Introduction à Labview

HAD ANAS

Introduction

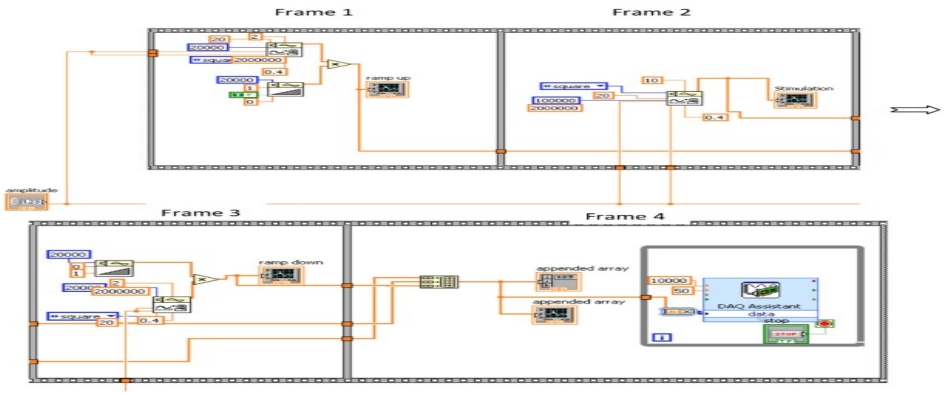
- Qu’est-ce que LabVIEW ?
  - LabVIEW :Laboratory of Virtual Instruments Engineering Workbench

- Système de programmation à usage général qui comporte des bibliothèques de fonctions pour toute tache de programmation.

- Bibliothèques dédiées à l’acquisition de données, le contrôle d’instruments, analyse, traitement et stockage de données
  - beaucoup de sous-programmes LabVIEW sont développés par les fabricants d’instruments de mesure

- Multiples versions pour applications ciblées :FPGA/Real Time / Professionnelle

- Distinction des autres logiciels : **Permet de faire de la programmation graphique** !

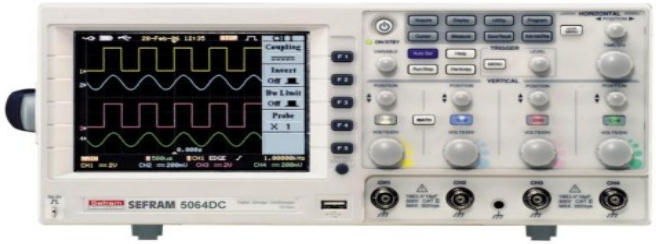


Introduction
Domaines d’application

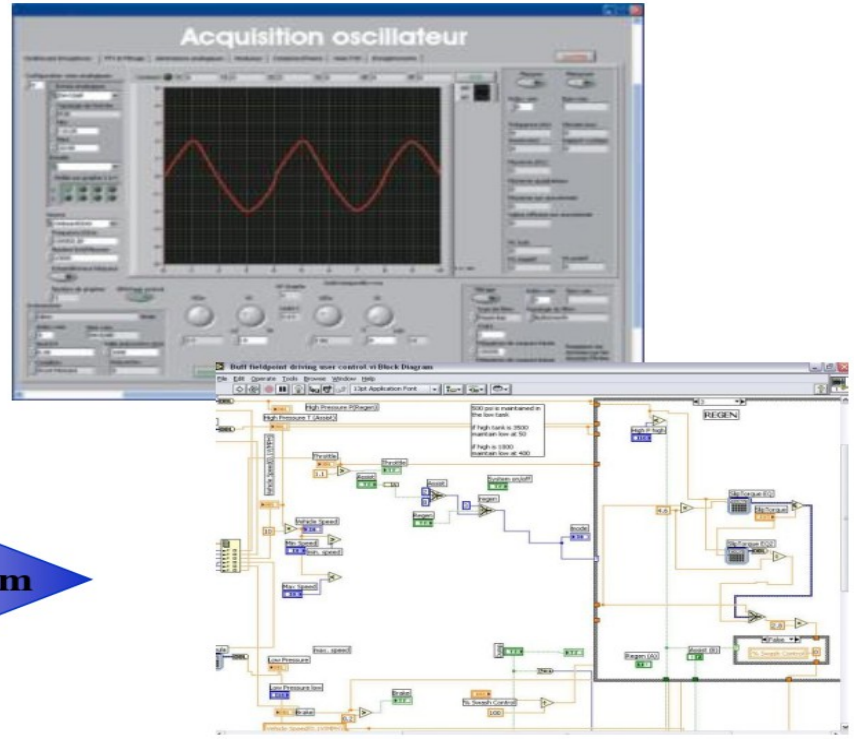
- LabVIEW permet de faire de la programmation graphique pour les mesures et l'automatisation.
- Domaines d'application traditionnels :
  - Acquisition, traitement et commande à partir d'un PC
- LabVIEW permet de développer toute une chaîne d’acquisition

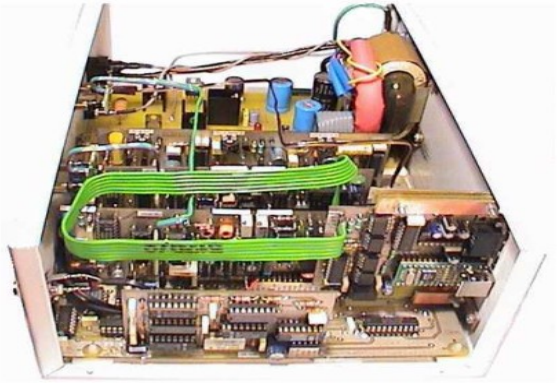
Logiciel

Présentation de l’interface



Front panel



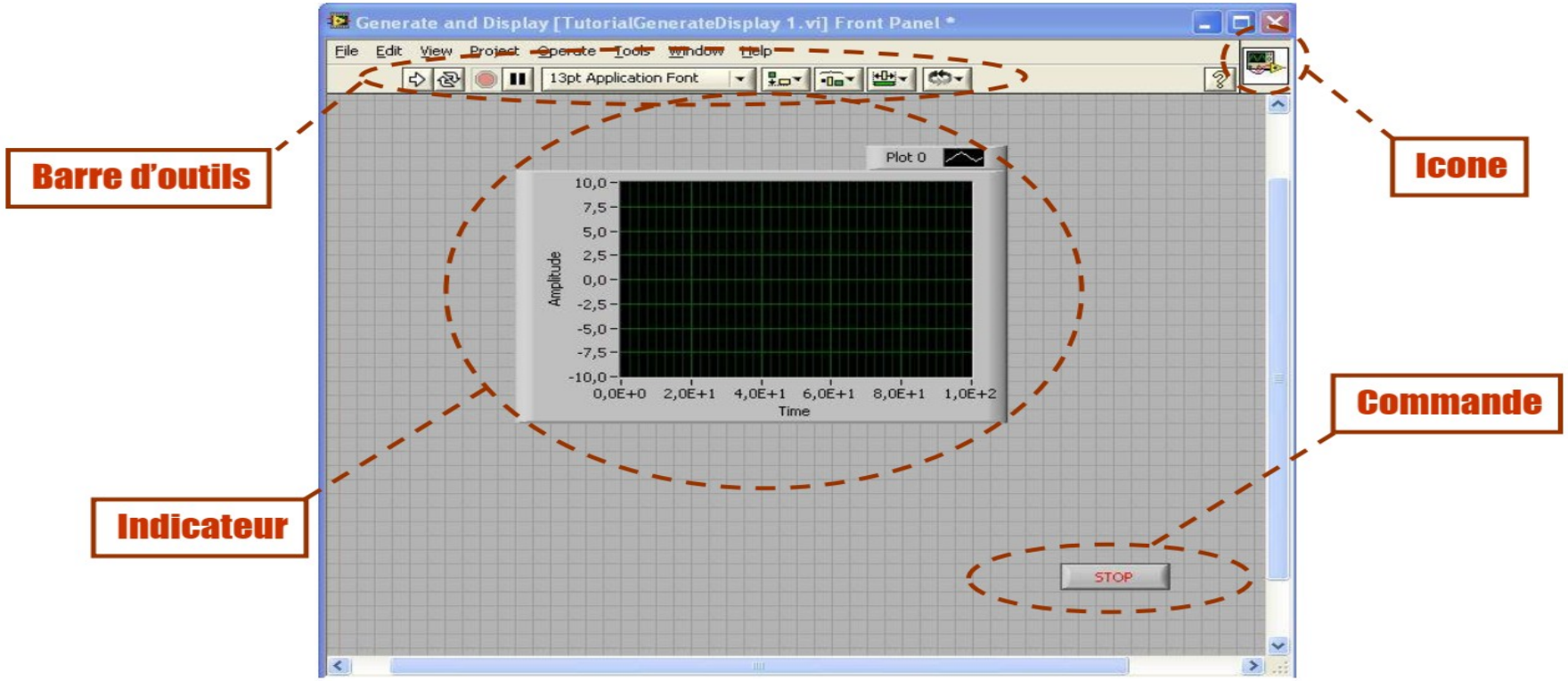


Block diagram

Logiciel

Présentation de l'interface

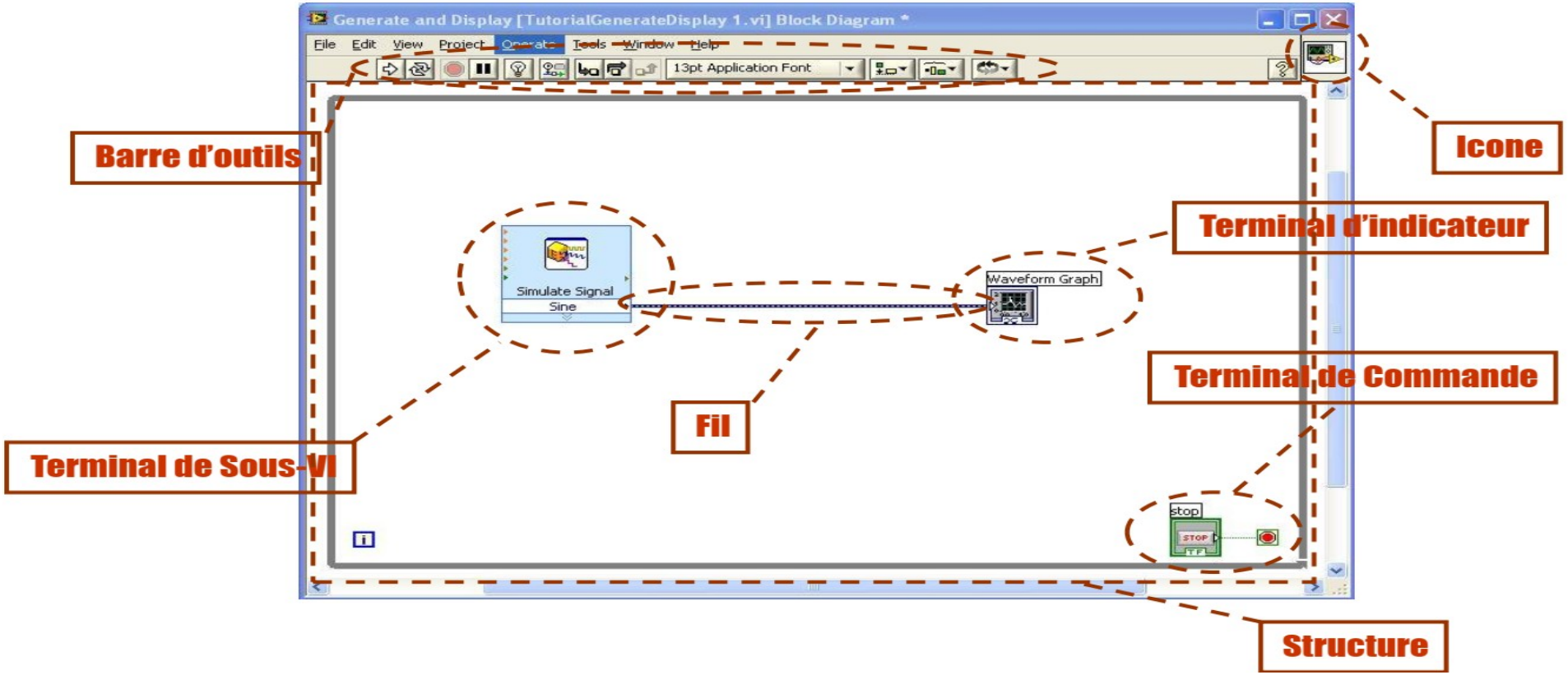
La face avant (*front panel*) ⇨ **Interface utilisateur**



Logiciel

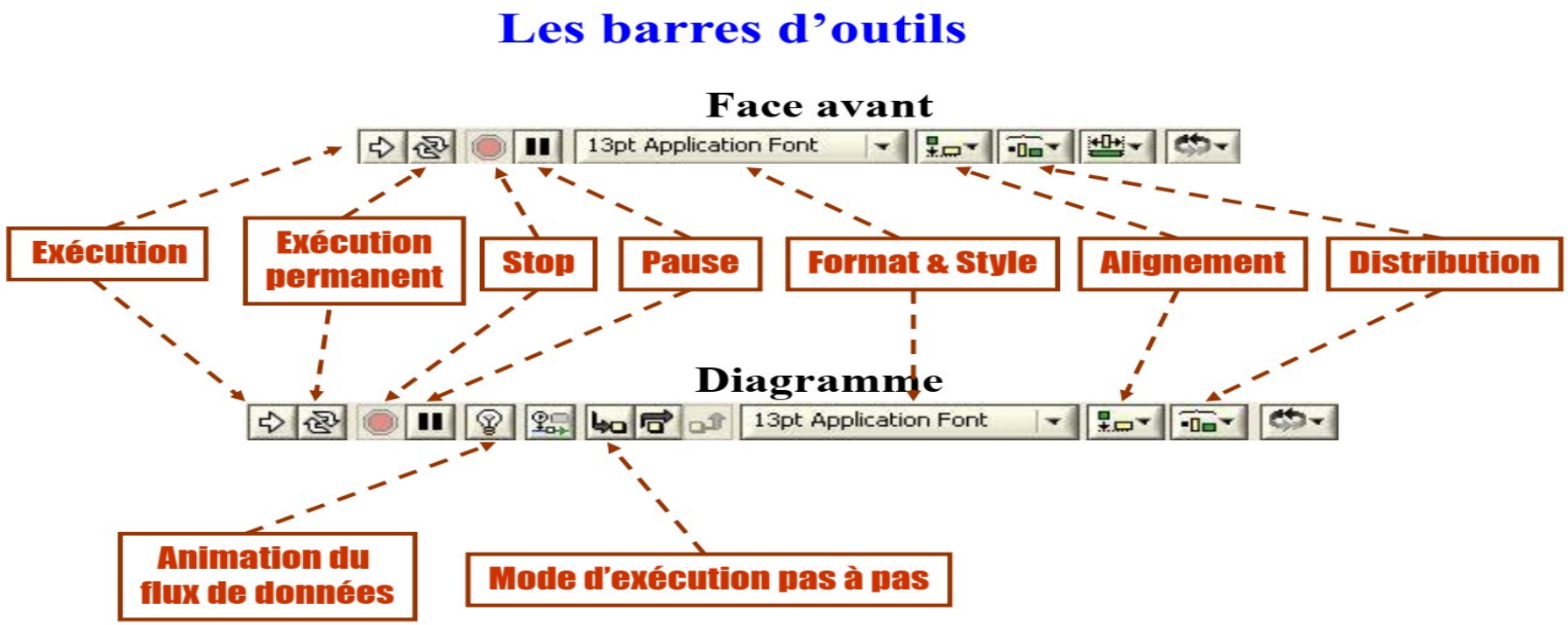
Présentation de l'interface

Le diagramme *(block diagram)* ⇨ **Interface programmeur**



Logiciel

Présentation de l'interface



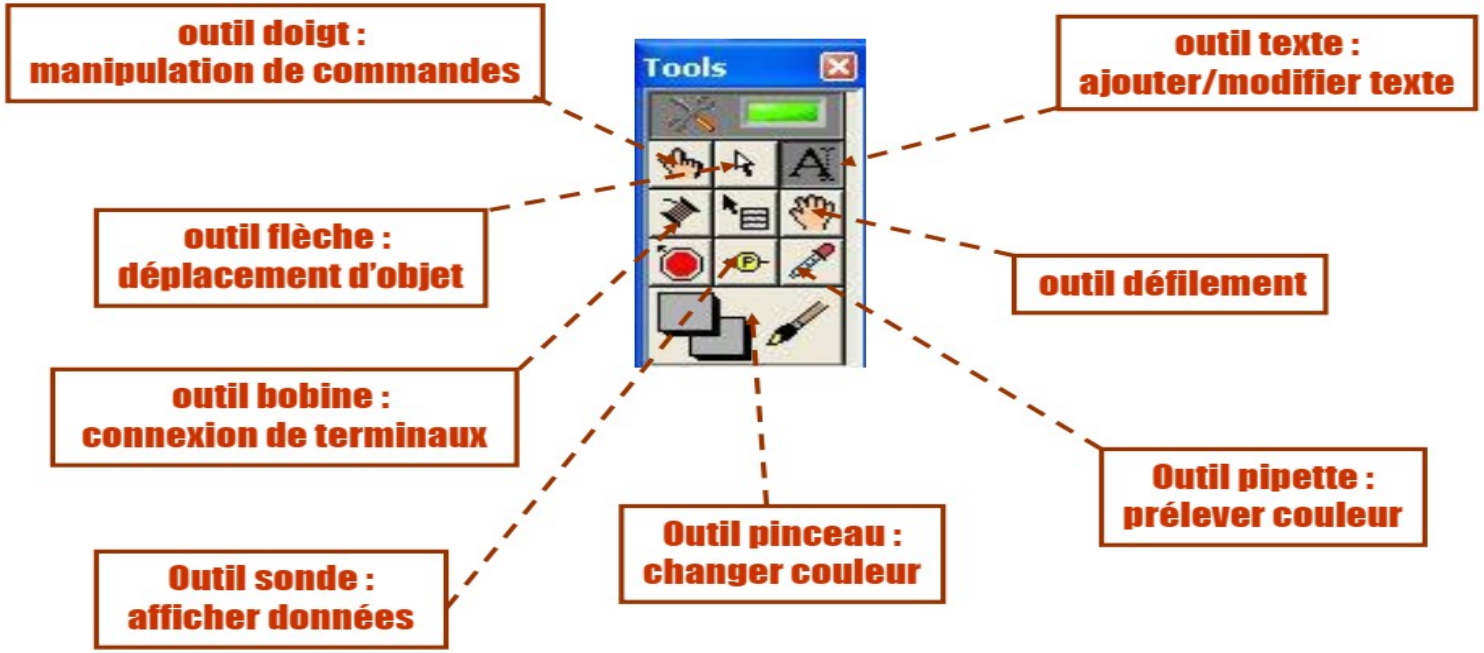
Logiciel

Présentation de l'interface

La palette Outils

⇨ commune à la face avant et au bloc diagramme !

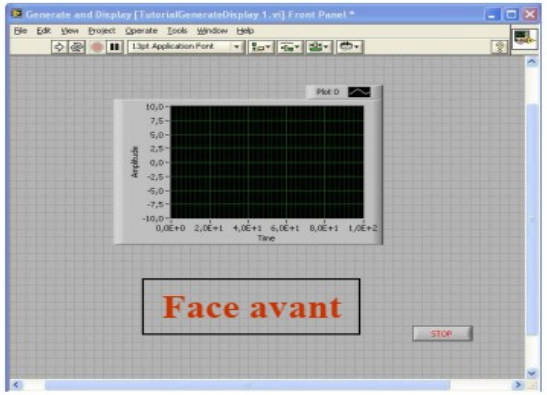
⇨ affichable à partir du menu *VIEW* de l'une ou l'autre fenêtre

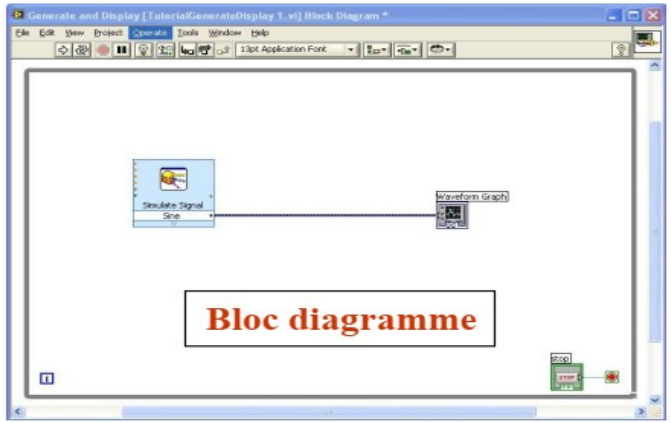


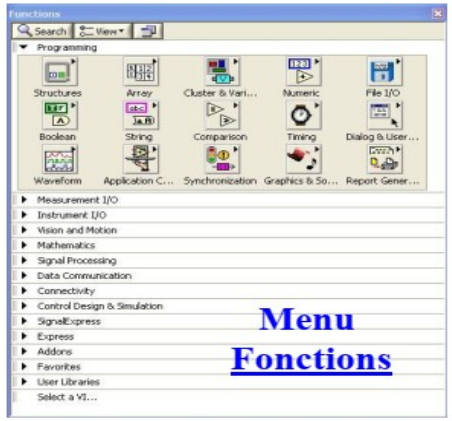
Logiciel

Menus et sous-menus

Les menus *Commandes* et *Fonctions*

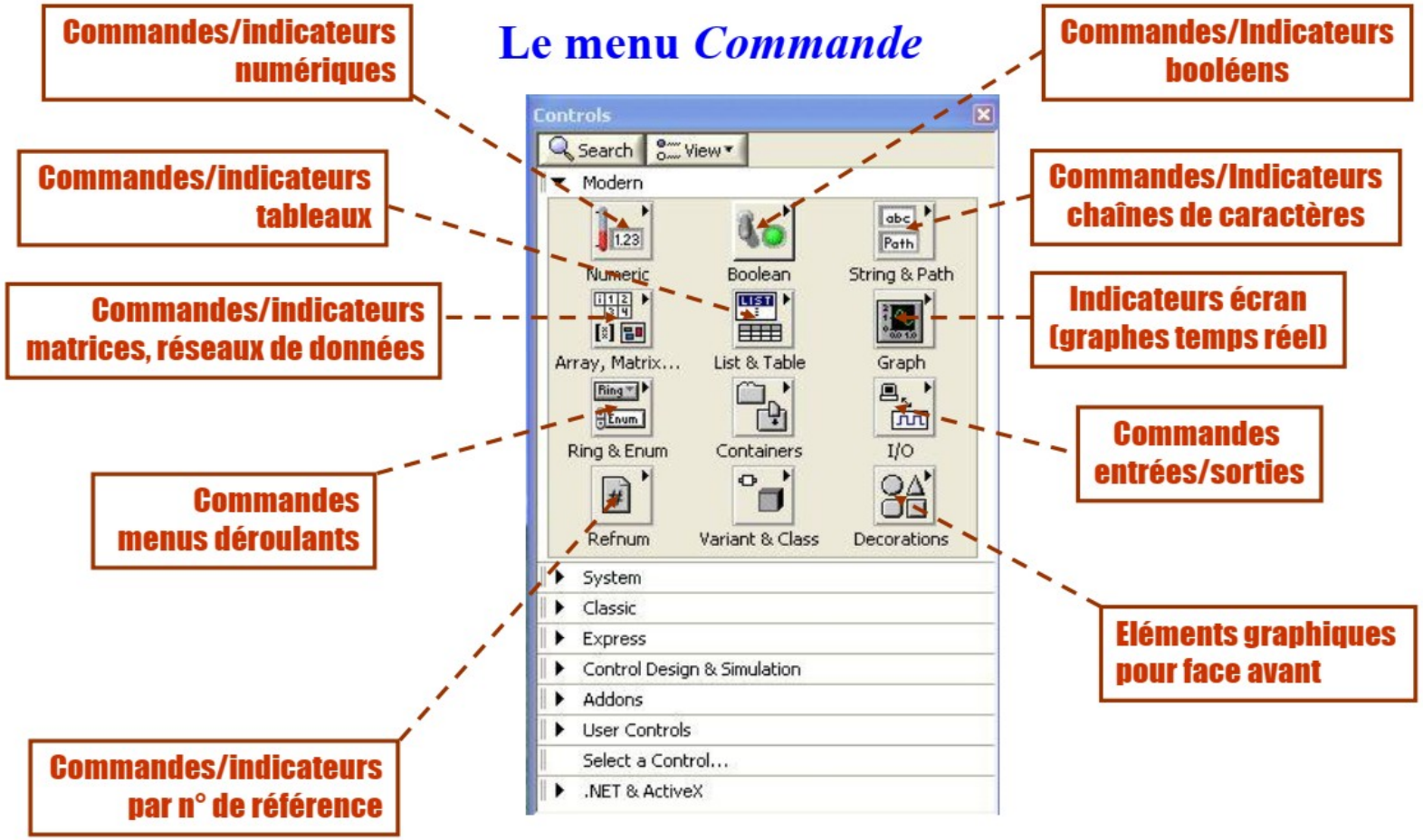






Logiciel

Menus et sous-menus

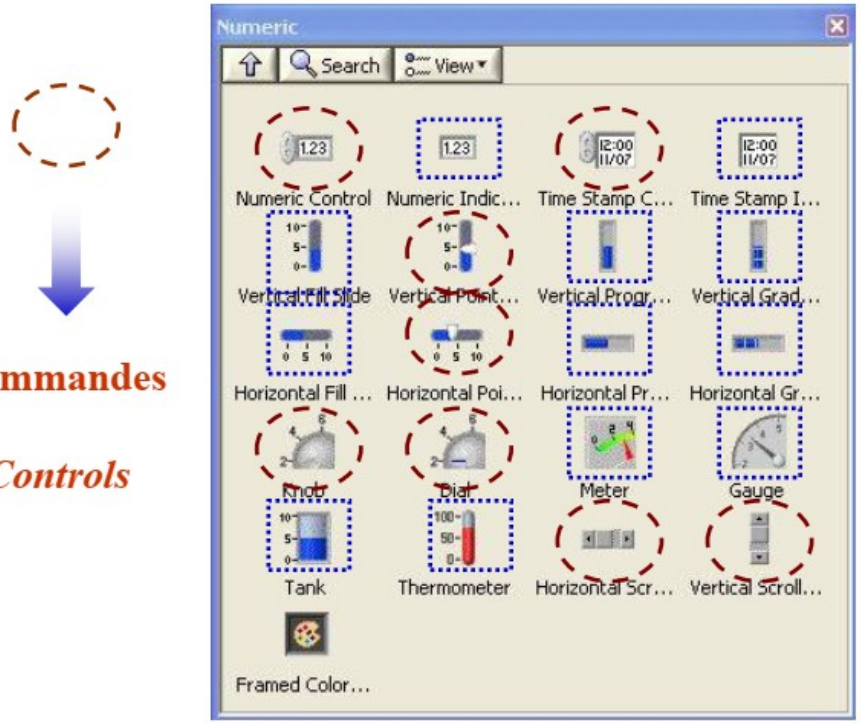


Logiciel

Menus et sous-menus

Exemple de sous-menu du menu *Commande…*

⇨ Bibliothèque Numérique …



Commandes

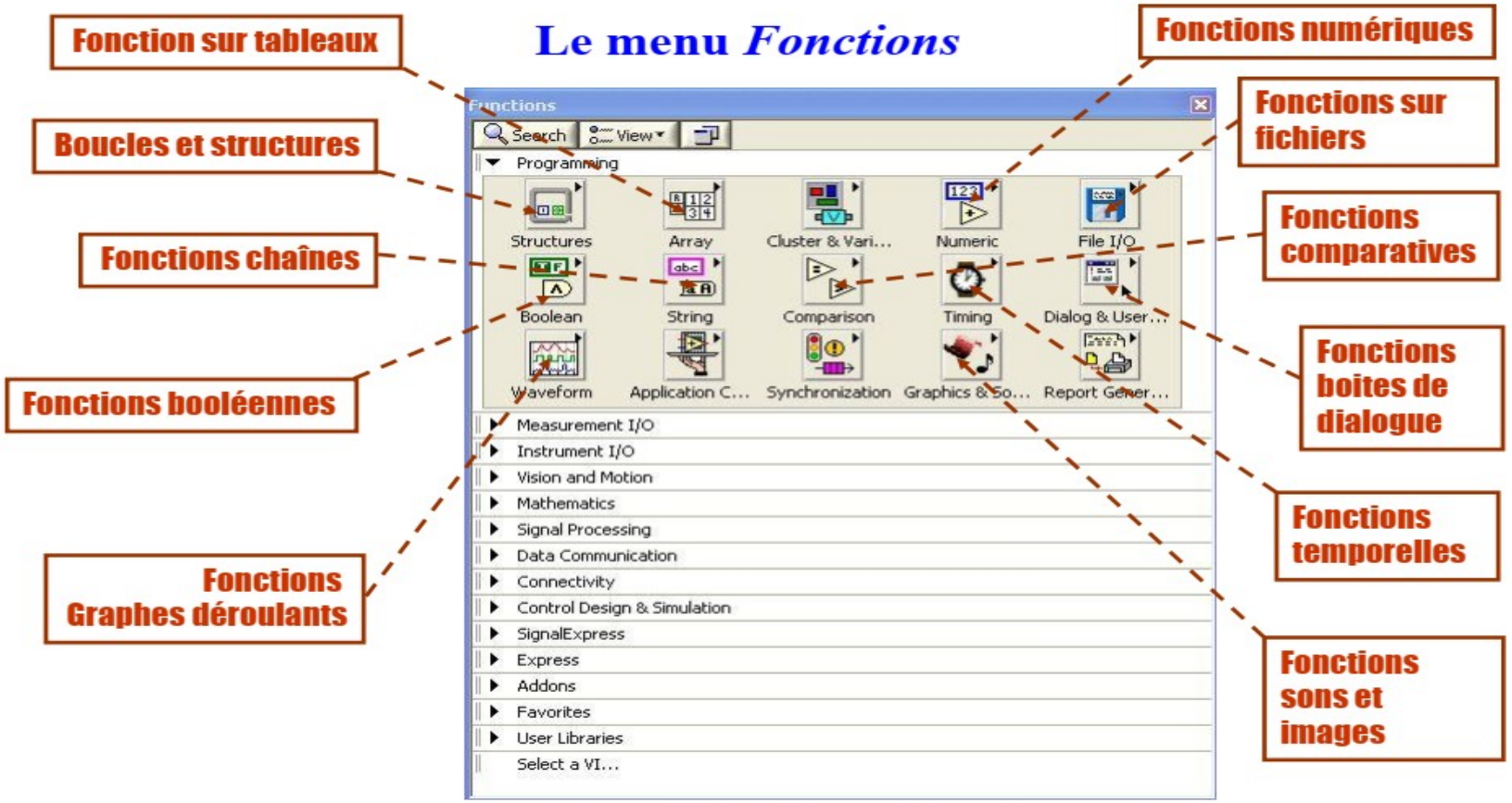
Controls

Indicateurs

Indicators

Logiciel

Menus et sous-menus

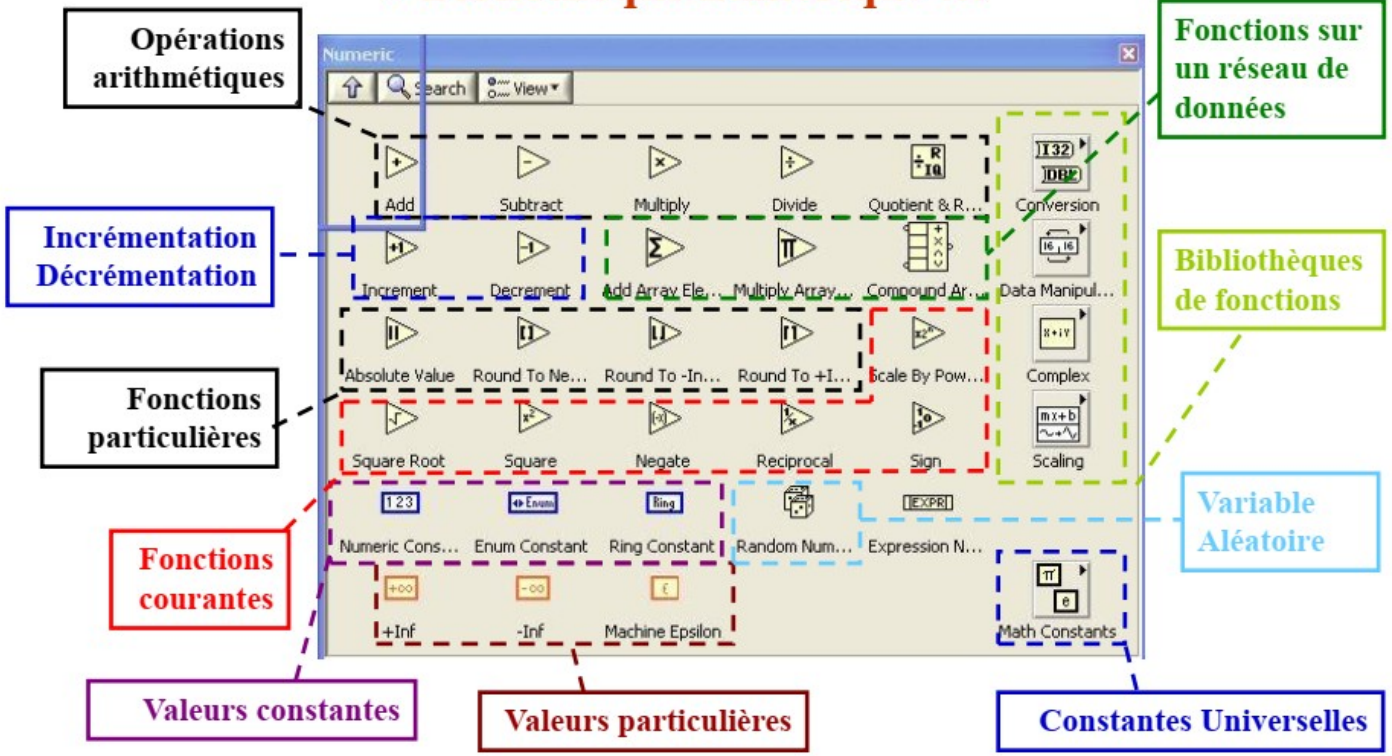


Logiciel

Menus et sous-menus

Exemple de sous-menu du menu *Fonctions…*

⇨ Bibliothèque numérique …

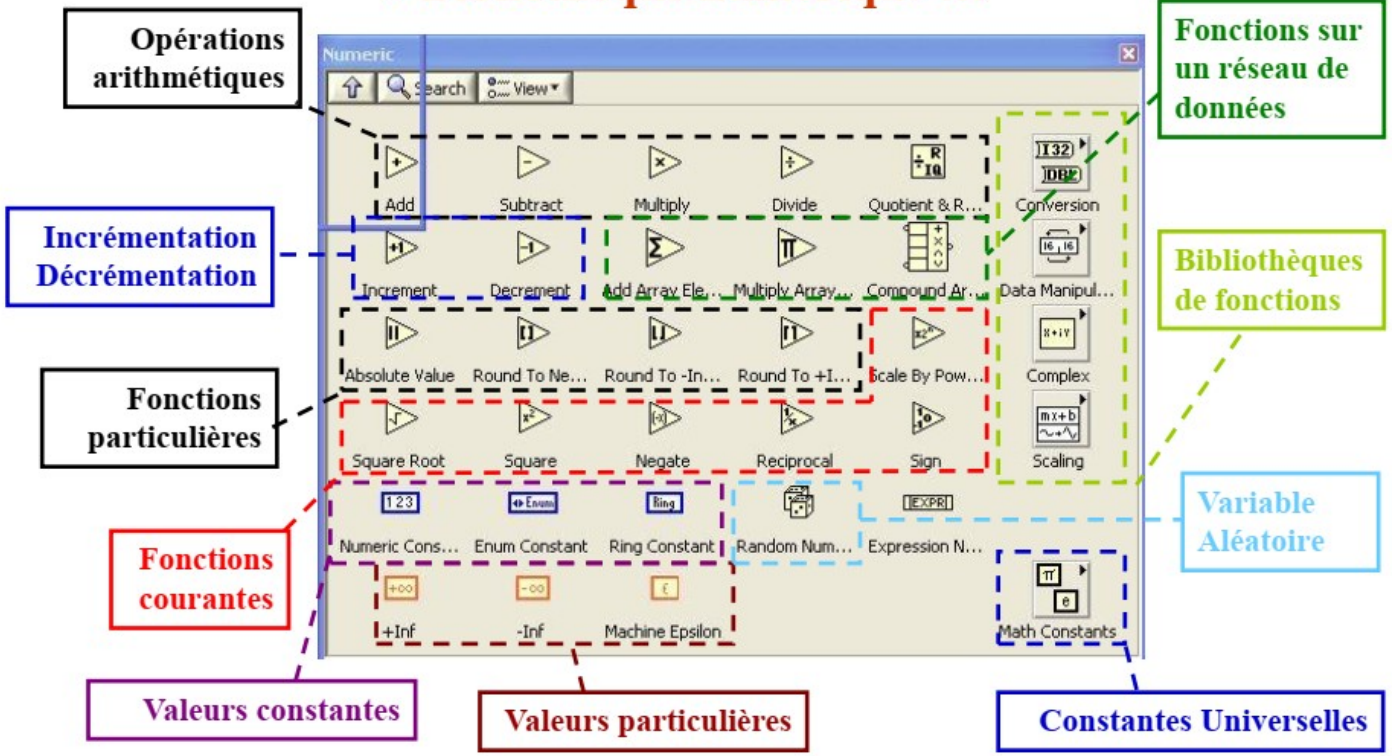


Logiciel

Menus et sous-menus

Exemple de sous-menu du menu *Fonctions…*

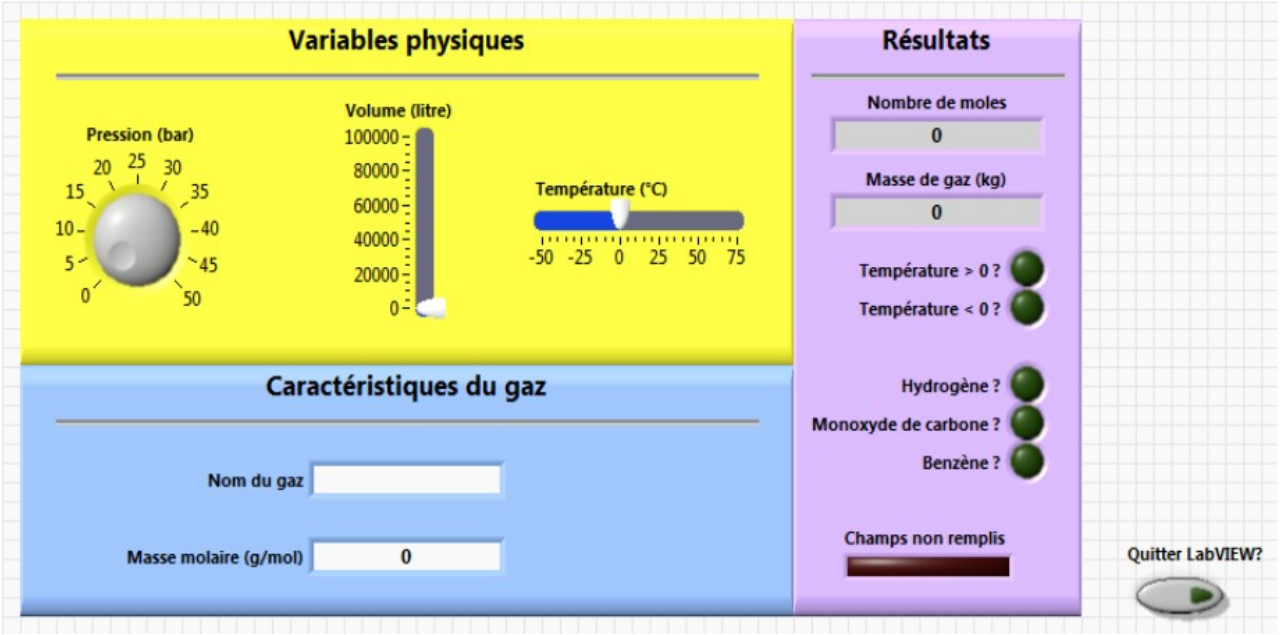
⇨ Bibliothèque numérique …

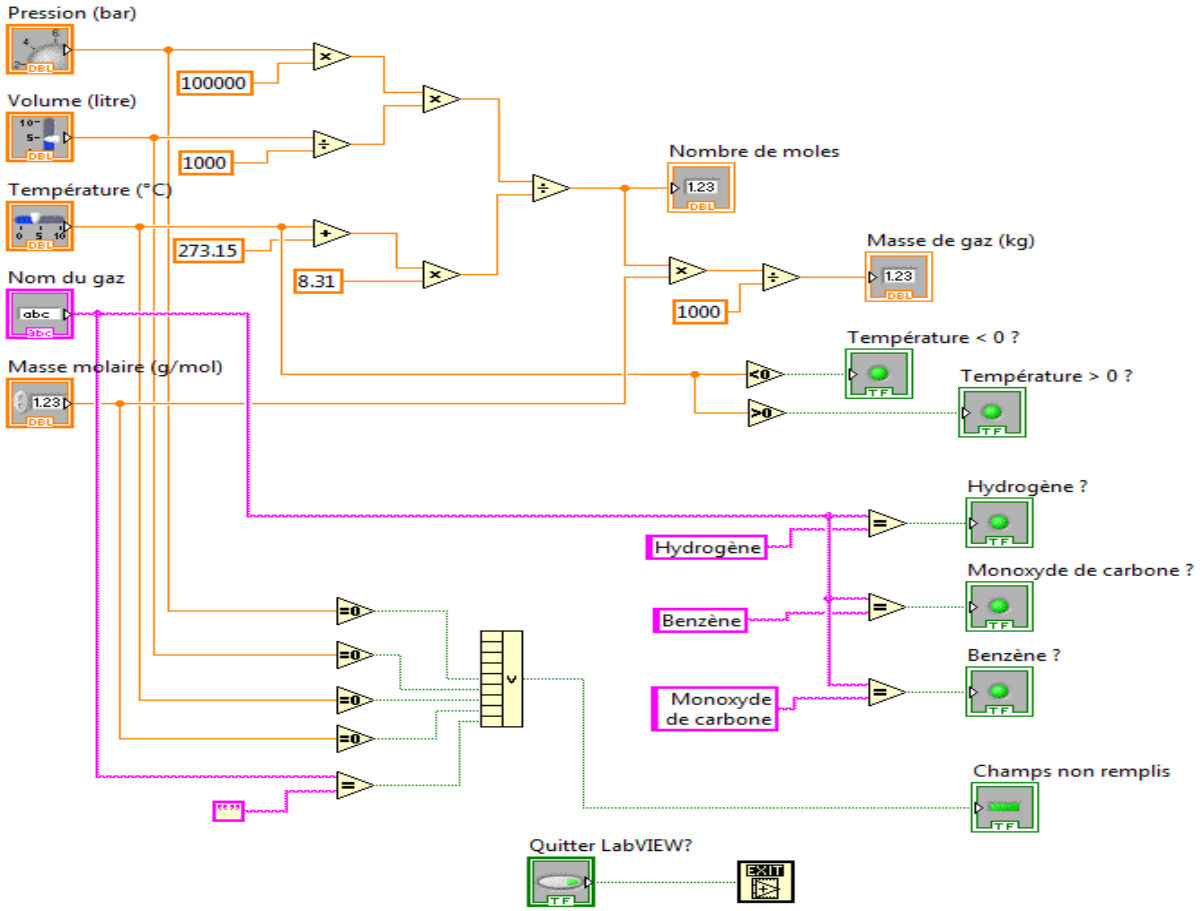


Logiciel

Menus et sous-menus

- A l'aide des sous-menus du menu Commande/ menu Fonctions, reproduire le diagramme et le graphe représentés ci-joint :





Logiciel
Les variables d'entrée

- Servent à l'acheminement des données entre terminaux vis les nœuds du V.I.
- Circulation unidirectionnelle : Commande Indicateur
- Épaisseur du fil en fonction de la nature des données (scalaire, tableau 1D, tableau 2D)

Type de variable	Représentation	Code couleur
Booléenne		Vert
Chaîne de caractère		Violet
Nombre réel		Orange
Nombre entier		Bleu

Logiciel
Technique de développement

Flèche brisée dans la barre d'outils = exécution impossible !

Recherche d'erreur ⇨ double-clic sur la flèche brisée !

Mise au point d'un diagramme ⇨ mode pas à pas !

Exécute le diagramme nœud par nœud !

Visualisation séquentielle du déroulement du diagramme ⇨ mode animation !

Permet de suivre à vitesse lente le flux de données au sein du diagramme !

Mise en place d'indicateur locaux ⇨ mode sonde !

Permet de visualiser des données dans un fil au fur et à mesure de leurs acheminement !

Logiciel
Notions de VI et de sous-VI

- Qu'est-ce qu'un VI ?
  - V.I. : Virtual Instrument
- V.I. = programme LabVIEW qui comprend :
  - l'interface utilisateur interactive (face avant)
  - le diagramme de flux de données (code source)
- V.I. présente une structure hiérarchique et modulaire
- 1 V.I. = 1 tache d'exécution !

- Qu'est-ce qu'un sous-VI ?
  - V.I. contenu dans un autre V.I. = sous-V.I.
- Sous-V.I.s = V.I.s de plus bas niveaux ou sous-programme
- V.I. principal = ensemble de sous-V.I. représentant une tâche spécifique
- Chaque sous-V.I. peut fonctionner indépendamment du reste de l'application
- Sous-V.I. = V.I. de bas niveaux, que l'on peut enregistrer dans des bibliothèques spécifiques

Programmation
Boucles et structures

Boucles ⇨ permettent l'exécution d'un programme, d'un sous-programme ou d'une partie de programme jusqu'à une action ou une valeur définie par l'opérateur.

Structures ⇨ contrôlent le flux de données dans un V.I.

Structures ⇨ permettent d'organiser, de séquencer ou de conditionner des éléments d'un V.I. ou un V.I. complet

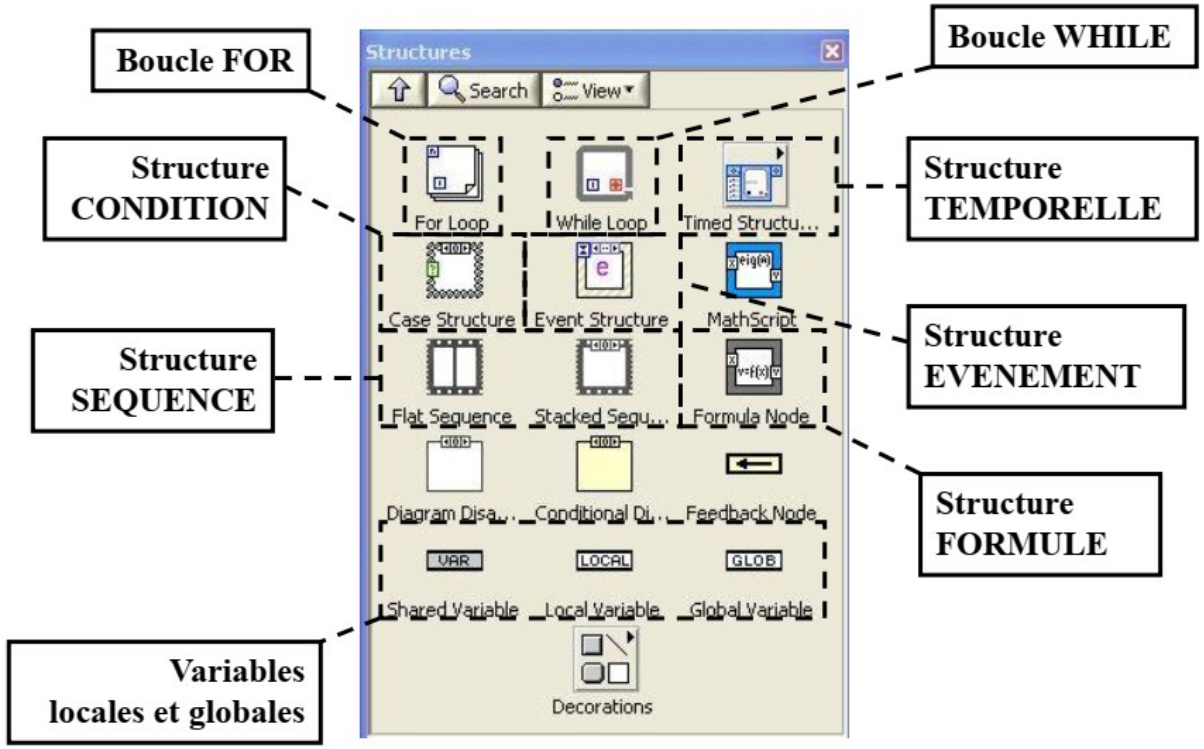
Boucles ⇨ identique à celles mises en jeu en programmation en langage C ou en langage BASIC

Structures
⇨ conditions = if, then, else…
⇨ temporelle = déroulement du programme (pas de flux de données)…

Programmation

Boucles et structures

⇨ **Bibliothèque boucles et structures …**

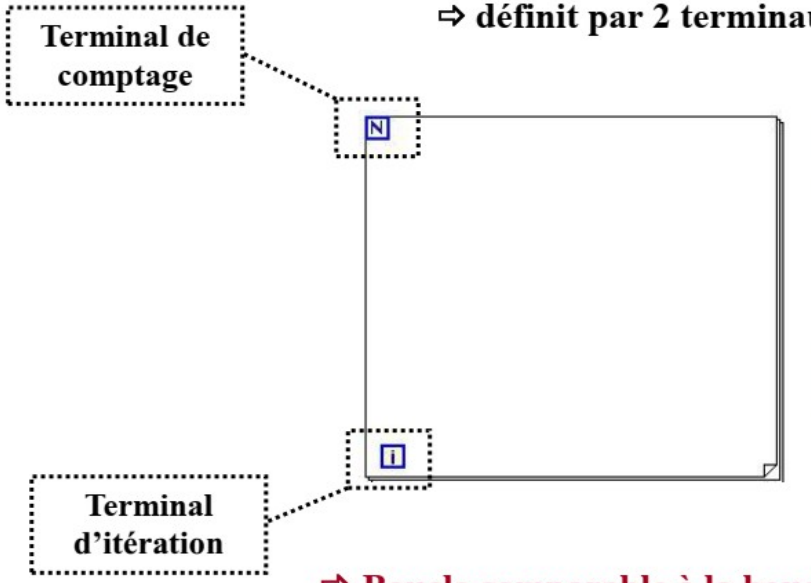


Programmation
Boucles et structures

Boucle FOR

⇨ Répète une partie du code diagramme un nombre déterminé de fois, ce nombre étant définissable par l'opérateur

⇨ définit par 2 terminaux



N — Entrée numérique

↳ nbre total d'éxécution
⇨ Définit par l'utilisateur

i — Sortie numérique

↳ nbre d'itération effectuée

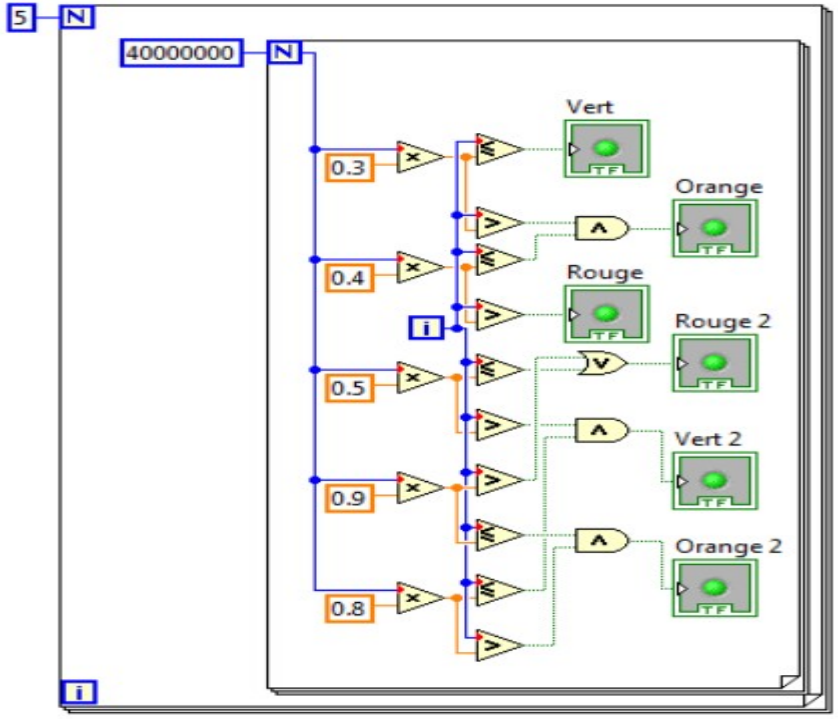
⇨ Boucle comparable à la boucle FOR en programmation en langage C ou BASIC

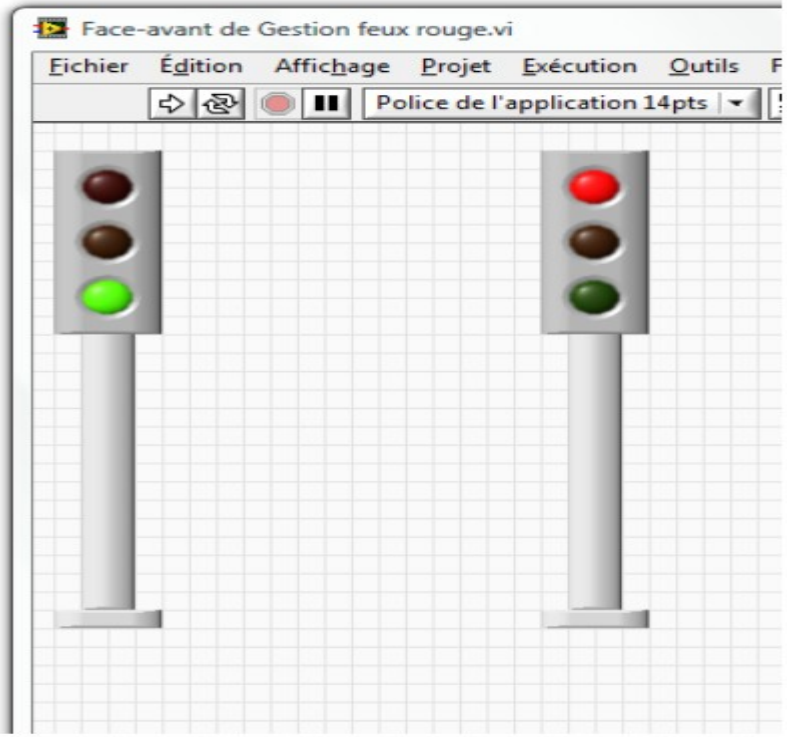
Programmation

Boucles et structures

Application d'une boucle FOR:

Reproduire le VI ci-dessous, identifier les fonctions et observer les résultats



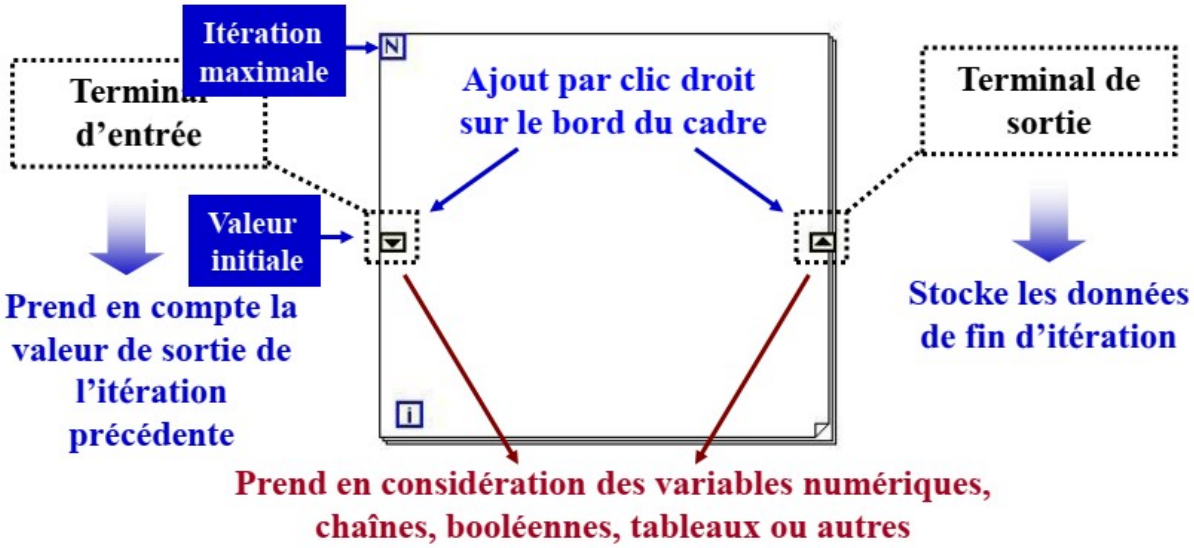


Programmation
Boucles et structures

Structure Registre à décalage

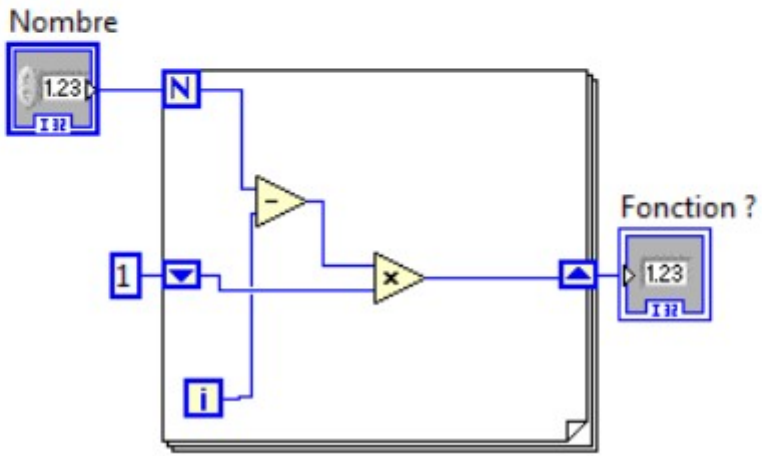
⇨ Variables locales qui transfèrent des données entre deux itérations consécutives d'une même boucle

⇨ 2 terminaux placés au bord droit et gauche de la boucle



Application d'un registre à décalage

Quelle est la fonction réalisée par un VI dont le diagramme est représenté ci-dessous?



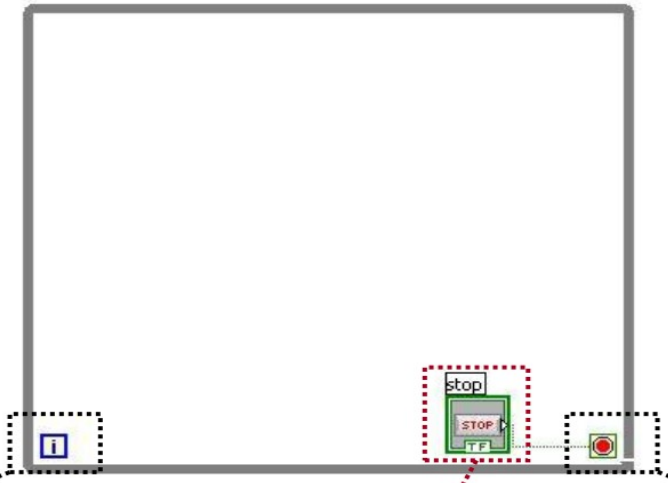
Programmation
Boucles et structures

Boucle WHILE

⇨ Répète le code diagramme contenu à l'intérieur de la boucle jusqu'à un changement d'état de la variable booléenne associé au terminal conditionnel

Teste la valeur du terminal conditionnel à chaque itération

Au moins une exécution !



⇨ Boucle comparable à la boucle DO…WHILE en programmation en langage C ou BASIC

Terminal d'itération

Terminal de commande booléenne

Terminal conditionnel

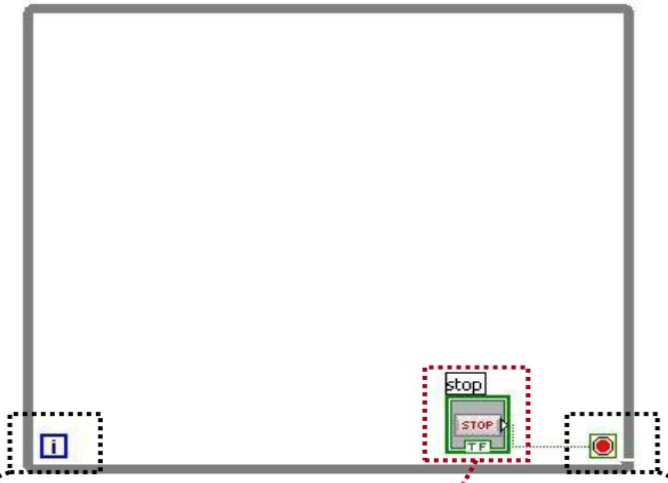
Programmation
Boucles et structures

Boucle WHILE

⇨ Répète le code diagramme contenu à l'intérieur de la boucle jusqu'à un changement d'état de la variable booléenne associé au terminal conditionnel

Teste la valeur du terminal conditionnel à chaque itération

Au moins une exécution !



⇨ Boucle comparable à la boucle DO…WHILE en programmation en langage C ou BASIC

Terminal d'itération

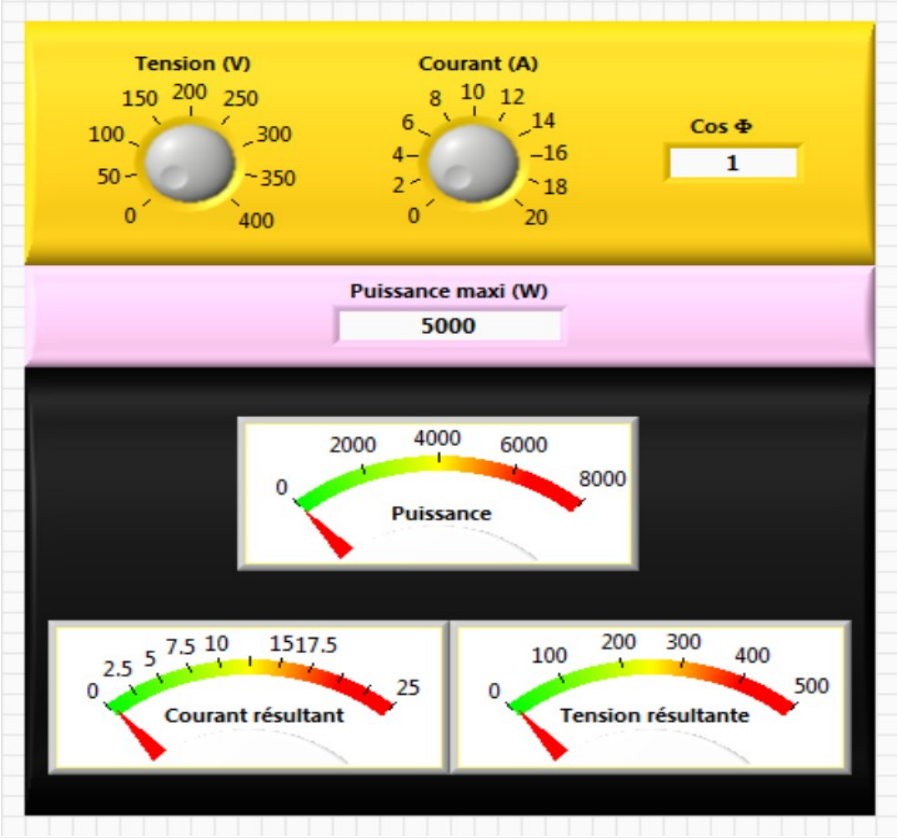
Terminal de commande booléenne

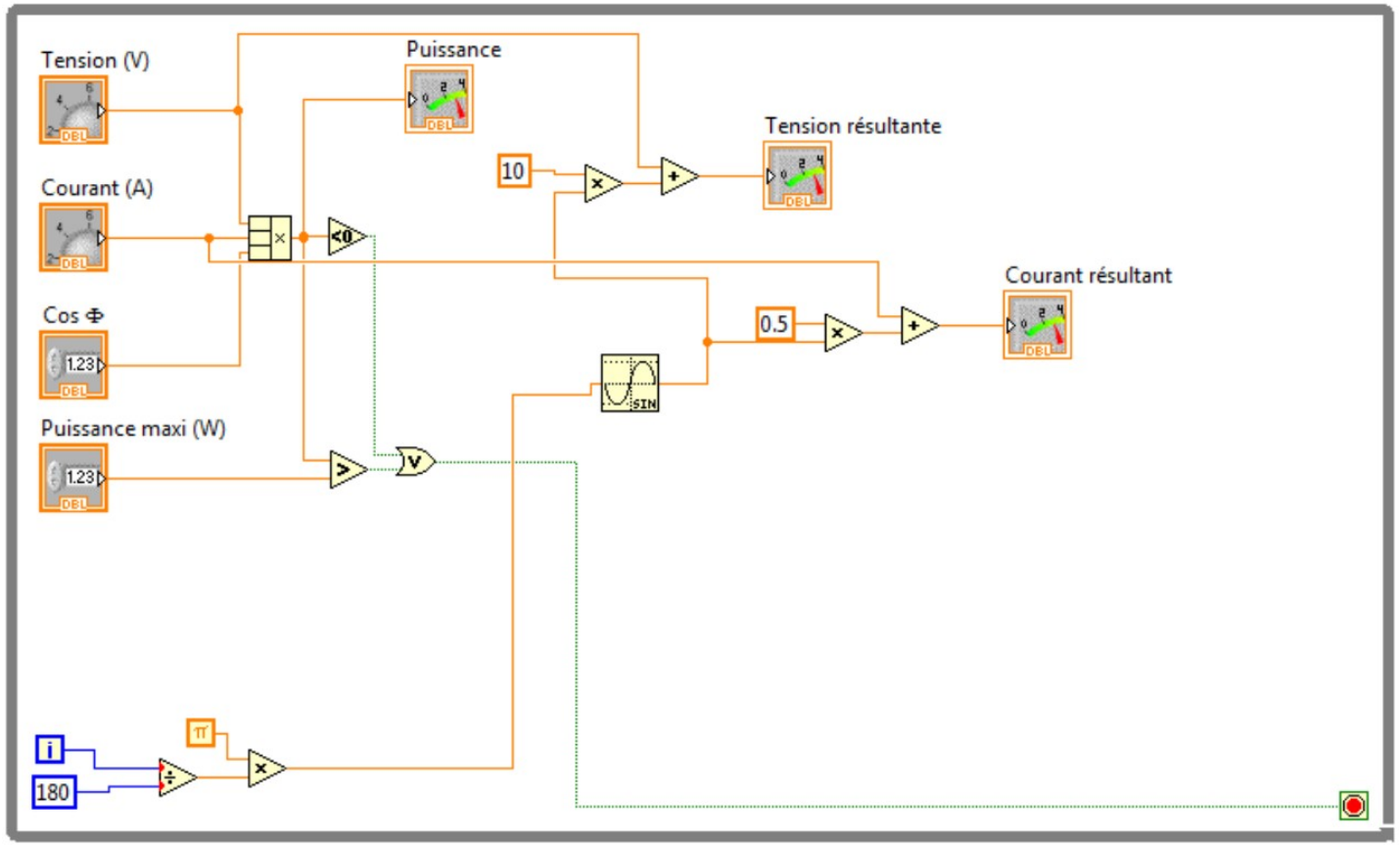
Terminal conditionnel

Programmation

Boucles et structures

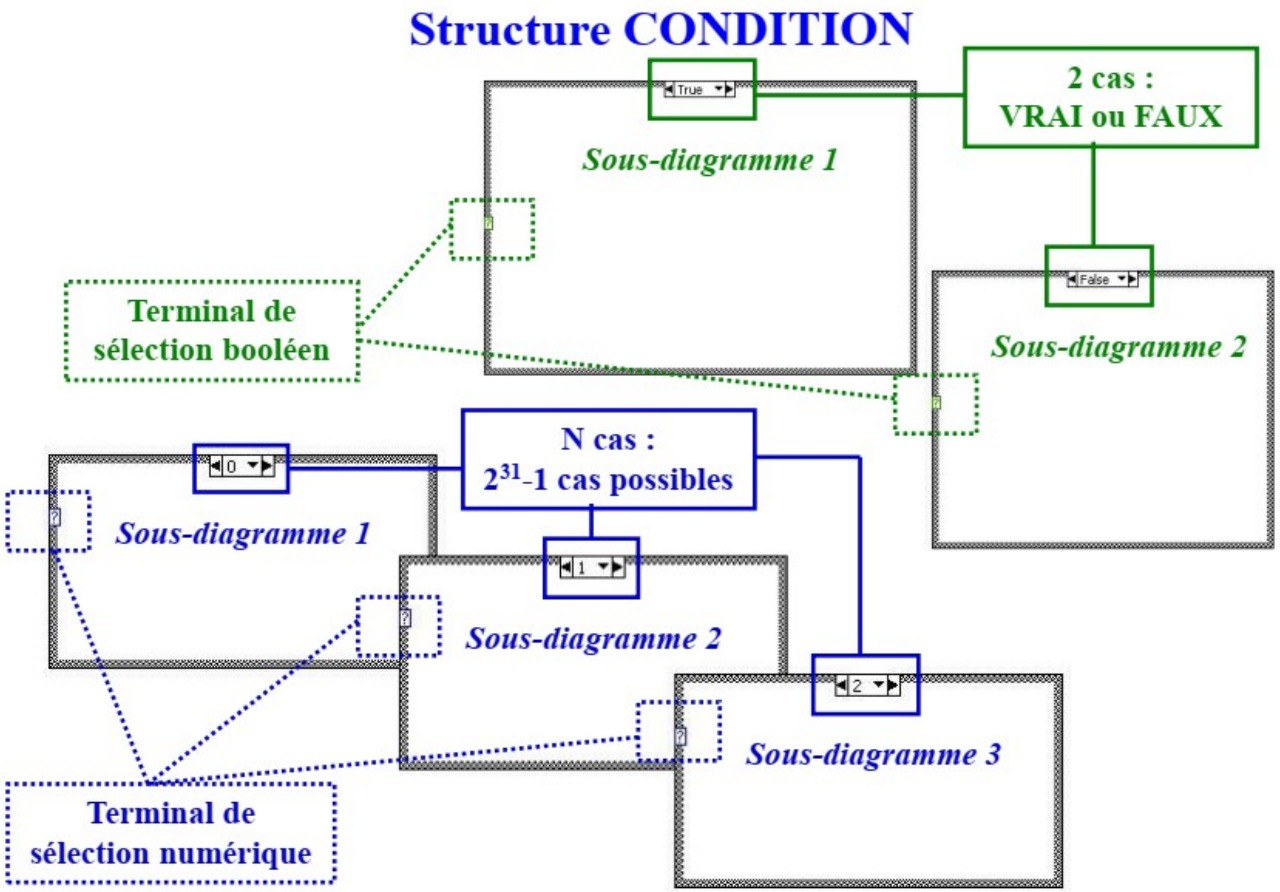
➢ Application d'une boucle While :





Programmation

Boucles et structures



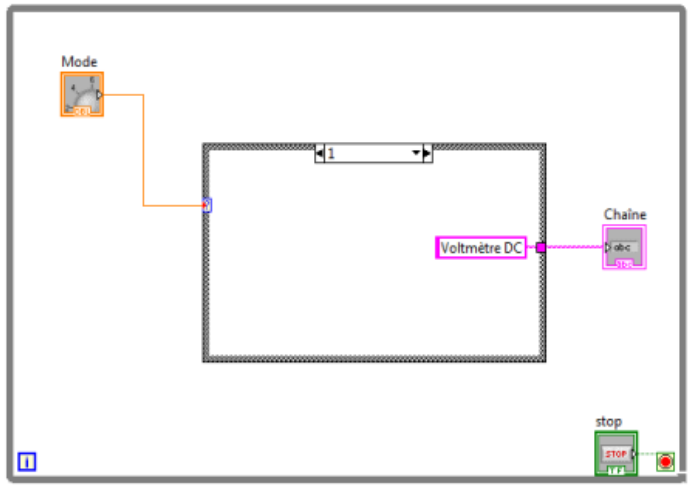
Programmation

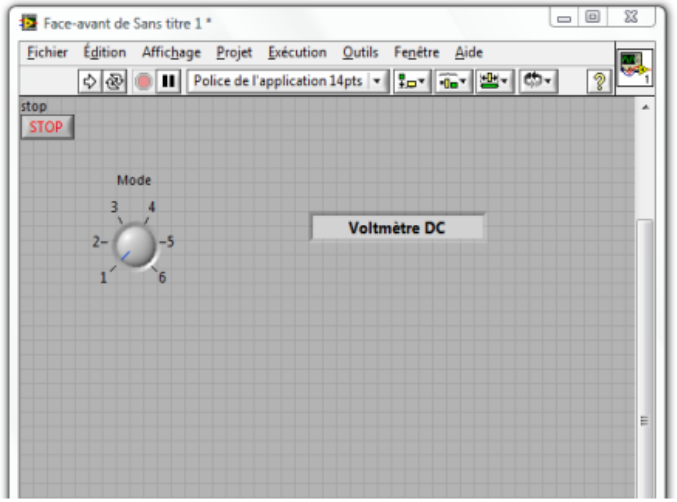
Boucles et structures

Application d'une structure condition à sélection numérique:

Reproduire le VI ci-dessous avec les différents mode suivants:

Choix	0, par défaut	1	2	3	4	5	6
Mode	Mode inconnu	Voltmètre DC	Voltmètre AC	Ampèremètre DC	Ampèremètre AC	Ohmètre	Wattmètre





Programmation
Graphes

⇨ **représentation bidimensionnelle d'un ou plusieurs tableaux de données appelés *tracés***

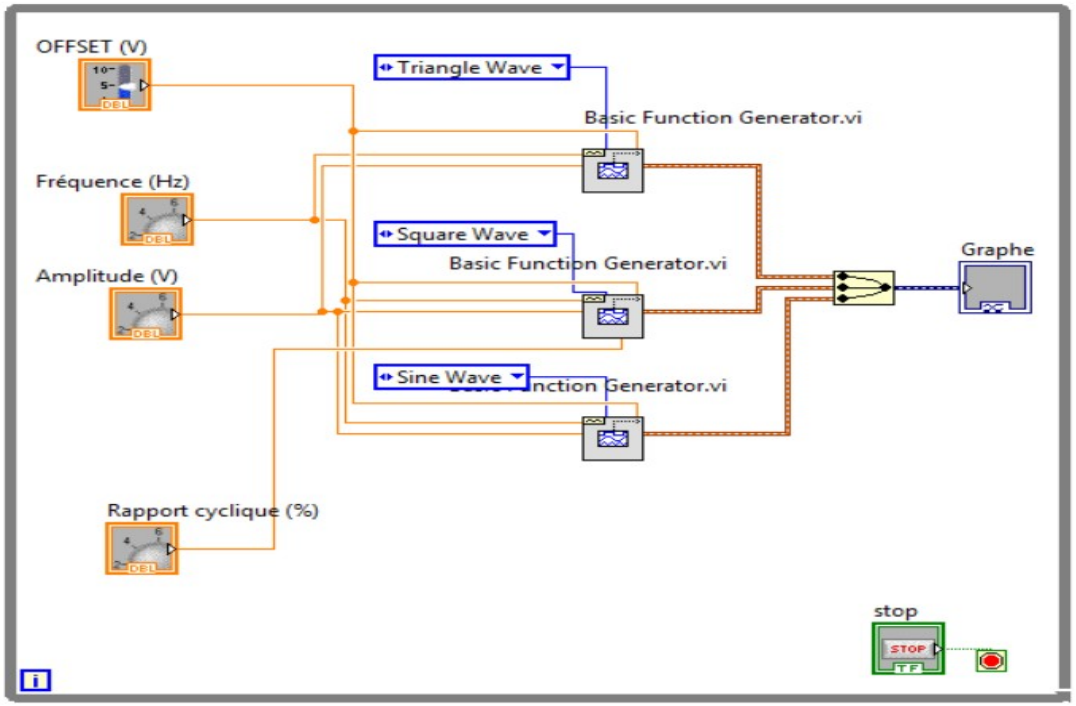
⇨ **2 types de graphes:**
Les graphes XY
Les graphes simples

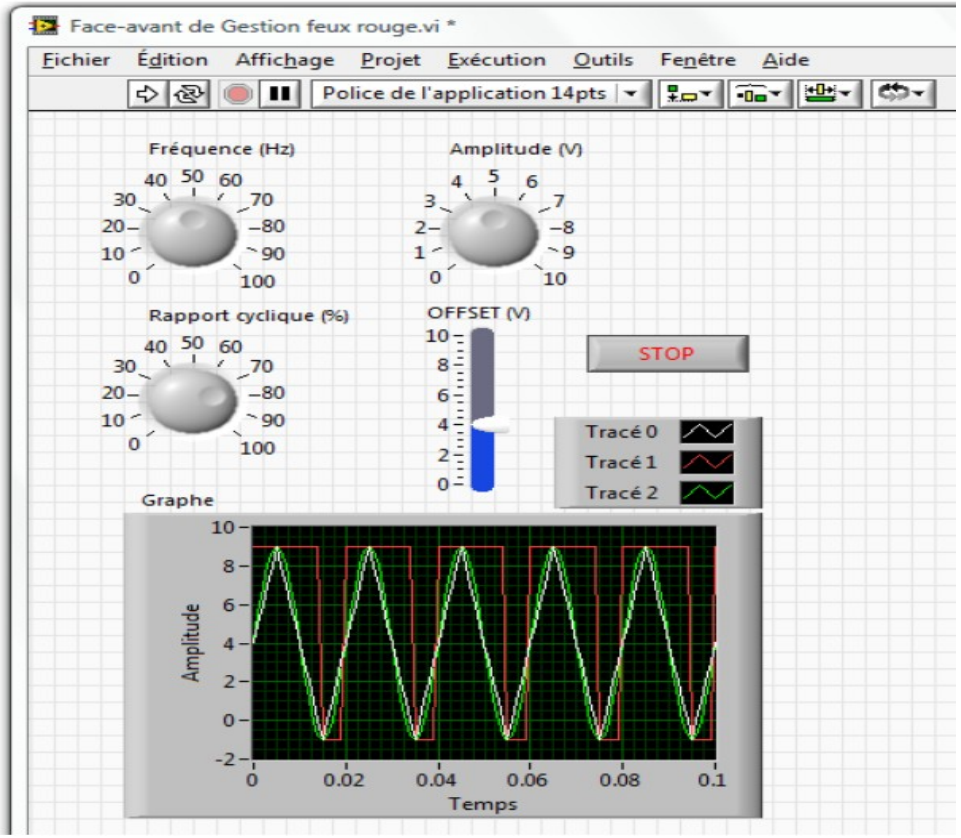
Graphes XY : indicateurs graphiques permettant de tracer des points répartis à intervalles réguliers
Ex : tracé de tableaux de données régulièrement espacées

Graphes simples : objet graphique d'usage général, idéal pour tracer des fonctions multivariables
Ex : tracé de forme circulaire ou courbes variant avec le temps

Programmation
Graphes

- Reproduire le VI ci-dessous et interpréter les résultats :





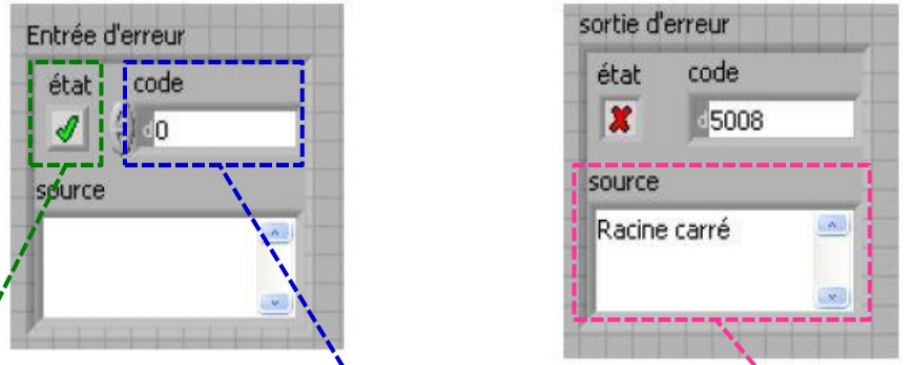
Programmation

Gestion des erreurs

Cluster d'erreur ⇨ permet la visualisation, la gestion et la transmission des erreurs entre terminaux au sein des V.I.s

Exemple :

Indicateur face avant



Pas d'erreur

Erreur

Information d'état
Variable type booléen

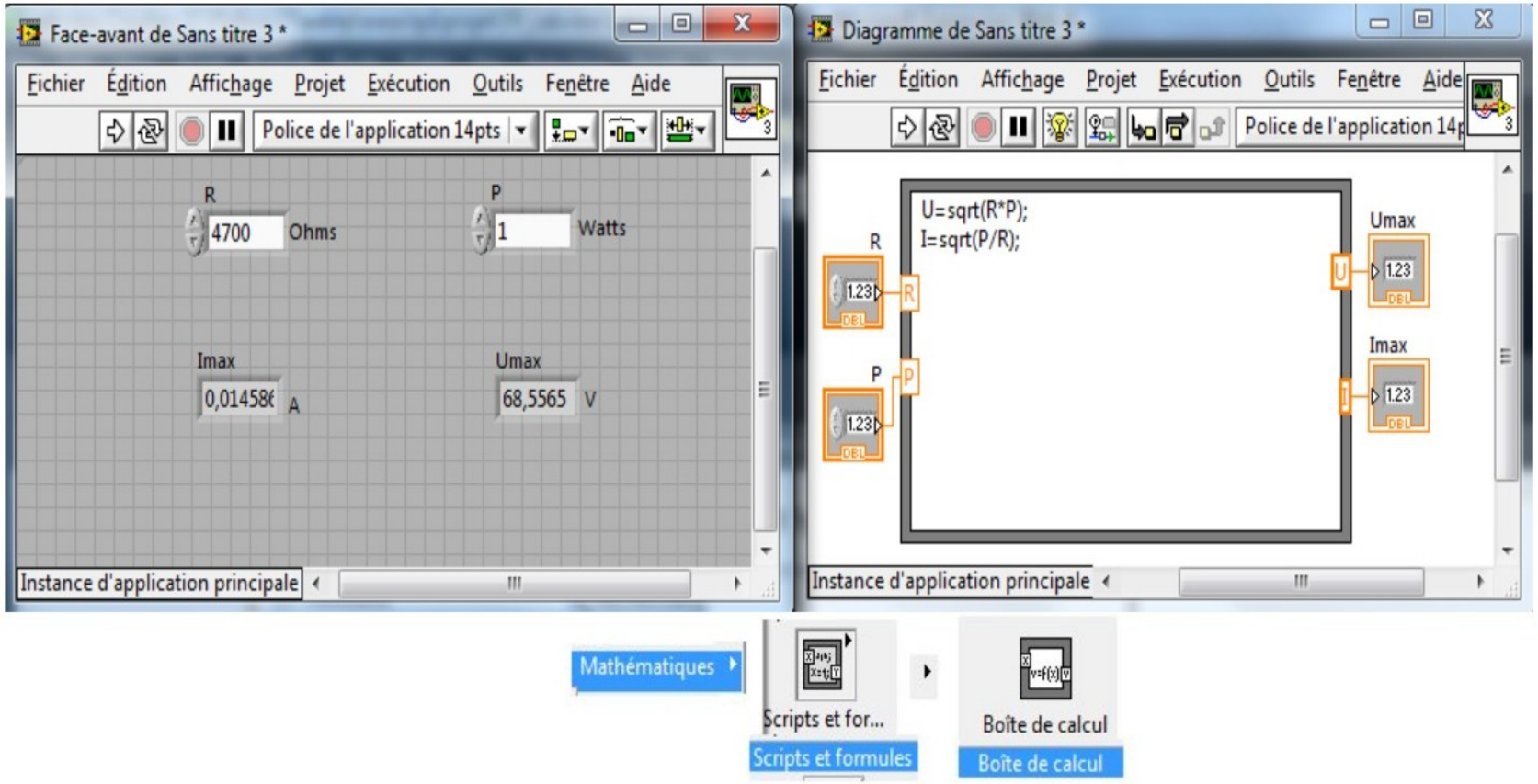
Code erreur
Variable type entier

Nature de l'erreur
Variable type chaîne

Programmation

Boite de calcul

- Si vous êtes familier avec la programmation sur Arduino, vous pouvez programmer des code dans la boîte de calcul.
- Exemple :Recopier le programme ci-joint:



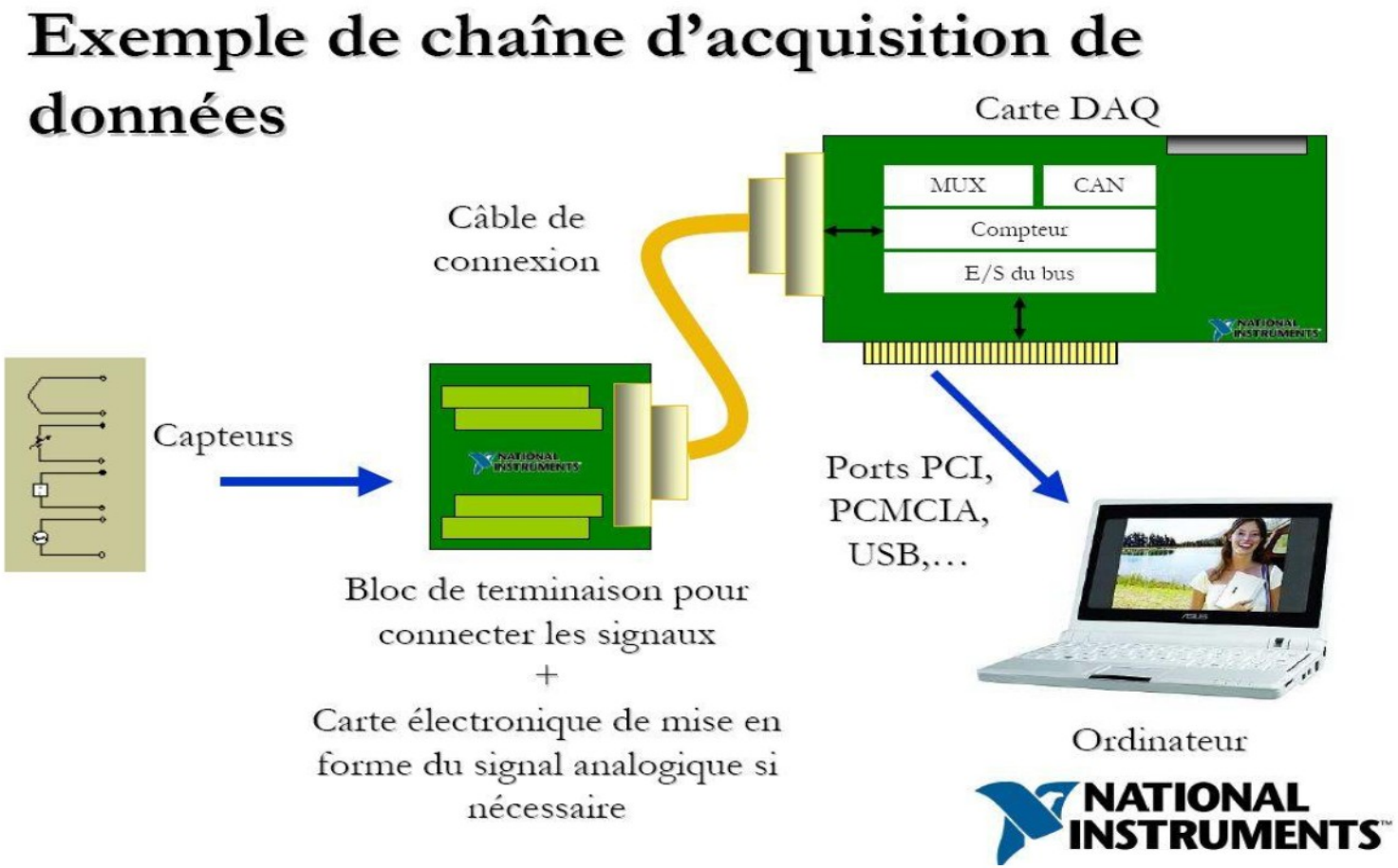
Programmation
Exercice

- Cahier des charges : Trouver les racines du polynôme du 2° degré
  - Construire une face avant permettant la saisie de et
  - A l'aide d'une boîte de calcul calculer le déterminant
  - Calculer et afficher les racines réelles si elles existent

```
DÉBUT
  LIRE a,b,c
   RÉPÉTER
   JUSQU'À Bouton Calculer
   SI a=0 ALORS
      ÉCRIRE 1er degré
      x=-c/b
      ÉCRIRE x=-c/b
   SINON
      delta=b²-4*a*c
      SI delta<0 ALORS
         ÉCRIRE Racines Complexes
         x1=(-b+i*racine(-delta))/(2*a)
         x2=(-b-i*racine(-delta))/(2*a)
      SINON
         ÉCRIRE Racines Réelles
         x1=(-b+racine(delta))/(2*a)
         x2=(-b-racine(delta))/(2*a)
      FINSI
      ÉCRIRE x1,x2
   FINSI
FIN
```

Acquisition de données

Chaîne d'acquisition de données



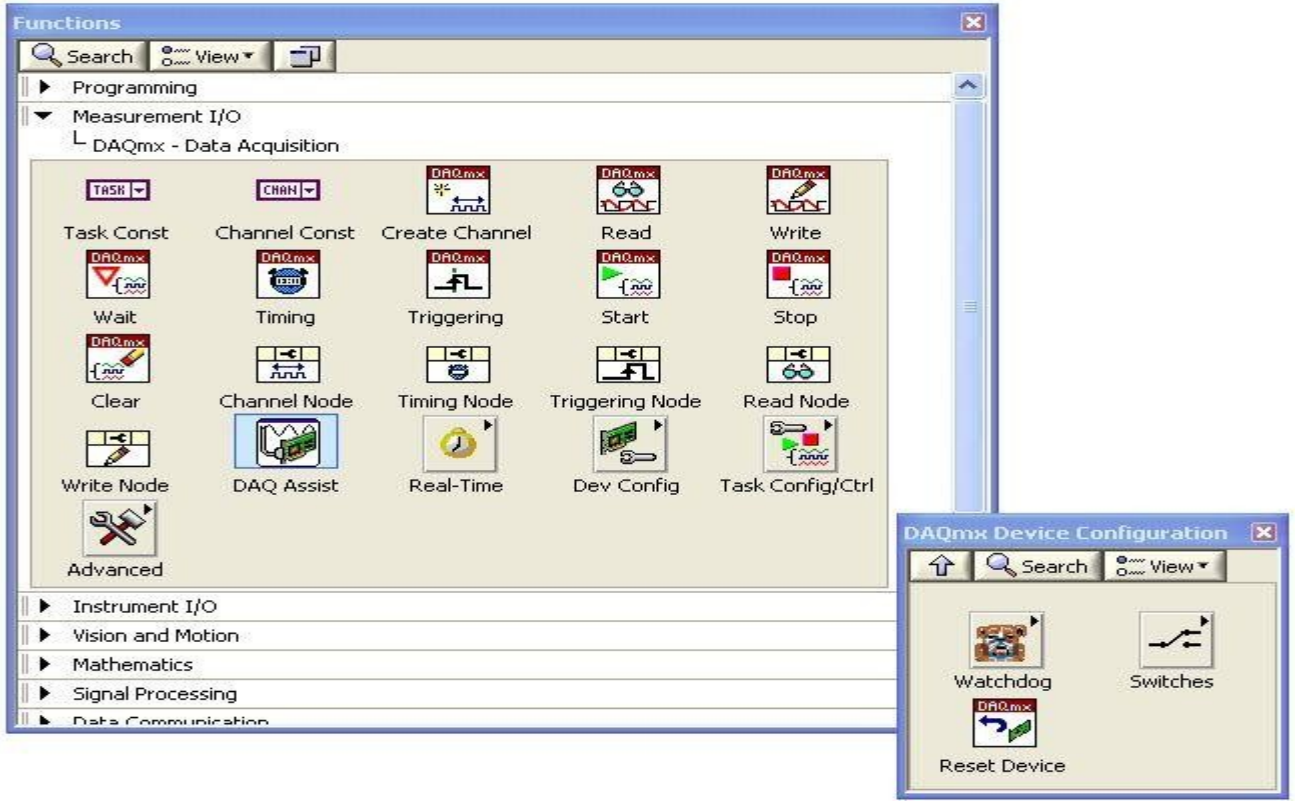
Acquisition de données

Chaîne d'acquisition de données

- Qu'est-ce que les NI-DAQmx ?
  - Ensemble de V.I.s permettant la gestion de signaux analogiques et numériques ainsi que leur gestion et leur traitement par l'environnement LabVIEW.
  - Permettent de traiter plusieurs portions de code dans des threads différents (parties du microprocesseur) et ainsi d'optimiser l'utilisation du processeur.
- 2 familles de V.I.s DAQmx:
  - V.I.s icône, polymorphes
  - V.I.s DAQ Express
- Intérêts :
  - programme de moindre densité
  - codage multithread donc vitesses d'exécution différentes compatibles

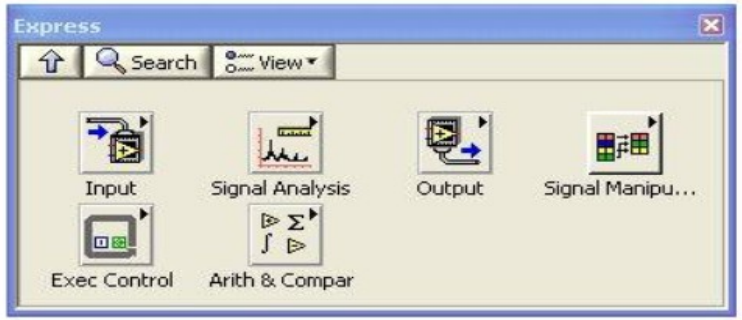
Acquisition de données
Menu NI-DAQmx

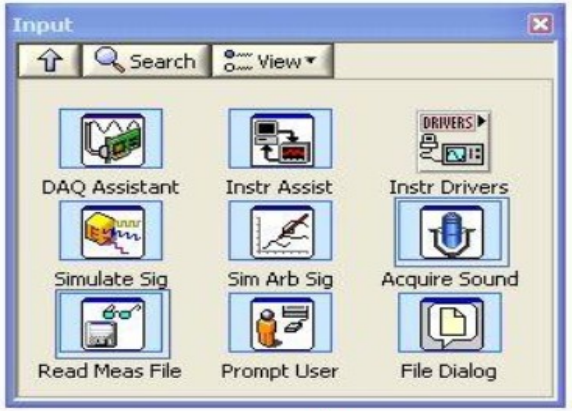
- Bibliothèques de fonctions de traitement et de configuration de signaux acquis par des cartes périphériques développée



Acquisition de données
Menu NI-DAQmx

- Les V.I.s DAQ express : sous-VIs permettant de gérer, de configurer et de traiter les entrées/sorties





Menu traitement des entrées

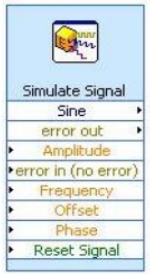
Menu traitement des sorties

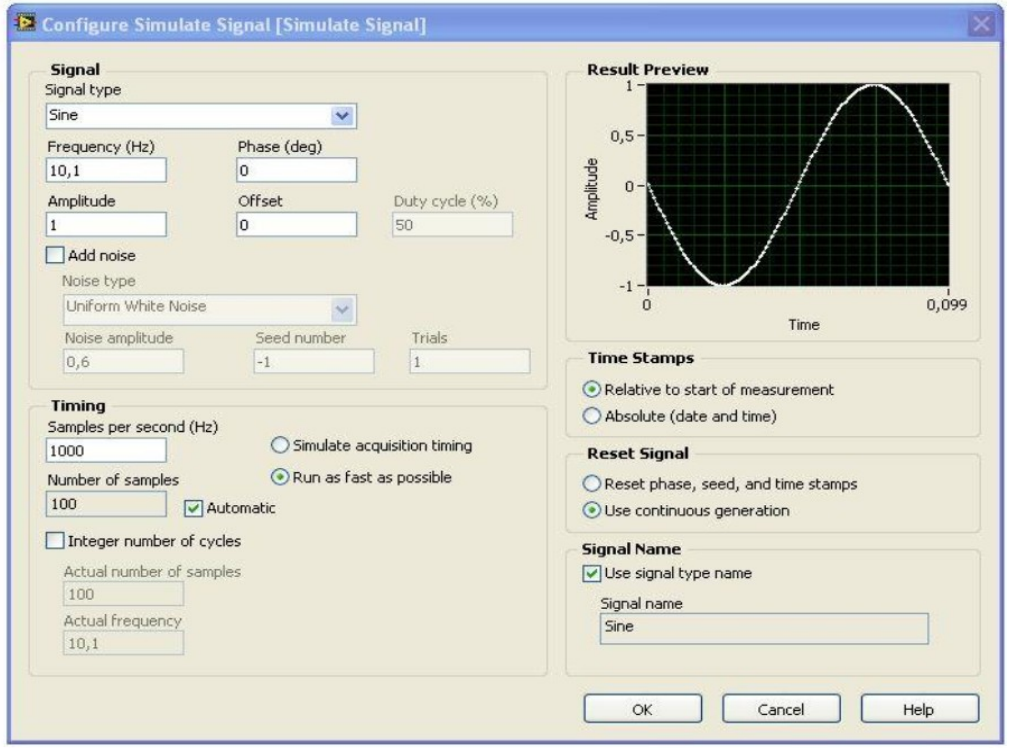
Acquisition de données

Menu NI-DAQmx

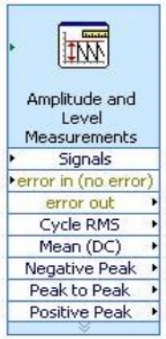
- Les V.I.s DAQ express : sous-VIs permettant de gérer, de configurer et de traiter les entrées/sorties

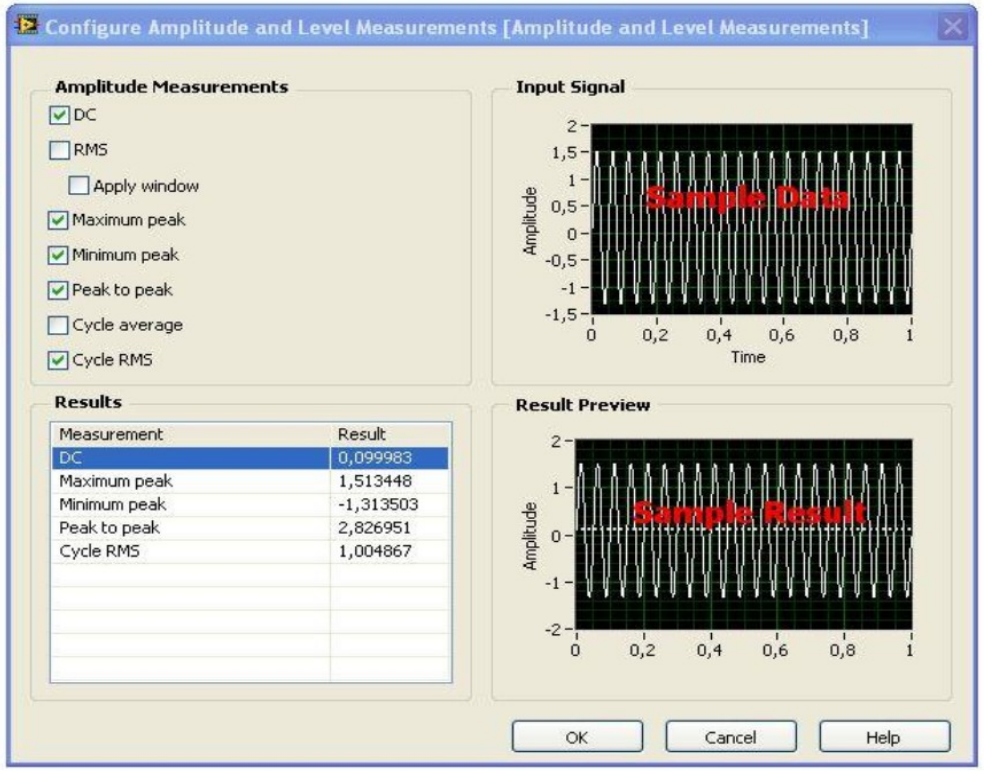
⇨ **Générer un signal électrique paramétrable…**





⇨ **Calculer les grandeurs électriques représentatives d'un signal…**

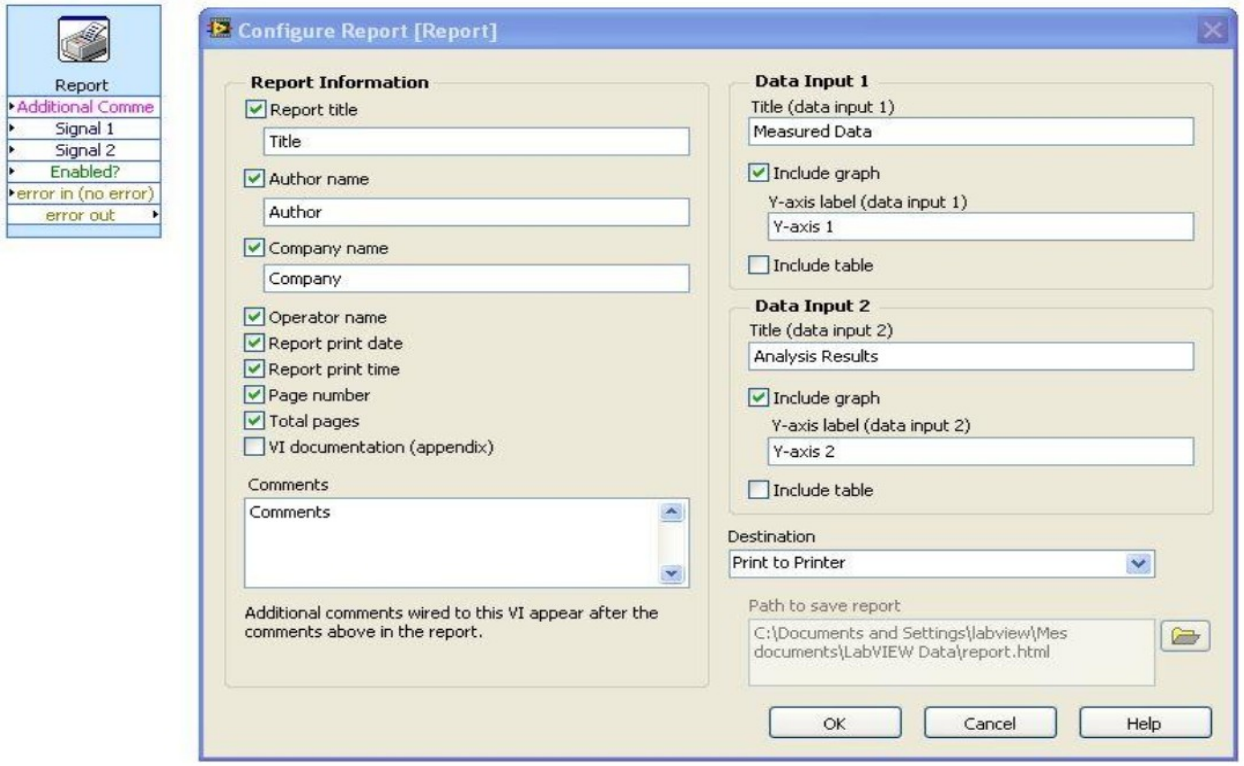




Acquisition de données
Menu NI-DAQmx

- Les V.I.s DAQ express : sous-VIs permettant de gérer, de configurer et de traiter les entrées/sorties

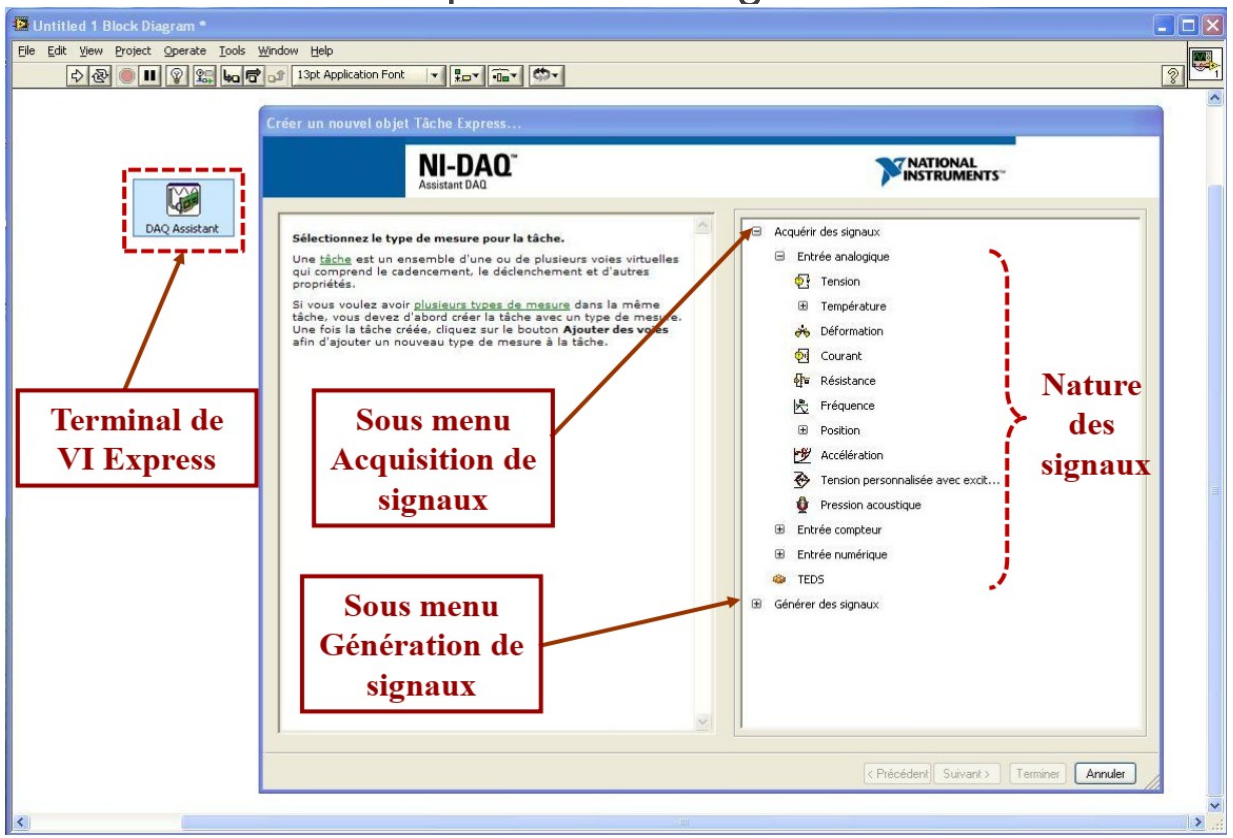
⇨ **Générer un rapport de mesures…**



Acquisition de données

Le DAQ assistant

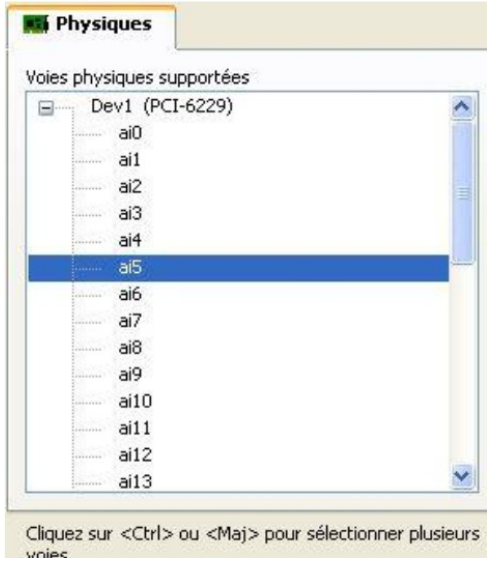
- Assistant à l'acquisition de signaux :



Carte PCI d'acquisition multifonction

Etape 2 :
Définir les voies physiques
de la carte interface d'acquisition

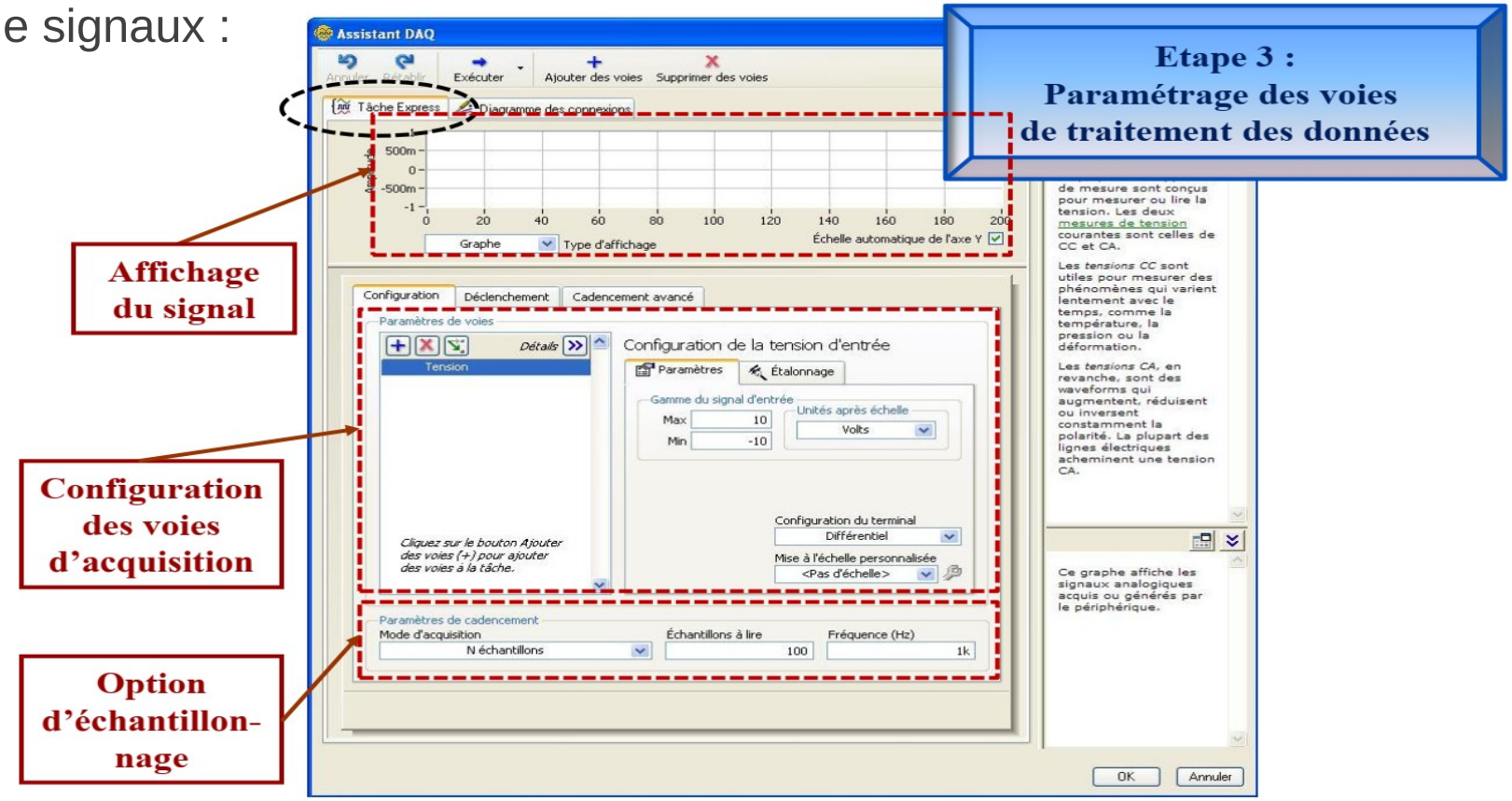
Déclaration de la connectique de la carte utilisée



Acquisition de données

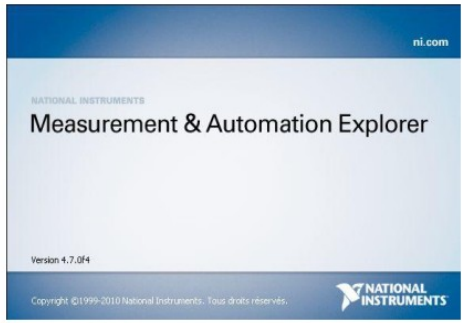
Le DAQ assistant

- Assistant à l'acquisition de signaux :

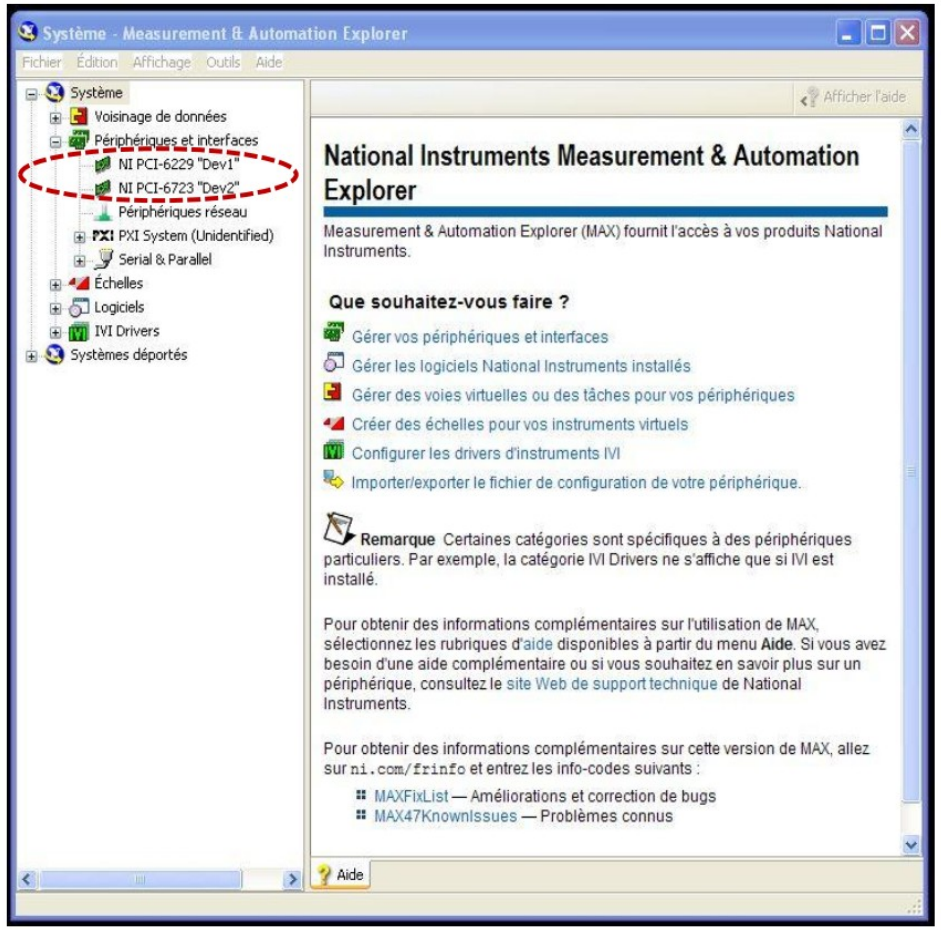


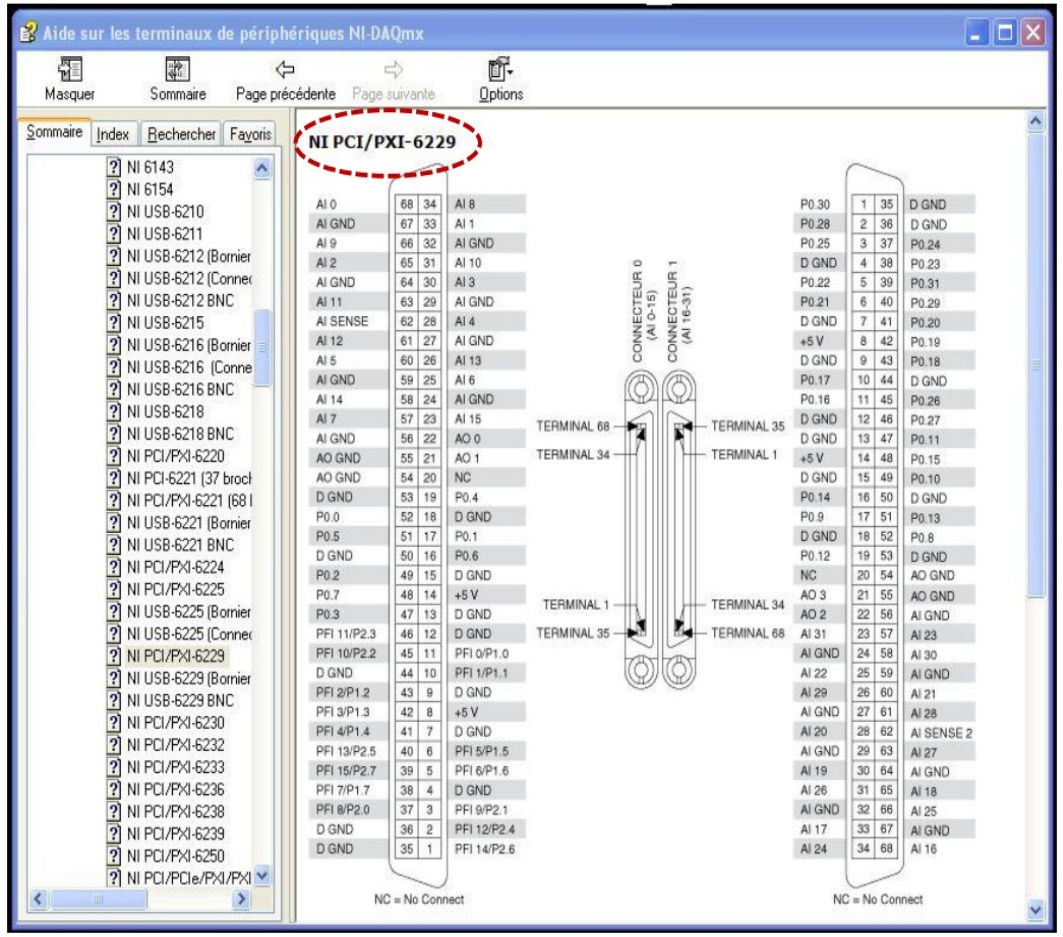
Acquisition de données

Measurement & Automation explorer



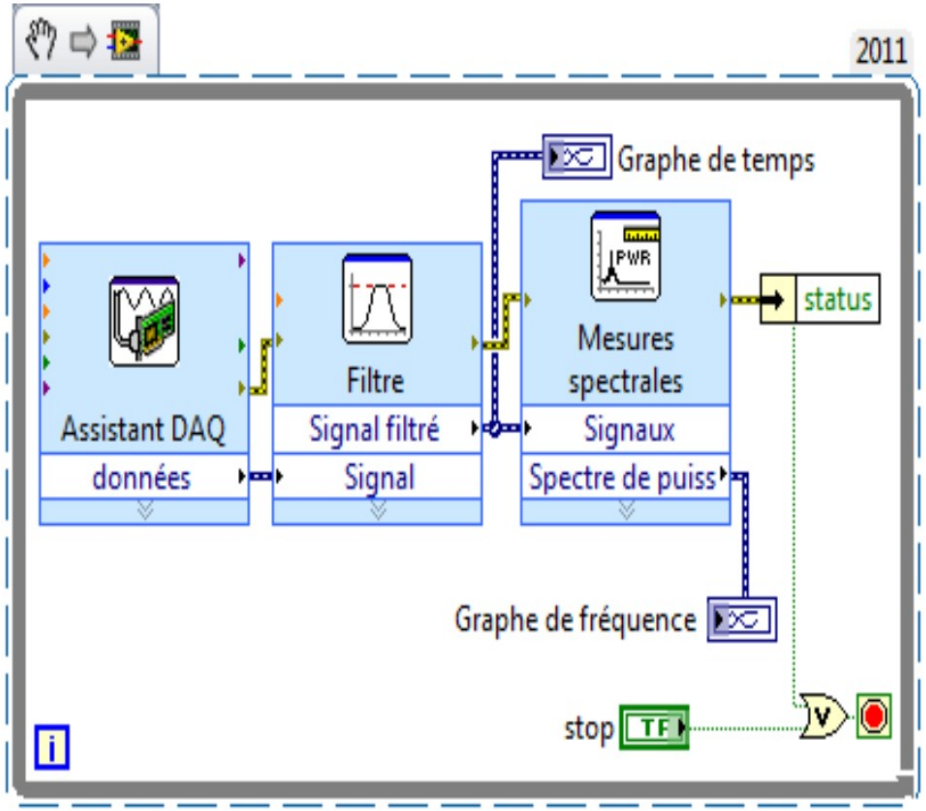
Accès aux ressources matérielles
Détection et communication avec les appareils GPIB
Test des cartes d'acquisition et de génération
Recalibration





Acquisition de données
Exercice 1

- Les carte d'acquisition:
  - Allumer le GBF et générer une tension analogique avec une fréquence 1.5kHz et d’amplitude 2.
  - Relier la sortie du GBF à la carte d'acquisition NI 9234 et brancher cette dernière sur PC.
  - Sur Labview, chercher les blocs affichée dans la figure ci-joint. Avant d’afficher votre signal, faire un filtrage du signal à une fréquence de coupure que vous choisissez.
  - Testez plusieurs fréquences de coupure et plusieurs valeurs pour la sensibilité. Interprétez les résultats.
  - Pour les signaux on prendra une fréquence d'échantillonnage de 51,2 kHz et une durée d'acquisition de 0.1 sec.



Acquisition de données
Exercice 2

- Créé un VI qui permet:
  - l’acquisition d’une tension sur l’entrée analogique 0 da la carte d’acquisition;
  - de mesurer la valeur efficace et la valeur moyenne de ce signal;
  - retranche au signal d’entrée sa valeur moyenne;
  - écrête ce signal à +/- sa valeur efficace;
  - génère ce signal sur l’une des sorties analogiques de la carte d’acquisition.

- En utilisera les cartes :
  - NI 9234 : Pour l'acquisition des signaux analogiques
  - NI 9204 : Pour la génération des signaux analogiques

Acquisition de données
Exercice 3

- Développer une Interface utilisateur/programme qui permet :
  - Spécifier la fréquence d'échantillonnage + durée des signaux à récupérer
  - Contrôler la vitesse d’un banc (on utilisera le banc moteur - réducteur)
  - Lancer / arrêter l’acquisition et le banc
  - Afficher les signaux acquis et leurs spectres
  - Afficher les valeurs des nivaux globaux
  - Enregistrer les signaux acquis sur des fichiers txt
- Ouvrir les fichiers sous matlab pour une analyse détaillée.

Merci pour votre attention