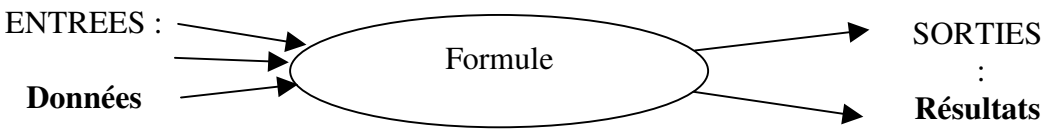
Version 2013



COURS LabVIEW

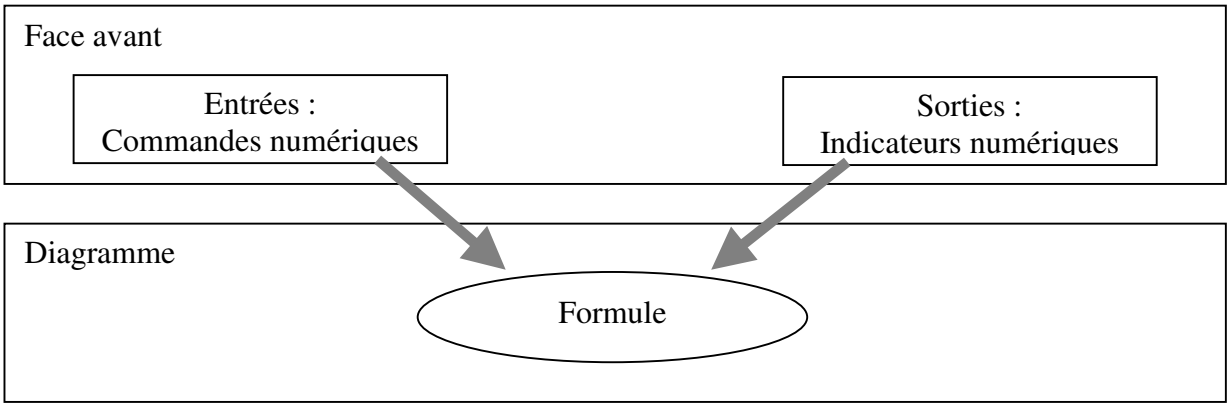
I – INTRODUCTION

Un calcul utilise des données pour fournir un résultat à partir d'une formule.

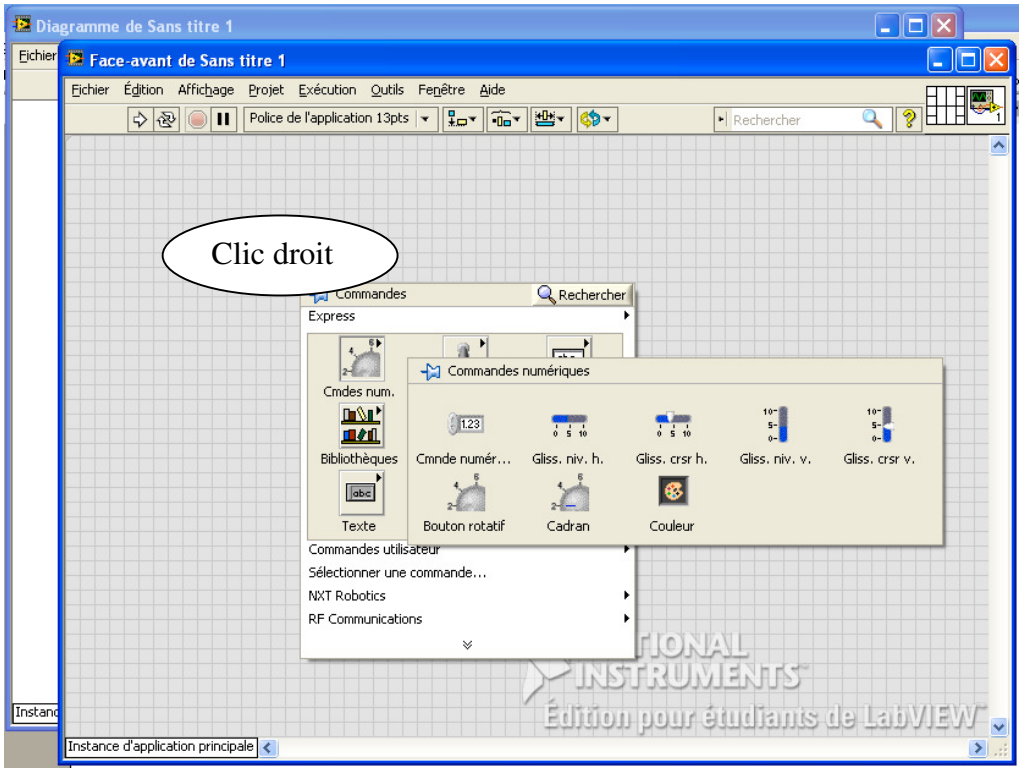


II – ENTREES SORTIES DANS LabVIEW

Les variables d'entrées/sorties sont déposées sur la face avant du VI.
La formule est construite dans le diagramme.



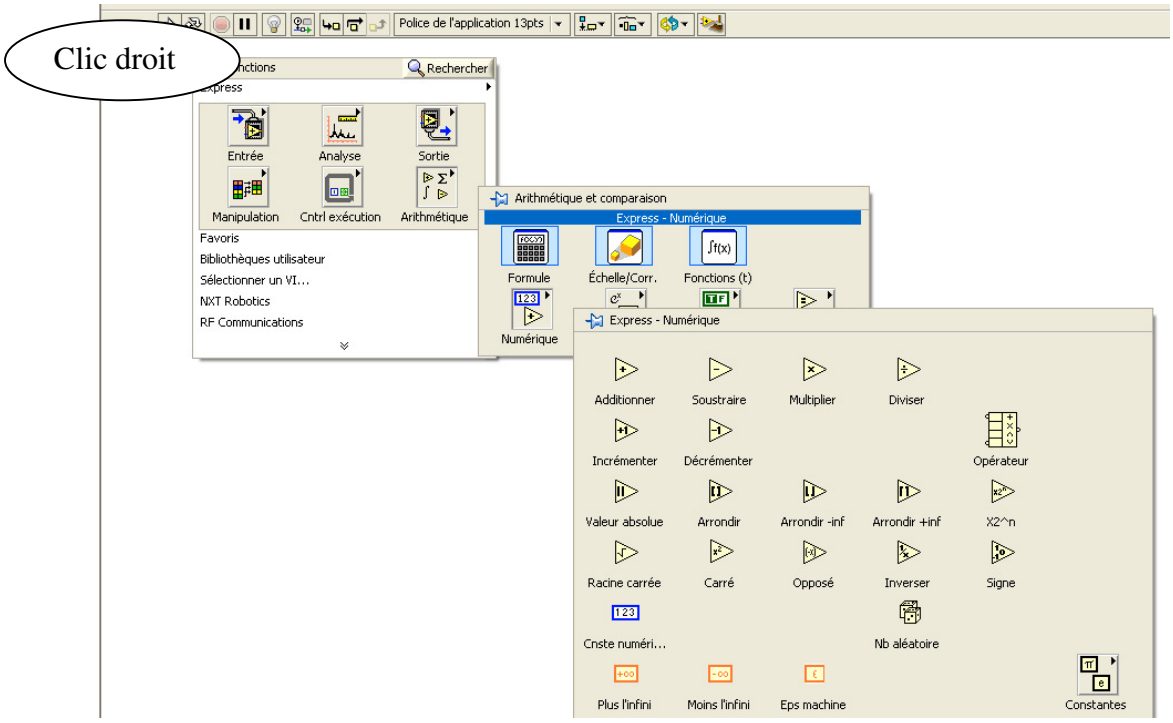
Les commandes et indicateurs numériques peuvent prendre différentes formes à sélectionner dans la palette visible par clic droit dans la fenêtre face avant.



III – FORMULE

1°/ OPERATEURS

La formule peut être écrite dans le diagramme à l'aide d'opérateurs disponibles dans la palette de fonctions visible par clic droit dans le diagramme.



EXERCICE 1-1

Réaliser un VI qui affiche la tension U obtenue à l'aide d'un pont diviseur.

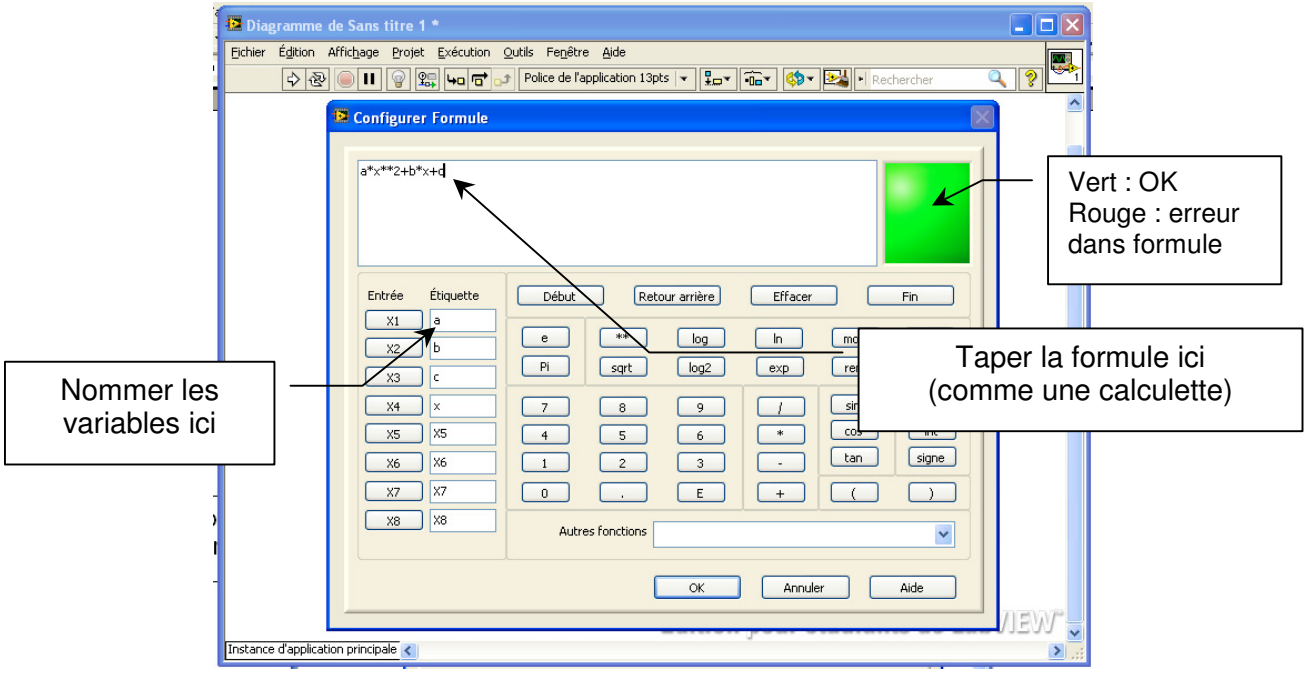
E, R1 et R2 sont des commandes numériques sur la face avant.

Rappel formule : U = R2 E / (R1 + R2)

2°/ BOITE DE FORMULE

La formule est écrite dans une boîte située dans la palette de fonctions/arithmétique/formule (voir figure § précédent.

Dans la boîte, il convient de nommer les variables en utilisant les mêmes noms que les données du programme.

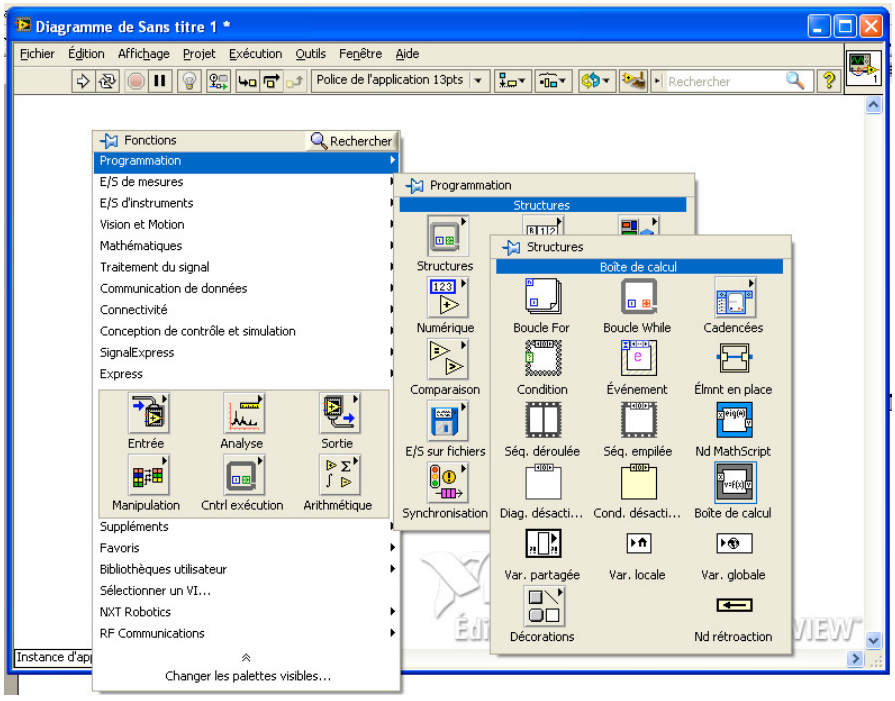


EXERCICE 2-1

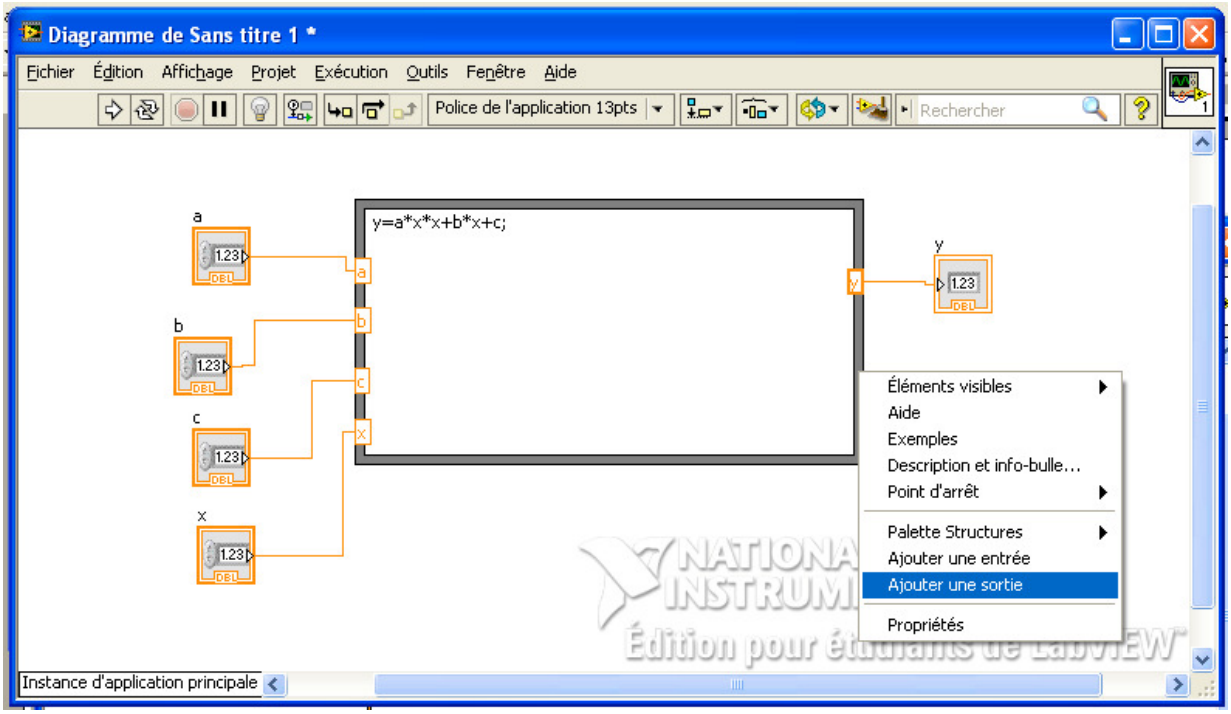
Reprendre l'exercice 1 en utilisant la boite de formule.

3°/ BOITE DE CALCUL

La boîte de formule ne peut réaliser qu'un calcul.
La boite de calcul, moins conviviale permet de réaliser plusieurs calculs.
On la trouve dans la palette de fonctions/programmation/structures/boîte de calcul



Exemple :



Utilisation :

On ajoute les entrées/sorties par clic droit sur le bord du cadre de la boîte.
Chaque formule est du type y = …… où y est une sortie

La variable y peut être réutilisée comme entrée dans la ligne suivante.

Terminer la ligne par un **;**

EXERCICE 3-1

Reprendre l'exercice 1 en utilisant la boite de calcul.

EXERCICE 4-1

La luminance par unité de longueur d'onde d'un corps noir à la température T est donnée par la loi de Planck :

$L(\lambda) = 2\,h\,c^2 / [\,\lambda^5\,(\exp(h\,c / \lambda\,k\,T) - 1\,)\,]$ **$h = 6{,}62.\,10^{-34}$ J/s** **$k = 1{,}38.\,10^{-23}$ J/K**
$c = 3.\,10^8$ m/s

T est une commande sur la face avant initialisée à 5770 K

λ est une commande en face avant comprise entre 0 et 2 µm, Afficher la valeur de la luminance sur un indicateur numérique.

EXERCICE 5-1

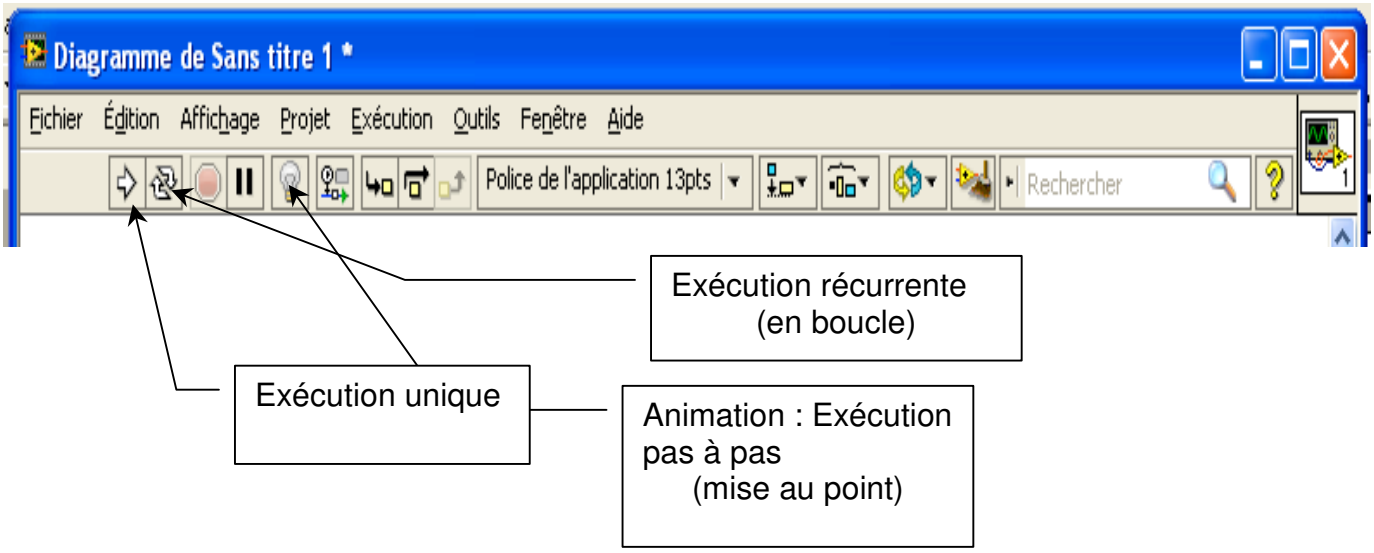
Ecrire un programme permettant d'afficher sur un terminal numérique la valeur du gain donné par un circuit RC :

$$G(f) = 20 \log [1 / \sqrt{ (1 + (RC.2\pi f)^2) }]$$

R, C et f sont modifiables en face avant grâce à des commandes numériques.

On prendra comme valeur test R = 4700 Ω, C = 10 nF et f = 5 kHz.

IV – MODES D'EXECUTION DU PROGRAMME



EXERCICE 6-1

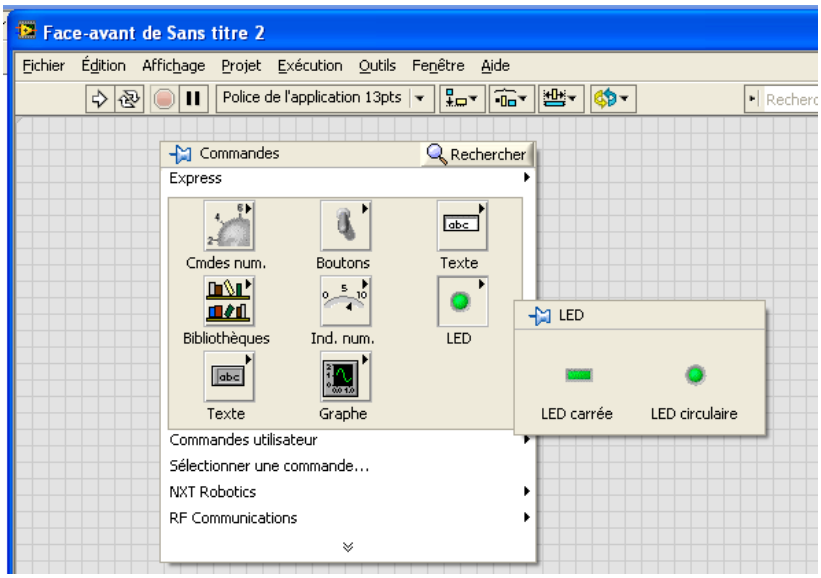
Modifier le programme précédent pour qu'il se répète jusqu'à l'appui sur un bouton stop lors du lancement par une exécution unique.

Chapitre 2 - BOOLEENS, OPERATEURS DE COMPARAISON STRUCTURE DE CHOIX

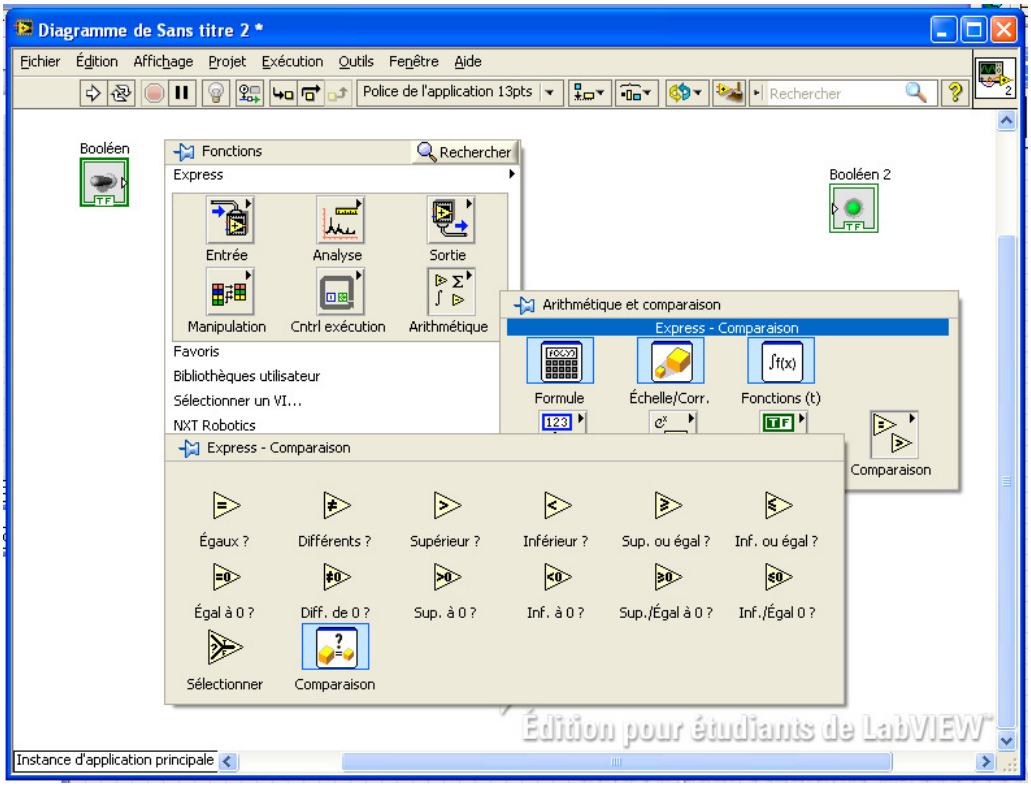
I – VARIABLE BOOLEENNE

Une variable booléenne est une variable pouvant prendre deux valeurs : Vrai (True : T) ou Faux (False : F).

Exemple : Interrupteur => Variable d'entrée — Commande booléenne
Led => Variable de sortie — Indicateur booléen



II – OPERATEURS DE COMPARAISON



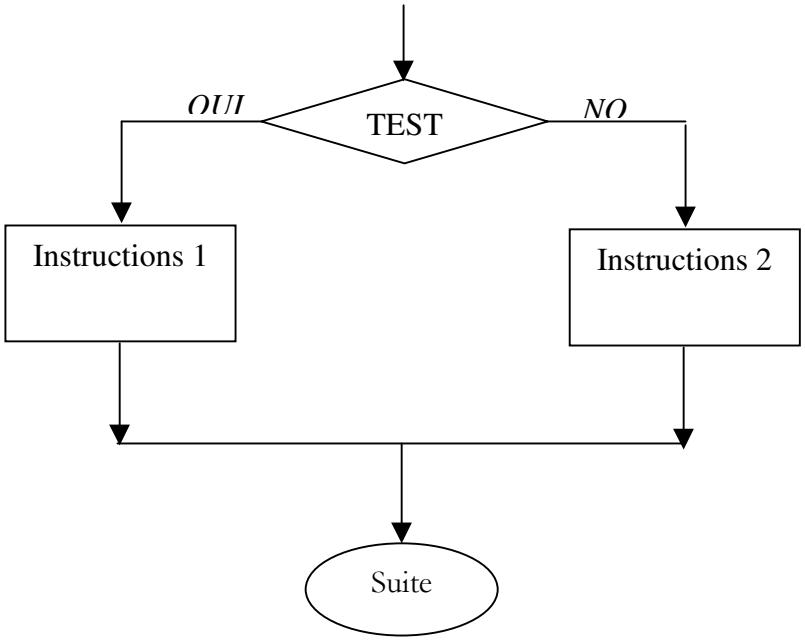
Les opérateurs de comparaison mettent en relation des variables d'entrées numériques et une sortie booléenne.

Exemple : a>b ? réponse T ou F



III– STRUCTURE CONDITION

Suivant le résultat d'un test, on réalise une série d'instructions ou une autre



Le test est la relation permettant de réaliser l'aiguillage.

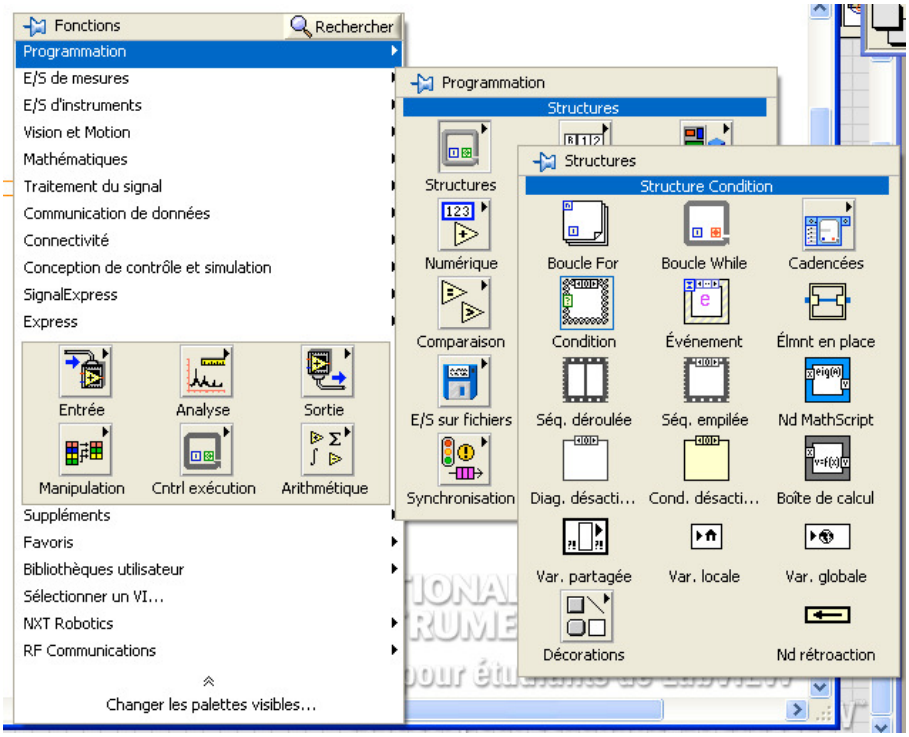
Le test met en œuvre des opérateurs de comparaison <, >, ≤, ≥, =, ≠.
Il peut être aussi l'évaluation d'une simple variable booléenne (test de l'appui sur un bouton poussoir)

Le résultat du test est un booléen : il ne peut prendre que 2 valeurs : OUI ou NON.

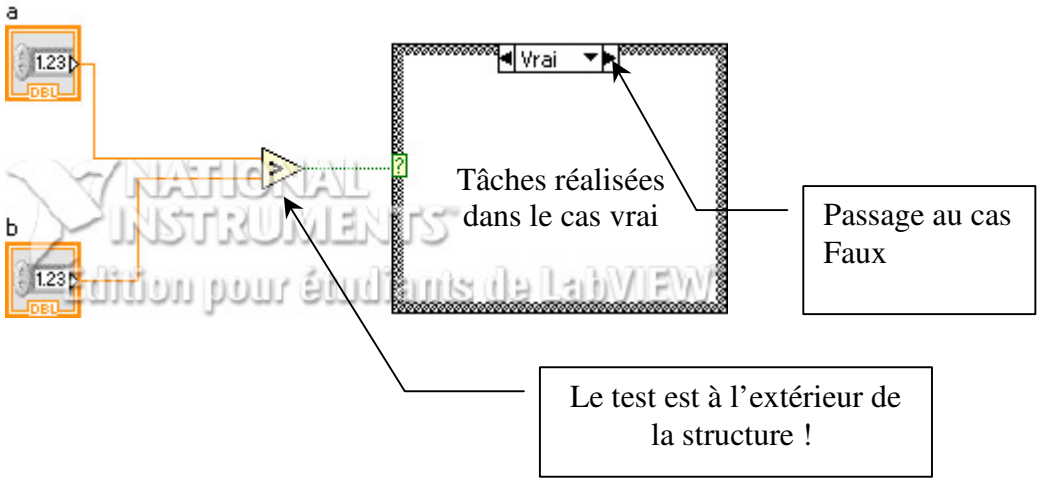
Le test peut être simple du type a < b ou plus complexe : (a < b) ET (a < c)

Les booléens issus des relations de comparaisons sont alors reliés entre eux par des opérateurs logiques : ET, OU, NON ET, NON OU, OU Exclusif, etc …

Dans le diagramme, la structure condition se trouve dans la palette de foncions/programmation/structures/condition



Exemple :



EXERCICE 1-2

La tension d'une batterie 12 V est simulée par un potentiomètre U à glissière en face avant. Le résultat du test de la batterie est affiché sur 3 leds différentes.

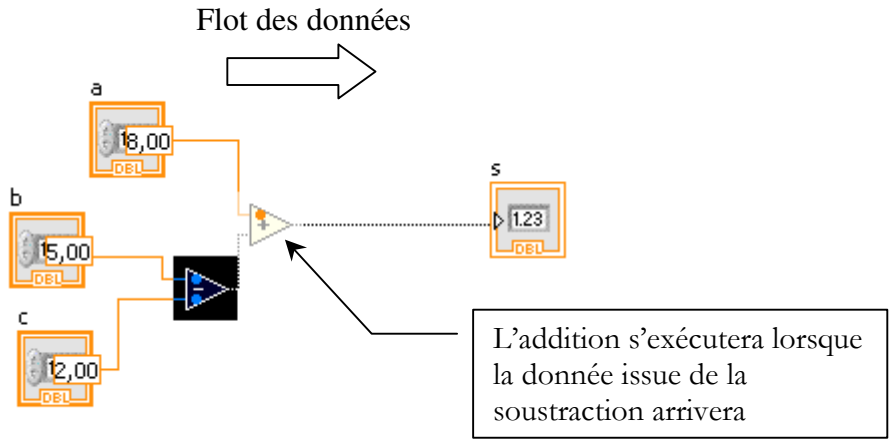
Si 10 < U < 12 => Led Verte allumée
Si 8 < U < 10 => Led Orange allumée
Si 0 < U < 8 => Led Rouge allumée

Le programme s'arrêtera par appui sur bouton stop

I – ORDRE D'EXECUTION D'UN PROGRAMME

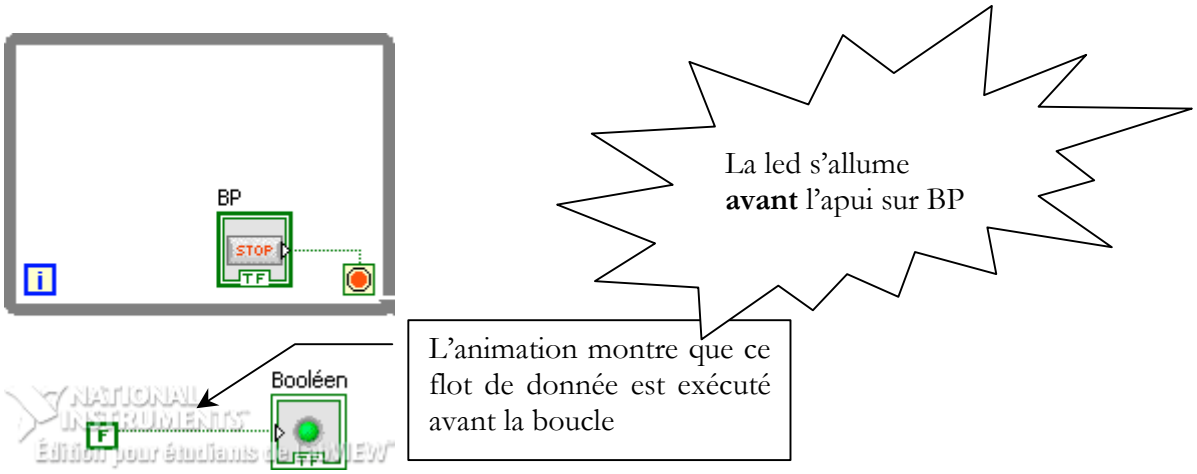
Un langage de programmation textuel classique s'exécute ligne par ligne et donc instruction après instruction.

Un VI labVIEW s'exécute en suivant le flot des données.



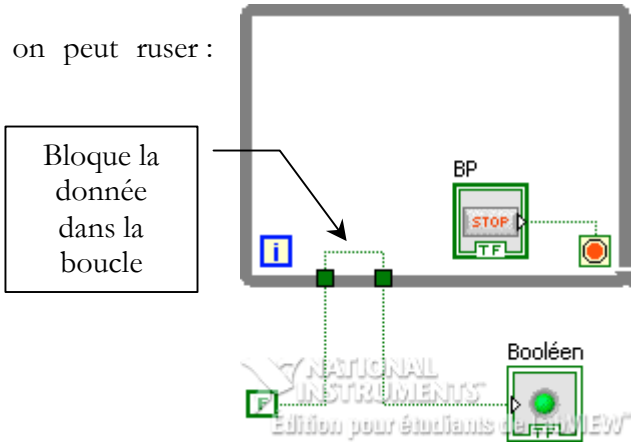
Le contrôle de l'ordre d'exécution du programme peut aussi nous échapper.

Exemple : Allumer une led **après** avoir appuyé sur un Bouton Poussoir :



Pour reprendre le contrôle de l'ordre de l'exécution séquentielle

on peut ruser :



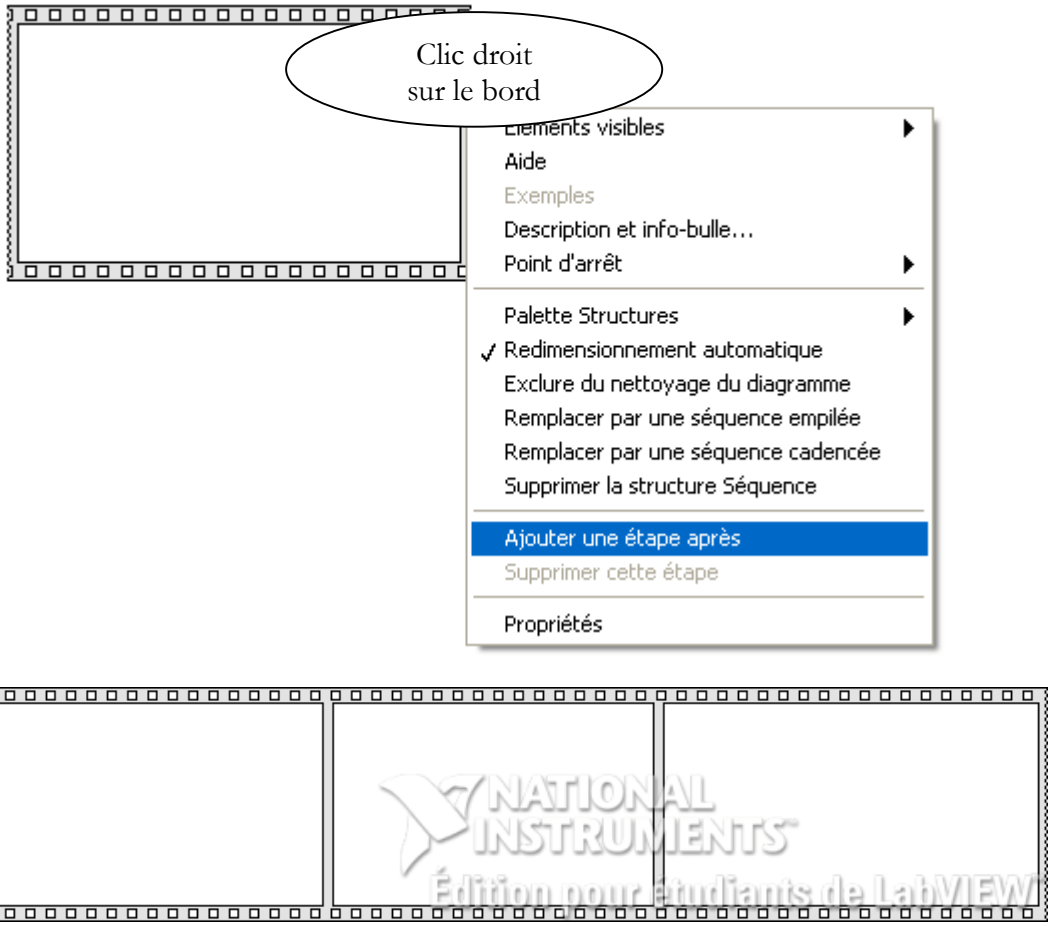
… ou utiliser la structure séquence :

Palette fonction/Express/Contrl exécution/Seq. déroulée

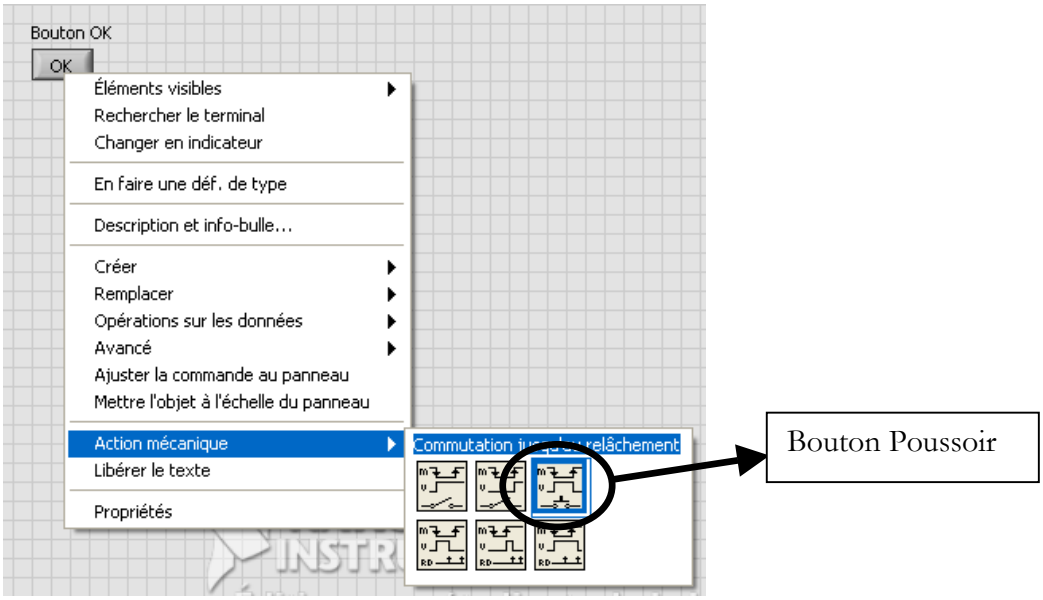
II - SEQUENCE

La structure séquence se trouve dans : Express/contrôle exécution / Seq déroulée
ou dans Palette fonction/programmation/structures/seq déroulée

L'ensemble des tâches devant être réalisées séquentiellement (les unes après les autres) sont placées dans des étapes.

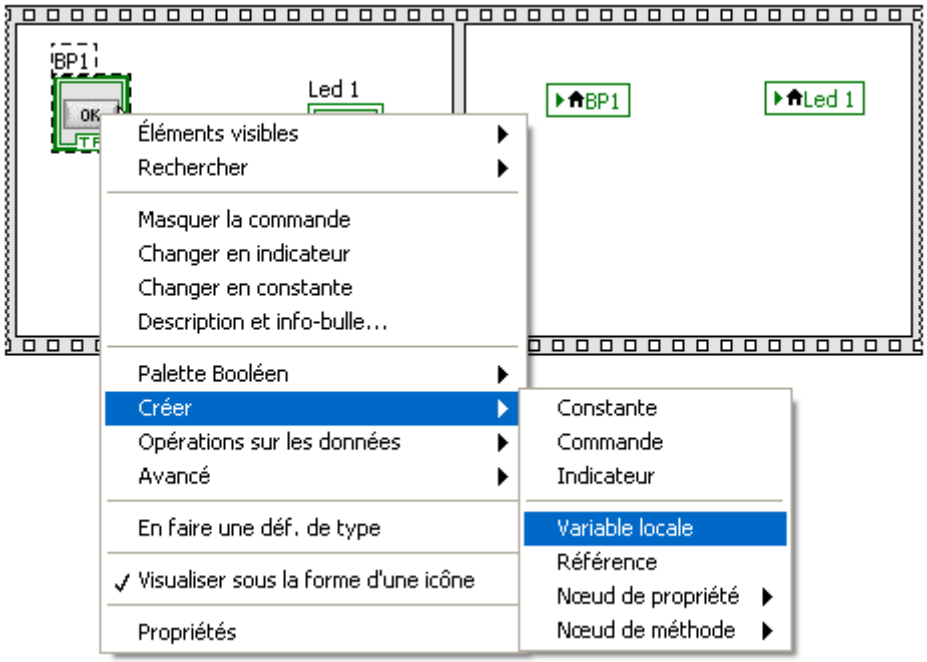


Dans les exercices suivants on utilise des boutons poussoirs et non des interrupteurs (pas d'accrochage). Un clic droit sur l'objet permet de changer l'action mécanique :

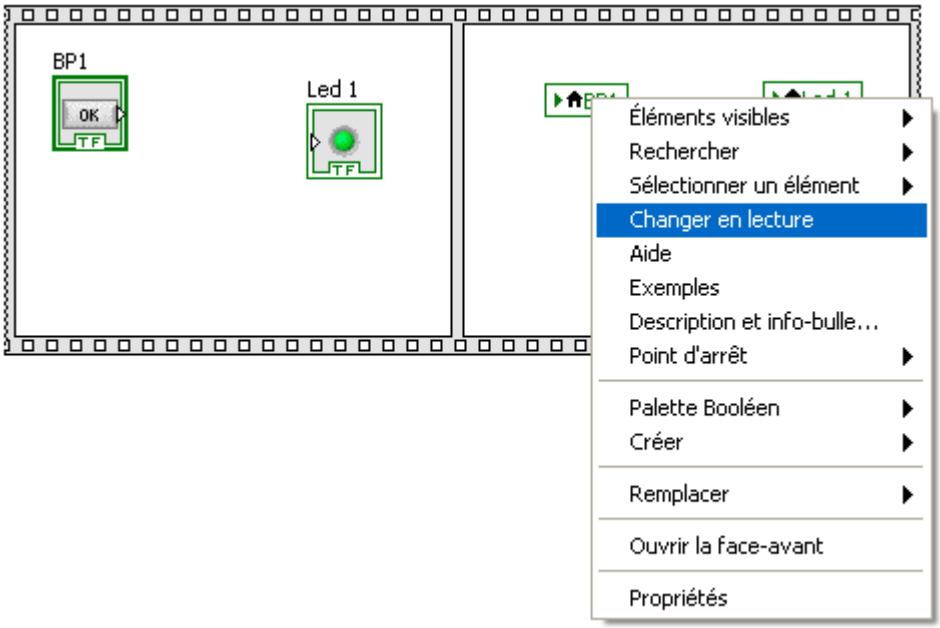


III – VARIABLE LOCALE

On peut avoir besoin dans les différentes étapes de la séquence de réutiliser une même variable.
On crée alors une variable locale



Pour un bouton poussoir la variable locale crée est en écriture, il convient de la remettre en lecture :

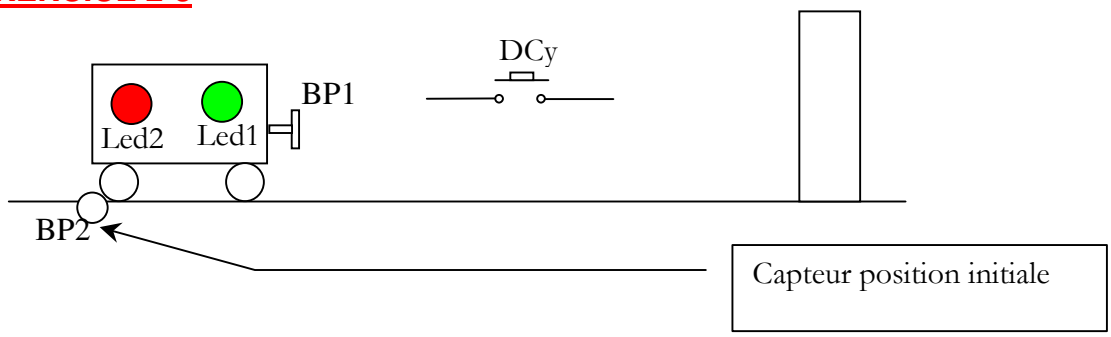


EXERCICE 1-3

Attendre l'appui sur BP pour allumer une led
Réaliser le VI correspondant aux deux versions données ci-dessus. Vérifier l'ordre d'exécution avec l'exécution animée.

Réaliser le VI en utilisant la structure séquence.

EXERCICE 2-3



La marche avant est simulée par l'allumage d'une led verte (led1) La marche arrière est simulée par l'allumage d'une led rouge (led2).
Le wagon étant en position initiale, l'appui sur le bouton DCy (départ cycle) lancera la marche avant.
Le contact avec le mur est simulé par appui sur un bouton poussoir BP1

Simuler le fonctionnement du chariot qui attend l'appui sur DCy pour avancer et qui doit faire marche arrière quand le bouton poussoir BP1 est enfoncé. Le retour en position initiale est détecté par le contact simulé par un bouton poussoir BP2.

Réaliser le vi en utilisant la structure séquence.

N. B : On ne lancera pas l'exécution du vi par l'exécution récurrente

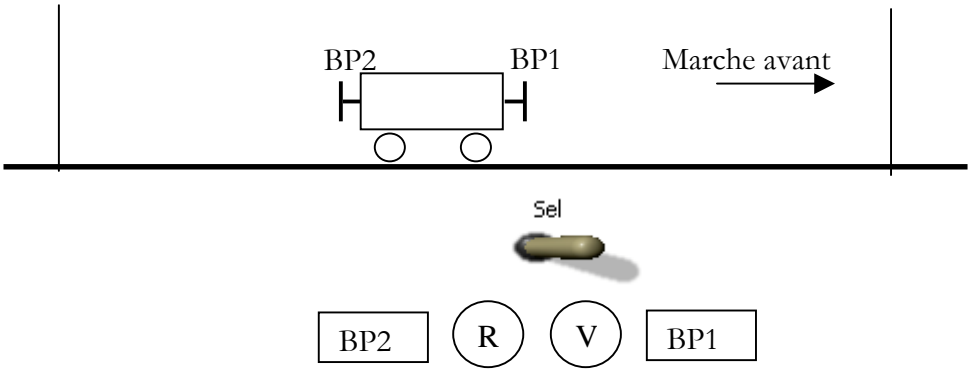
EXERCICE 3-3

Même dispositif. Simuler le fonctionnement du chariot qui attend l'appui sur DCy pour lancer 10 allers-retours.

EXERCICE 4-3

Réaliser le programme suivant :

Si Sel basculé à droite :
Marche avant (led Verte allumée) jusqu'à appui sur BP1
Marche arrière ((led Rouge allumée) jusqu'à appui sur BP2
Si Sel basculé à gauche :
Marche arrière ((led Rouge allumée) jusqu'à appui sur BP2
Marche avant (led Verte allumée) jusqu'à appui sur BP1



Chapitre 4 – CODAGE DES VARIABLES NUMERIQUES

I – CODAGE

1°/ NOMBRE CODE EN DECIMAL

Un résultat de mesure
La valeur d'une commande numérique } Information numérique utilisant le système décimal
Le résultat d'un calcul

Exemple : X = 241

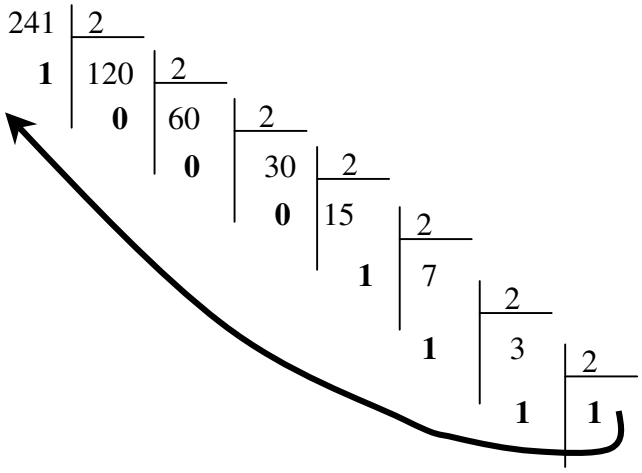
$10^2 = 100$	$10^1 = 10$	$10^0 = 1$
2	4	1

Système décimal : $241 = 2 * 10^2 + 4 * 10^1 + 1 * 10^0$

En réalité, les données manipulées par l'ordinateur sont codées en binaire.

2°/ CODAGE EN BINAIRE

Conversion Décimal Binaire :



241 = 0b11110001

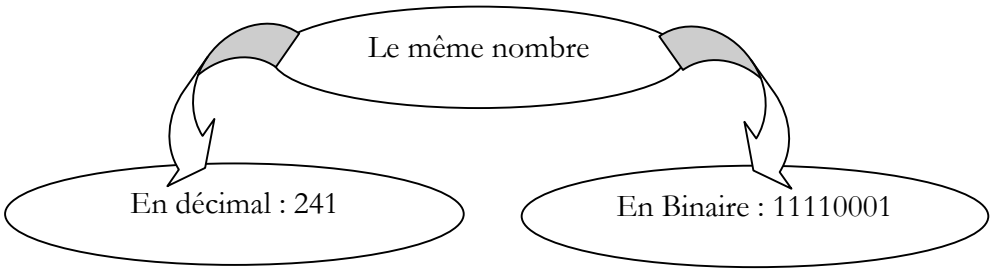
Pour indiquer qu'il s'agit d'un nombre codé en binaire

Conversion Binaire Décimal :

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^4	2^1	2^0
1	**1**	**1**	**1**	**0**	**0**	**0**	**1**

$11110001 = 1*2^7 + 1*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 0*2^1 + 1*2^1 = 241$

En résumé :



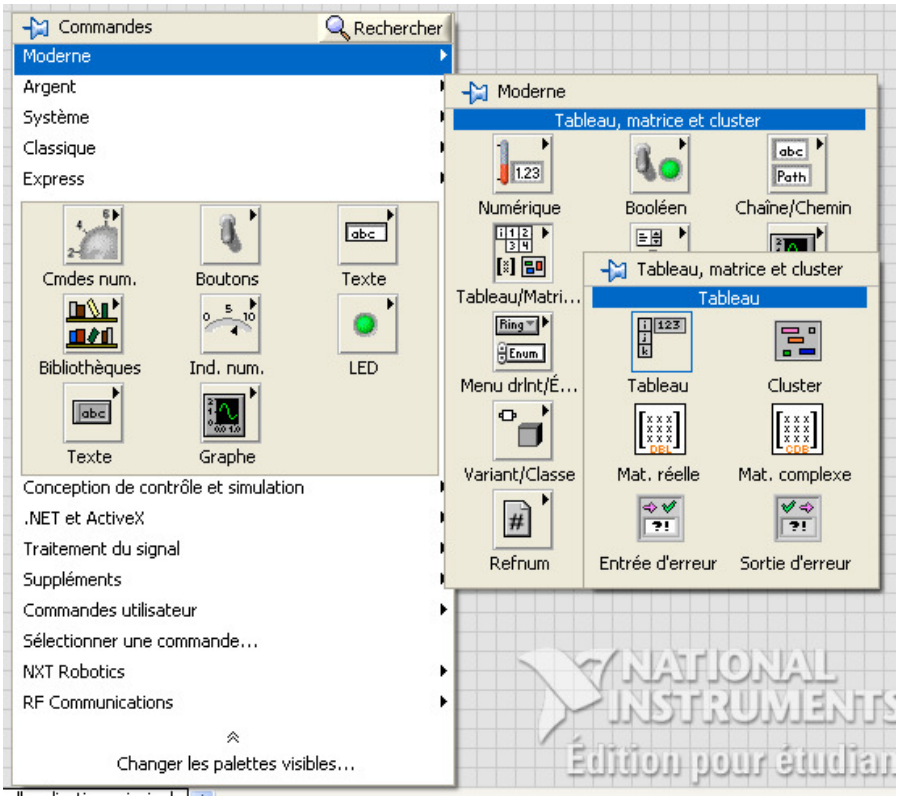
Un mot binaire est un ensemble de bits (**Bi**nary Uni**ts**)

Chaque bit apparaît comme une information booléenne (0 ou 1).

II – VARIABLE NUMERIQUE POUR COMMANDER DES INDICATEURS BOOLEENS

Comment commander des indicateurs booléens à partir d'une variable numérique ?

Il faut regrouper les variables booléennes dans un tableau :

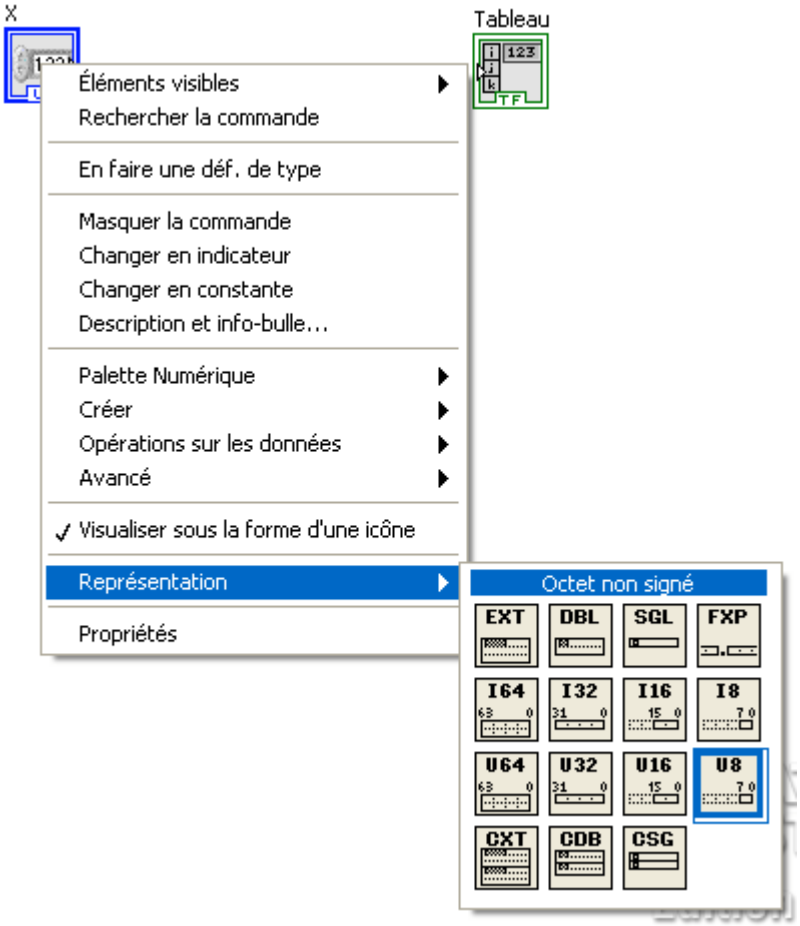


Remplir ensuite le tableau avec une led pour créer un tableau de led. 

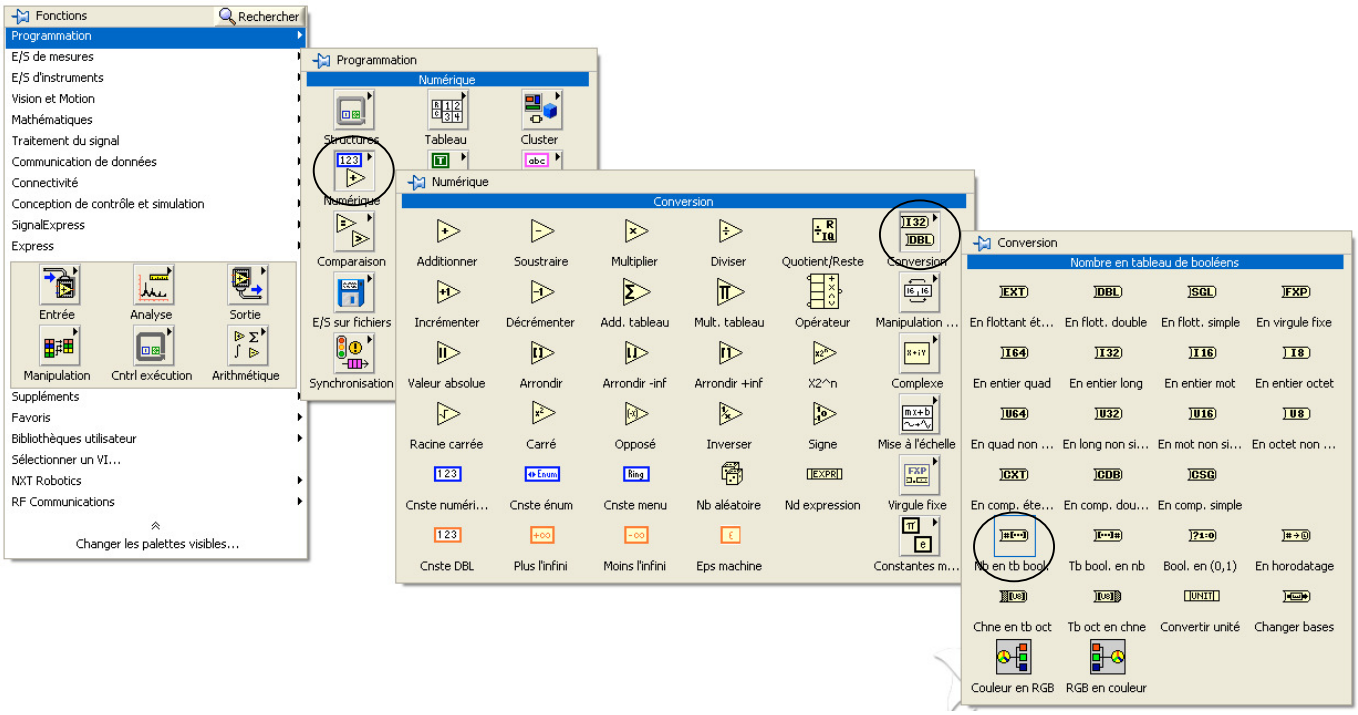
Utilisons la variable X pour allumer les leds correspondant à sa conversion en binaire.

La variable X doit être un entier non signé (uniquement positif, entier naturel).

Pour cela il faut changer la représentation de cette variable et choisir U8 : Entier non signé de 8 bits.



Dans le diagramme convertir le nombre en tableau de booléens :





EXERCICE 1-4

Afficher sur un tableau de leds le comptage en binaire de 0 à 250.

EXERCICE 2-4

On utilise un tableau de 8 leds pour créer un jeu de lumière appelé chenillard : chaque led s'allume successivement à la cadence de 0,25 s, l'allumage d'une led éteint la précédente.
Le chenillard s'arrête par appui sur un bouton stop

EXERCICE 3- 4

Un tableau de 10 leds constitue un chenillard.

Dans un premier temps, les leds s'allument successivement à la cadence de 0,25 s, chaque led restant allumée.
Ce cycle recommence trois fois consécutives.

Dans un deuxième temps les 10 leds clignotent 4 fois simultanément (allumage pendant ¼ seconde).

Dans un troisième temps, pendant ¼ s les leds de n° impair sont allumées pendants que celles de n° pairs sont éteintes puis inversement pendant ¼ s, ce cycle recommençant 3 fois.

Les trois phases doivent se succéder jusqu'à l'arrêt par appui sur un bouton Stop.

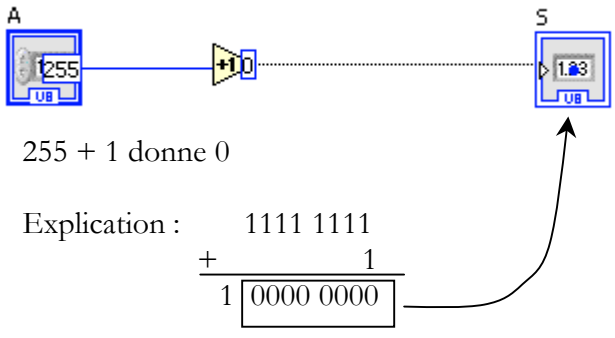
III – REPRESENTATION DES VARIABLES ENTIERES

1°/ EXEMPLE U8 : Nombre entier positif

Min : 0000 0000 = 0

Max : 1111 1111 = 1 0000 0000 – 1 = 2^8 – 1 = 256 – 1 = 255

Attention :



255 + 1 donne 0

Explication : 1111 1111

\+ 1

1 0000 0000

2°/ EXEMPLE I8 : Nombre entier relatif, codage complément à deux

Exemple : - 95

Coder 95 en binaire : 95 = 0 1 0 1 1 1 1 1

Complémenter 1 0 1 0 0 0 0 0
Ajouter 1 + 1

Résultat **1 0 1 0 0 0 0 1**

Bit de signe

Exemple : - 128

Coder 128 en binaire : 128 = 1 0 0 0 0 0 0 0

Complémenter 0 1 1 1 1 1 1 1
Ajouter 1 + 1

Résultat **1 0 0 0 0 0 0 0**

Bit de signe

On constate que pour un nombre négatif le 7ème bit est à 1
Le codage de la valeur n'utilise plus que 7 bits

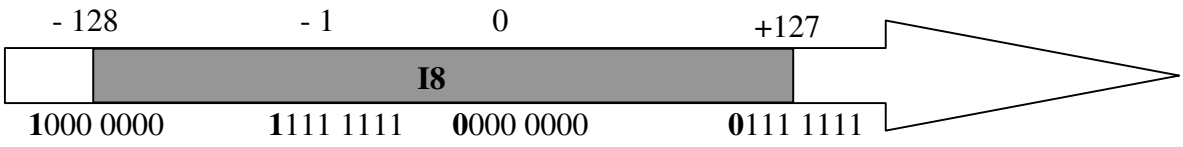
Exemple : + 127

Coder 127 en binaire : 127 = 0 1 1 1 1 1 1 1

Bit de signe

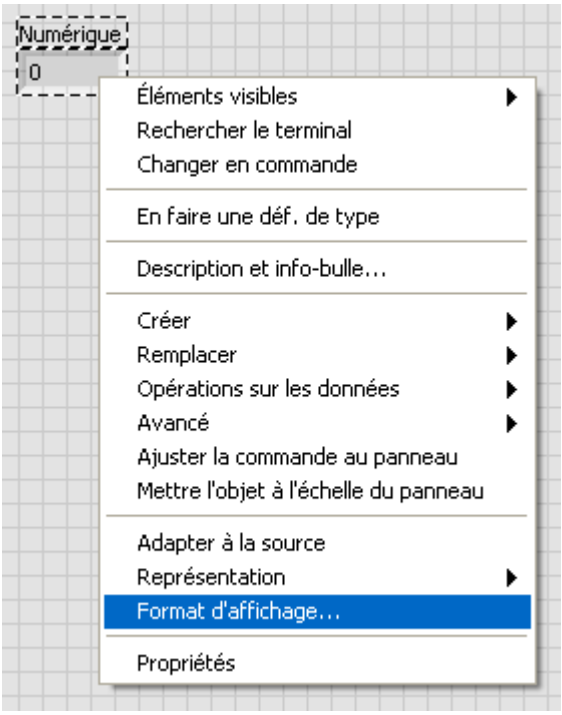
C'est le plus grand nombre pouvant être codé sur 7 bits.

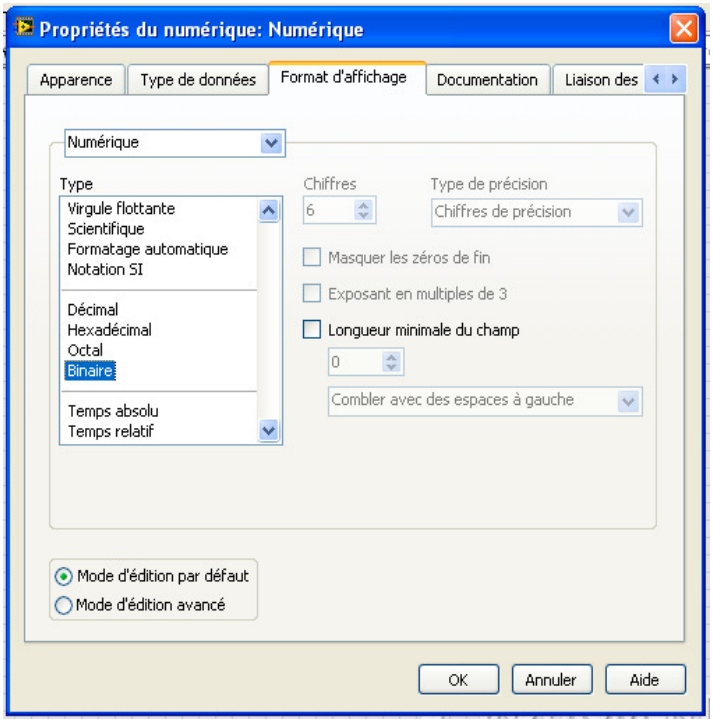
Conclusion I8 :



3°/ FORMAT D'AFFICHAGE

On peut modifier le format d'affichage d'une commande ou indicateur numérique entière





EXERCICE 4-4

Afficher sur des indicateurs numériques les valeurs binaires de commandes numériques de type U8, I8, U16, I16 affichées en décimal

Inversement afficher sur des indicateurs numériques les valeurs décimales de commandes numériques de type U8, I8, U16 et I16 affichées en binaire

Vérifier les valeurs extrêmes de chaque représentation

Que se passe t-il si on dépasse les valeurs extrêmes d'une représentation ?

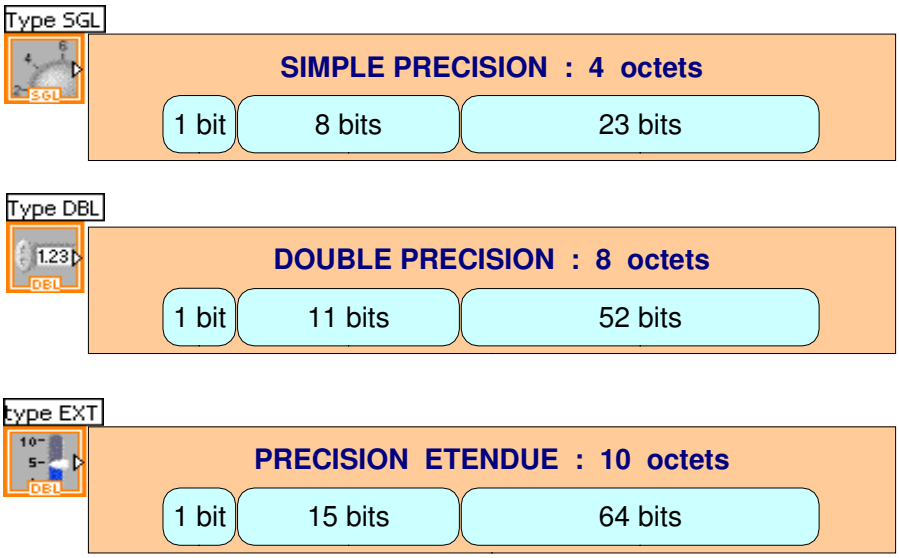
IV – REPRESENTATION DES VARIABLES NUMERIQUES NON ENTIERES

1°/ NOMBRES A VIRGULE FLOTTANTE

C'est la représentation qu'utilise LabVIEW par défaut lorsque l'on dépose une commande ou indicateur numérique sur la face avant.

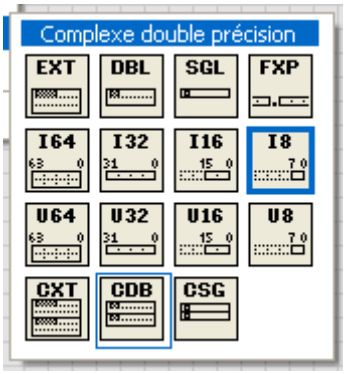
Cette représentation permet le codage des nombres décimaux appelés nombres à virgule flottante.

Le codage utilisé ne sera pas étudié.
Selon le nombre de bits utilisé par ce codage, on peut obtenir des valeurs extrêmes différentes.



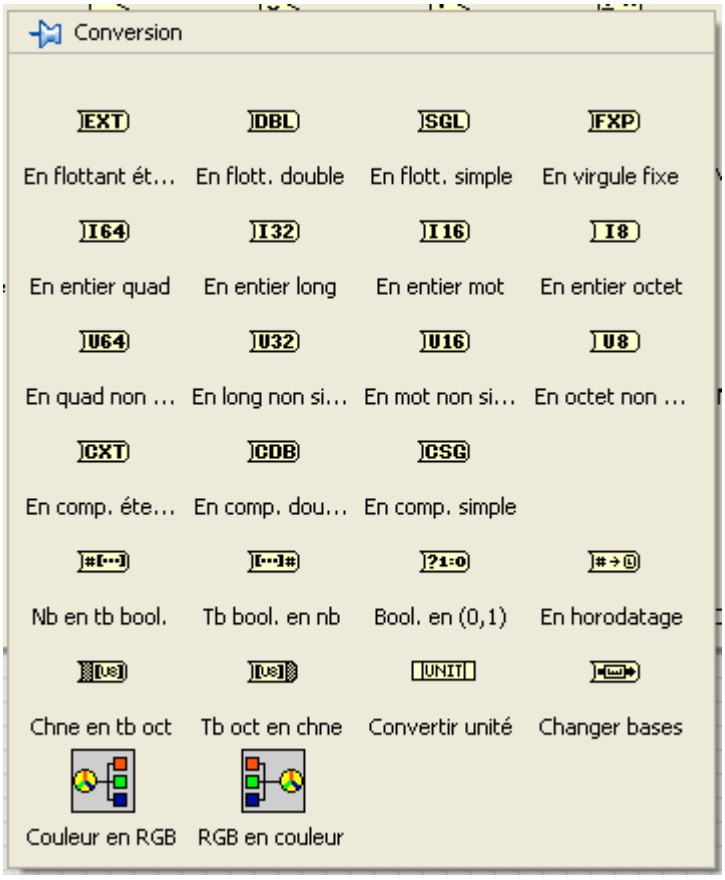
2°/ NOMBRES COMPLEXES

Dans labVIEW il est possible de manipuler des nombres complexes en simple, double précision et précision étendue.

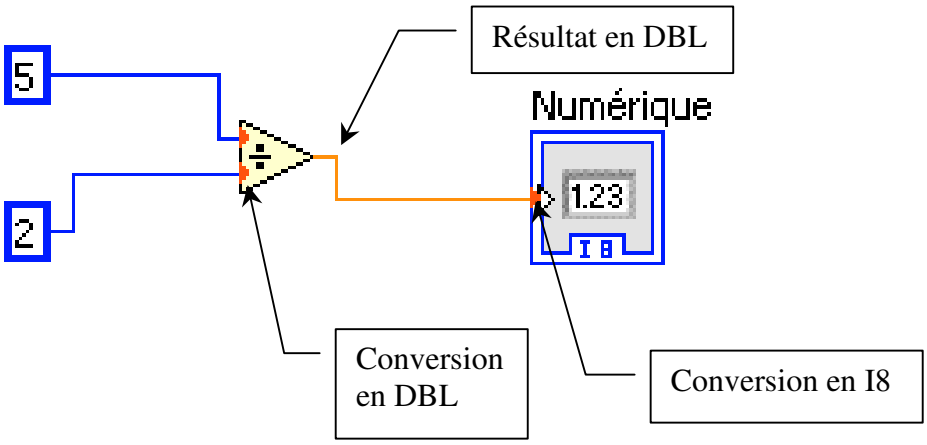


V – CONVERSION DE TYPE

Il est possible de convertir la représentation d'une variable par programmation dans le diagramme.



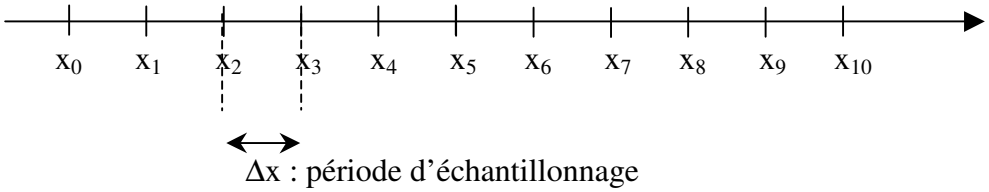
Dans certains cas LabVIEW force la conversion à l'entrée d'un opérateur.
Il apparaît alors un point rouge sur l'entrée correspondante.



Chapitre 5 – GRAPHES

I – ECHANTILLONNAGE

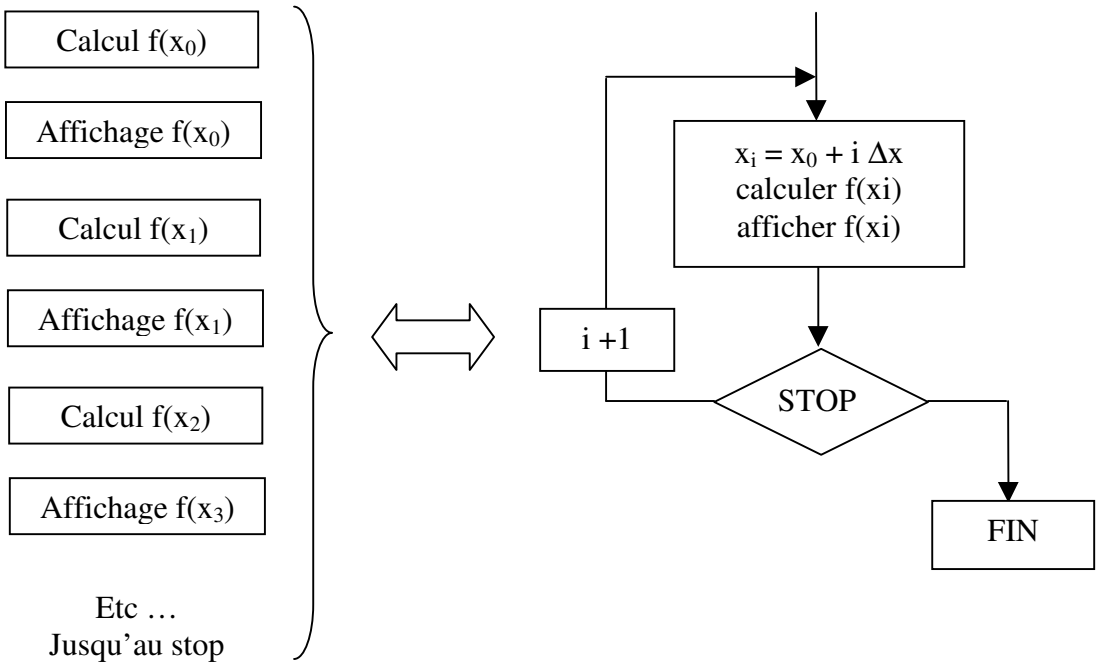
Pour tracer un graphe il faut discrétiser (échantillonner) la variable d'abscisse.



On calcule ainsi f(x) seulement pour les valeurs x_i : $\mathbf{x_i = x_0 + i\,\Delta x}$ ou $\mathbf{x_i = x_{i-1} + \Delta x}$

L'ensemble des valeurs x_i constitue une suite arithmétique de premier terme x_0 et de raison Δx.

II – GRAPHE DEROULANT

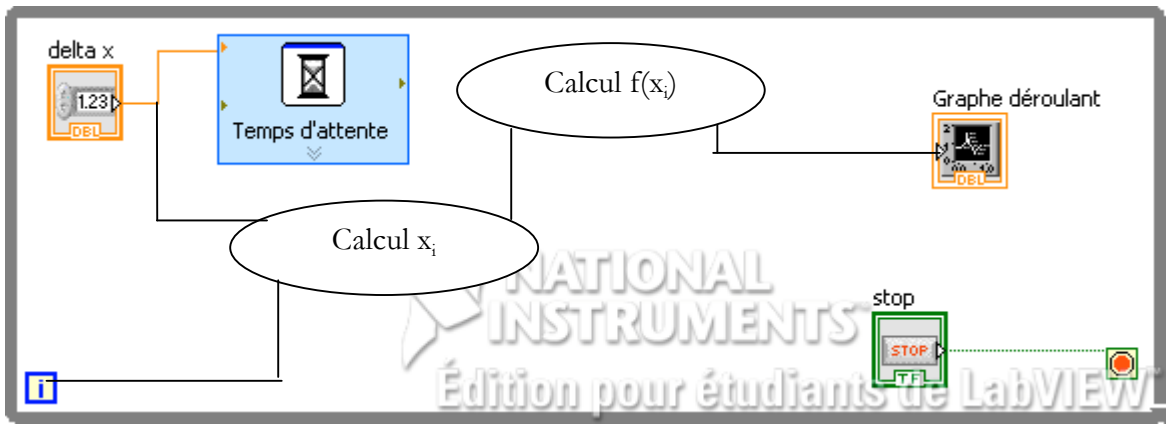


Cette solution est réalisée à l'aide d'un :

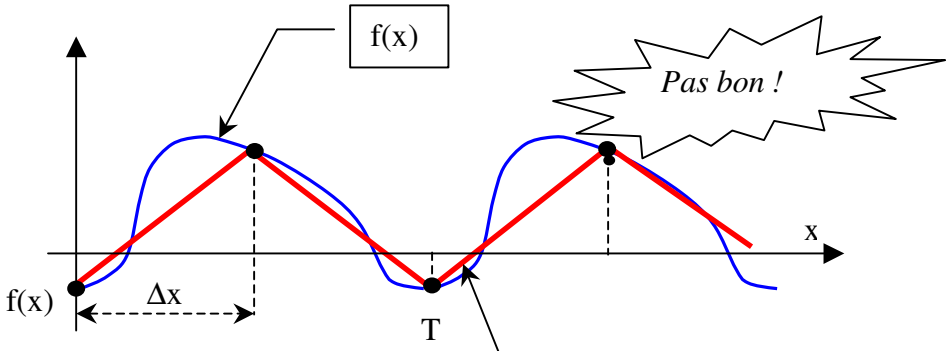
graphe déroulant dans une boucle while.

A chaque itération, on calcule et affiche un seul point.

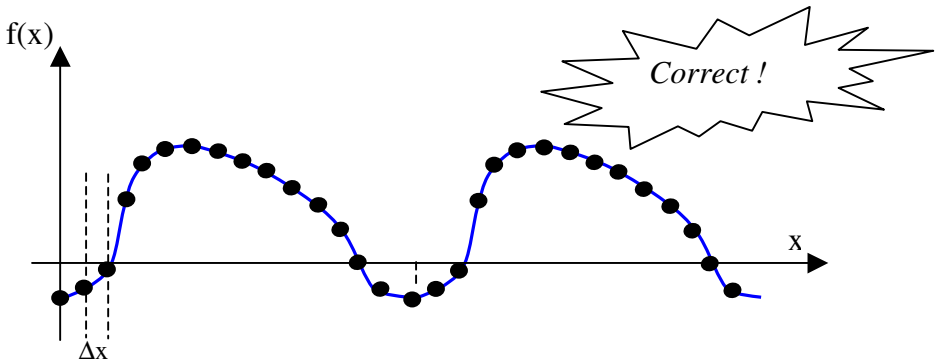
Dans le diagramme LabVIEW :



Choix période d'échantillonnage :



La représentation graphique après échantillonnage ne correspond pas à la fonction f car **Δx est trop grande**



Pour échantillonner correctement il faut suffisamment d'échantillons par période ! => **choisir Δx << T**

EXERCICE 1-5

Afficher sur un graphe déroulant la fonction f(t) = sin(t).

La période d'échantillonnage est une commande en face avant, elle sera choisie et initialisée correctement. Elle cadencera également le graphe déroulant.

EXERCICE 2-5

On veut représenter sur un graphe déroulant la fonction s définie par :

$$s(t) = A \sin(2\pi f_1 t) \sin(2\pi f_2 t)$$

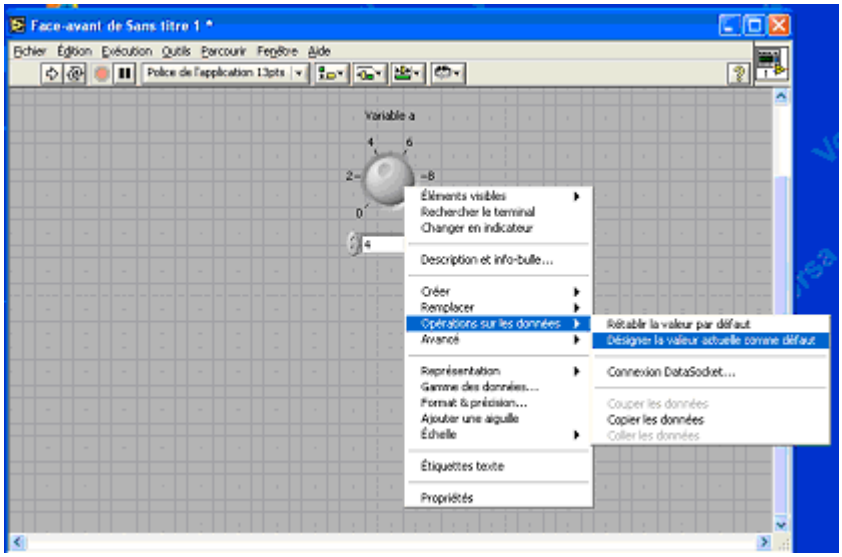
pour laquelle l'utilisateur choisira $f_2 > f_1$. (Cette considération n'est pas à prendre en compte par programmation).

A, f_1 et f_2 sont des commandes numériques sur la face avant.

La période d'échantillonnage est une commande en face avant, elle sera choisie et initialisée correctement. Elle cadencera également le graphe déroulant.

Initialisation de f_1 et f_2 : f_1 = 0,1 Hz et f_2 = 3 Hz

Remarque : Initialisation de variable. La valeur initialisée est conservée lors de l'ouverture du VI.



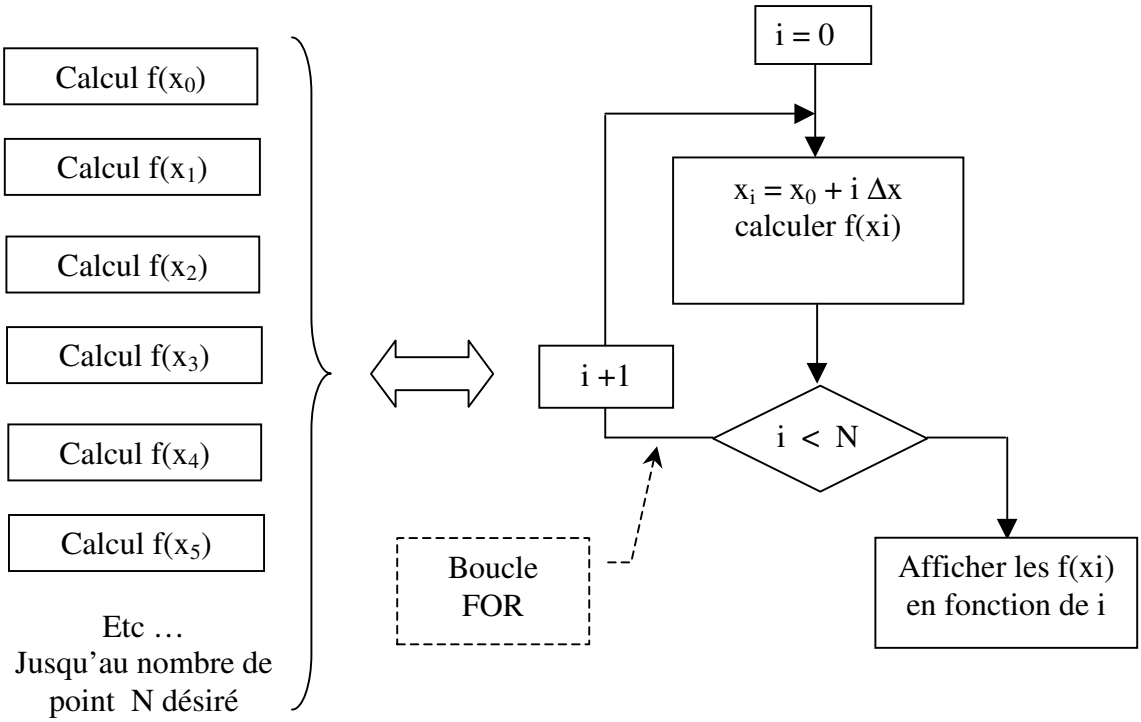
EXERCICE 3-5

La montée en température de 80 à 90 °C d'une enceinte isolée suit la loi de variation suivante :

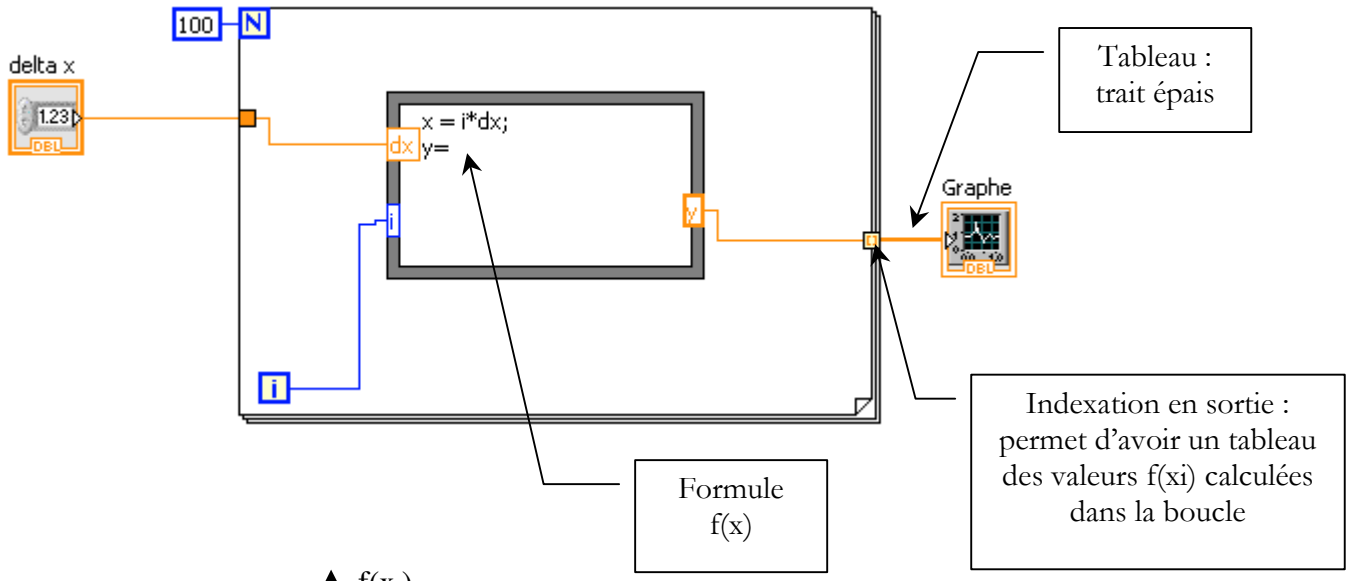
$$T(t) = 10(1 - \exp(-t/\tau)) + 80 \quad \text{avec} \quad \tau = 10 \text{ s}$$

Représenter sur un graphe déroulant la simulation de cette montée en température.
La période d'échantillonnage Te est une commande sur la face avant. Choisir et initialiser correctement la période d'échantillonnage par rapport à la constante de temps du système.

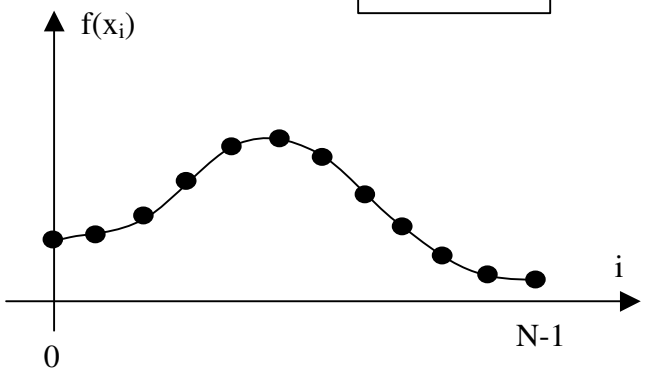
III – GRAPHE



Cette solution est réalisée à l'aide d'un graphe à l'extérieur d'une boucle for



Le graphe obtenu est :



N points pour i allant de 0 à N-1

EXERCICE 4-5

Afficher sur un graphe 100 points de la fonction f(t) = sin(t).

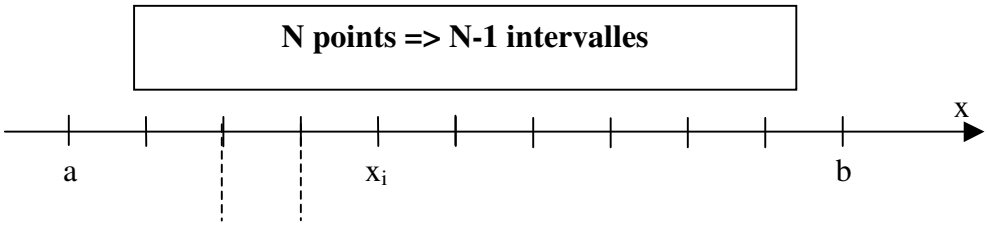
La période d'échantillonnage est une commande en face avant, elle sera choisie et initialisée correctement.

EXERCICE 5-5

Ecrire un programme qui génère un tableau de N valeurs de la fonction y(t) = sin (x) / x .

On représentera cette fonction sur l'intervalle [-4π ; 4π] avec N points pour la représentation. (N=100). Δx n'est donc pas une commande en face avant, il faut calculer Δx en suivant l'indication ci-dessous.

Remarque :

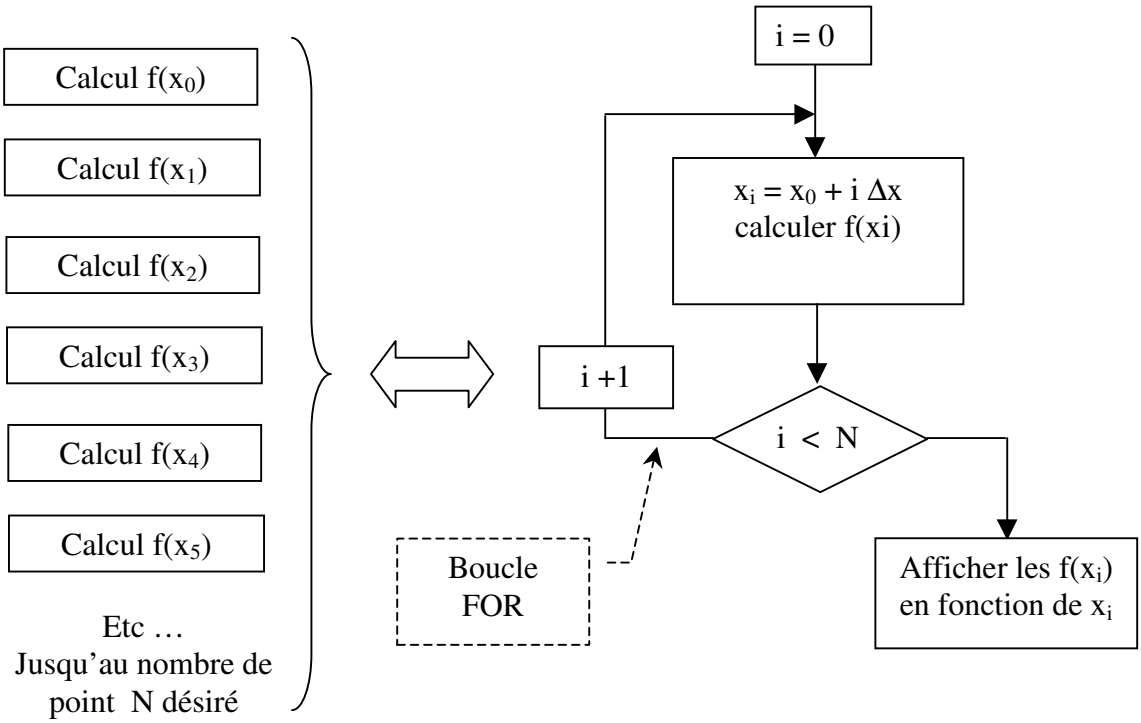


Δx : période d'échantillonnage

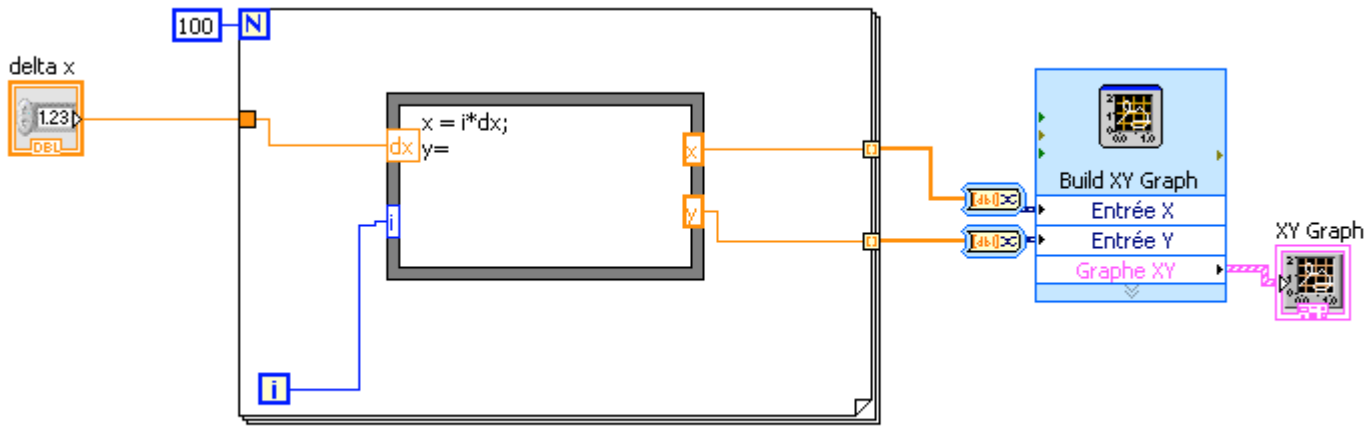
$$\Delta x = (b - a) / (N-1)$$

$$x_i = a + i\,\Delta x$$

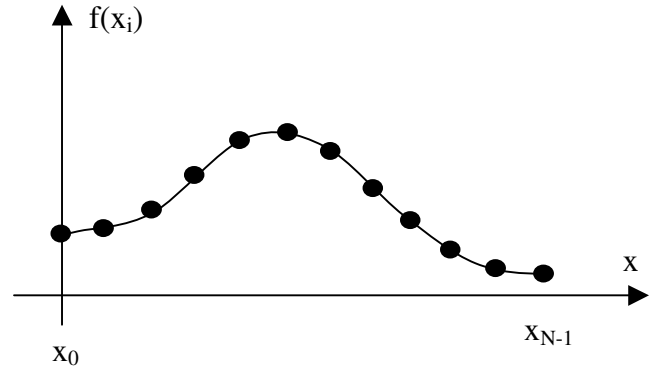
III – GRAPHE XY



Cette solution est réalisée à l'aide d'un graphe XY à l'extérieur d'une boucle for.



Le graphe obtenu est :



N points pour x_i allant de x_0 à x_{N-1}

EXERCICE 6-5

La luminance par unité de longueur d'onde d'un corps noir à la température T est donnée par :

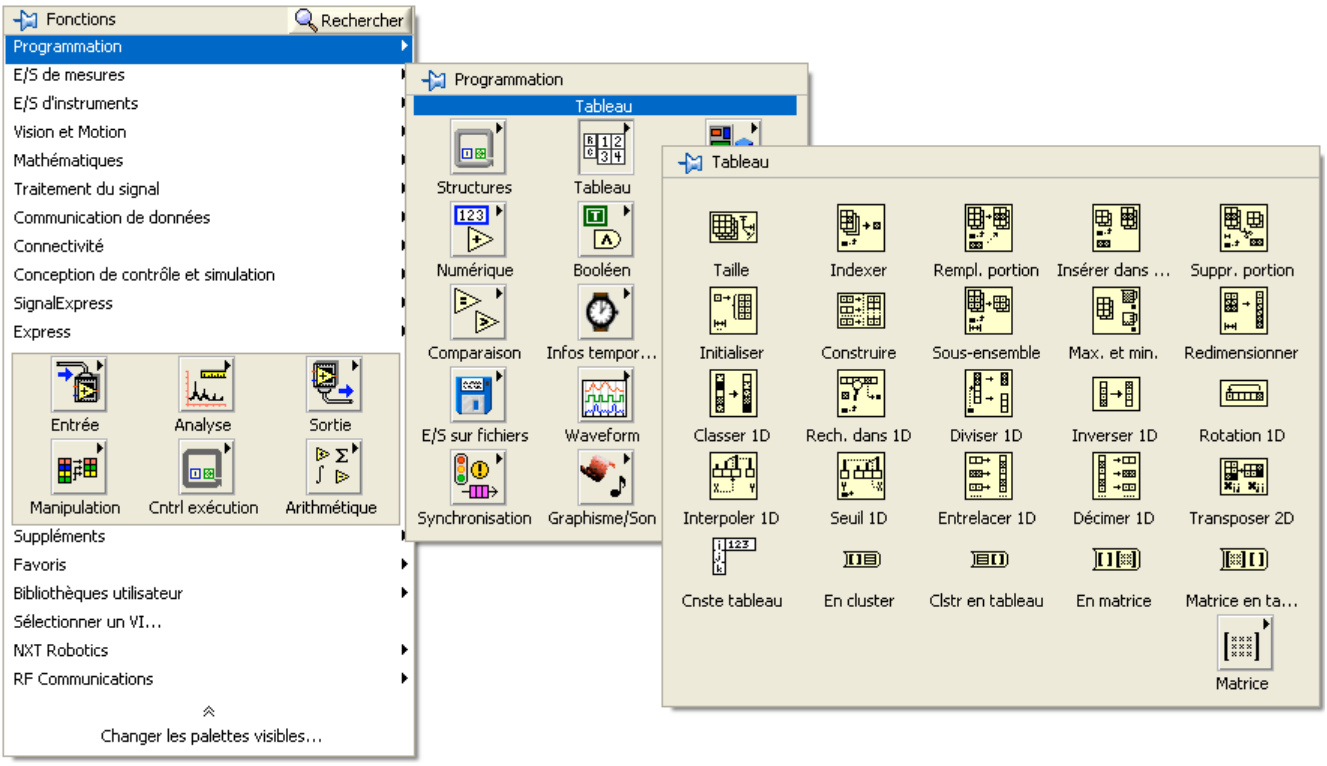
$L(\lambda) = 2\,h\,c^2 / [\,\lambda^5\,(\exp(h\,c / \lambda\,k\,T) - 1\,)\,]$ **$h = 6{,}62.\,10^{-34}$ J/s** **$k = 1{,}38.\,10^{-23}$ J/K**
$c = 3.\,10^8$ m/s

T est une commande sur la face avant initialisée à 5770 K

Pour λ compris entre 0 et 2 µm, représenter sur un graphe XY les 100 points de la courbe de L(λ).

En utilisant les outils de manipulation des tableaux, déterminer la longueur d'onde donnant le maximum de luminance.

Outils de manipulation de tableaux : Uiliser l'aide pour comprendre ces différentes fonctions.



EXERCICE 7-5

Le centre d'inertie d'un mobile de masse m lancé d'un point O avec une vitesse initiale V_0 dans une direction faisant un angle α avec l'horizontale décrit une trajectoire dont les équations paramétriques sont :

$$y(t) = -\tfrac{1}{2} g t^2 + (V_0 \sin \alpha)\, t \qquad \text{avec } g = 9{,}81 \text{ m s}^{-2}$$

$$x(t) = (V_0 \cos\alpha)\, t$$

V0 et α sont des commandes en face avant initialisées à :

$V_0 = 50$ m/s $\quad \alpha = 0{,}96$ radians (soit 55 °)

Pour t compris entre 0 et 10 s représenter sur un graphe XY les 100 points de la trajectoire décrite par le mobile.

En utilisant les outils de manipulation des tableaux, déterminer et afficher sur un indicateur numérique la hauteur maximale atteinte par le mobile, la valeur de x correspondante ainsi que la portée du tir.

IV – GRAPHE XY ECHELLE LOG

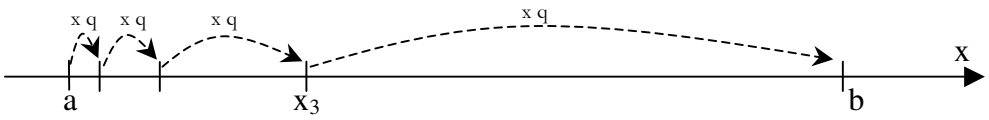
Utilisation échelle log : quand x varie dans un large intervalle.

Principe : La période d'échantillonnage augmente, on écarte les points au fur et à mesure que x augmente.

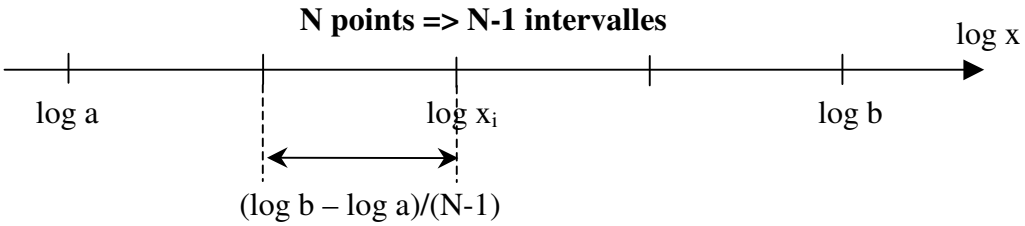
1er point : $x_0 = a$
2ème point : $x_1 = a*q$
3ème point : $x_2 = x_1*q = a*q^2$
4ème point : $x_3 = x_2*q = a*q^3$
etc …..

$$x_i = a\ q^i$$

On constate que l'ensemble des points xi constitue une suite géométrique de premier terme a et de raison q.

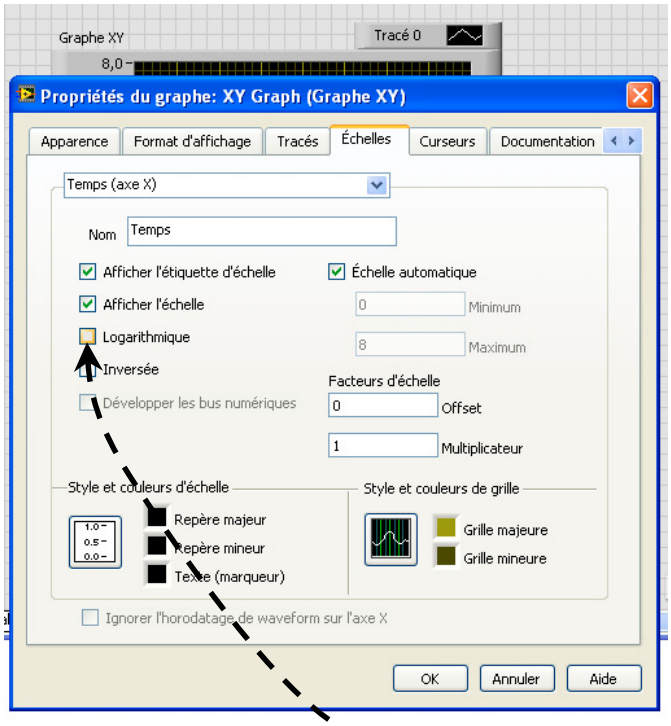


On comprime ensuite l'axe des abscisses en appliquant la fonction log décimal *(voir annexe ci-après)*



Les points deviennent régulièrement espacés.

Expression de q : $\log q = (\log b - \log a)/(N-1) = [1/(N-1)] \log(b/a) = \log [(b/a)^{1/N-1}] \Rightarrow$ $q = (b/a)^{1/N-1}$



En résumé pour créer une échelle d'abscisses logarithmique :

$$x_i = a * (b/a)^{i/N-1}$$

Puis demander sur le graphique une échelle log pour les abscisses.

Clic droit sur le graphe : échelle des x /formatage

EXERCICE 8.5 - Graphe XY échelle logarithmique

Ecrire un programme permettant d'afficher sur un graphe la courbe de réponse en fréquence d'un circuit RC :

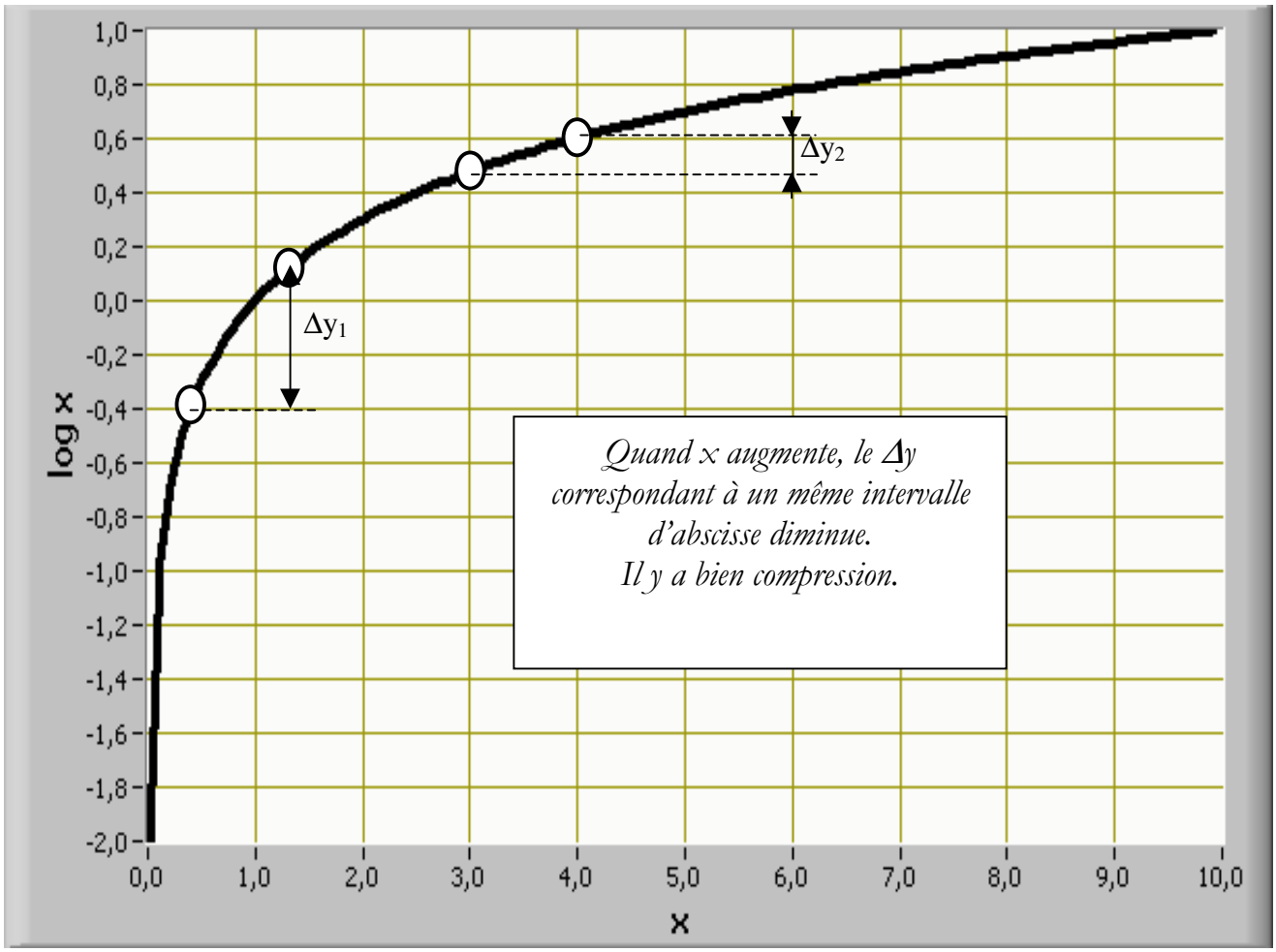
$$G(f) = 20 \log \left[1 / \sqrt{ (1 + (RC.2\pi f)^2) } \right]$$

R et C sont modifiables grâce à une commande numérique.
Le nombre de points N est modifiable également sur la face avant.
La courbe sera tracée de f_a = 10 Hz à f_f = 100000 Hz

On prendra R = 4700 Ω, C = 10 nF.

Le graphe utilisera en abscisses une échelle logarithmique (points régulièrement espacés) qui sera graduée en Hz.

Annexe : La fonction log comprime



Chapitre 6 – REGISTRE A DECALAGE

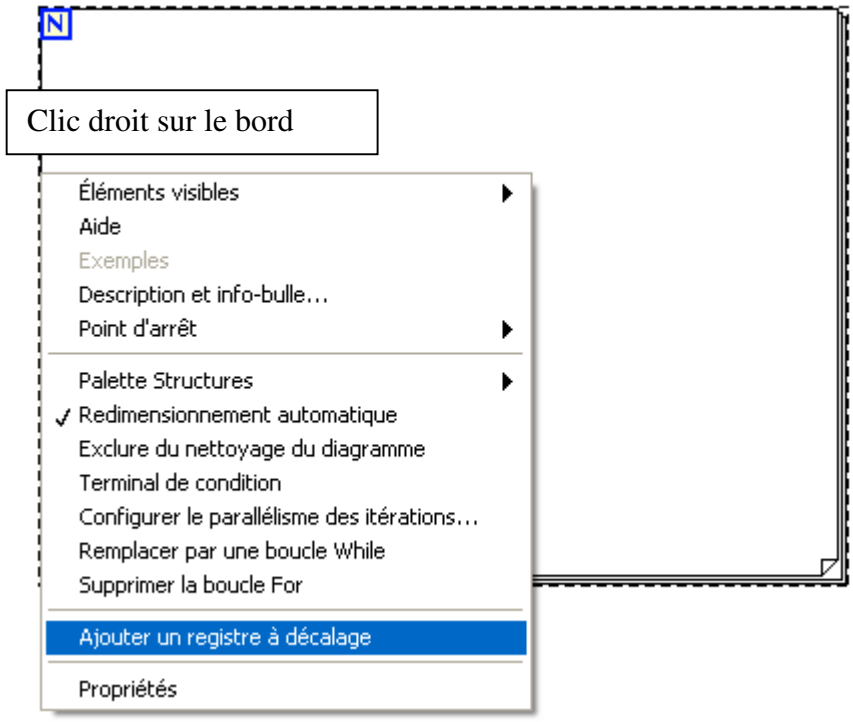
I – ROLE ET UTILISATION

Dans une boucle, le registre à décalage permet d'accéder aux résultats obtenus lors des itérations précédentes en vue de leur réutilisation.

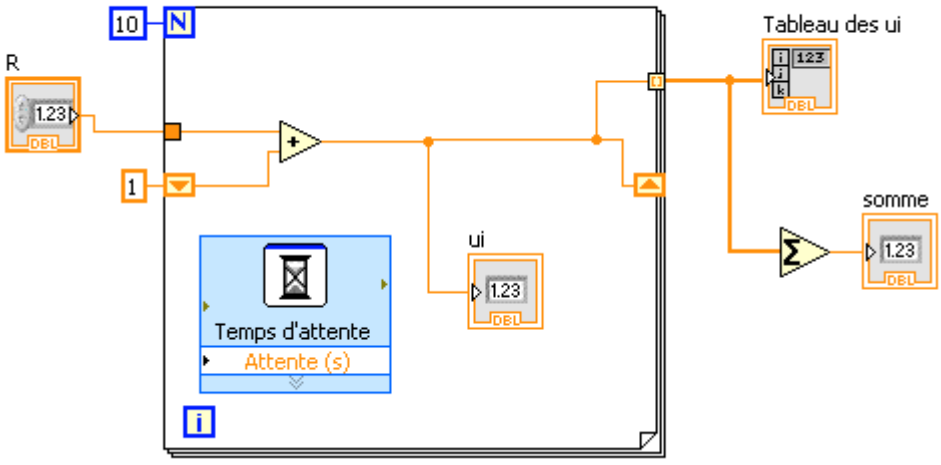
On en a besoin par exemple pour calculer les éléments d'une suite définie par une formule de récurrence.

Exemple : $u_i = u_{i-1} + r$

II – MISE EN OEUVRE



Exemple : Suite arithmétique de premier terme $u_0 = 1$ et de raison R (commande en face avant)
Les 10 premiers termes de la suite sont calculés et affichés chaque seconde.
Lorsque la boucle est exécutée, les u_i sont affichés dans un tableau et on affiche la somme des termes.



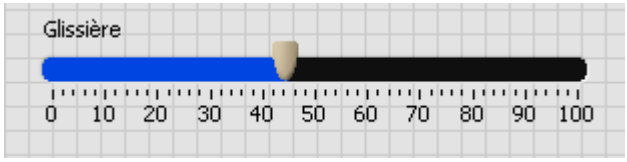
EXERCICE 1-6

1°/ Réaliser un programme qui affiche successivement à chaque seconde les 20 premières valeurs de la suite géométrique définie par son premier terme u_0 et sa raison q. *Rappel : $u_i = u_0 * q^i$*
U_0 et q sont des contrôleurs numériques sur la face avant.

2°/ Même chose en utilisant la formule de récurrence : $u_i = u_{i-1} * q$

EXERCICE 2-6

Déposer un potentiomètre à glissière sur la face avant d'un VI.



Allumer une led verte si on déplace la glissière à droite
Allumer une led rouge si on déplace la glissière à gauche

EXERCICE 3-6

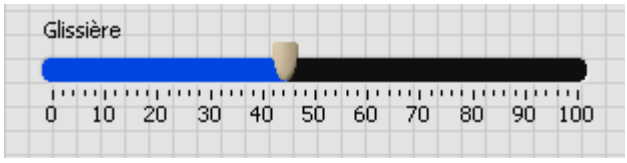
Soit la suite définie par la relation de récurrence :

$$u_n = -0{,}7u_{n-2} + u_{n-1}, \quad \text{avec} \quad u_{-1} = 3 \quad \text{et} \quad u_{-2} = 0$$

Représenter sur un graphe déroulant l'évolution de la suite u_n au cours du temps. On affichera un point à chaque seconde. Le programme s'arrêtera par appui sur STOP.

EXERCICE 4-6

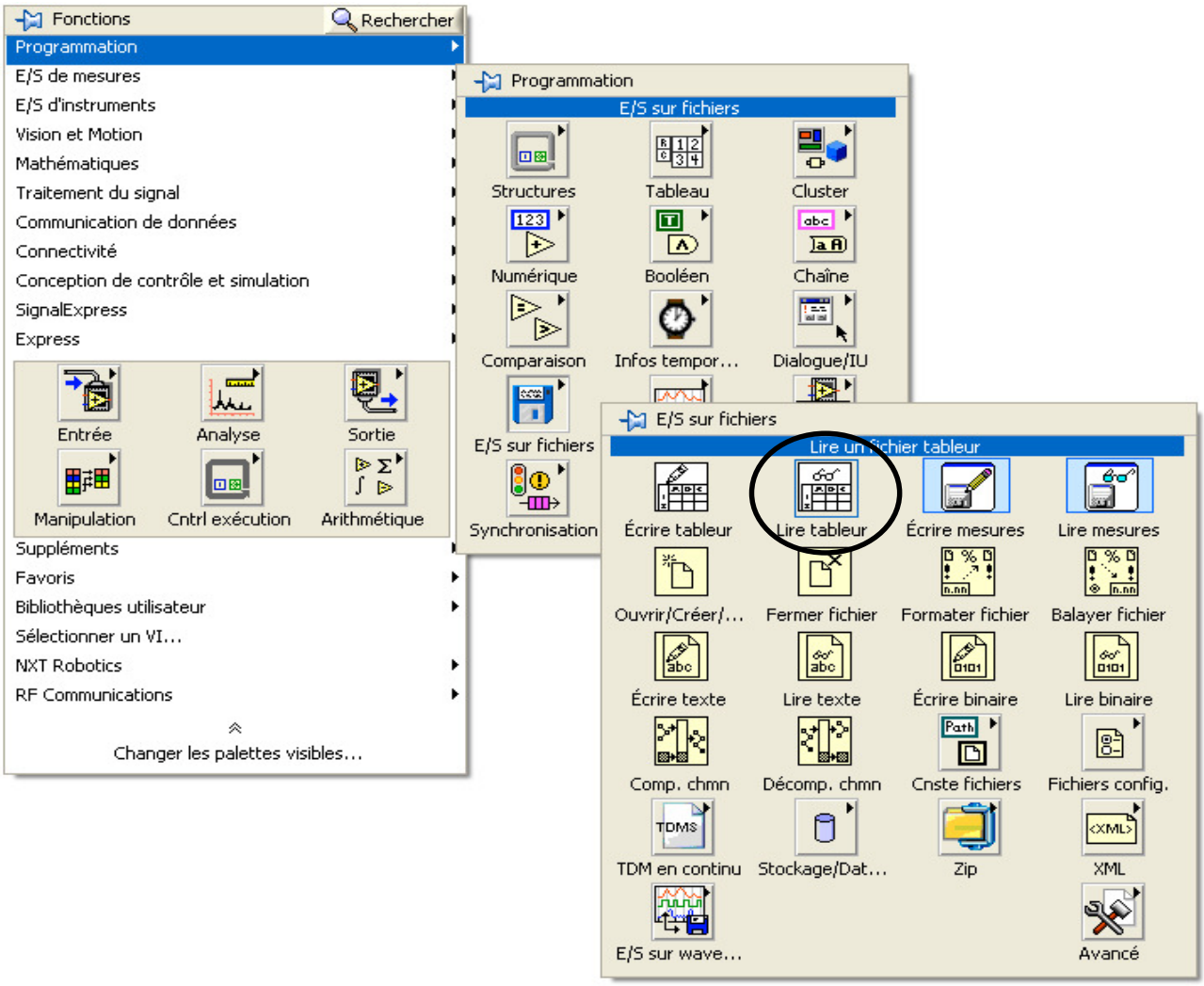
Déposer un potentiomètre à glissière sur la face avant d'un VI.



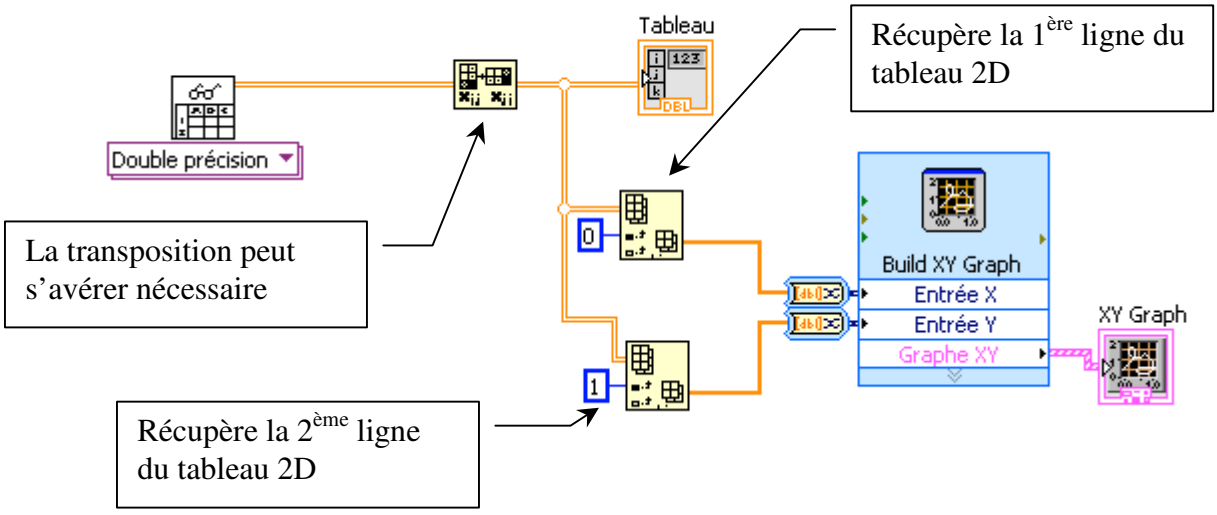
Allumer une led verte si on dépasse 70 en déplaçant la glissière à droite
Eteindre la led si on passe en dessous de 30 en déplaçant la glissière à gauche

I – LECTURE DE FICHIER TABLEUR

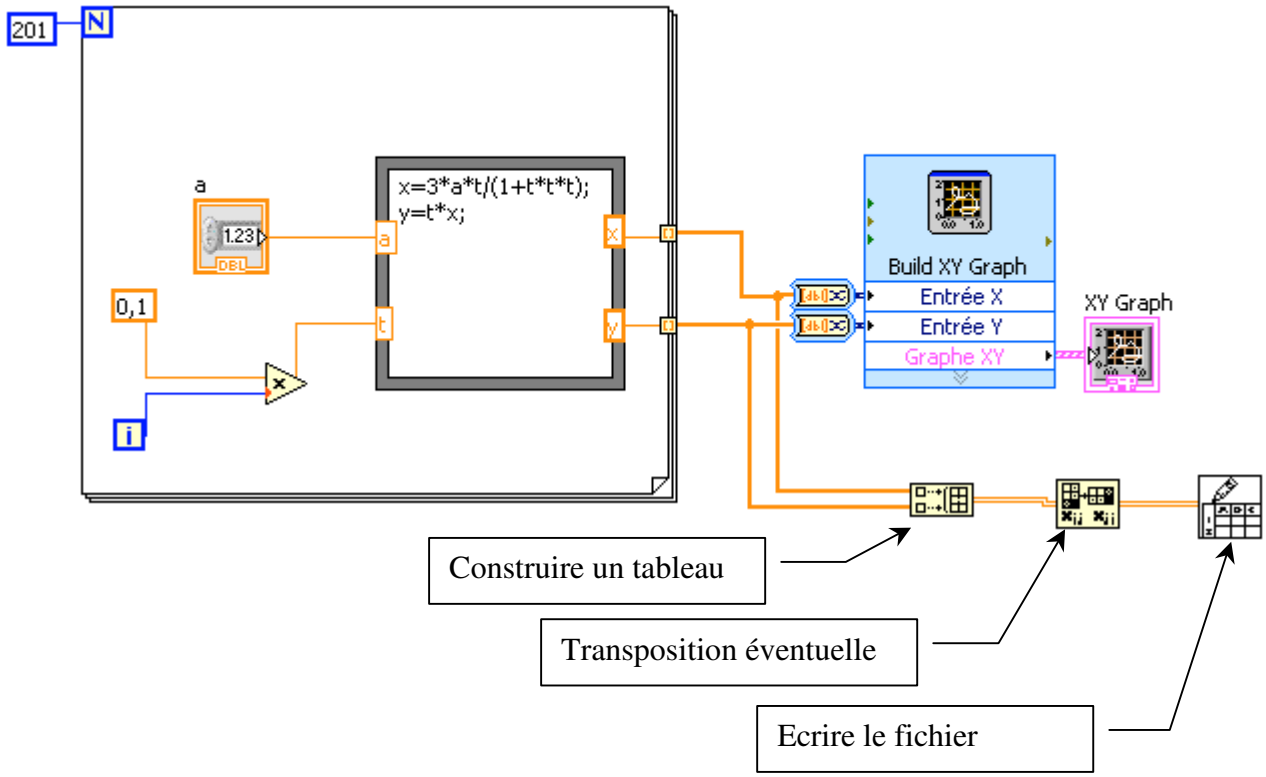
Dans le diagramme :



Ce diagramme permet d'ouvrir un fichier tableur (boite de dialogue pour retrouver le fichier), de le transposer (les lignes deviennent les colonnes) et de l'afficher dans un tableau à deux dimensions.



II – ECRITURE DE FICHIER TABLEUR



EXERCICE 1-7

Créer un VI qui ouvre ce fichier tableur et qui trace la courbe.

Cette courbe correspond à L'enregistrement de la vitesse de rotation d'une roue de voiture au cours du temps.

Compléter le VI de façon à traiter les données de la façon suivante :

- **Déterminer l'accélération du véhicule supposé se déplacer en ligne droite.**
- **Déterminer la distance parcourue**

Tracer ces deux courbes.

En utilisant les outils de manipulation de tableau, déterminer les instants où les roues ont patiné ou se sont bloquées.

Toujours avec les outils de manipulation de tableau, éliminer du tableau des vitesses les phénomènes de patinage ou de blocage des roues.

Exporter les données traitées dans un nouveau fichier tableur.

EXERCICE 2-7

Enregistrer ce fichier avec l'extension .csv dans votre espace personnel.

Ce fichier de données donne la mesure de la pollution de l'air relative au polluant PM10 (poussières microscopiques 10 µm) sur 48h pour Montbéliard centre.

Utiliser l'outil LabVIEW qui permet d'ouvrir un fichier tableur. Placer ces données dans un tableau.
Tracer le graphe correspondant.
Tracer également sur ce même graphe la moyenne glissante sur 8h.
Quel type de filtrage sur les données introduit la moyenne glissante ?

La moyenne glissante ou moyenne mobile est un type de moyenne statistique utilisée pour analyser des séries ordonnées de données, le plus souvent des séries temporelles, en supprimant les fluctuations transitoires de façon à en souligner les tendances à plus long terme. Cette moyenne est dite mobile parce qu'elle est recalculée de façon continue, en utilisant à chaque calcul un sous-ensemble d'éléments dans lequel un nouvel élément remplace le plus ancien ou s'ajoute au sous-ensemble.

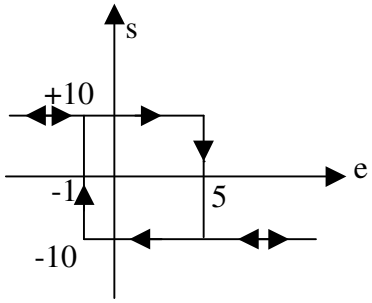
Ce type de moyenne est utilisé généralement comme méthode de lissage de valeurs, en particulier dans le domaine financier pour l'analyse technique de cours boursiers.

EXERCICES DE SYNTHESE

EXERCICE 1-8 : Comparateur à hystérésis

Un comparateur a hystérésis comporte deux seuils différents selon le sens de variation de la grandeur d'entrée.

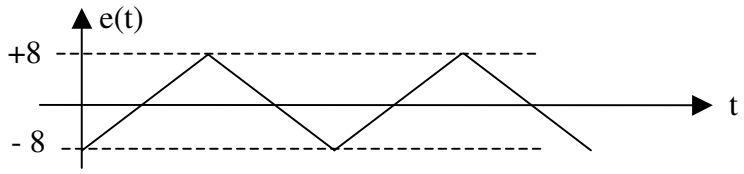
Le fonctionnement est décrit par la caractéristique de transfert s(e) donnée ci-dessous.



Quand e(t) augmente, s bascule à –10 quand e > 5.

Quand e(t) diminue, s bascule à +10 quand e < -1

1°/ Réaliser un programme qui génère le signal triangulaire périodique (période T par exemple de 20 ms) décrit ci-dessous et l'affiche sur un graphe déroulant. (Ne pas utiliser les fonctions de génération de signaux « clef en main » !



2°/ Modifier le programme pour afficher également sur le même graphe déroulant la sortie s(t)

3°/ Faire apparaître la caractéristique de transfert s(e) sur un graphe XY.

EXERCICE 2-8

On veut représenter sur un graphe déroulant la fonction :

$$s(t) = A \sin \{ 2\pi [f_p + k \sin(2\pi f_m t)] t \}$$

pour laquelle l'utilisateur choisira $f_p > f_m$. (Cette considération n'est pas à prendre en compte par programmation).

A, f_p, f_m et k sont des commandes numériques sur la face avant.

La période d'échantillonnage sera choisie correctement (50 points par période de fp). Elle cadencera également le graphe déroulant.

Initialisation : f_p = 1 Hz f_m = 0,01 Hz k = 1 A = 10