TP M1 - SIA CMD ESET ISTR RMGBM STP. INSTRUMENTATION - LABVIEW 14hTP

Déroulement des séances : 4 Séances de 3,5 heures

3,5h : TP1 Initiation Instrumentation Labview

- Les bases (constitution d'un VI + structures + calcul + tableaux +graphes +E/S fichiers). Pour cette partie voir avec le polycopié distribué.
- Exercices pour amener la corrélation. Le VI qui fait cela est donné, il reste à le faire tourner et à le comprendre.

3,5h : TP2 Carte E/S USB Ni6008/6009 ou Ni6221 ou USB-Ni6361

- Partie analogique Matériel : GBF + oscilloscope + PC et carte.
  * Entrée analogique (modes RSE et différentiel, fréquence d'échantillonnage, plage d'entrée … gain variable, déclenchement, mesure pas de quantification), travail à réaliser en lien avec la doc des cartes.
  (Matériel : GBF+ carte acquisition…+oscilloscope pour savoir ce que l'on doit récupérer)
  * Sortie analogique (fréquence d'échantillonnage, plages de sortie, quantification)
  * Faire aussi la FFT du signal et voir le repliement du spectre.
- Partie numérique (entrée / sortie) … utiliser des exemples déjà faits dans l'aide Labview pour illustrer cette partie. (En fonction du temps restant)

2,5h : TP3 GPIB Pilotage d'instruments avec LABVIEW

- La base avec MAX (converser avec un GBF ou oscilloscope, … utiliser la doc de l'appareil que l'on veut piloter)
- Analyser un fichier qui va avec votre appareil pour comprendre ce qu'il fait. Le modifier à la marge.
Eventuellement en fonction du temps disponible, l'enseignant montrera l'utilisation du pilotage à distance via le réseau LAN (1/2 h).

4,5h : Mini-projet : Mesure de la distance avec les E/R à ultrasons

Pour cette partie un compte rendu est demandé, il sera évalué et noté.

Le compte-rendu (∼7-8 pages au total ; ne pas dépasser 10 pages) devra contenir au minimum les points suivants :

1. Objectifs/moyens/résumé du TP
2. Mise en oeuvre :
   (a) justification du choix des réglages du GBF
   (b) interprétation du signal observé à l'oscilloscope
3. LabVIEW :
   (a) explication/justification du choix des paramètres d'acquisitions
   (b) explication de la réalisation de la corrélation croisée
   (c) interprétation (critique) de la cc.
4. Organigramme du VI (= représentation schématique des liens fonctionnels du programme)
5. Impression du VI (face-avant et diagramme) : soignez votre face-avant ! (ex. : titres des axes, etc)

Université Paul Sabatier, Toulouse

INITIATION À L'INSTRUMENTATION NUMÉRIQUE :

Utilisation du logiciel LabVIEW 2012 pour la mise en œuvre de cartes multifonctions et le pilotage d'instruments

TD d'initiation

4 Introduction à LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un logiciel de développement d'application d'un genre nouveau. En effet, contrairement aux autres langages de programmation, LabVIEW n'utilise pas un langage textuel, mais un langage de programmation graphique : le langage G. Les différentes fonctions du programme sont repérées par des icônes reliées entre elles par des fils. LabVIEW permet en outre : (1) de piloter des cartes multifonctions et des instruments munis d'interfaces séries ou GPIB ; (2) d'analyser, de présenter, de traiter et de stocker les données. Ce langage de programmation permet de développer des applications dans le domaine de l'instrumentation numérique sans connaissances approfondies de l'informatique. Les programmes LabVIEW sont appelés "**VI**" (pour "Virtual Instrument") car il s'agit réellement de "transformer" son ordinateur en un "instrument virtuel" qui regroupe les capacités d'un ou plusieurs oscilloscopes, d'un ou plusieurs générateurs de fonction, ... Un VI utilisé à l'intérieur d'un autre VI est appelé "**sous-VI**". Chaque sous-VI peut être exécuté de façon autonome.

Les VI sont organisés de la manière suivante :

— Une **face avant** sur laquelle le programmeur va créer l'interface graphique visible (tracé de courbes, indicateurs de valeurs, témoins de mise en service, boutons de commande, commutateurs rotatifs, des LED, ...

— Un **diagramme** dans lequel le programmeur va dessiner le code de l'application. Pour cela, le programmeur dispose d'une grande quantité de bibliothèques contenant des sous-programmes (ou sous-VI), déjà écrits, qui permettent d'effectuer une grande partie des fonctions les plus utilisées dans le test et la mesure ains que le contrôle de processus (acquisition de données, contrôle d'instruments...). Il existe également des bibliothèques contenant les fonctions classiques des langages usuels (boucles FOR, WHILE, manipulation de tableaux, de fichiers, ...).

Lorsque vous démarrez LabVIEW, vous obtenez une fenêtre similaire à celle de la figure 1.

Figure 1 – Fenêtre de démarrage LabVIEW.

Depuis la version 8.0, LabVIEW offre la possibilité de créer et gérer des projets via l'Explorateur de projet. Celui-ci offre aux utilisateurs une visualisation, dans l'environnement de développement, des fichiers dont ils ont besoin pour une application.

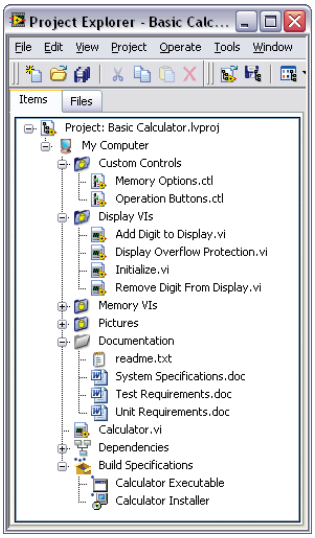


Figure 2 – Exemple de projet LabVIEW.

Pour accéder à l'explorateur de projet, cliquer sur "Create project" dans la fenêtre de démarrage, puis sur "Blank Project".

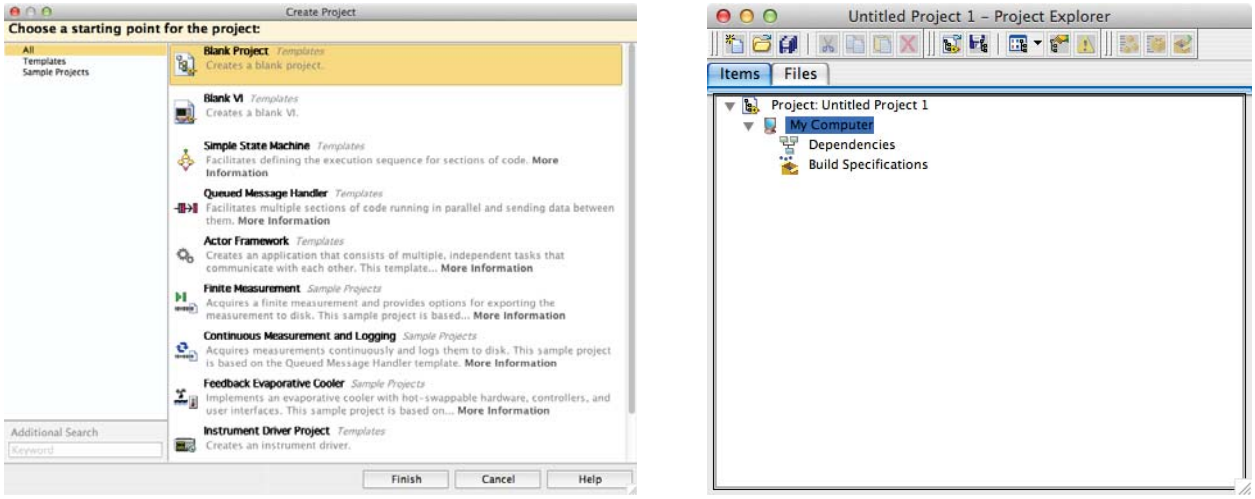


Figure 3 – Accès à l'explorateur de projet LabVIEW.

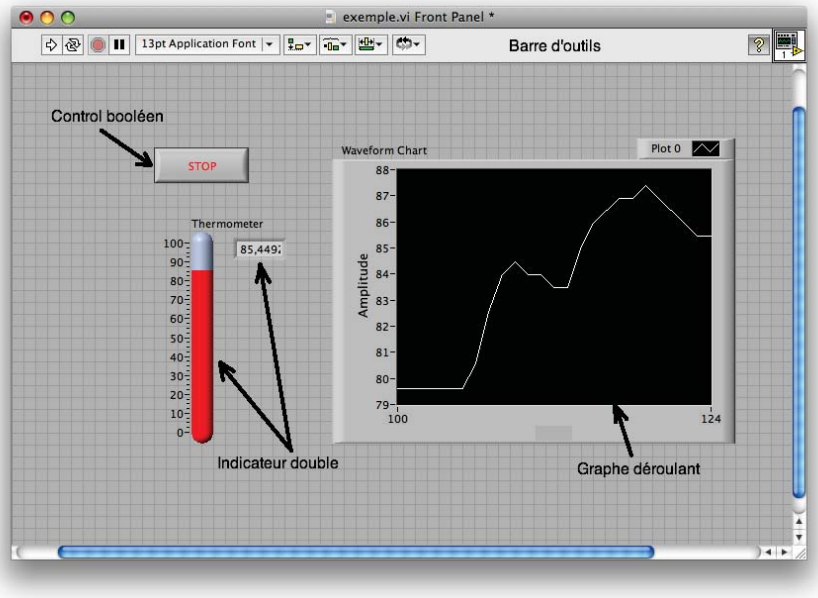
Pour les séances de TD, la création de projet n'est pas nécessaire, nous nous contenterons d'un simple "Blank VI".

4.1 Constitution d'un VI

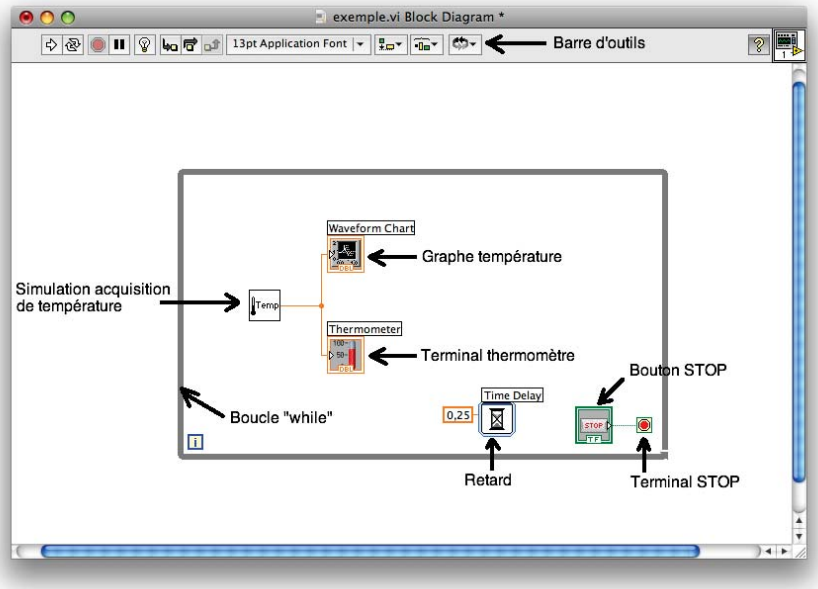
Un VI possède donc 2 fenêtres essentielles :

— La **face avant** sur laquelle on dispose les commandes, les indicateurs, les graphes... (similaire à la face avant d'un instrument). Les différents éléments utiles pour la face avant sont regroupés dans la **palette de commande**.

— Le **diagramme** qui regroupe les différents éléments de programmation G décrivant l'action du VI. Les différents éléments utiles pour la face avant sont regroupés dans la **palette de fonctions**.



(a) La face avant du VI

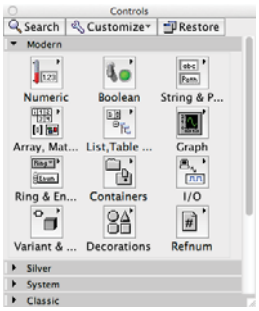


(b) Diagramme du VI

Figure 4 – Exemples de face avant et de diagramme d'un VI LabVIEW.

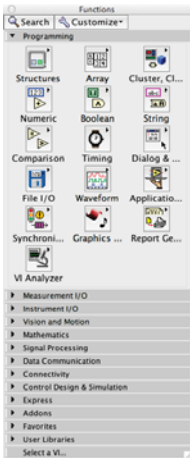
Les outils nécessaires à la programmation et l'exécution des VIs sont regroupés dans la **palette d'outils**. Pour changer d'outils, on peut soit cliquer dans la barre d'outils, soit taper sur la touche "ESPACE". Si on active la petite icône verte en haut de la palette d'outils, la sélection de l'outil se fait automatiquement (par exemple si on approche d'un connecteur, la bobine de liaison apparaît). Une illustration de ces différents éléments est donnée dans les figures 5–7.

Palette de commandes
(associée à la face avant)



Utilisée pour disposer des commandes ou des indicateurs sur la face avant du VI.

Palette de fonctions
(associée au diagramme)



Utilisée pour disposer des opérateurs mathématiques, des outils statistiques, des sous programmes ou des constantes, sur le diagramme du VI.

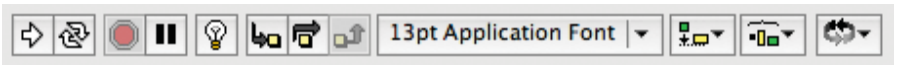
Figure 5 – Palettes de commandes et de fonctions.

Sélection automatique

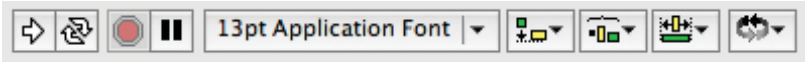
- Changer un paramètre
- Déplacer / changer la taille
- Ecrire
- Connecter
- Raccourci
- Scrolling tool
- Mettre un point d'arrêt
- Mettre une sonde
- Changer la couleur
- Choisir la couleur

Figure 6 – Palettes d'outils : outils nécessaires à la programmation ou à l'exécution d'un VI

Diagramme :



Face avant :



- Exécution du VI
- Exécuter en continu
- Arrêt exécution
- Pause/Continue
- 13pt Application Font — Attributs du textes
- Aligner et ranger des objets
- Ranger

Pour le diagramme :

- Exécution pas à pas animée
- Exécution spéciale

Figure 7 – La barre de status

LabVIEW propose également de nombreux programmes (VIs) qui "simulent" le comportement et les performaces d'instruments disponibles dans le commerce, grâce à une carte multifonctions connectée dans l'ordinateur. Dans le cas présenté, le VI correspond à un générateur de fonction Hewlett Packard 34401A. On appelle ces programmes des VI "Instruments".

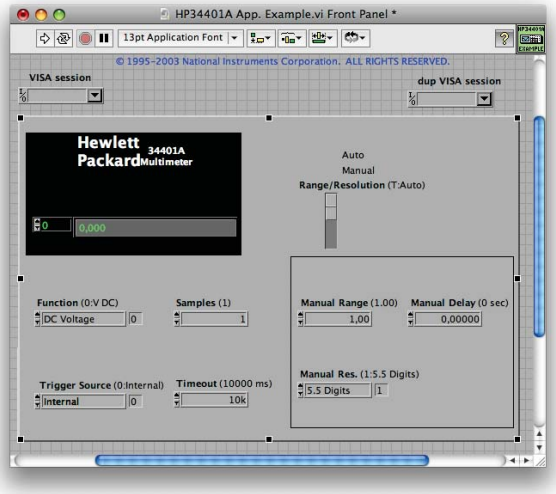


Figure 8 – Un exemple de VI "instrument". Les VI Instruments servent à : (i) faire l'interface homme-machine (ii) simuler un instrument existant.

4.2 Définitions importantes

1. **Commande** : paramètre d'entrée du VI ou terminal d'entrée du VI. Une commande apparaît à la fois sur la face avant (sa valeur est modifiée en interactif avec l'outil "doigt" ou "caractère" A) et sur le diagramme où elle sert de terminal d'entrée au VI (voir figure 9). A l'exécution du VI, la commande prend la valeur indiquée.

2. **Indicateur** : paramètre de sortie du VI ou terminal de sortie du VI. Un indicateur apparaît à la fois sur la face avant et sur le diagramme (voir figure 10). Un indicateur permet de visualiser le résultat de l'exécution d'u VI ou le contenu d'une variable en cours d'exécution. Il existe des indicateurs de toutes sortes : numériques, tableaux, caractères, graphes, ...

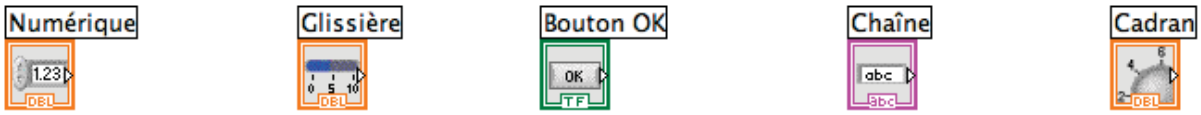
3. **Constantes** : valeurs fixes qui servent lors de l'exécution du VI. Elles n'apparaissent que sur le digramme (voir figure 11).

4. **Fonctions** : sous-VI prédéfinis, disponible dans la palette de fonctions. Exemple : fonction > numérique > multiplier. Voir figure 12.

   Il est possible de faire apparaître le **menu local** d'une fonction, d'une commande ou d'un indicateur en cliquant sur le bouton droit de la souris tout en la maintenant sur l'icône correspondant. Ce menu très important permet d'effectuer de nombreuses opérations :
   - **aide en ligne** : description des terminaux et de la fonction
   - **description** : rentrer sa propre description de la fonction
   - **visualiser** : les étiquettes, les terminaux
   - **remplacer** : substituer une fonction par une autre
   - **créer** : une constante, un indicateur, une commande avec un câblage automatique sur un terminal

Ce dernier point est très important : tout objet peut être câblé automatiquement grâce à son menu local (créer un indicateur ou une commande selon le cas). Ceci permet d'éviter toute une série d'erreur (erreur de type, confusion commande-indicateur, etc). Faire plusieurs tests de ce principe.

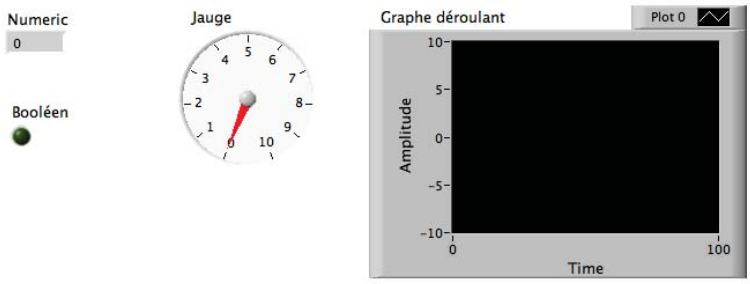
(a) Exemples de commandes sur la **face avant**. On peut redéfinir les bornes d'une commande numérique avec l'outil "doigt" ou "A" : sur le cadran, on a changé 0,10 en -50,50.



(b) Exemples de terminaux des commandes sur le **diagramme**. Les commandes sont encadrées en traits **gras**. L'icône permet de repérer la commande, la couleur traduit le type :

- orange : réels (DBL : double précision)
- bleu : entiers (I8 : entiers de 8 bits)
- vert : logique (TF = True-False ou Vrai-Faux)
- rose : chaînes de caractères (abc)

Figure 9 – Exemples de commandes (dans "palette de commandes").



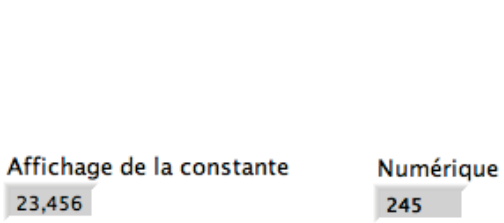
(a) Exemples d'indicateurs sur la **face avant**. On peut redéfinir les bornes d'un indicateur numérique avec l'outil "doigt" ou "A".

(b) Exemples de terminaux d'indicateurs sur le **diagramme**. Les commandes sont encadrées en traits **fins**. L'icône permet de repérer la commande, la couleur traduit le type :

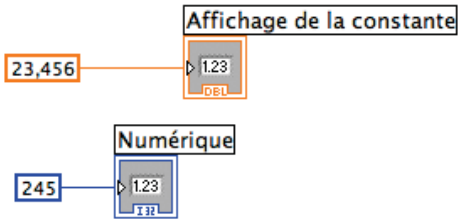
- orange : réels (DBL : double précision)
- bleu : entiers (I8 : entiers de 8 bits)
- vert : logique (TF = True-False ou Vrai-Faux)
- rose : chaînes de caractères (abc)

Figure 10 – Exemples d'indicateurs (dans "palette de commandes").

5. **Liaisons** : elles permettent le transfert de l'information entre les sous-VI. Elles s'effectuent à l'aide de l'outil "**bobine**" (voir figure 13). Les caractéristiques des liaisons (forme, épaisseur, couleur) reflètent le type d'information transportée (entier, réel, logique, vecteur, tableau, caractère, ...). Une liaison incorrecte apparaît en tirets (voir figure 14 — attention de ne pas confondre avec la liaison en pointillés qui correspond à une liaison logique, cf. figure 13), barrés d'une croix rouge. On peut détruire tous les mauvais câblages avec **CTRL-B**.



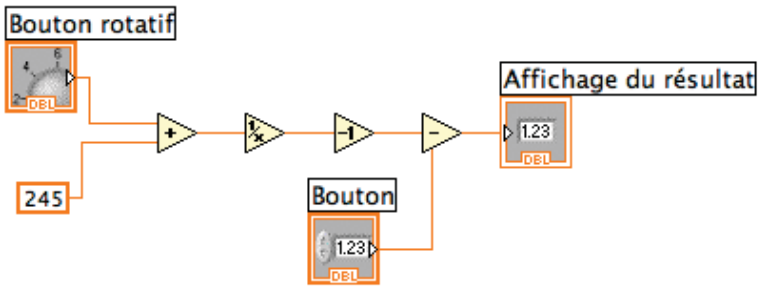
(a) Face avant : les constantes n'y apparaissent pas (seuls les indicateurs de ces commande apparaissent).



(b) Diagramme (code des couleurs identique ; pour changer le type et créer un indicateur par "clic-droit", menu local)

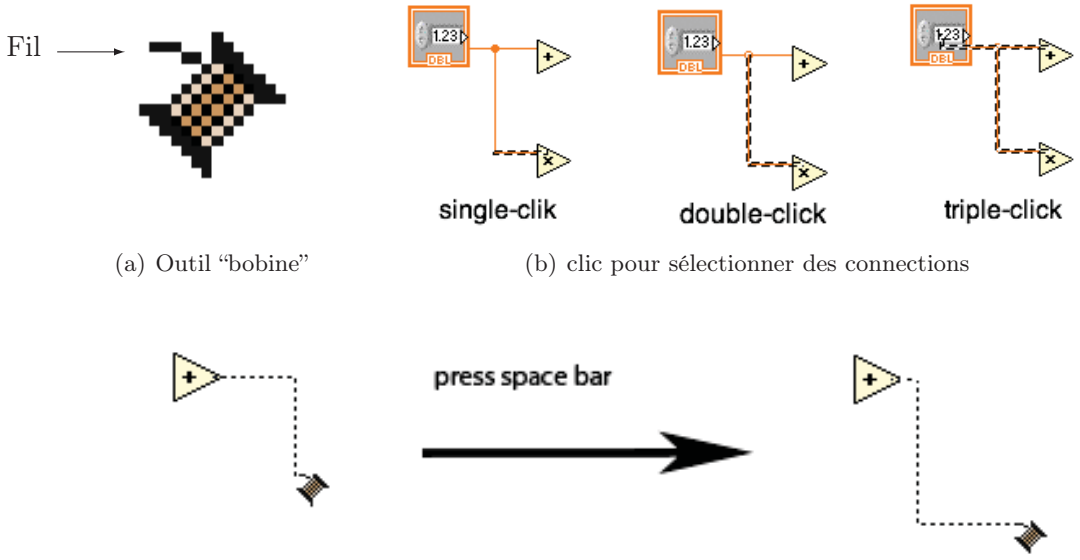
Figure 11 – Exemples de constantes, ici, par exemple, les valeurs 23,456 et 245 (dans "palette de fonctions").

(a) Face avant. A l'exécution, l'indicateur affiche le résultat du calcul.



(b) Diagramme

Figure 12 – Exemples de fonctions (Fonctions > Programmation > Numérique).



(a) Outil "bobine" (b) clic pour sélectionner des connections

(c) Barre "espace" change l'orientation

Figure 13 – Liaisons : outil bobine et astuces sur les connections

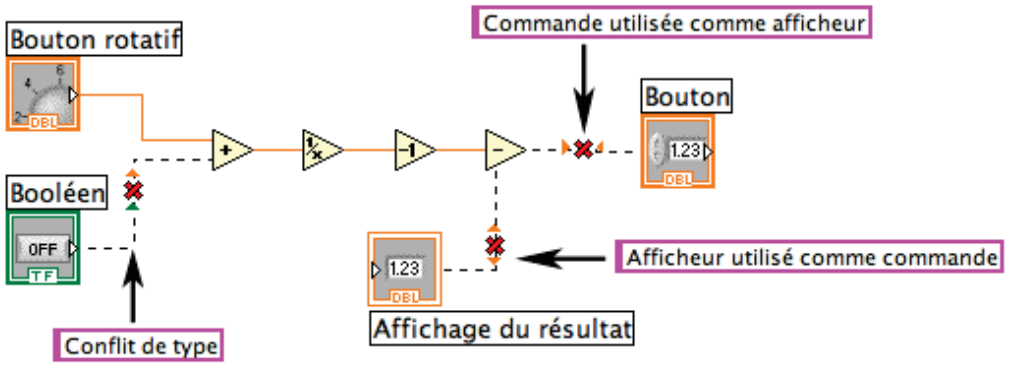


Figure 14 – Exemples de câblages incorrects sur le diagramme

4.3 Exercices

1. Ecrire un VI qui calcule le rapport de deux nombres réels x et y (division réelle) que l'on spécifie par des commandes (**numériques**). On spécifiera x=76.2 et y = 4.5

   Mode opératoire :

   — Disposer deux commandes numériques que l'on renomme "x" et "y". Disposer un opérateur "diviser". Cabler ces trois éléments. Faire apparaître le menu local de la sortie de l'opérateur "diviser". Actionner : **créer ... un indicateur**. Cette opération crée un indicateur (par défaut un afficheur numérique) qui a automatiquement le type correct et qui est déjà câblé à l'opérateur. Exécuter votre programme.

   — Faire apparaître le menu local de l'indicateur qui affiche le résultat. Actionner : **remplacer ...** pour remplacer celui-ci par n afficheur à aiguille. Modifier les bornes de l'indicateur à aiguille pour que celui-ci affiche correctement le résultat. Faire apparaître le menu local de la commande x. Actionner : **représentation ... I8** pour transformer cette commande en une commande de type entier de 8 bits. Vérifier que cette commande est bien entière. Exécuter votre programme.

   — Exécuter votre programme "en continu" (les deux flèches). Modifier les valeurs des paramètres en cours d'exécution. Observer le résultat.

2. Faire un VI qui calcule $y = ax^2 - bx + 3/x$ avec x, a et b des réels double précision

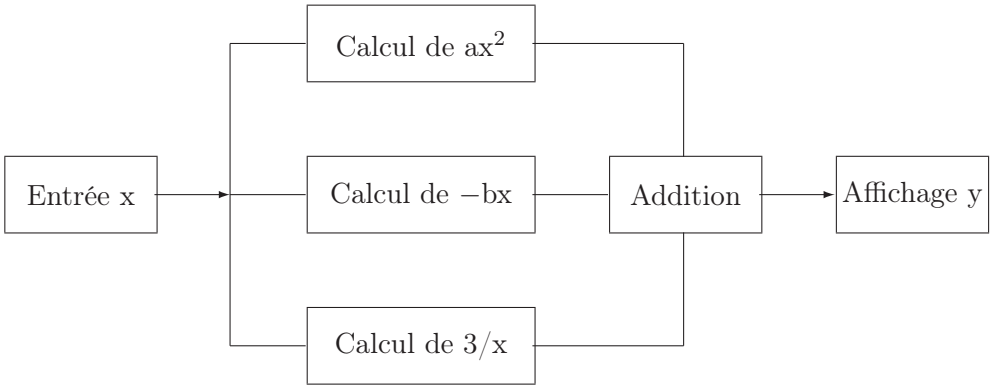
   — x est une commande de type numérique

   — a est une commande de type glissière

   — b est une commande de type bouton

   — y sera représenté sur un vu-mètre (cadran à aiguille)

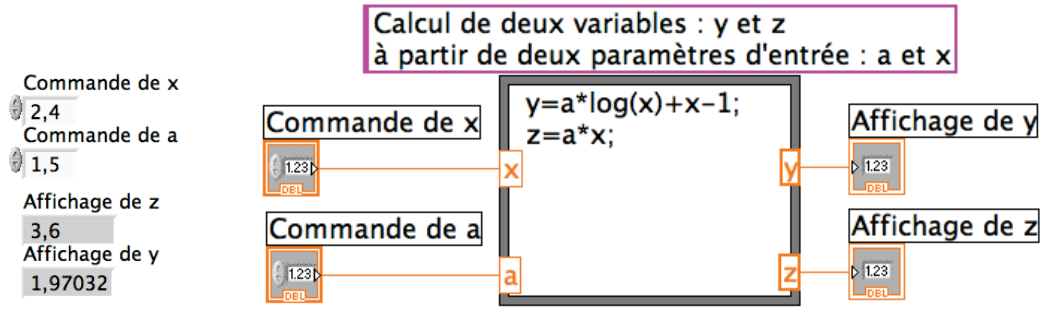
   Mode opératoire (organigramme simplifié) :



   Respecter la disposition des blocs de calcul ci-dessus afin d'obtenir un VI clair, symétrique et harmonieux (et facile à débugger ...). Exécuter ce programme en continu en modifiant les paramètres.

4.4 La structure "boîte de calcul"

Elle permet d'implémenter une équation mathématique sans utiliser les fonctions numériques. La figure 15 montre un exemple d'utilisation de cette structure. On entre les équations avec l'outil A (éditeur de texte). Les entrées et les sorties (les terminaux) sont ajoutées avec le **menu local de la structure** (clique-droit de la souris placée sur le cadre de la structure).
Chaque ligne de l'équation doit se terminer par " ;" .



(a) Face avant (b) Diagramme

Figure 15 – Calculs avec une boîte de calcul.

4.5 Les structures boucles et registres à décalage

Comme pour les langages de programmation classique (Matlab, C, ...), il est possible de faire des boucles avec LabVIEW. Les objets graphiques permettant de réaliser les boucles sont dans la palette fonction > structures. Les deux principales boucles qui seront utilisées sont la boucle "FOR" et la boucle "WHILE". Ces boucles ainsi que les registres à décalage sont décrits ci-dessous.

1. La structure "**WHILE**" : implémente le pseudo code "Exécuter ce qui est à l'intérieur de la boucle tant que la condition est vraie". Cette structure est munie d'un terminal d'itération qui indique le nombre de passage dans la boucle. Un "booléen" (bouton "stop") est automatiquement connecté au terminal conditionnel. Il peut bien sûr être remplacé par un test sur une valeur. Voir les figures 16 et 17.

2. La structure "**FOR**" : implémente le pseudo code "Pour i valant de 0 à N par pas de 1, exécuter ce qui est à l'intérieur de la boucle". Le nombre d'itération (N) est défini à l'extérieur de la boucle. Voir les figures 18 et 19.

3. **Les registres à décalage** : ces registres permettent de stocker des informations (scalaires, vecteurs, tableaux, chaînes de caractères) qui circulent et/ou sont modifiées dans des boucles au cours des itérations (voir figures 20, 21 et 22). La mise en place d'un registre à décalage s'effectue en plaçant le curseur de la souris sur le bord de l'objet boucle et en cliquant sur le bouton droit (**menu local de la structure**).

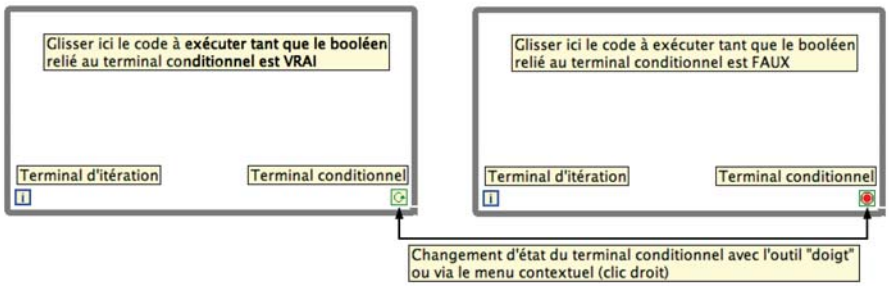
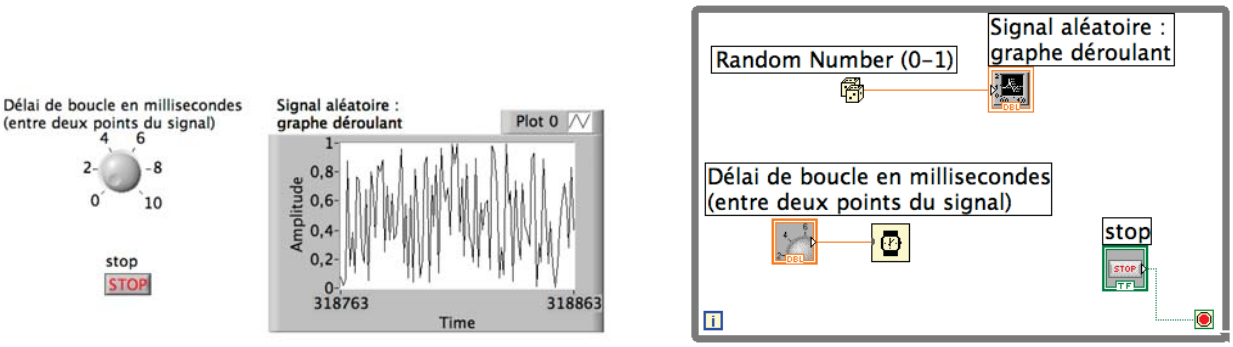


Figure 16 – La structure WHILE.



(a) Face avant. (b) Diagramme.

Figure 17 – Exemple d'utilisation de la boucle WHILE.

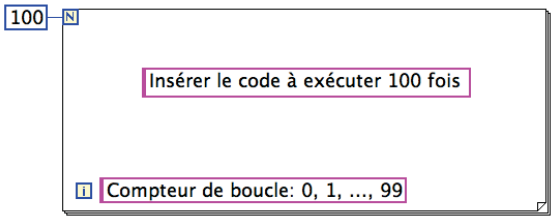
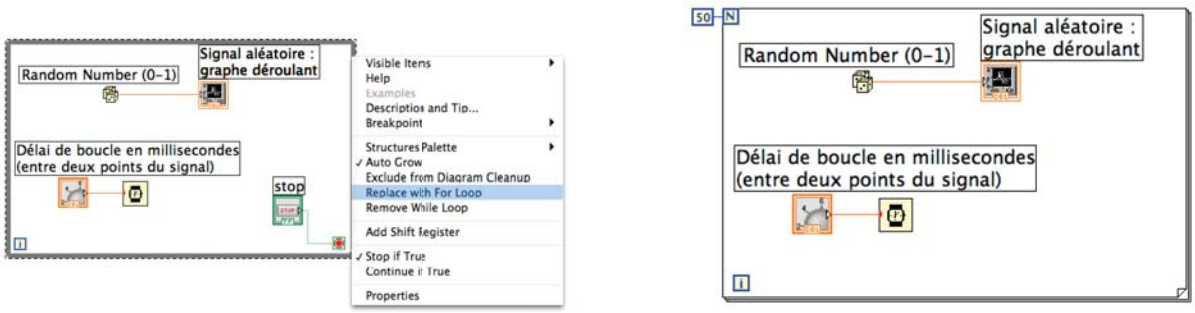


Figure 18 – La boucle FOR.



(a) Remplacement de la boucle WHILE via le menu contextuel (clic droit sur le bord de la structure). (b) Diagramme.

Figure 19 – Équivalent de l'exemple de la figure 17 en remplaçant la boucle WHILE par une boucle FOR.

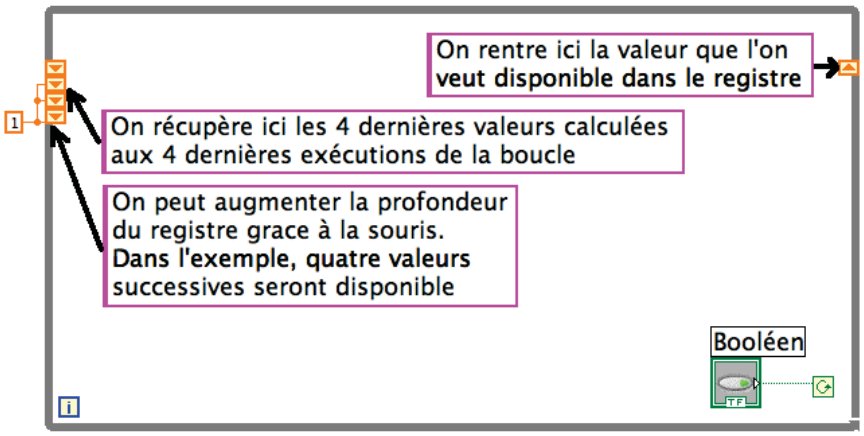


Figure 20 – Registre à décalage

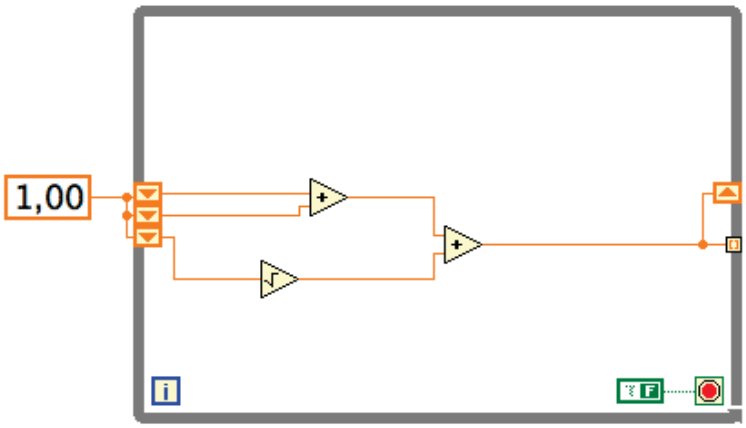
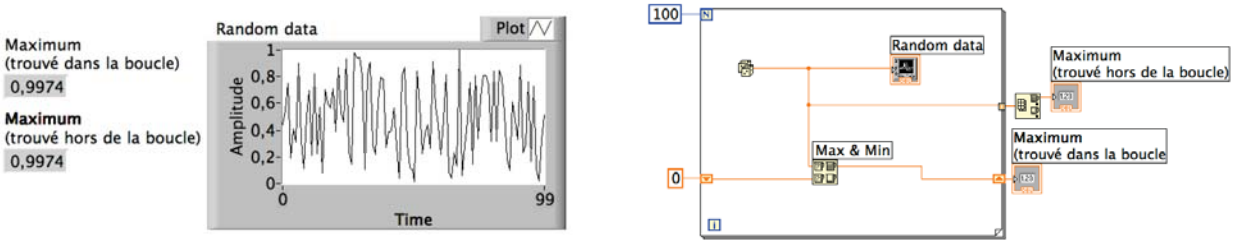


Figure 21 – Exemple d'utilisation de registre à décalage. **Exercice : que fait ce programme ?**



(a) Face avant (b) Diagramme

Figure 22 – Exemple d'utilisation de la boucle FOR avec des registres à décalage.

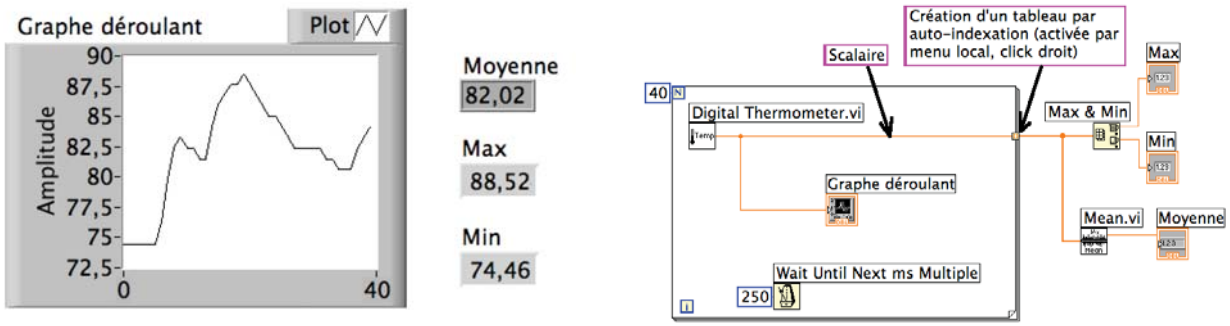
4.6 Exercice

On considère les éléments de la suite $x(n) = x(n-1)+x(n-2)$ avec $x(0) = 1$ et $x(1) = 1$. Cette suite est appelée "suite de Fibonacci".

1. Utilisation d'une boucle "WHILE" :
   - calculer un élément toutes les 0.1s
   - arrêter le calcul quand l'utilisateur le souhaite, à partir d'une commande STOP
   - afficher la suite x dans un graphe déroulant
2. Modifier votre VI pour utiliser une boucle "FOR" :
   - calculer les 40 premiers éléments de la suite
   - afficher la valeur de $x(n)/x(n-1)$. En déduire la limite de cette valeur lorsque n tend vers l'infini

4.7 Les tableaux

Plusieurs méthodes peuvent être utilisées pour définir un tableau de valeurs. Certaines fonctions (ex. : sinus, impulsion, ...) fournissent directement un tableau de N valeurs. Les valeurs calculées dans une boucle (FOR en particulier) peuvent être placées dans un tableau si l'auto-indexation est activée (voir figure 23). Un scalaire dans une boucle qui est sorti de cette boucle par un fil devient automatiquement un tableau contenant toutes les valeurs successives de ce scalaire. Ce principe s'appelle : **auto-indexation**. Pour obtenir uniquement le dernier élément en sortie de boucle, il est nécessaire de désactiver cette auto-indexation par le **menu local du terminal** de sortie de boucle. Ces méthodes sont illustrées dans les figures ci-dessous (figure 24) de même que quelques remarques sur le polymorphisme des fonctions (figure 25) et quelques outils spécifiques aux tableaux, par exemple sous-ensemble d'un tableau, taille d'un tableau (figure 26).

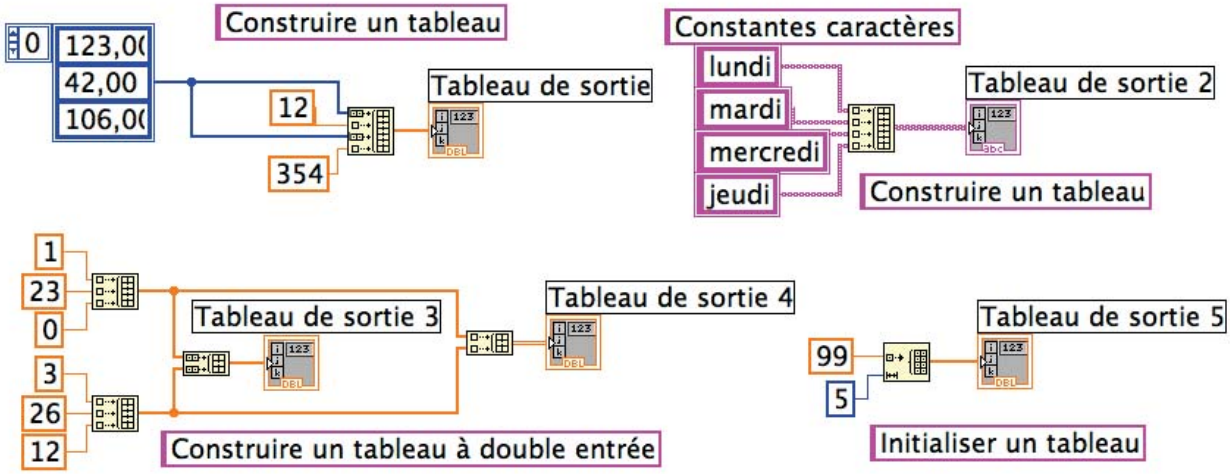


(a) Face avant.

(b) Diagramme.

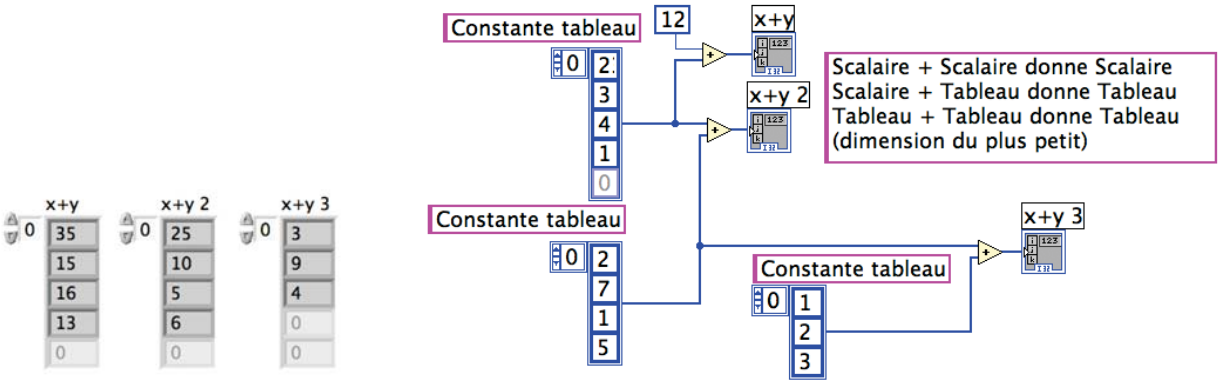
Figure 23 – Exemple de création d'un tableau avec la boucle FOR.

(a) Face avant.



(b) Diagramme.

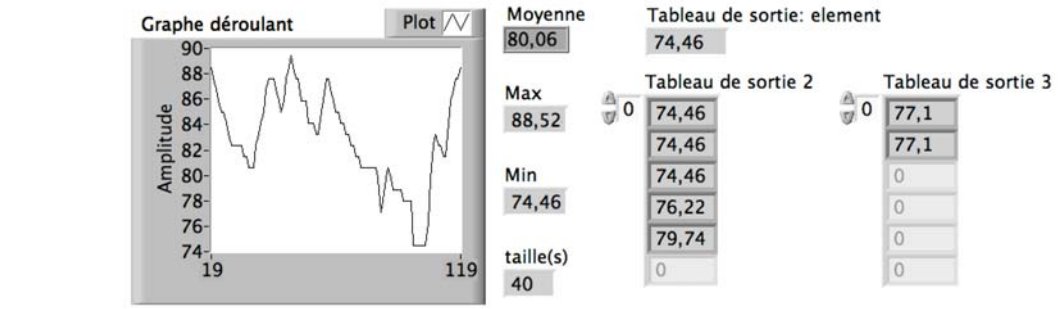
Figure 24 – Exemple de définition d'un tableau.



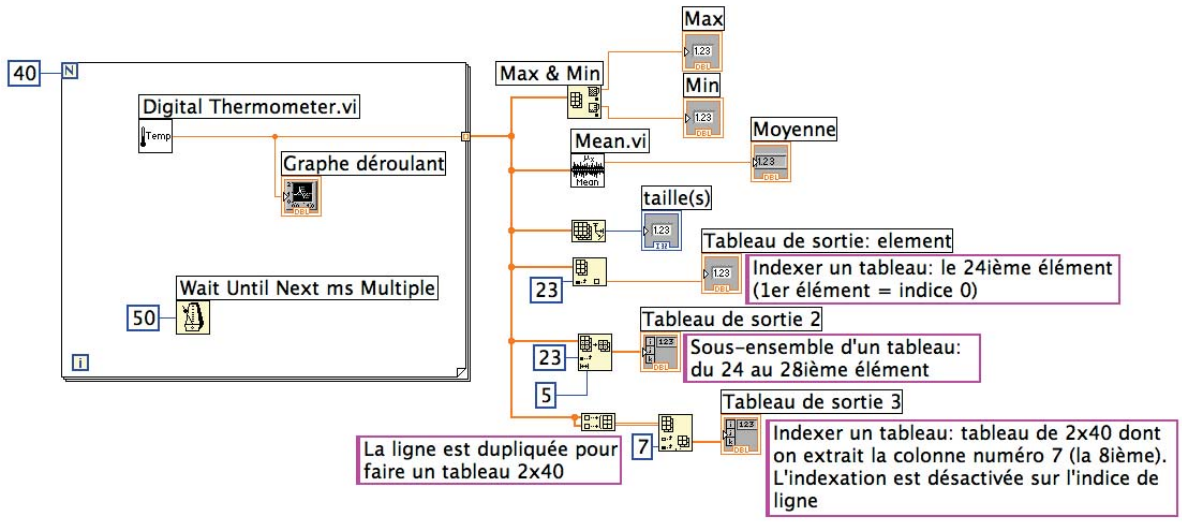
(a) Face avant

(b) Diagramme

Figure 25 – Remarque sur le polymorphisme des fonctions.



(a) Face avant



(b) Diagramme

Figure 26 – Outils de manipulation des tableaux.

4.8 Exercices

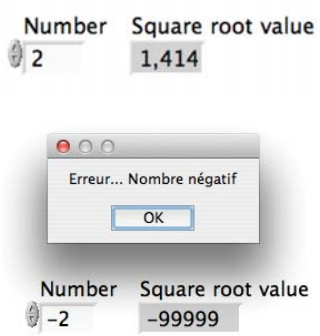
On considère à nouveau la suite de Fibonacci : $x(n) = x(n-1) + x(n-2)$ avec $x(0) = 1$ et $x(1) = 1$. Ecrire un VI de manière à obtenir :

- un tableau qui contient les 15 premiers éléments de la suite (par auto-indexation)
- un tableau qui contient les 5 premiers éléments de la suite (par extraction du tableau précédent)
- un indicateur qui contient l'élément numéro 8
- un indicateur qui contient la taille du tableau
- un indicateur qui contient la moyenne des éléments du tableau

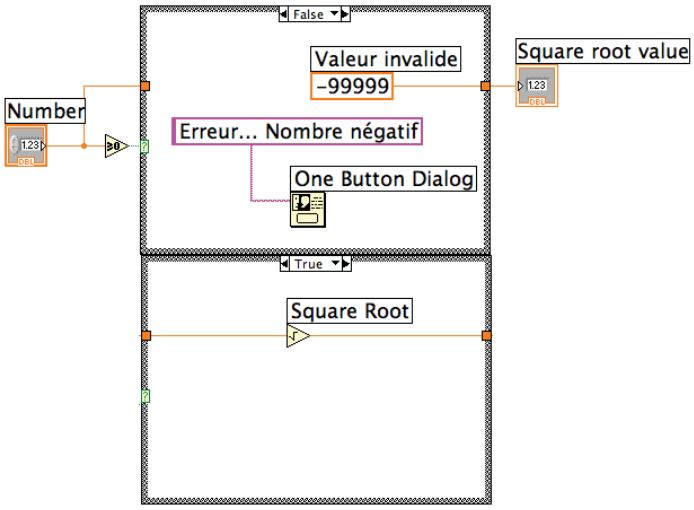
4.9 Les structures "condition" et "séquence"

1. La structure "condition"
   Dans le cas le plus simple, cette structure possède deux volets exécutés selon une condition.

L'entrée de la condition ou terminal de sélection est repéré par le symbole " ?". Cette entrée reçoit soit un booléen pouvant résulter d'une opération logique (voir l'exemple présenté figure 27), soit un entier.



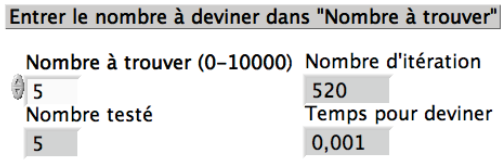
(a) Faces avant pour les cas "VRAI" et "FAUX".



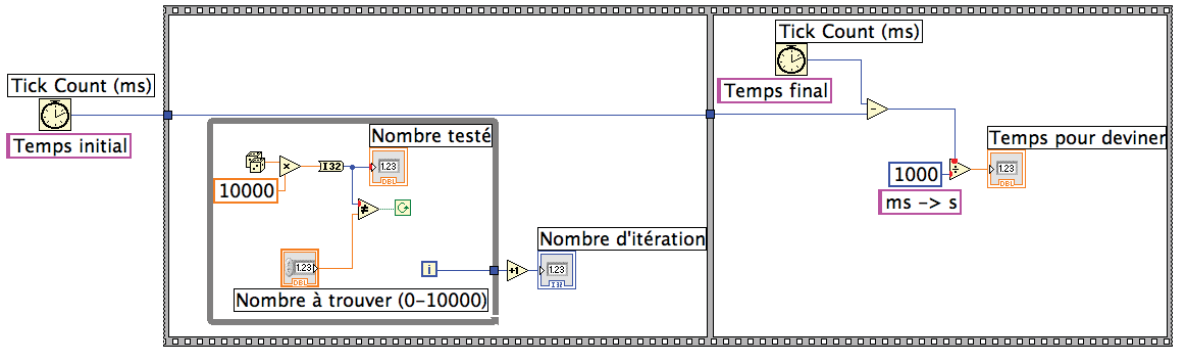
(b) Diagramme.

Figure 27 – Exemple d'utilisation de la structure "condition".

2. La structure "séquence"
   Certaines opérations doivent être efectuées l'une après l'autre – séquentiellement. La structure "séquence" permet de définir les différentes étapes des opérations sur différents volets. La figure 28 présente un exemple d'utilisation de cette structure.



(a) Face avant.



(b) Diagramme.

Figure 28 – Exemple d'utilisation de la structure "séquence" en mode déroulé.

4.10 Exercices

1. Calculer le temps mis par un programme pour évaluer la fonction $f(x) = \sin^2(x) + \sin(x) - 1$, en 10^5 point couvrant uniformément l'intervalle [-2 ;+2].
2. Ecrire un VI qui calcule :
   $\begin{cases} y = -x & \text{si } -0.5 < x < 0.5 \\ y = x - 1 & \text{sinon} \end{cases}$
   pour x variant de -2 à 2 par pas de 0.01.

4.11 Graphes

Les graphes sont des indicateurs accessibles par la palette de commande. Il en existe plusieurs types : graphe déroulant, graphe, graphe X-Y, graphe 3D, ... La figure 29 donne un exemple des trois premiers types, qui seront les plus utilisés :

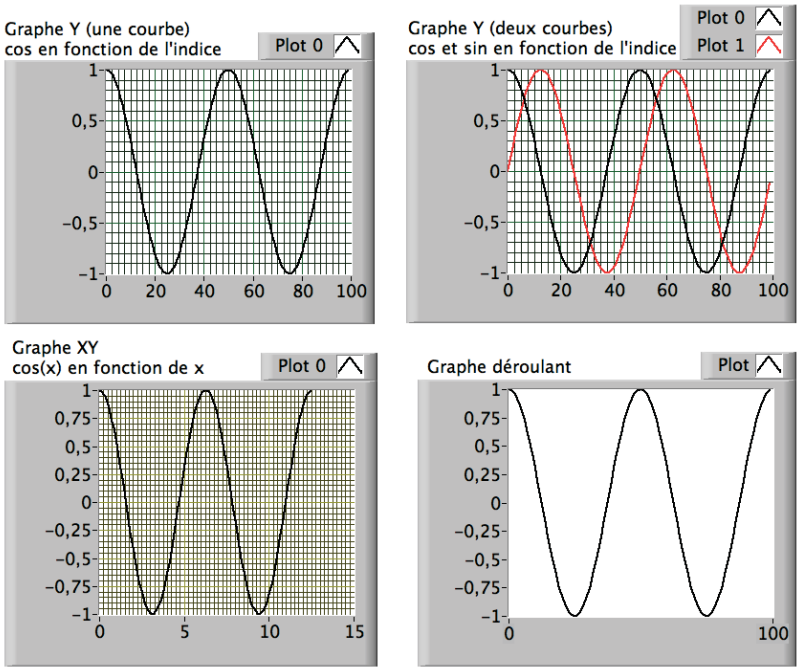
— le **graphe déroulant** ("waveform chart") trace les valeurs d'un élément au fur et à mesure, c'est-à-dire à chaque itération ou bien à chque exécution du VI principal ;

— le **graphe** ("waveform graph"), nommé **graphe Y** par la suite, trace les valeurs d'un vecteur en fonction de son indice. Si le graphe Y est branché sur un tableau (un ensemble de vecteurs), les différents vecteurs du tableau sont représentés indépendamment ;

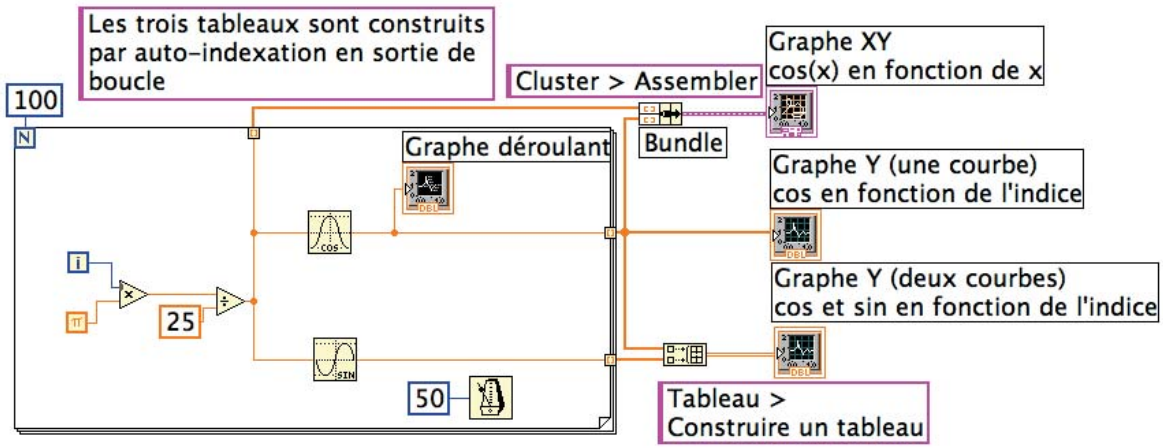
— le **graphe X-Y** permet de tracer un vecteur en fonction d'un autre vecteur de même dimension. Pour l'utiliser, on assemble ("bundle") un cluster contenant les deux vecteurs des deux axes X et Y (dans cet ordre). Le cluster est alors envoyé au "graphe X-Y".

Les option de représentation des courbes (couleur, style de tracé, grille...) et axes (format des labels, échelle linéaire ou logarithmique, ...) sont accessibles par différents menus locaux. L'échelle des axes peut être modifiée en changeant une des valeurs avec l'outil A. Un des menus locaux permet de créer un curseur pour visualiser les valeurs du graphe avec la souris.

Une façon alternative de spécifier l'axe horizontal quand celui-ci est régulièrement espacé est de bâtir un cluster contenant les informations : valeur initiale sur x, incrément sur x et vecteur y (dans cet ordre) et de fournir ce cluster à un graphe comme dans l'exemple suivant (figure 30). Cette méthode est très pratique pour éviter de devoir créer des vecteurs inutiles.

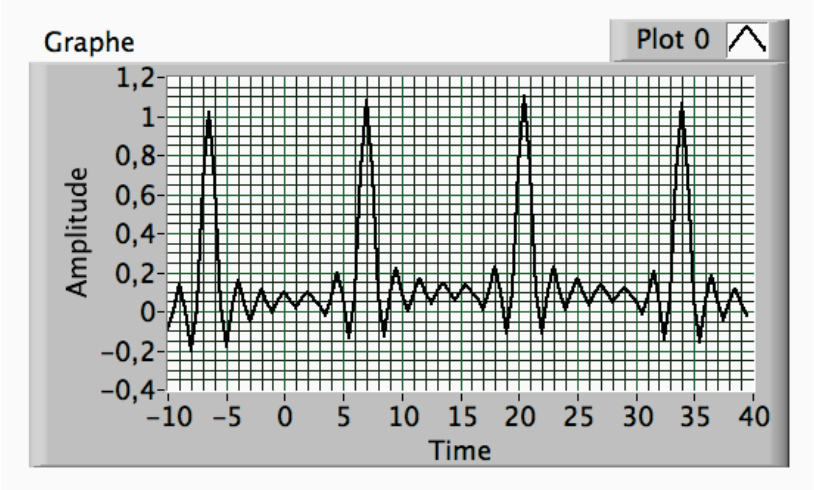


(a) Face avant.

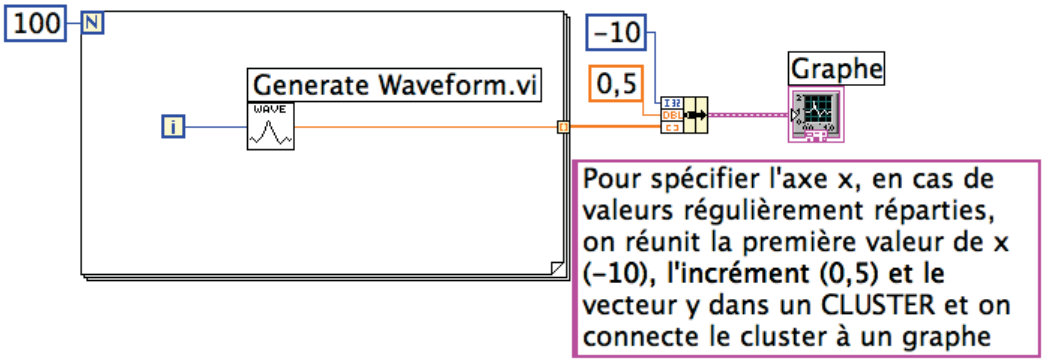


(b) Diagramme.

Figure 29 – Exemple de réalisation de graphes Y, XY et déroulant.



(a) Face avant.



(b) Diagramme.

Figure 30 – Exemple de réalisation de graphes $y = y(x)$ par cluster.

4.12 Exercices

1. Réaliser les exemples de la figure 29 et 30 et analyser leur fonctionnement
2. Représenter la fonction $y = 1/(1 - x^2)$ pour x variant de -20 à 20 par pas de 0,1 de deux méthodes différentes :
   - en calculant les deux vecteurs y et x et en utilisant un graphe XY
   - en calculant y uniquement et en utilisant un graphe

4.13 Transformée de Fourier

La transformée de Fourier se trouve dans le menu Traitement du signal > Transformée de la palette de Fonctions. N'oubliez pas que la transformée de Fourier d'un signal donne des coefficients complexes. Pour obtenir le module et la phase de ces coefficients, vous devez utiliser la fonction qui extrait ces valeurs d'un nombre complexe, située dans le menu Numérique de la palette Fonctions.

Il existe également plusieurs fonctions dans le menu Traitement du signal > Analyse Spectrale de la palette de Fonctions qui permettent l'analyse spectrale de signaux (FFT de signaux réels, FFT de signaux complexes, FFT power spectrum, ...).

4.14 Exercices

Soit la fonction :

$$f(t) = f_1(t) + f_2(t), \quad \text{avec } f_1(t) = \sin\left(2\pi\nu_1 t + \frac{\pi}{2}\right) \quad \text{et } f_2(t) = \sin(2\pi\nu_2 t)$$

1. Calculer, en utilisant une boucle FOR, le tableau des valeurs de t, $f_1(t)$, $f_2(t)$, et $f(t)$ pour t variant de 0 à 100 secondes par pas de 0,01 seconde. On impose les contraintes suivantes :
   - une commande pour chaque fréquence ν_1 et ν_2
   - une commande permettant de fixer le pas de temps, dt
   - une commande fixant le temps d'acquisition T_{acq}
   - un indicateur donnant la valeur de la fréquence d'échantillonnage f_e
2. Tracer les graphes suivants :
   - graphes superposés de f, f_1, et f_2 en fonction de l'indice
   - graphe de $f(t)$, **en utilisant le tableau des temps** t
   - *(Facultatif)* f_1 en fonction de f_2 (figure dite de Lissajoux, utilisée pour déterminer le rapport de fréquence)
3. Réaliser l'analyse spectrale de f : spectre de phase, d'amplitude et de puissance en fonction de la fréquence
4. Faire apparaître, en cliquant sur le menu local du graphe, la palette graphe et la palette curseurs. Utiliser ces outils pour caractériser les structures : domaine affiché (dépend de quoi ? — le vérifier) ; symétries (par rapport à quoi ? valeur ?) ; position des raies (dues à quoi ?)
5. Que se passe-t-il si f_e diminue (à T_{acq} constant = 100) ? Comparer chaque fois fréquence signal réel et positions en fréquence des raies. Conclusion ?

4.15 Autocorrélation et mesure d'un retard

L'autocorrélation d'un signal continu est donnée par :

$$C_{xx}(\tau) = \frac{1}{T}\int_0^T x(t)x(t-\tau)dt$$

Cette fonction a des propriétés que nous rappellerons sans les démontrer :

— fonction paire
— fonction maximale pour un décalage nul ($t = 0$)
— $\lim_{t\to\infty} C_{xx} = 0$
— si le signal x est à moyenne nulle ($\langle x\rangle = 0$), alors $\langle C_{xx}\rangle = 0$

Un exemple typique est donné par la figure 31.

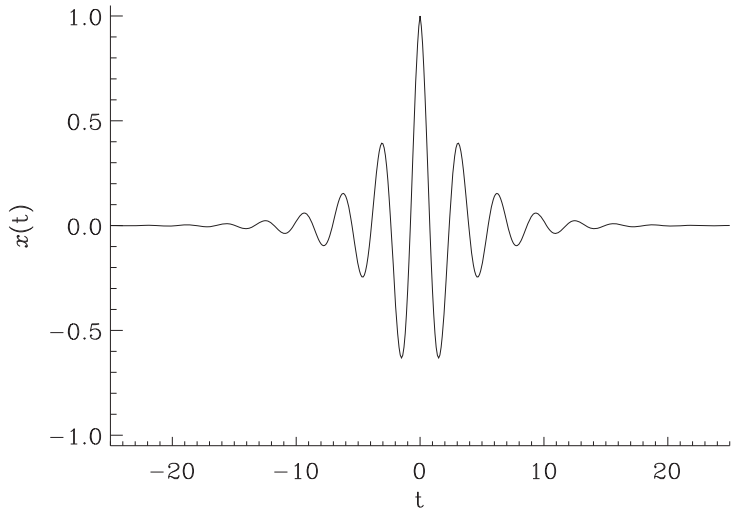


Figure 31 –

Envisageons maintenant le cas d'un signal $x(t)$ et une version de ce signal retardé de t_0 et multipliée par une constante k. On nomme ce nouveau signal $y(t)$: $y(t) = k \times x(t + t_0)$
On peut former une nouvelle fonction, la fonction d'intercorrélation entre x et y qui s'écrit :

$$C_{xy}(\tau) = \frac{1}{T}\int_0^T x(t)y(t-\tau)dt = \frac{k}{T}\int_0^T x(t)x(t-\tau+t_0)dt = kC_{xx}(\tau'), \quad \text{avec } \tau' = \tau - t_0$$

L'autocorrélation C_{xx} est maximale (et donc l'intercorrélation C_{xy} l'est également) pour $\tau' = 0$, càd lorsque $\tau = t_0$. La représentation graphique de C_{xy} est similaire à celle de C_{xx}, mais décalée le long de l'axe horizontal. Cette propriété peut nous fournir nune méthode pour mesurer le décalage de deux signaux similaires et qui présentent un retard :

1. on calcule la fonction d'intercorrélation C_{xy} de ces deux signaux
2. on cherche opur quel décalage cette fonction C_{xy} est maximale
3. le retard cherché est égal à la valeur de ce décalage

4.16 Exercices

Soit la fonction :

$$x(t) = \mathrm{e}^{-t^2} \times \cos(2t)$$

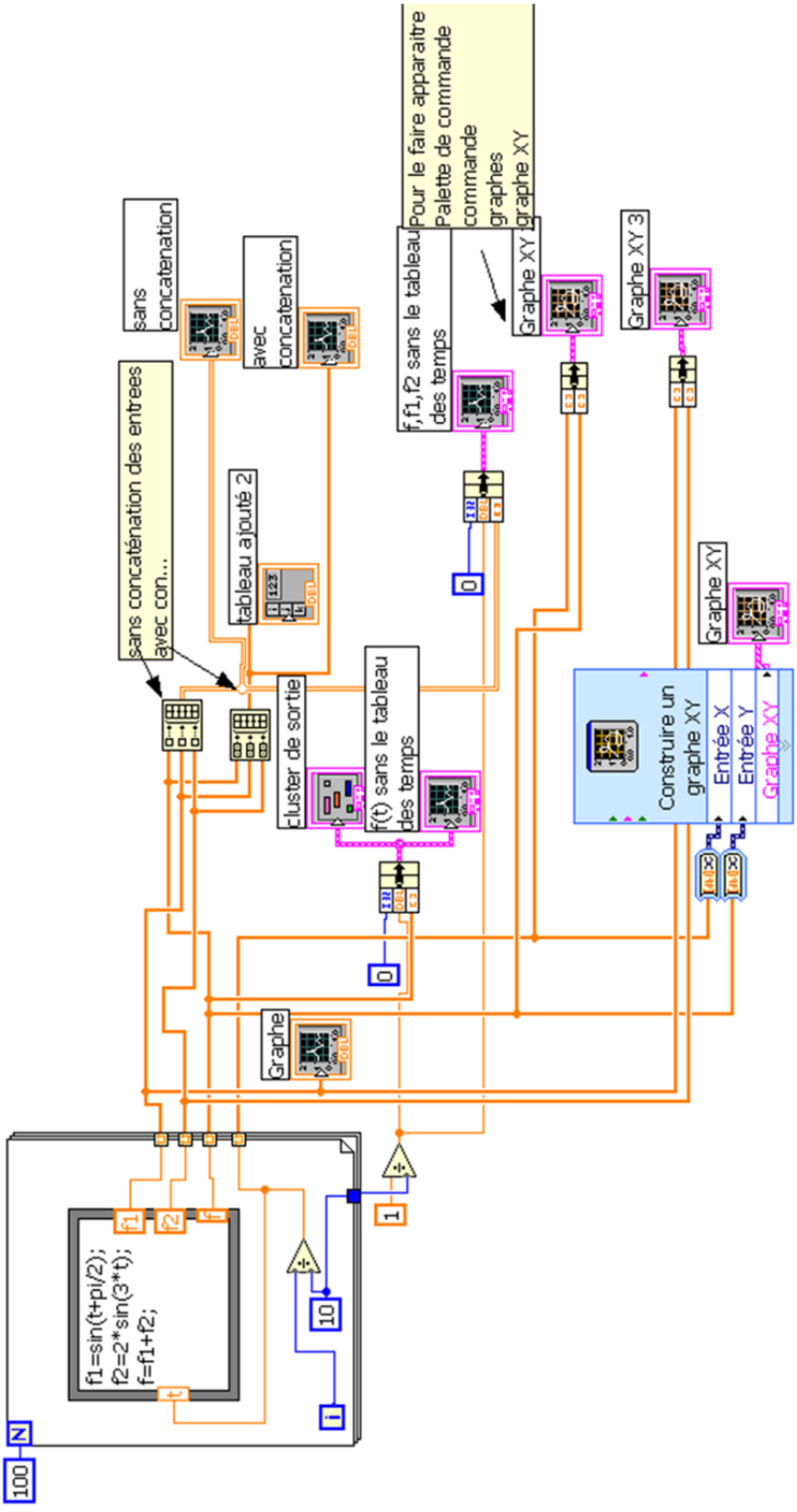
- Représenter $x(t)$ en échantillonnant cette fonction sur 5000 points entre $t = -10$ et $t = +10$
- Représenter $y(t) = x(t + 1.5)/2.0$
- Calculer l'intercorrélation de ces deux signaux (après soustraction de leur moyenne)
- Retrouver la valeur du décalage (1.5 s) en utilisant cette intercorrélation

L'intercorrélation se trouve dans Fonctions > Traitement du signal > Opérations sur signaux. Bien lire l'aide en ligne de cette fonction.

TP1 Initiation Instrumentation Labview M1 SIA CESE ISTR ESET 14H

EXEMPLES DE GRAPHES

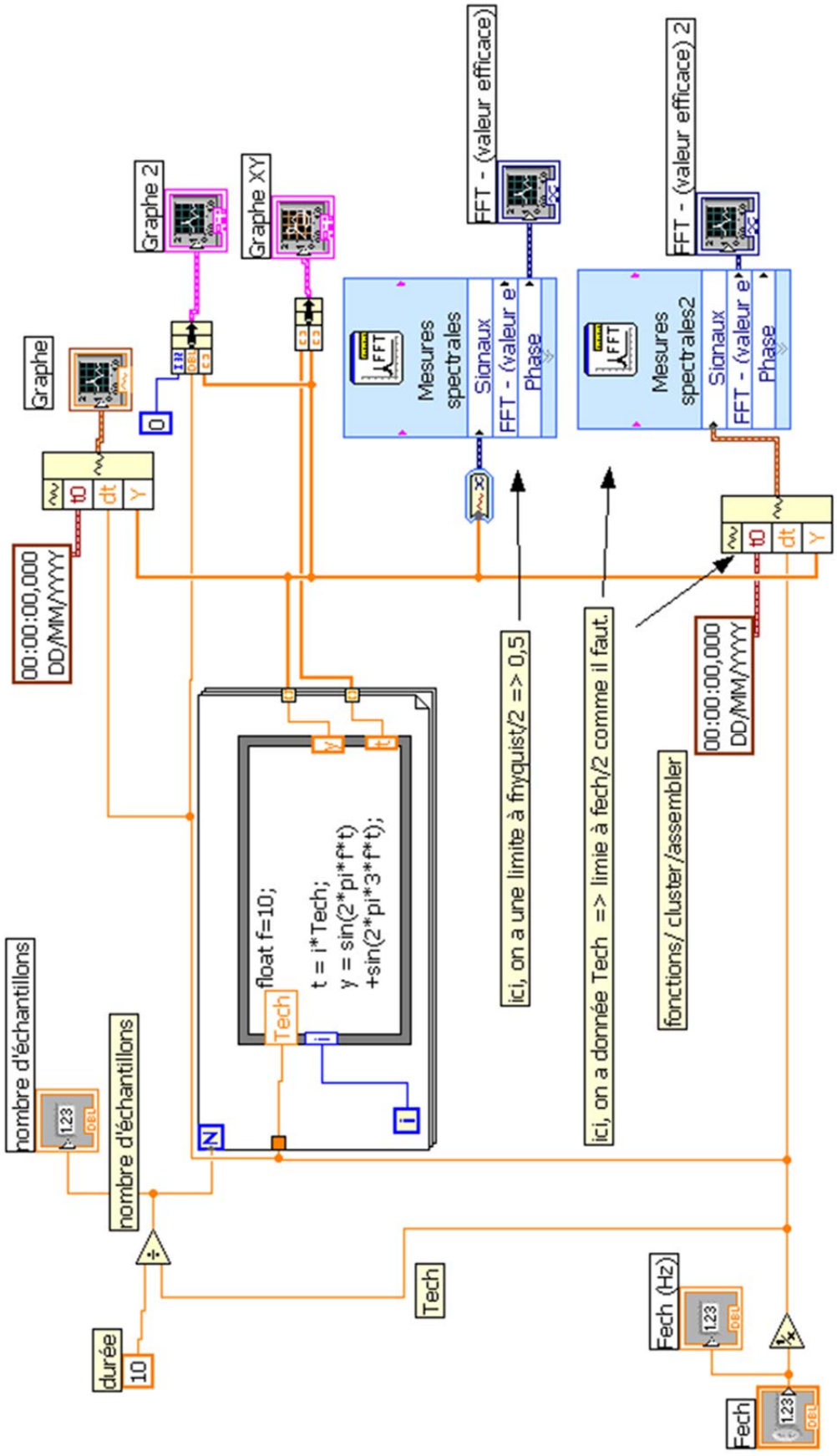
graphes en fonction de l'indice des tableaux x(n) x1(n), x2(n) sur le même graphe	graphes en fonction du temps x(t) x1(t), x2(t) sur le même graphe



Boitier V. vincent.boitier@univ-tlse3.fr Perisse T. thierry.perisse@univ-tlse3.fr Dedieu J. joel.dedieu@univ-tlse3.fr Renaud C. cedric.renaud@laplace.univ-tlse.fr
Site Web : http://thierryperisse.free.fr/

TP1 Initiation Instrumentation Labview M1 SIA CESE ISTR ESET 14H

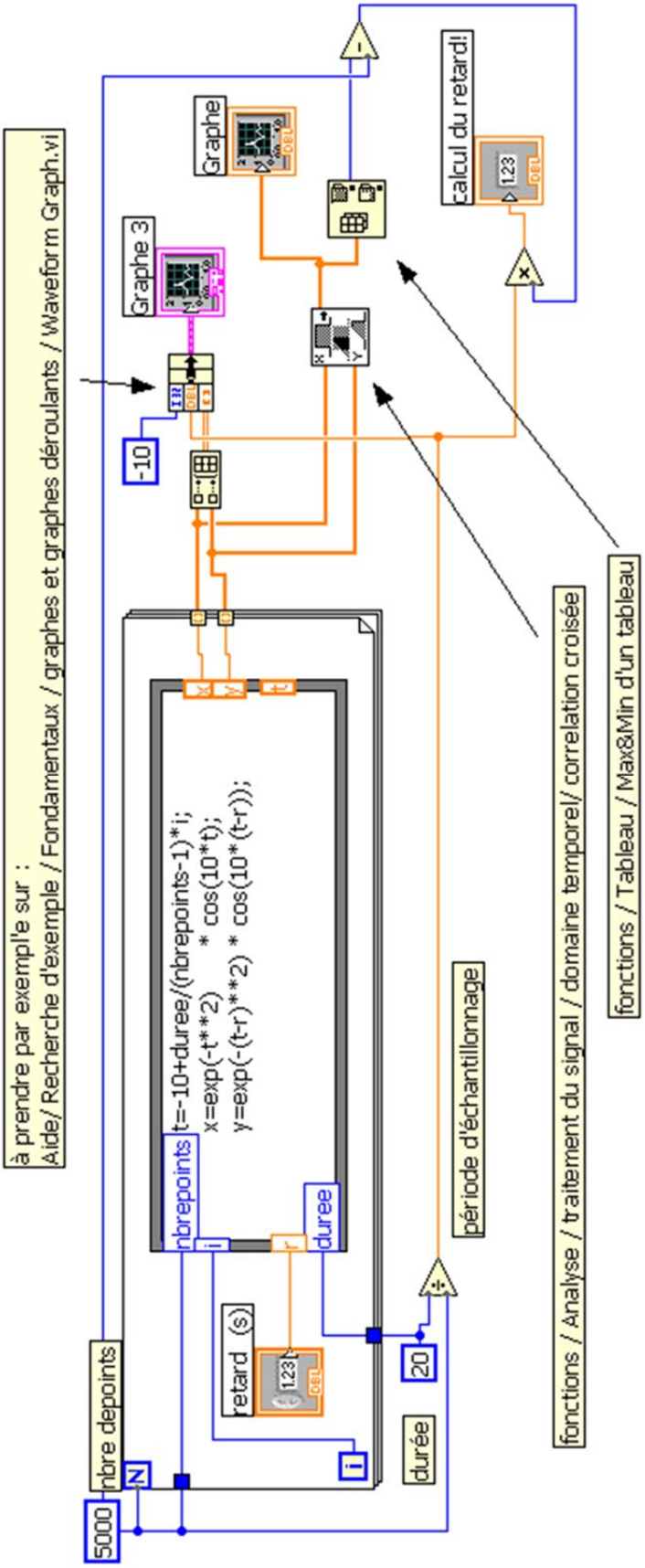
FFT



Boitier V. vincent.boitier@univ-tlse3.fr Perisse T. thierry.perisse@univ-tlse3.fr Dedieu J. joel.dedieu@univ-tlse3.fr Renaud C. cedric.renaud@laplace.univ-tlse.fr
Site Web : http://thierryperisse.free.fr/

TP1 Initiation Instrumentation Labview M1 SIA CESE ISTR ESET 14H

AUTO CORRELATION.



Boitier V. vincent.boitier@univ-tlse3.fr Perisse T. thierry.perisse@univ-tlse3.fr Dedieu J. joel.dedieu@univ-tlse3.fr Renaud C. cedric.renaud@laplace.univ-tlse.fr
Site Web : http://thierryperisse.free.fr/

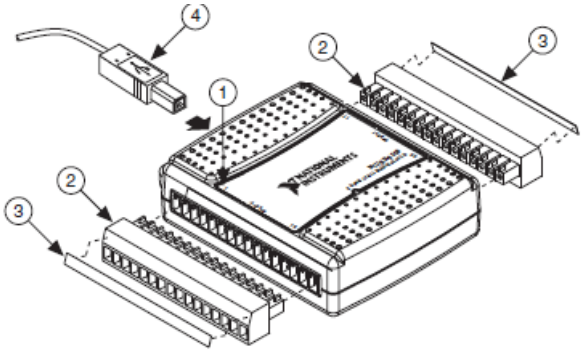
TP2 Labview - Utilisation des Carte E/S USB-6008/6009

Voir aussi doc carte USB 6008/6009 pour les spécifications techniques.

Objectifs du TP :

Utiliser les cartes d'acquisitions et de commande avec Labview. Maitriser les paramètres indispensables lorsqu'on souhaite faire des acquisitions.

Il est demandé **impérativement** de faire un schéma **préalable** des connexions que vous souhaitez réaliser avant de réaliser ces dites connexions.



1. Identifier le matériel et tests de base sous MAX

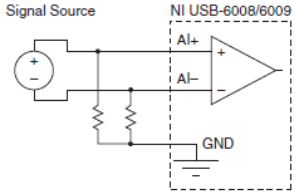
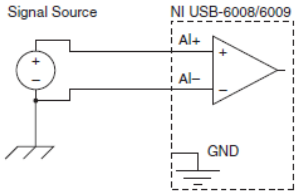
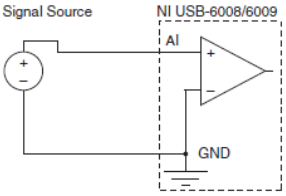
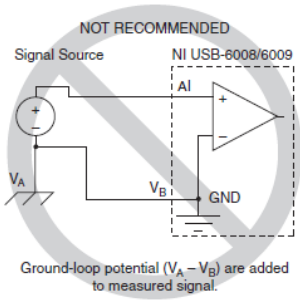
sous **NI** Double-cliquer sur **Périphériques et Interfaces** pour explorer l'arborescence de ce menu du logiciel MAX : vérifier que la carte NI USB-6008 est bien reconnue par le système d'exploitation. Pour cela, cliquer sur le nom de la carte puis sur "Auto-test" ; MAX doit indiquer
"L'auto-test a réussi" Aller dans *Panneaux de tests*

Test **générer un signal** depuis la carte d'acquisition *Câblage* USB 6008 → Oscillo *Connections* AO 1 *réglage oscillo* : main delayed /Roll calibre : 5V/carreau	Test **acquérir un signal** sur la carte d'acquisition *Câblage* GBF (continu, 1Vpp) → Oscillo mode roll → USB 6008 *Connections :* AI 0 configuration RSE.

N.B. : Lors de la mise sous tension du GBF, celui-ci considère par défaut que les signaux sont débités sur une charge d'impédance caractéristique de 50Ω. Ce n'est pas le cas dans le cadre de ce TP. Le GBF affiche alors le double de la valeur réelle de l'amplitude et de l'offset. Vous pouvez modifier le mode de fonctionnement du GBF, en lui indiquant que vous travaillez sur des impédances élevées (voir le mémo sur votre paillasse).

“Floating source” ou “ground referenced source”

Avec les cartes **NI6008/6009**, les cas d’utilisation des différents modes de configuration d’entrée : **DIFF** pour différentielle ou **RSE** (“Referenced Single-Ended”) sont résumés dans le tableau ci-dessous.

Analog Input Mode	Floating Signal Sources (Not Connected to Building Ground) Examples: • Ungrounded thermocouples • Signal conditioning with isolated outputs • Battery devices	Ground-Referenced Signal Sources Example: • Plug-in instruments with non-isolated outputs
Differential (DIFF)		
Referenced Single-Ended (RSE)		

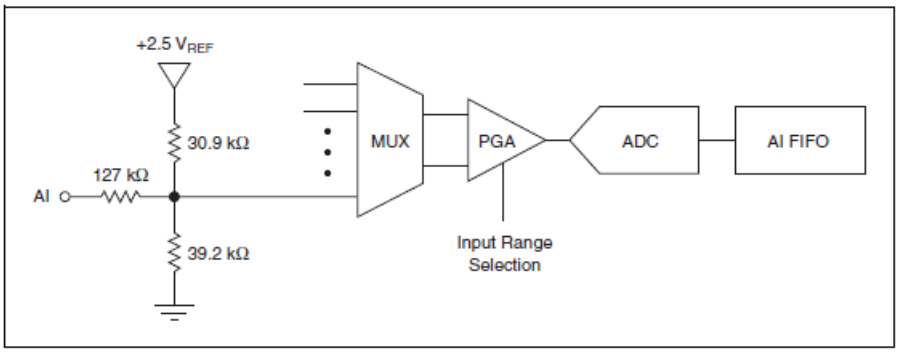


Figure 7. NI USB-6008/6009 Analog Input Circuitry

Fermer le logiciel Max.

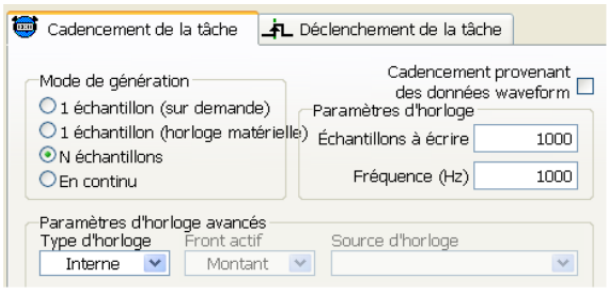
2. Programmation d'une application simple sous Labview

a. Générer un signal depuis la carte d'acquisition

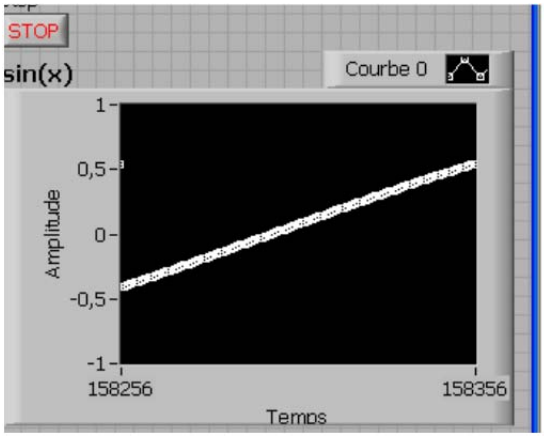
câblage :	*Programmation* :
USB 6008 AO1 Oscillo voie 1 Oscillo en mode roll	Labview / nouveau / vi vide puis dans le diagramme : clic droit / fonctions / express / entree /assistant DAQ Réaliser un VI qui permet de générer un ***signal continu variable***. Réaliser un VI qui permet de générer un sinus : Pour ceci aller dans les *express* récupérer le vi *simuler* pour créer un signal, le programmer pour générer un sinus de fréquence **1 Hz** fait avec **1000pts** à **1000Hz** (on a alors une période complète).

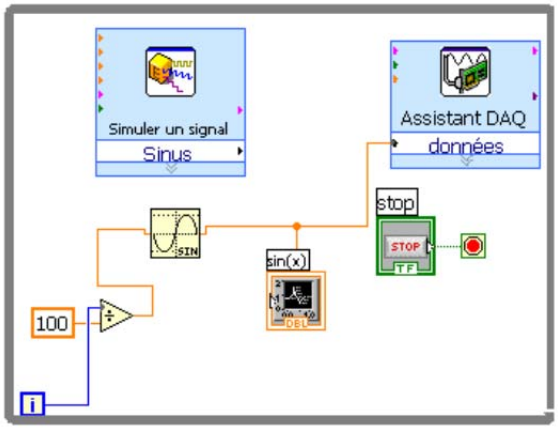
Dans l'assistant DAQ, on va jouer sur les paramètres suivants :

Choix de la voie de sortie :
Vérifier que la voie 1 a bien été choisie (cliquer sur le double chevron bleu)
Mode de génération :

Cocher *: cadencement provenant des données waveform* Tester *N échantillons* Décocher : *cadencement provenant des données waveform.* Tester avec les paramètres suivants : Mode de génération : *En continu.* Fréquence : quelle est la f max possible? Mode de génération : *N échantillons.* N= 1000 N= 200	

On essaye maintenant: *1 échantillon sur demande* … pour cela il faut créer le sinus (cf figure ci-dessous) ça fonctionne mais attention, ce n'est pas du tps réel, il y a parfois des tâches plus prioritaires (irrégularités sur la courbe vue à l'oscillo)!





b. Acquérir un signal sur la carte d'acquisition.

GBF → Oscillo mode main voie1 → USB 6008 AI0 Paramétrer le GBF pour générer un triangle à 1 kHz 4Vpp *mode différentielle*	En sortie de l'assistant DAQ, on câble un graphe (on fera des zooms sur ce graphe). Dans l'assistant DAQ, pour retrouver les performances de la carte données dans la doc, on va jouer sur les paramètres suivants:

Tension d'entrée / paramètres :
Que signifie *échantillons à lire* et *fréquence*?
Quelles sont les valeurs max pour ces deux paramètres ? (on ne lit qu'une seule voie ici!!)
Gamme du signal d'entrée.
Plages d'entrée / quantification
Plages d'entrée :
Quelles plages d'entrées sont proposées dans la doc ?
Pour une plage d'entrée de ±10V puis de ±1V visualiser le triangle.
Mesure du quantum en fonction de la plage d'entrée.
Mettre le GBF en continu sortie 0V. Le bruit qui se rajoute donne assez de variabilité pour mesurer la quantification. (si le blindage est bien réalisé, il n'y aura pas assez de bruit ⇒ on rajoute dans ce cas une toute petite composante sinusoïdale)

Plage d'entrée	±10V	±2V	±1V
Quantum			

L'erreur de quantification est-elle la seule à prendre en compte pour déterminer l'incertitude liée à la carte. Comment doit-on choisir la plage d'entrée?

Revenir à une gamme +10 -10 V pour la suite
Acquisition multiples.
Rajouter une voie de mesure et envoyer le même signal triangulaire (1 kHz 4Vpp) sur les voies 0 et 1. L'acquisition est-elle simultanée?
Rajouter ensuite 1 voie de mesures (il y en a alors 3) et envoyer le même signal sur la 0 et la 1 puis sur la 0 et la 2.
L'acquisition est-elle simultanée? Valeur du décalage temporel?
Faire un VI qui récupère la voie 0 1 et 2 et qui permet de ressortir ces 3 voies séparément.
Express / manipulations /

Cadencement de la tâche

câblage :	Programmation :
On garde le câblage précédent et on rajoute : Sortie SYNC du GBF → Oscillo voie 2 → USB 6008 PFI0 (ou une entrée PFI) : le signal doit avoir un niveau TTL (« 0 »<0,8V, « 1 » entre 2,2 et 5V.	Dans l'assistant DAQ, utiliser l'onglet *déclenchement* *Front numérique sur PFI0*

On doit alors récupérer la même chose sur le scope et sur le graphe.

Pour comprendre l'importance du choix de la fréquence d'échantillonnage, nous allons regarder simultanément le signal temporel et sa représentation fréquentielle. Pour cela on prendra :
carte acquisition : fech = 10kHz et 10k points (il faut assez de points pour une « belle » FFT, c'est-à-dire des raies fines pour des signaux sinusoïdaux)
GBF : Vin start = sinus à 1kHz 1Vpp puis faire varier la fréquence du GBF par pas de 1kHz

Faire la FFT des signaux (*fonctions / express / analyse / mesures spectrales /)*
conclure sur le choix de Fech par rapport à la fréquence max du signal pour reconstituer correctement le signal et pour retrouver la fréquence du GBF.

Il reste du temps, je prépare la suite :

Exciter l'émetteur US via le GBF avec un sinus de fréquence variable et récupérer sur l'oscillo le signal du GBF et le signal émis par le récepteur.
L'ensemble se comporte comme un filtre de quel type ? Comment le caractériser simplement.