

Ce partie décrit l'acquisition de données (DAQ ) à l'aide de cartes et de modules National Instruments.

### Vous apprendrez:

- A. Comment utiliser les cartes et le logiciel MAX
- B. Comment sont organisés les VIs DAQ
- C. Comment réaliser une entrée analogique
- D. Comment réaliser une sortie analogique
- E. Comment balayer plusieurs voies analogiques
- F. Comment contrôler les lignes d'E/S numériques
- G. Comment réaliser une acquisition de données bufférisée.

### A. Aperçu

La palette **Acquisition de données** contient les VIs pour contrôler les cartes DAQ de National Instruments. Ces cartes sont souvent multifonctions : conversion analogique numérique & numérique analogique, entrée/sortie numérique et compteur/timer. Il est indispensable de bien connaître les fonctionnalités de la carte utilisée pour pouvoir la programmer.

### Configuration matérielle

Sur plateformes PC, les cartes actuelles sont toutes Plug&Play, le port PCI attribue automatiquement les adresses et les interruptions requises par le matériel.



National propose un utilitaire d'exploration et de configuration de ses matériels, nommé MAX pour Measurement & Automation eXplorer.

Lorsque MAX est lancé, il détecte les périphériques qui lui sont connus et les placent dans la section **Périphériques et Interfaces**. Il est alors possible de tester la ressource matérielle et sa connexion au monde extérieur, de façon manuelle (toujours utile avant d'incriminer autre chose...).

MAX permet également d'indiquer aux VI d'acquisition comment est configurée la carte insérée dans le PC, c'est-à-dire par exemple, sa tension de référence, le mode bipolaire, entrées référencées en différentielles... Cette configuration permet aux VIs d'acquisition d'assurer la correspondance entre mots binaires et valeurs réelles.

Enfin et surtout MAX permet d'assigner un **numéro** à une **carte**, c'est ce numéro qui identifiera la carte dans LabVIEW. Cette indirection permet de changer de carte (si elles ont des possibilités semblables) sans modifier le code.

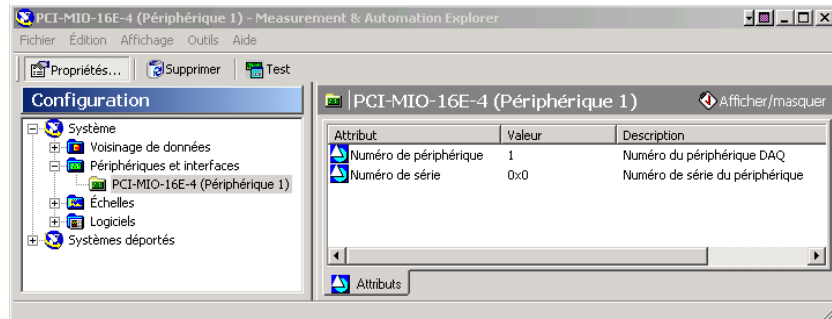
### Exercice 8-1 : Utilitaire MAX

Objectif: Utiliser Measurement & Automation Explorer pour reconnaître la configuration actuelle.

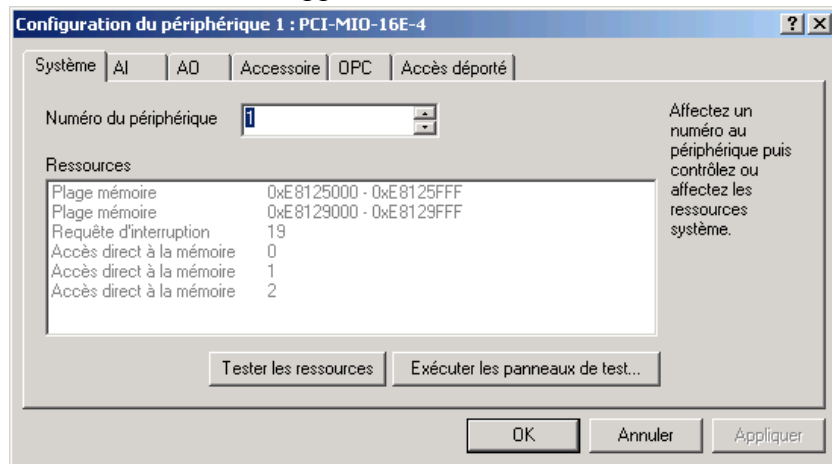
MAX va vous permettre de connaître le type de carte installée dans la machine mise à votre disposition et de vérifier le fonctionnement de l'ensemble.

## Examen de la configuration

1. Lancer MAX depuis le raccourci bureau.
2. Développer **Périphériques et interfaces** pour voir les cartes installées, vous devriez voir des LabPC ou des PCI 1200 qui sont des cartes E/S multifonction “low cost”.



3. Vous pouvez obtenir des informations sur la carte installée en sélectionnant **Propriétés** dans le menu contextuel associé à la carte. Une fenêtre apparaît :



Cette fenêtre possède plusieurs onglets. Le premier onglet, **Système**, indique les ressources système assignées à votre carte et le **numéro du périphérique** qui vous servira dans LabVIEW. Les autres onglets permettent de configurer les paramètres des entrées/sorties analogiques. Appuyez sur le bouton **Tester les ressources** pour vérifier la configuration de la carte.

## Tester les E/S de la carte.

4. Appuyez sur le bouton **Exécuter les panneaux de test**. Ces dernier vous permettent de tester les fonctionnalités de votre périphérique DAQ, telles que les entrées et les sorties analogiques et numériques. Examinez les différents panneaux et leurs possibilités.

## Fin de l'exercice 8-1

## B. VIs d'acquisition

Les VIs d'acquisition sont situés dans la palette **Acquisition de données**, vous y trouverez pour les VIs **Entrées analogiques**, **Sorties**



**analogiques, E/S numériques et compteur**, l'arborescence suivante :

- VIs Simples
- VIs Intermédiaires
- VIs Utilitaires
- VIs Avancés

### VIs simples

Les VIs simples font des opérations basiques sur une ou plusieurs lignes E/S. Ils opèrent à chaque exécution, la configuration de la carte, le lancement des opérations E/S, la libération de la carte... à utiliser dans toutes les acquisitions pas trop pointues.

### VIs intermédiaires

Les VI intermédiaires sont plus proches du matériel et permettent un contrôle plus fin, notamment en termes de gestion d'erreurs.

### VIs utilitaires

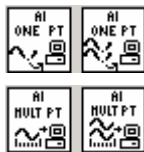
Ils sont à mi-chemin entre simples et intermédiaires.

### VIs avancés

Ils permettent d'accéder aux horloges, aux tampons, multiplexeurs... bref d'opérer très près du matériel, usage réservé aux pros.

## C. Entrées analogiques

Quatre VIs simples pour cette partie :



- Deux pour l'acquisition d'une **seule** valeur sur une ou plusieurs voies.
- Deux pour l'acquisition d'une **suite de valeurs** avec une période d'échantillonnage fixe sur une ou plusieurs voies.

Pour l'ensemble de ces VI, **périphérique** est le numéro de périphérique défini dans MAX, **voie** est une liste de voies à scruter, **limite haute et basse** sont les valeurs maxi et mini que peuvent lire les voies, elles définissent le gain. Les VIs fournissent les résultats sous la forme d'un **waveform** qui regroupe au minimum l'instant de départ, la période d'échantillonnage, le tableau de valeurs. Consultez l'aide pour de plus amples renseignements

### Nom de voies DAQ

Pour nommer les voies d'acquisition, utilisez le contrôle **Nom de voie DAQ** situé dans la palette **E/S**. Ce contrôle interroge l'enregistrement de configuration de MAX et propose les voies adaptées à la configuration du PC. L'option **Autoriser les noms non définis** permet de créer des numéros de voies sans passer par les « voies virtuelles » définies dans MAX ou en l'absence de cartes d'acquisition. Si une seule voie est utilisée, on saisit le numéro correspondant, s'il s'agit d'une liste, on sépare, par des virgules, les différents numéros des voies.

### Exercice 8-2 : Voltmètre

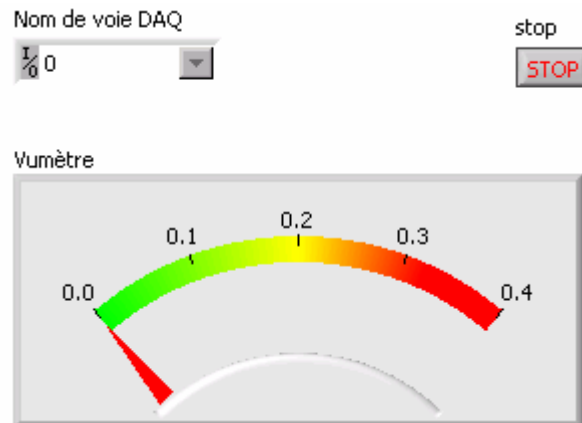


Objectif: acquérir un signal analogique en utilisant une carte d'acquisition de données.

Vous allez construire un VI qui mesure la tension générée par le capteur de température intégré dans la boîte de démonstration. Ce capteur génère une tension proportionnelle à la température ( $10\text{mV}/^\circ\text{C}$ ). Il est câblé physiquement sur la voie 0.

## Face Avant

1. Par exemple :



- a. Le capteur fournit une tension comprise entre 0 et 0.4V. Calibrez l'échelle du vue mètre en conséquence.
- b. Activez l'option **Autoriser les noms non définis** du contrôle E/S.

## Diagramme



2. Dans une boucle while dont l'arrêt est assujéti au bouton STOP, faites l'acquisition d'un point toutes les 100 ms
  - a. La fonction « **AI Echantillonner une voie** » lit la tension sur la voie concernée. Elle retourne une donnée de type waveform, mais le polymorphisme des indicateurs (cas particuliers des waveforms) fait qu'ils extraient et affichent la valeur mesurée.
3. Enregistrez votre VI sous le nom `Voltmetre.vi`. Exécutez le, en posant le doigt sur le capteur de température vous devriez voir évoluer la tension mesurée.
4. Dans le menu contextuel du contrôle E/S, sélectionnez **Nouvelle voie DAQ**. Cette option permet de créer une « voie virtuelle », C'est-à-dire une voie connue par un nom dans LabVIEW et dont les propriétés sont définies dans MAX, cette indirection permet de changer de matériel (carte ou capteur) sans changer le code...

Suivez les panneaux de configuration de MAX :

- Entrée analogique.
- Nom de la voie : T, Description : capteur intégré.
- Capteur de température fournissant une tension.
- Gamme 0 à 100 degrés C.

- Echelle personnalisée.
- Nom : Celsius, échelle linéaire.
- $Y = 100 X + 0$ .
- Carte DAQ xxxx(sélectionnez la carte disponible dans le PC), voie 0, Asymétrique référencée par rapport à la masse.

Une fois ces indications entrées, sous MAX, vous devriez voir apparaître dans « **Voisinage de données** », une voie **T**, dans « **Echelles** », une échelle **Celsius**

5. Dans la liste déroulante du contrôle E/S, apparaît la voie **T**, sélectionnez la, changez l'échelle de l'indicateur de température pour qu'elle corresponde à l'étendue 0->100 °C. Supprimez le numéro de périphérique connecté au VI d'acquisition. Relancez le VI, tout doit fonctionner, sans préciser dans le code, ni la carte, ni la voie, ni le facteur d'échelle, magique non ?

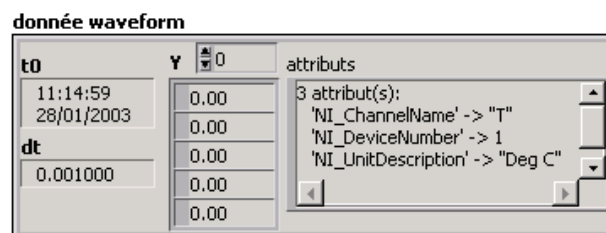
## Fin de l'exercice 8-2

## D. Entrées analogiques type Waveform

Dans de nombreux cas, l'acquisition point par point n'est ni assez rapide, ni suffisamment précise en temps. Dans ce cas, il convient d'utiliser les VIs **Acquérir un signal** ou **Acquérir des signaux**. Ces VIs font l'acquisition d'une rafale de points à une fréquence d'échantillonnage déterminée. L'opération étant gérée par le matériel de la carte, on peut compter sur des débits très importants sans perte de points. Le format natif renvoyé est de type Waveform.

### Donnée type Waveform

Un type waveform est un cluster qui contient les données d'acquisition, les éléments temporels et des attributs. Il peut s'agir aussi d'un tableau de clusters en cas d'acquisitions sur plusieurs voies.



Il est possible de les câbler directement aux Graphes, les échelles de temps sont fournies par le waveform.

Il existe une multitude de fonctions pour traiter ces données, elles sont situées dans les palettes **Analyse** et **Waveform**, consulter ces bibliothèques pour vous faire une idée de la richesse des traitements possibles.

## Exercice 8-3 : Acquisition traitement enregistrement d'une Waveform

Objectif: Ecrire une acquisition dans un fichier texte.

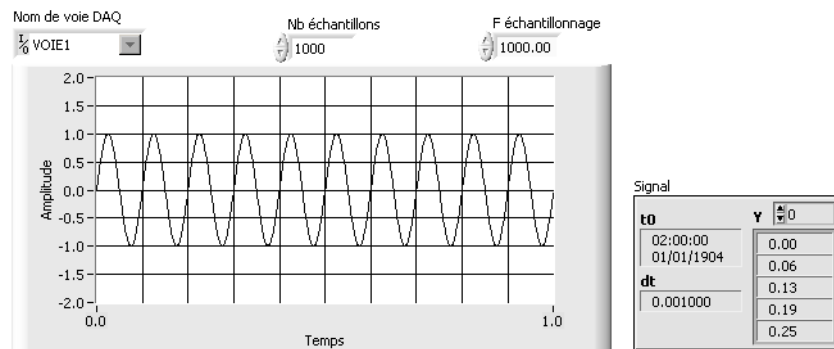
On se propose de réaliser une acquisition de X points à la vitesse de Y

échantillons/seconde et d'enregistrer le résultat dans un fichier texte.

Sur la boîte de démonstration, connecter l'entrée analogique 1 à la sortie sinus du générateur de fonctions.

## Face avant

1. Créez une face avant ressemblant à:



**Nombre d'échantillons** spécifie le nombre de points à acquérir et **fréquence d'échantillonnage** la vitesse d'acquisition.

- a. Créez une voie virtuelle de gain 1 appelée VOIE1, faisant l'acquisition sur l'entrée analogique 1.

## Diagramme



2. Construisez le diagramme en tenant compte des éléments suivants :
  - a. **AI Acquérir un signal** permet de lancer une acquisition avec les horloges internes à la carte.
  - b. **Exporter des waveforms dans un fichier tableur**, écrit les données au format txt en créant une entête.
  - c. Pensez à la gestion d'erreurs.
3. Enregistrez le VI sous le nom `Fichier_Waveform.vi`.
4. Lancez le code, enregistrez les données sous le nom `acq.txt`.
5. Ouvrez `acq.txt` à l'aide du bloc note ou d'Excel, examinez les données et l'entête.

## Fin de l'exercice 8-3.

## E. Sorties analogiques



Les fonctions de sorties analogiques ont un fonctionnement identique aux entrées, elles travaillent soit sur une donnée unique, soit sur une waveform, et cela sur une ou plusieurs voies.

Si une erreur survient, un message d'alerte apparaît et demande de stopper l'exécution ou de l'arrêter.

## Génération de Waveform

Dans les applications où le paramètre temps est crucial, il faut utiliser la fonction **AO-Générer un signal** qui génère une forme d'onde d'après un tableau de valeurs ou une waveform, en cadencant la mise

à jours des valeurs par une horloge propre à la carte.

### Exercice 8-4 Sortie en tension



Objectif: Créer un signal périodique en sortie de carte DAQ.



Reprenez l'exercice 8-3, modifiez le câblage en connectant l'entrée analogique 1 à la sortie analogique 0. Changez le code pour générer une forme d'onde que vous choisirez dans la palette **Génération de waveform**.

#### Face avant

1. Vous pouvez ajouter un graphe pour visualiser la forme d'onde avant la sortie sur la carte.

#### Diagramme



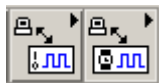
2. Tenez compte des éléments suivants :
  - a. **AO Générer un signal** demande en entrée un numéro de voie et de carte, ces éléments peuvent être inclus dans une voie de sortie virtuelle comme pour les entrées analogiques. La fréquence de mise à jour est donnée par l'entrée associée, les éléments t0 et le dt du waveform sont ignorés.
3. Vérifiez le fonctionnement de la chaîne complète.

#### Fin de l'exercice 8-4

## F. Les compteurs et entrées/sorties analogiques

Les compteurs sont des ensembles de registres capables de s'incrémenter ou de se décrémenter sur des fronts venant de l'extérieur ou d'une horloge. Ils sont utilisés pour le comptage, la mesure de fréquence, de période, pour générer des impulsions...

Les entrées sorties numériques sont des lignes logiques, physiquement disponibles. Elles permettent de commander des actionneurs tout ou rien, et de lire l'état de capteurs binaires.



Les fonctions propres aux compteurs et aux lignes numériques sont dans les palettes **Acquisition de données»E/S numériques** ou **Compteurs**.

Nous n'utiliserons que deux fonctions, l'une de comptage, l'autre d'écriture dans un port.

### Exercice 8-5 Comptage et affichage

Objectif: Créer un VI de comptage et d'affichage.

Cet exercice met en œuvre le codeur incrémental du pupitre de démonstration, le compteur 0 de la carte ainsi que les quatre LED du pupitre, associées au port 0. Le codeur fournit 24 impulsions par révolution, nous nous proposons de les compter, puis de les afficher (1) sur la face avant du Vi dans un entier et sur quatre indicateurs binaires, (2) sur les quatre LED du pupitre de démonstration.

## Front Panel

1. La face avant peut ressembler à :



Les quatre LED du pupitre doivent refléter l'état des indicateurs binaires de la face avant (tableau de LEDs), qui codent eux même la valeur décomptée modulo 16. (Rappel :  $18 \equiv 2 \text{ modulo } 16$ )

## Diagramme

2. Le diagramme comporte une boucle While dont la condition d'arrêt est liée à l'état du bouton STOP. Le Vi compteur doit être initialisé avant l'entrée dans la boucle de comptage et arrêté en fin de boucle.
3. Vous utiliserez les éléments suivants:



- **Compteur » Compter des événements ou un temps**, configure un ou deux compteurs pour compter un nombre d'événements ou un temps écoulé, codé par un entier long sur 32 bits. Un événement externe est une transition du signal sur la broche SOURCE spécifiée du compteur. Cette broche du Vi sera connectée à la sortie A ou B du codeur. **Démarrer/redémarrer** permet de configurer et démarrer le(s) compteur(s), **Stop** permet d'arrêter le(s) compteur(s)



- **E/S numériques simples » Écrire sur un port numérique** Produit un motif binaire de sortie sur une voie numérique spécifiée. Par exemple, une valeur binaire de 15 produit un motif de sortie 00001111 (pour un port 8 bits).



- **Numérique » Quotient & Reste** donne accès à la division entière. Cette fonction permet de connaître la valeur à afficher modulo 16



- Pour convertir un entier 32 bits (correspondant au type disponible en sortie de la division entière) en un entier 8 bits, utilisez la fonction de conversion **Numérique » Conversion » En entier octet non signé**. Dans même palette, **Nombre en tableau de booléens** permet d'obtenir un tableau de booléens (à afficher) à partir d'un entier.

- **Tableau » Sous-ensemble d'un tableau** permet d'extraire les quatre bits de poids faible. **Renverser un tableau 1D** permettra d'afficher l'élément de poids fort à gauche.

4. Enregistrez le VI et exécutez le, vérifiez l'incrémentation de l'indicateur **décompte**, l'affichage des LED du pupitre et de la face avant.

## Fin de l'exercice 8-5



## Chapitre 9 Contrôle d'instruments

---

Cette partie décrit les fonctionnalités de LabVIEW spécifiques au contrôle d'instruments distants, connectés par liaison GPIB ou série.

### **Vous apprendrez:**

- A. Généralités
- B. Configuration et communication par GPIB
- C. Communication avec les instruments
- D. Utilisation des pilotes
- E. Les VISA
- F. Utilisation des VISA
- G. Configuration et communication par liaison série.

### **A. Généralités**

---

Les instruments de mesure professionnels sont en général pourvus d'une interface permettant de les relier à un ordinateur. Cette connexion permet de configurer l'instrument et de récupérer les données acquises. Deux grands standards se partagent le marché de l'instrumentation, le bus GPIB et la liaison série. La liaison GPIB est mieux définie par la norme. Plus robuste, elle offre des avantages précieux en terme de synchronisation. Son coût est cependant nettement plus élevé.

La liaison série équipe en standard les ordinateurs, mais les problèmes de connexion et de configuration sont parfois déconcertants !

### **B. Configuration et Communication GPIB**

---

La norme ANSI/IEEE 488.1-1987, plus connue sous le nom General Purpose Interface Bus (GPIB), décrit les spécifications électriques, mécaniques et fonctionnelles de l'interface. La norme 488.2 étend ces spécifications au langage utilisé.

La liaison GPIB est une liaison parallèle 8 bits, capable de débits de  $\geq 1 \text{ Mo/s}$ . Chaque interlocuteur du bus est repéré par une adresse unique (comprise en 0 et 30). Les spécifications générales sont les suivantes :

- Un maximum de 15 appareils connectés au bus,
- Une longueur de câble de 4m au maximum entre 2 appareils,
- Une longueur totale de câble maximale de 20m,
- Au minimum, les 2/3 des appareils connectés doivent être sous tension.

### **Architecture du logiciel**

L'implémentation des ressources GPIB est comparable à celle utilisée pour les cartes d'acquisition. Le pilote (DLL contenant l'ensemble des commandes GPIB) est installé automatiquement par le module « plug & play » lors du démarrage de la machine. Le logiciel « Measurement & Automation eXplorer – aussi appelé MAX » permet de configurer et de tester la liaison.

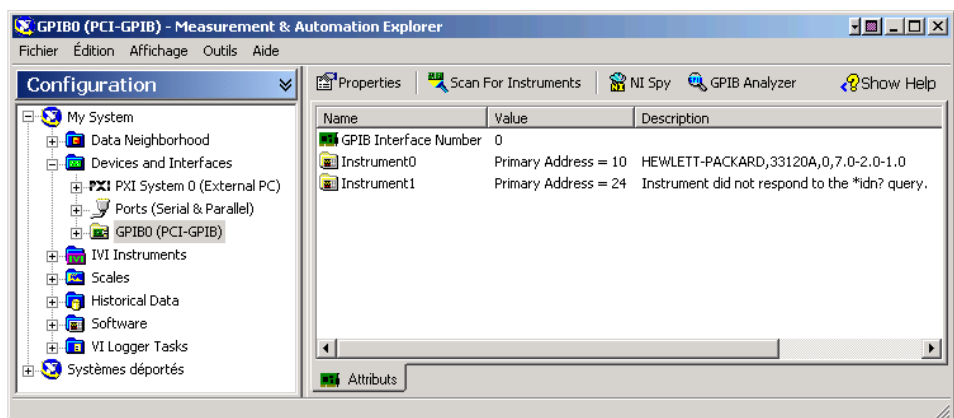
## Configuration du logiciel

Pour configurer la carte une fois le pilote installé, lancer MAX. Développer la section **Périphériques et interfaces**, une section **GPIB0** doit apparaître. Cette dernière peut également être développée indiquant ainsi tous les appareils sous tension, physiquement connectés au bus. Un clic droit sur un instrument spécifique permet de vérifier son adresse et de communiquer avec lui de façon sommaire pour vérifier que la communication fonctionne correctement.

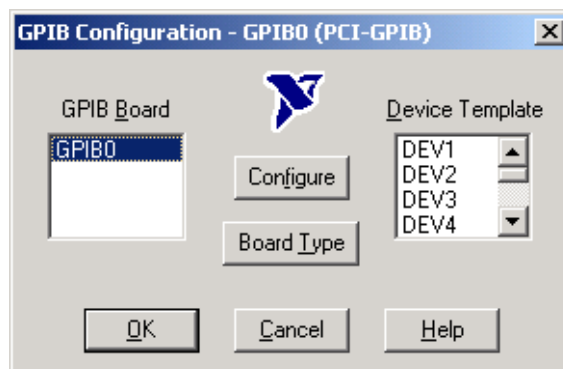
La façon la plus simple de vérifier que le périphérique communique correctement avec le PC, consiste à demander au périphérique son nom.

Pour cela il faut envoyer (écrire) la commande « \*idn ? », puis dans un deuxième temps, lire la réponse de l'appareil. Certains appareils anciens, ne possédant pas de nom propre, renvoient une donnée en réponse à la commande. L'appareil ayant reconnu que le PC lui parle doit, sur sa propre face avant, allumer ses voyants « Remote » et « Listen ».

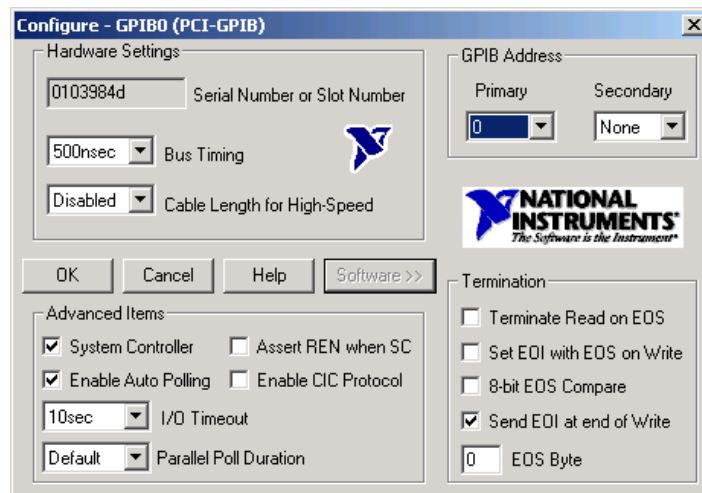
La copie d'écran présentée ci-dessous vous donne le résultat de cette opération réalisée par MAX en cliquant sur le bouton **Scan For Instruments**. Vous pouvez constater que le générateur HP33120 répond à la commande <\*idn?>, mais l'appareil d'adresse 24 ne comprend pas cette commande car il n'est pas conforme à la norme IEEE 488.2 (appareils fabriqués avant 1992). Notez aussi que la carte porte le numéro 0, il est possible d'utiliser simultanément plusieurs cartes GPIB dans le PC.



Un clic sur le bouton **Properties** de la carte GPIB ouvre le panneau de configuration, il est possible de configurer indépendamment les cartes interfaces et les périphériques.



Dans un premier temps, étudions la configuration de la carte GPIB0. Les valeurs présentées ci-dessous sont les valeurs par défaut.



Le cadre **Hardware Setting** définit le timing du bus, avec des périphériques standard et une carte GPIB non « High Speed », il est préférable de garder la configuration par défaut.

Le cadre **GPIB Address** définit l'adresse de la carte sur le bus IEEE (comme tout interlocuteur connecté au bus), la valeur par défaut est 0, l'étendue valide est de 0 à 30. Aucun autre périphérique ne doit avoir cette adresse. Les adresses secondaires sont rarement utilisées.

Le cadre **Advanced Items** définit :

- **System Controller** le PC est le maître des échanges, il définit qui parle et qui écoute. Il n'y a qu'un seul contrôleur sur le bus!
- **Enable Auto Polling** autorise la scrutation automatique des appareils lorsque l'un d'entre eux fait une demande de service
- **Assert REN when SC** bascule automatiquement la ligne Remote enable lorsque le système est contrôleur. Il est parfois utile de cocher cette case avec certains appareils anciens qui ne passent pas spontanément en mode remote, à la reconnaissance de leur adresse.
- **Enable CIC Protocol** active une procédure de reprise de contrôle automatique du bus.
- **Time Out** est la durée maximale d'attente de réponse d'un périphérique ; passé ce délai, le contrôleur reprend la main de façon autoritaire (évite tout blocage du bus, attention cependant à ne pas mettre des valeurs trop faibles pour la RAZ de certain appareils complexes).
- **Parallel Poll Duration** est le temps maximum d'attente à une demande de scrutation parallèle, en standard 2  $\mu$ s, plus si des extensions de bus sont utilisées.

Le cadre **Termination** configure la façon dont l'échange entre deux appareils connectés au bus se termine. La norme GPIB prévoit qu'une ligne (un fil physique du bus) particulière, nommée EOI (End of identify), peut basculer pour indiquer la fin d'une transmission. Les périphériques utilisent souvent un (des) caractère(s) terminateur(s) (EOS End of String) en plus du basculement d'EOI. Les plus fréquents :

'CR', 'LF' et bien sûr la combinaison 'CR LF'. Une dernière méthode pour connaître la fin d'un message, connaître le nombre de caractères attendus (de loin la plus mauvaise). La configuration matérielle de la carte propose les options suivantes :

- **Terminate Read On EOS** cette option termine la lecture sur réception du caractère terminateur
- **Set EOI With EOS on Write** Bascule la ligne EOI au moment de l'envoi du terminateur. Les terminateurs hard et soft sont ainsi envoyés.
- **8 Bits EOS Compare**, par défaut les codes ascii sont sur 7 bits, et le terminateur est cherché sur les 7 bits de poids faible, cette option force la comparaison sur 8 bits.
- **Send EOI At End Of Write** force la ligne EOI en fin d'écriture
- **EOS byte** valeur décimale de l'EOS

## C. Communication avec les instruments

### Caractéristiques propres d'un appareil

Chaque appareil possède des caractéristiques propres dont il faut prendre connaissance avant de commencer à programmer l'appareil.

#### Adresse

L'adresse d'un appareil permet de l'identifier sur le bus. En conséquence deux appareils ne peuvent pas avoir la même adresse. Chaque appareil possède une adresse par défaut qui peut être changée sur le périphérique lui-même, en utilisant (généralement) les menus accessibles manuellement par sa face avant.

Un périphérique, reconnaissant son adresse cheminant sur le bus, passe en mode commande à distance (remote) ; en conséquence sa face avant est inhibée. Un appui sur le bouton « local » de cette même face avant peut la rendre à nouveau active

#### Mot de programmation

Pour commander à un périphérique d'exécuter des tâches, il est nécessaire de lui envoyer des ordres sous la forme de chaînes de caractères ASCII. Ces derniers sont généralement écrits dans un protocole intitulée SCPI (Standard Code for Programming Instruments). Un effort de normalisation du langage de commande a permis l'apparition du langage SCPI. Celui-ci est accepté par tous les appareils postérieurs à 1990. Chaque phrase du message correspond à une chaîne de caractères (ex : « \*idn? » ou « :FUNC:MEAS »). La compréhension par le programmeur des nombreuses commandes SCPI et leur assemblage dans une syntaxe très stricte requiert une lecture très approfondie des notices des appareils. Elle permet théoriquement d'écrire un code correspondant exactement à l'application recherchée. La phase de développement peut être assez longue et fastidieuse.

Les codes correspondent à des tâches génériques. Ainsi le code de deux appareils, de fonction identique, sont-ils très proches et souvent

compatibles. Les mots de programmation sont précisés dans la notice de chaque appareil.

### Format des données

Les données sont exprimées en ASCII ou parfois en binaire, dans un format propre à chaque appareil. Une fois choisi le format, celui-ci reste invariant – en particulier le nombre de caractères- quelle que soit l'amplitude de la donnée.

Le format peut être totalement numérique : 1.23456 ou  $1.23^e-4$ . Il peut également comporter une partie alpha-décimale : NDCV 1.234<sup>E</sup>-3, spécifiant la nature de la donnée (normal DC voltage). Dans les 2 cas, la donnée reste sous forme de chaîne et doit être transformée en nombre pour autoriser des traitements numériques.

### Etapes essentielles d'une communication PC <=> périphérique

Un programme informatique de commande à distance d'un instrument de mesure contient un certain nombre de tâches standards qui sont abordées ci dessous.

### Remise à zéro

Remettre un périphérique à zéro consiste à le configurer dans un état initial unique, prévu en entreprise, et normalement identique à celui observé lors la mise sous tension (conditions par défaut ou « default factory conditions »)..

La remise à zéro (RAZ) prenant souvent plusieurs secondes, il est nécessaire d'introduire dans le programme une temporisation permettant d'attendre la fin de cette RAZ.

### Configuration

Les appareils de mesures actuels possèdent un très grand nombre de modes de fonctionnement. En mode commande à distance (remote) ceux-ci sont imposés exclusivement par voie informatique. L'usage de la face avant des appareils reste alors limitée à la visualisation de résultats ou des status.

A chaque mode de fonctionnement correspond une **chaîne de caractères** (ou mot de programmation) qui devra **être écrite** par le PC sur le périphérique, pour lui imposer la ou les fonctions correspondantes. Les commandes des périphériques étant très nombreuses, il est nécessaire d'envoyer exclusivement celles qui diffèrent des conditions par défaut.

### Déclenchement

La configuration d'un appareil prépare la mesure, mais ne la déclenche généralement pas. L'exécution de la mesure ne s'effectue qu'après apparition des conditions de déclenchement.

A un ordre de déclenchement unique peut être associée une ou plusieurs mesures. Le déclenchement peut être :

- **Interne** : automatique donc non maîtrisé.

- **externe** : associé au basculement d'une ligne ou à la fermeture d'un interrupteur.
- **logiciel** : associé à l'expédition d'une commande informatique spécifique.

La maîtrise des conditions de déclenchement est souvent une étape critique de la commande à distance.

## Lecture

Une fois la mesure effectuée, la donnée est disponible dans la mémoire du périphérique. Le transfert de la donnée vers le calculateur est une opération de lecture.

La fin du transfert est associée à la détection du caractère de fin de chaîne, au basculement d'une ligne spécifique (EOI), à l'obtention d'un nombre de caractères ou enfin au dépassement du temps maximum autorisé (time out).

## Traitement des données

Les données peuvent être l'objet de traitements ultérieurs, une fois mises sous forme numérique. La bibliothèque de traitement de LV6.1 est riche de nombreux outils : visualisation, filtrage, régression, FFT, ....

Les traitements effectués par les appareils eux-mêmes : mise à l'échelle, FFT... relèvent de la configuration de l'appareil.

## Sources d'erreurs classiques

Les 3 points ci dessous étant responsables d'un grand nombre d'erreurs, il est souhaitable de préciser leur concept.

### Time out

La fiabilité des échanges impose de ne transmettre une nouvelle commande qu'après acceptation de la précédente. Pour ne pas bloquer le bus et le PC indéfiniment, on convient a priori d'arrêter l'échange en cours si celui-ci dure plus de X secondes (classiquement 10s). Le calculateur reprend alors la main, mais la commande ou la donnée en cause est perdue.

### Terminateur

Le terminateur indique la fin d'une commande ou d'une donnée. Sous peine d'erreur, le terminateur envoyé par le parleur doit être le même que celui attendu par l'écouteur (qu'il soit matériel ou (et) logiciel).

### Séparateur décimal

Le séparateur décimal reconnu par les appareils de mesures est le point alors que classiquement en français, la virgule est utilisée. Pour changer ce dernier, aller sur le bureau, dans le menu Paramètre/Panneau de configuration/ Options régionales /Nombre.

## D. Bibliothèques de contrôle d'appareils dans LabVIEW

### Vi's spécifiques à l'interface GPIB ou RS 232C



Etant avant tout un langage destiné à l'instrumentation, LabVIEW est doté de toutes les fonctions permettant de piloter les interfaces. Il dispose en particulier de jeux de Vi's spécifiques au GPIB (IEEE488) et à la liaison série RS232C. Les fonctions de commandes spécifiques sont situées dans la palette **E/S Instruments**.

La norme IEEE488 ayant évolué de la version 1 vers 2, deux bibliothèques spécifiques correspondent à chacune des étapes. Aujourd'hui il est possible de programmer exclusivement en IEEE488.2, pour autant que l'appareil à contrôler réponde à cette version de la norme.

### Visa

Virtual Instrument Software Architecture (VISA) est une couche logicielle de bas niveau qui permet de communiquer avec les pilotes d'entrées/sorties installés sur le système. Ce niveau d'abstraction supplémentaire permet de s'affranchir, dans une certaine mesure, de la nature de l'interface. Le code est plus facilement réutilisable pour d'autres types d'interfaces, et d'autres langages. VISA n'est pas un élément de LabVIEW, mais une interface d'application (API) standardisant les appels aux drivers. LabVIEW possède un jeu de fonctions permettant ces appels.

### Terminologie

Les fonctions VISA font appel à une **Ressource** du système terme générique désignant un appareil connecté sur un port (série, parallèle, GPIB, USB, VXI...).

Cette ressource possède un **Nom**, qui identifie son interface, son adresse et le type de **session**. Une **session** VISA est un identificateur logique unique utilisé par VISA pour communiquer avec une ressource, elle peut être de type E/S ou événement.

Les noms de ressources valides sont les suivants :

Interface	Syntaxe
VXI INSTR	VXI[carte]::adresse logique VXI[::INSTR]
VXI MEMACC	VXI[carte]::MEMACC
VXI BACKPLANE	VXI[carte][::adresse logique VXI]::BACKPLANE
VXI SERVANT	VXI[carte]::SERVANT
GPIB-VXI INSTR	GPIB-VXI[carte]::adresse logique VXI[::INSTR]
GPIB-VXI MEMACC	GPIB-VXI[carte]::MEMACC
GPIB-VXI BACKPLANE	GPIB-VXI[carte][::adresse logique VXI]::BACKPLANE
GPIB INSTR	GPIB[carte]::adresse primaire[::adresse secondaire][::INSTR]

GPIB INTFC	GPIB[carte]::INTFC
GPIB SERVANT	GPIB[carte]::SERVANT
PXI INSTR	PXI[carte]::périphérique[::fonction][::INSTR]
Serial INSTR	ASRL[carte][::INSTR]
TCPIP INSTR	TCPIP[carte]::adresse hôte[::nom du périphérique LAN][::INSTR]
TCPIP SOCKET	TCPIP[carte]:: adresse hôte::port::SOCKET

Nom de ressource VISA



GPIB0::10::INSTR  
GPIB0::24::INSTR

Si vous utilisez un appareil GPIB d'adresse 21 connecté à la carte n°0 la ressource a pour nom **GPIB0::21::INSTR**. Si vous connectez un périphérique à la liaison COM1, la ressource prend pour nom : **ASRL0::INSTR**.

Si les ressources sont présentes sur la machine lors du développement, elle apparaissent automatiquement dans la liste déroulante associée à un contrôle de type VISA. Si ce n'est pas le cas, il faut entrer le nom de la ressource au clavier.

## Utilisation des VISA

Les fonctions VISA les plus couramment utilisées pour communiquer avec les instruments de mesure sont les fonctions VISA : VISA Write et VISA Read.

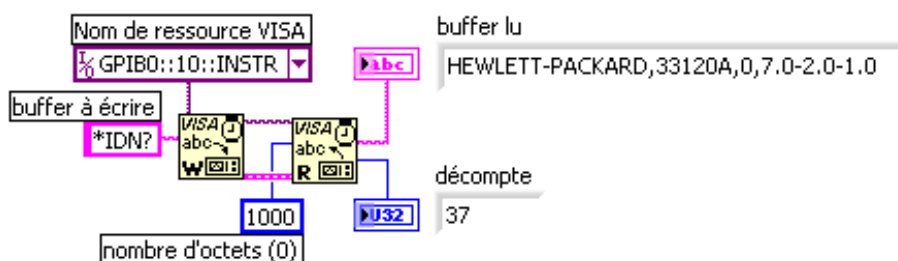


**VISA Write** écrit le contenu d'un tampon vers le périphérique désigné par le nom de la ressource VISA.



**VISA Read** lit les données en provenance du périphérique. Le VI demande le nombre de caractères à lire. Cette valeur doit être supérieure ou égale au nombre d'octets à transmettre. Si la fin de message est implémentée par le matériel (GPIB) la lecture s'arrête sur le terminateur, sinon elle s'arrête au nombre de caractères ou au time out.

L'exemple suivant fait une identification d'un appareil connecté sur une carte GPIB.



## Exercice 9-1 : Programmation d'un voltmètre et lecture d'une résistance

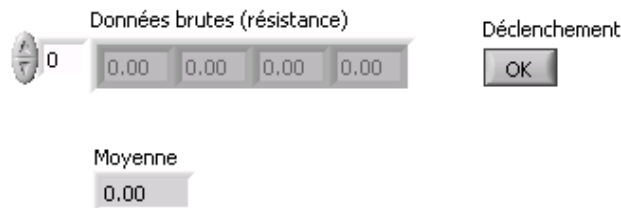
Objectif: Configurer un appareil de mesure, déclencher la conversion et rapatrier la donnée.

Dans cet exercice, vous configurez en utilisant le protocole SCPI un multimètre en ohmmètre 4 fils, avec un déclenchement sur GET (Group exécute trigger) de 4 mesures successives puis vous transférez les données vers le ordinateur.



## Face avant

1. En voici une représentation, l'indicateur à gauche est un tableau, le bouton déclenchement envoie un ordre GET.



## Diagramme

2. Le diagramme contiendra une boucle infinie dans laquelle on entre après une phase d'initialisation du multimètre. Lisez attentivement la documentation de l'appareil pour trouver les mots de programmation. Vous aurez besoin principalement des éléments suivant :



- Les fonctions **VISARead/Write** pour configurer l'appareil et lire les données. Parmi les éléments à écrire dans la phase d'initialisation, il faut :
    - a. définir la configuration de l'appareil (ohmmètre 4 fils)
    - b. spécifier la source de déclenchement (ici le bus)
    - c. imposer le nombre de mesures réalisées à chaque déclenchement
  - La fonction **VISA Assert Trigger** assure le déclenchement logiciel de la mesure par le périphérique (le mode de déclenchement dépend du type de la ressource VISA ; en GPIB envoie la commande \*trg).
  - Vous allez certainement avoir besoin des fonctions de conversion chaîne↔chiffres ( Fonction/conversion chaîne→Nombre/chaîne décimale en nombre).
3. Enregistrez le VI sous le nom `acq1eee.VI` et exécutez le.

## Fin de l'exercice 9-1

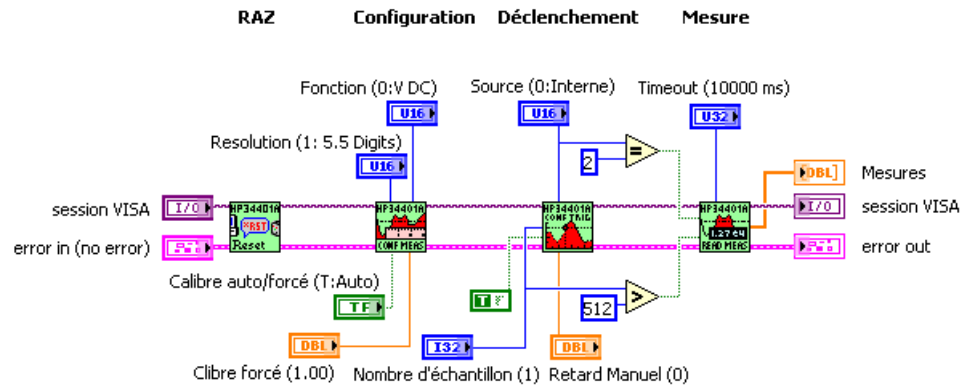
## E. Driver d'instrument

Pour faire gagner du temps aux utilisateurs, les grands constructeurs écrivent eux même et pour chaque appareil, un driver permettant d'effectuer la majorité des tâches susceptibles d'être demandées à celui-ci. La taille du VI correspondant croît, mais la mise en application devient alors beaucoup plus simple et rapide. La tâche du programmeur consiste alors à chaîner entre eux les Vi's issus du driver pour construire sa propre application.

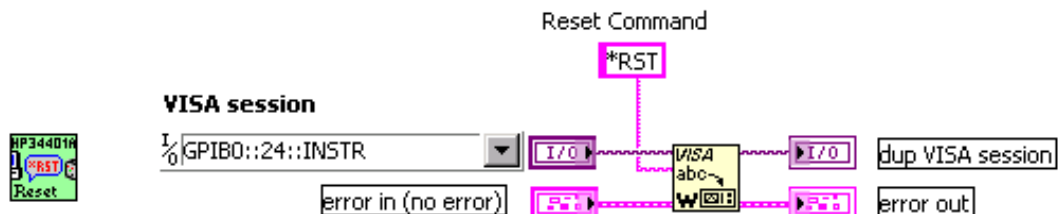
Un driver est une librairie LabVIEW (\*.llb) faisant appel, soit aux fonctions VISA de LabVIEW (drivers standards), soit à une DLL écrite dans un autre langage (driver IVI). La librairie est généralement téléchargeable depuis le réseau. Pour être visibles depuis LV6.1, les librairies standards **doivent** être déposées dans le répertoire `.../LV6.1/Inst.lib`, les librairies IVI ont un installateur qui se charge de tout.

## Exemple d'application simple

Voici un exemple des VIs disponibles pour le multimètre HP34401. Le programme ci dessous comporte quatre parties bien séparées

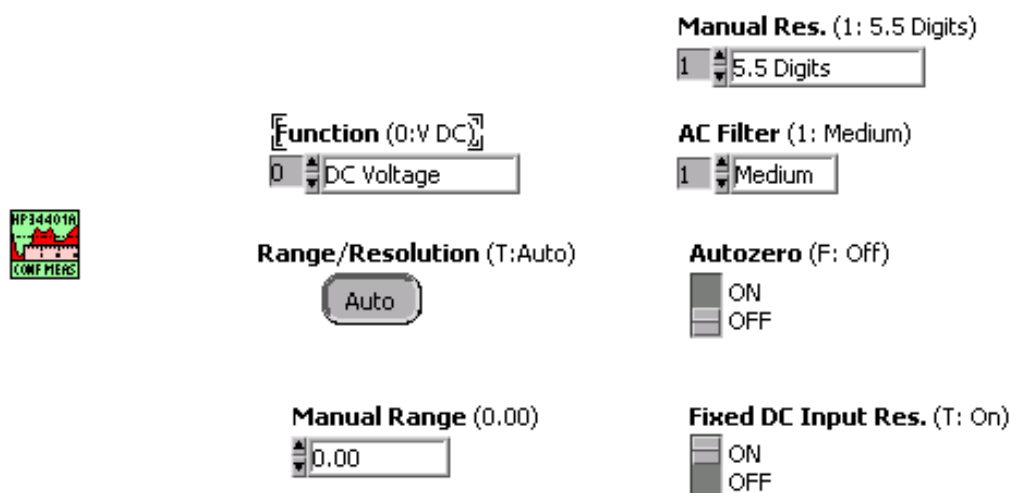


## Initialisation



Ce Vi définit le nom de la session visa ( ici GPIB0 ::10 ::INSTR) puis écrit la commande \* RST sur ce périphérique pour le remettre dans les conditions par défaut. Ainsi l'usage du Vi <Reset>, extrait du driver, rend l'action de remise à zéro totalement transparente.

## Configuration




Ce Vi reçoit du Vi précédent le nom de la session VISA et la gestion des erreurs

- Il permet définir toutes les configurations possibles du multimètre Hp34401 voltmètre continu ou alternatif, ampèremètre, ohmmètre, thermomètre...

- Le choix du calibre peut s'effectuer manuellement ou automatiquement (booléen vrai → choix automatique)
- En mode manuel, la résolution varie entre 1 et 5.5 digits, imposée par une liste déroulante.
- Dans le cas des mesures sur des grandeurs alternatives, l'efficacité du filtrage est accessible (forte, moyenne, faible) est accessible.


## Déclenchement



<b>Source (0: Internal)</b> <input type="button" value="0"/> Internal	<b>Delay (F: Auto)</b> <input type="button" value="Manual"/> <input type="button" value="Auto"/>
<b>Samples (1)</b> <input type="button" value="1"/>	<b>Manual Delay (0)</b> <input type="button" value="0.00000E+0"/>
<b>Trigger Count (1)</b> <input type="button" value="1"/>	

- Le contrôle « trigger count » fixe le nombre de déclenchement associé à une exécution du Vi « Conf Trig »
- Le nombre d'échantillons et le retard introduit entre chaque prise de points sont également accessibles de façon évidente. Par exemple si « trigger count = 2 » et si « sample count = 100 » alors le nombre de points de mesures est de 200.

## Lecture



<b>Timeout (60000 ms)</b> <input type="button" value="60000"/>	<b>Start Meas. (T)</b> <input type="button" value="Start Measurement"/> <input type="button" value="Read Block"/>	<b>Measurements</b> <input type="button" value="0"/> <input type="button" value="0.0000E+0"/>
<b># Points (50000)</b> <input type="button" value="50000"/>	<b>Software Trigger (F: OFF)</b> <input type="button" value="ON"/> <input type="button" value="OFF"/>	
	<b>Buffer (F: Internal)</b> <input type="button" value="Output"/> <input type="button" value="Internal"/>	<b>End of Data</b> <input type="button" value="End of Data"/> <input type="button" value="End of Block"/>
	<b>Data Storage (T: ON)</b> <input type="button" value="ON"/> <input type="button" value="OFF"/>	

La face avant permet entre autre de :

- Autoriser la prise des points de mesures tout en combinant les 8 modes de stockage et de transfert correspondant aux états des boutons à glissière
- S'adapter aux différences de fin de message suivant que la communication se fait par GPIB ou par RS 232C
- Préciser la durée du time out pour prendre en compte fait que l'acquisition de 1000 ou 2000 points de mesures peut prendre un temps non négligeable et que le calculateur doit patienter pendant tout ce temps là.
- NB : la complexité du Vi est évidente, la simplicité de sa mise en œuvre aussi ! C'est bien là l'intérêt de faire appel à un driver d'instrument.

**Exercice 9-2 : Programmation d'un multimètre et lecture d'une résistance via un pilote**

Objectif: Configurer un appareil de mesure, déclencher la conversion et rapatrier la donnée en utilisant un pilote d'appareil.

Reprenez l'exercice 9-2 en utilisant cette fois ci le pilote d'instrument installé sur votre machine.

**Fin de l'exercice 9-2****Que vérifier si le programme minimal ne fonctionne pas**

Si un périphérique ne fonctionne pas dans son environnement, il est conseillé écrire un 'petit bout de code' dit programme minimum, n'ayant rien avoir avec le programme (ou la chaîne de mesures) à déverminer mais effectuant une entrée sortie via l'interface incriminée. Cette approche permet de prendre du recul par rapport au problème avant de consulter une <hot line>. Il est en outre conseillé de :

- Vérifier si, depuis le logiciel, la carte est reconnue : pour cela observer visuellement l'état des voyants REM, Lstn ( listen) et Tlk (talker). Après un ordre d'écriture REM et Lstn doivent être allumés.
- Vérifier compatibilité des adresses sur le périphérique et dans le code. Si l'adresse est bonne le voyant REM doit être allumé. Dans le cas contraire, Forcer éventuellement la ligne REN en envoyant la commande <Visa GPIB Contrôle REM>
- Vérifier la compatibilité des terminateurs à l'émission et à la réception,
- Changer le câble ou le brancher si ce n'était pas le cas !
- Essayer des Mots de programmation simples qui doivent entraîner des changements de fonction et d'affichage. (préférer les déclenchements automatiques les plus simples).
- Vérifier si le programme minimum fonctionne en mode pas à pas .Dans l'affirmative, introduire 'judicieusement ! 'une temporisation ≈500ms derrière des taches un peu longues à exécuter
- Contrôler l'état du témoin de synchronisation ou de prise de mesure : si celui ci est toujours éteint, il y a vraisemblablement un problème de mode déclenchement,
- Compatibilité Time OUT avec tache longue asynchrone
- Tester l'état d'adressage du périphérique (Lstn ou Tlk) avant toute nouvelle tâche,
- Rechercher des exemples types dans la documentation papier ou sur Internet

**Que faire si une tache spécifique ne fonctionne pas**

- Débrancher tous les autres périphériques,
- Relire la documentation et en particulier les modes déclenchement,
- Ecrire un programme à part qui teste cette fonction spécifique,
- Rechercher une tache précédente mal terminée,
- Eteindre et rallumer le périphérique qui peut être bloqué. Cette méthode brutale vide efficacement son tampon de sortie.
- Quitter LV6.1 ou dans un deuxième temps éteindre le PC,

- Changer ordre d'allumage,
- Lancer un logiciel espion sur le bus,
- Remplacer le périphérique par un périphérique équivalent.

## F. Communications et configuration série

### Paramétrages possibles

La liaison série est mode de communication répandu dans le domaine informatique. Initialement prévue pour connecter de terminaux distants, elle s'est ouverte, en raison de son caractère économique, au contrôle d'instruments.

Il n'existe pas de protocole permettant un paramétrage automatique de la transmission. Il est donc nécessaire de les régler soi même. Parmi ces paramètres citons :

- La vitesse de transmission donne le nombre de bits par seconde, transmis par la ligne de communication (entre 110 et 56000 bauds).
- Le nombre de bits de données (7 ou 8).
- Eventuellement un bit de parité pour fiabiliser la transmission. Il s'agit d'un bit rajouté artificiellement aux bits constituant le message. Ce bit complémentaire, dit de parité, est forcé à 0 ou 1 de sorte que le nombre binaire transmis soit systématiquement pair (ou systématiquement impair), en anglais : even / odd respectivement.
- Le nombre de bits de stop (1, 1.5 ou 2) définit une durée pendant laquelle le signal reste bas après une transmission.
- Le contrôle de flux (handshake) qui peut exister, soit par le biais de lignes physiques (RTS/CTS), soit par un protocole logiciel (XON/XOFF), ou ne pas être géré.

Tous ces paramètres sont indépendants ; chacun d'entre eux peut rendre la liaison inopérante ! Ils doivent impérativement être configurés de la même façon pour les 2 partenaires de la communication série

### Connexion matérielle

Il existe deux normes principales de liaison série :

#### RS-232

La RS-232 est la liaison la plus commune sur les PC. Elle utilise des tensions référencées par rapport à une masse commune, ce qui la rend sensible aux perturbations et limite les longueurs de câbles (  $\approx 15$  mètres). Beaucoup d'appareils de mesures en sont équipés.

#### RS-449, RS-422, RS-423

La RS-449, et ses variantes RS-422, et RS-423 utilisent des paires différentielles qui permettent des communications sur de longues distances. Elles ont la faveur des fabricants de contrôleurs de process (capteurs intelligents, régulateurs, automates) répartis dans des usines en environnement très bruyé. (  $\approx 1$  km si combiné avec une boucle de courant 4-20mA )

## Câblage RS-232

Deux prises coexistent pour la connexion RS232, la DB25 et la DB9. Les appareils connectés sont soit des DTE (Data Terminal Equipment), soit des DCE (Data Communications Equipment). Les DTE sont des calculateurs, des terminaux ou des appareils de mesure, ils ont des prises mâles, les DCE sont des modems, des tables traçantes..., ils ont des prises femelles. Il faut croiser les fils de transmission et de handshake lorsque l'on connecte deux DTE ou deux DCE entre eux. Généralement les câbles Male/Femelle sont câblés fils à fils, les Femelle/Femelle et Male/Male sont croisés.

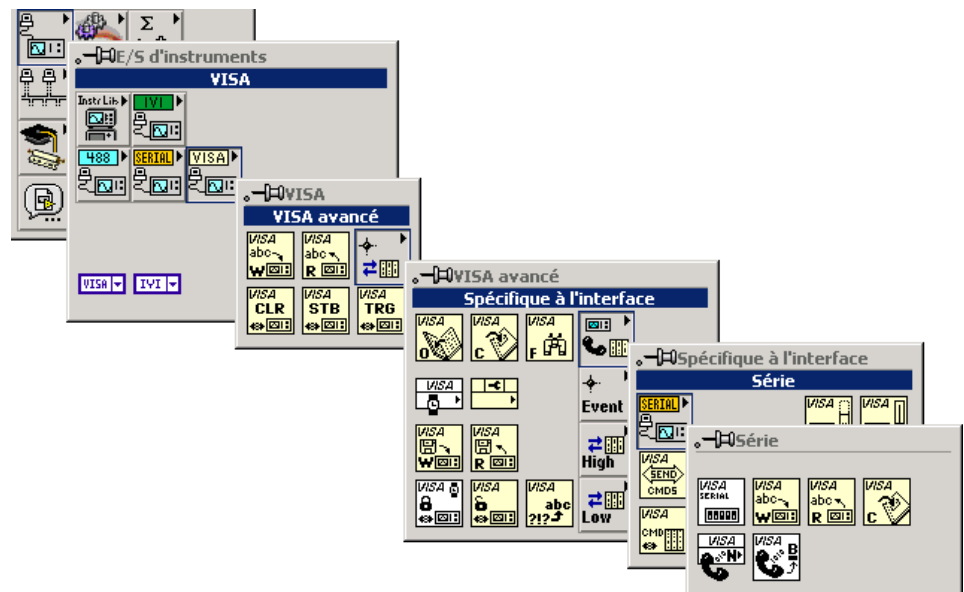


**Note** Pour simplifier les choses, les fiches DB-9 et DB25 ont les broches 2 et 3 inversées (2=Tx et 3=Rx sur DB25 alors que 2=Rx et 3=Tx sur une DB9 !). Bref c'est pas toujours facile pour câbler tout ça, mais de nombreux sites existent sur le net pour faciliter la tâche. En cas d'erreur, cela ne fonctionne pas mais cette situation n'est dangereuse ni l'interface ni pour le périphérique.

## VISA et liaison série

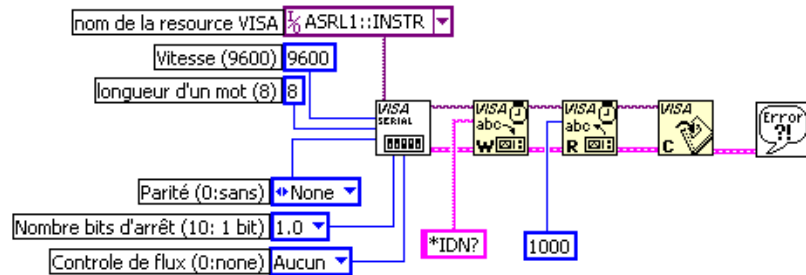
### VISA spécialisés

Il existe dans la palette des fonctions VISA avancées une sous palette dédiée à la liaison série.



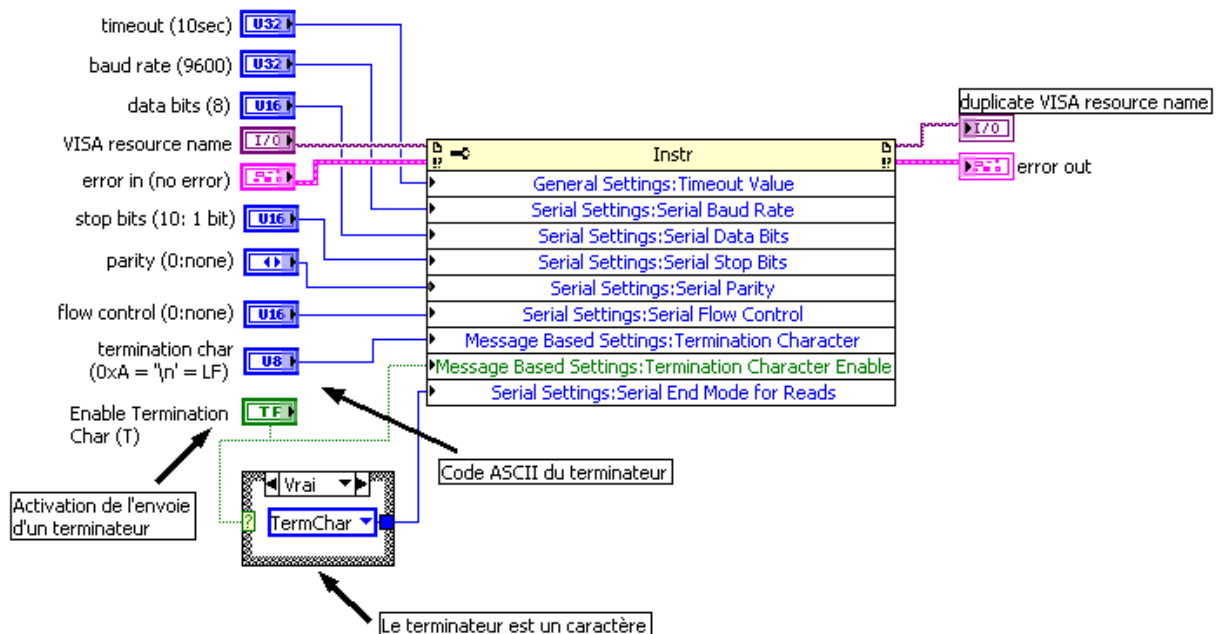
Cette palette regroupe des VIs non spécifiques (en jaune) et trois fonctions dédiées : à la configuration de la liaison, à la lecture du nombre d'octets et à l'envoi d'une pose.

L'exemple suivant reprend le premier exemple pour l'interface GPIB revue et corrigée pour la RS. Comme vous pouvez le constater, la différence n'est pas marquante hormis la configuration de la liaison. Par défaut l'envoi d'un caractère terminateur est activée.



## Nœud de propriété

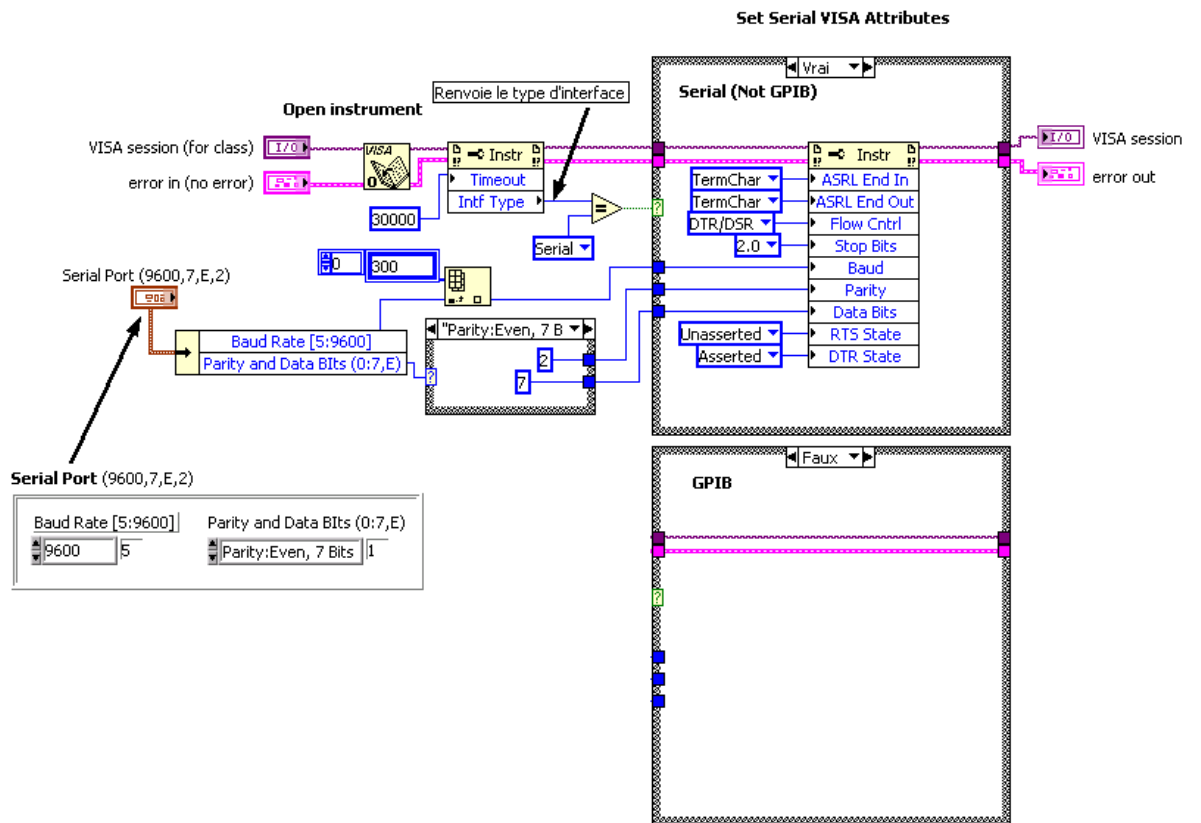
Si vous regardez comment est écrit le VI de configuration vous verrez ceci dans le diagramme :



L'élément central de ce VI est un nœud de propriété. De façon générale, un nœud définit l'ensemble des propriétés d'un objet quelconque ; son contenu dépend donc du type d'objet auquel il est connecté. Par exemple s'il s'agit d'un élément de la face avant, la propriété **Visible** doit être dans la liste. Il est possible de redimensionner la liste, de choisir les items uns à uns de les paramétrer en lecture ou en écriture.

Les nœuds de propriétés sont disponibles dans la palette **E/S d'instruments»VISA »VISA avancées** ainsi que dans la palette **Contrôle d'applications**.

L'exemple ci-dessous est extrait du pilote du HP34401. Il montre comment la spécificité du support (RS ou du GPIB) est assurée lors de l'initialisation de l'objet Visa.



En fonction du type d'interface, un paramétrage spécifique peut être introduit. Une fois ce VI exécuté, plus aucune autre modification n'est nécessaire à la prise en compte du mode de transmission (GPIB ou série) magique...