



Table Des Matières

Table Des Matières	1
Introduction Générale	2
Présentation de LabVIEW	2
I. INTRODUCTION A LABVIEW	3
II. Interface de labview	4
TP 1 : Création d'un VI simple Fonctions mathématiques de base	6
TP 2 : Opération et test avec les LEDs	11
Manipulation 1	11
Manipulation 2	12
Manipulation 3	13
TP3 : Calcule matricielle a l'aide de LaBVIEW	14
TP4 : Générateur de fonctions	23
Conclusion	26



Introduction Générale

Dans le cadre des travaux pratiques dispensés au département industriel à l'école nationale des sciences appliquées de Fès, nous sommes amenés à réaliser un TP en utilisant le logiciel LabVIEW.

Si les principaux atouts de ce logiciel sont sa programmation graphique intuitive ou l'importante bibliothèque de fonctions fournies, un avantage décisif réside pour nous dans sa capacité d'être multiplateformes matérielles, temps réel et communiquant avec tout type de processus.

LabVIEW n'est plus seulement un logiciel d'instrumentation virtuelle mais surtout un **Environnement Graphique Multi-Cibles Temps Réel sur Réseau**.

Présentation de LabVIEW



I. INTRODUCTION A LABVIEW

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

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

Pour l'acquisition de données et le contrôle d'instruments, LabView supporte les standards RS-232/422, USB, IEEE 488 (GPIB) et VXI/PXI, ainsi que les cartes d'acquisition de données.

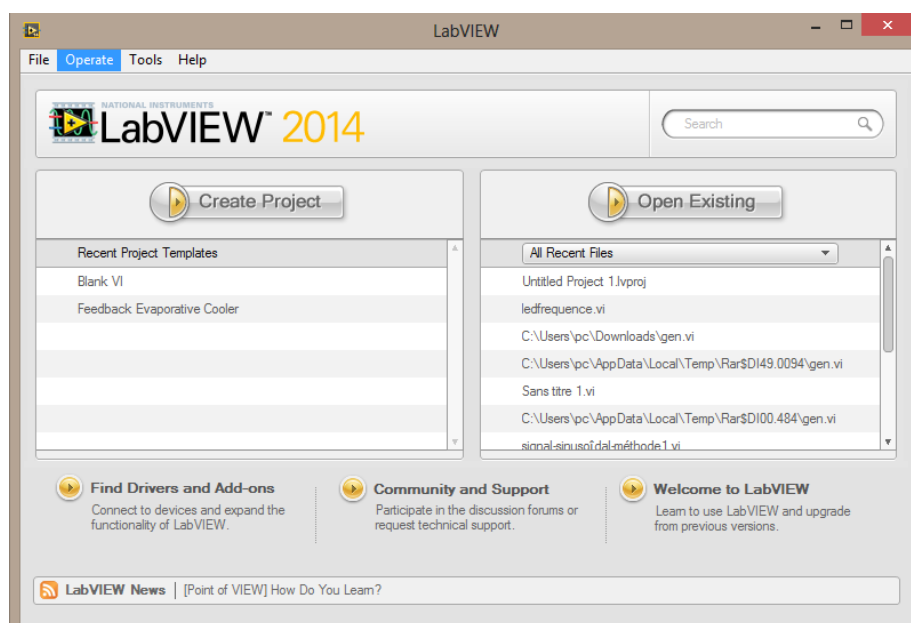
Pour l'analyse et le traitement des données, la bibliothèque d'analyse étendue contient les fonctions pour la génération et le traitement de signaux, les filtres, les fenêtres, les statistiques, la régression, l'algèbre linéaire et l'arithmétique matricielle.

LabView intègre un grand nombre d'éléments de présentation tels les graphes déroulants, des graphes XY, des abaques de Smith, jauges, cadrans à aiguille...

C'est un outil d'acquisition, d'analyse et de présentation de données.



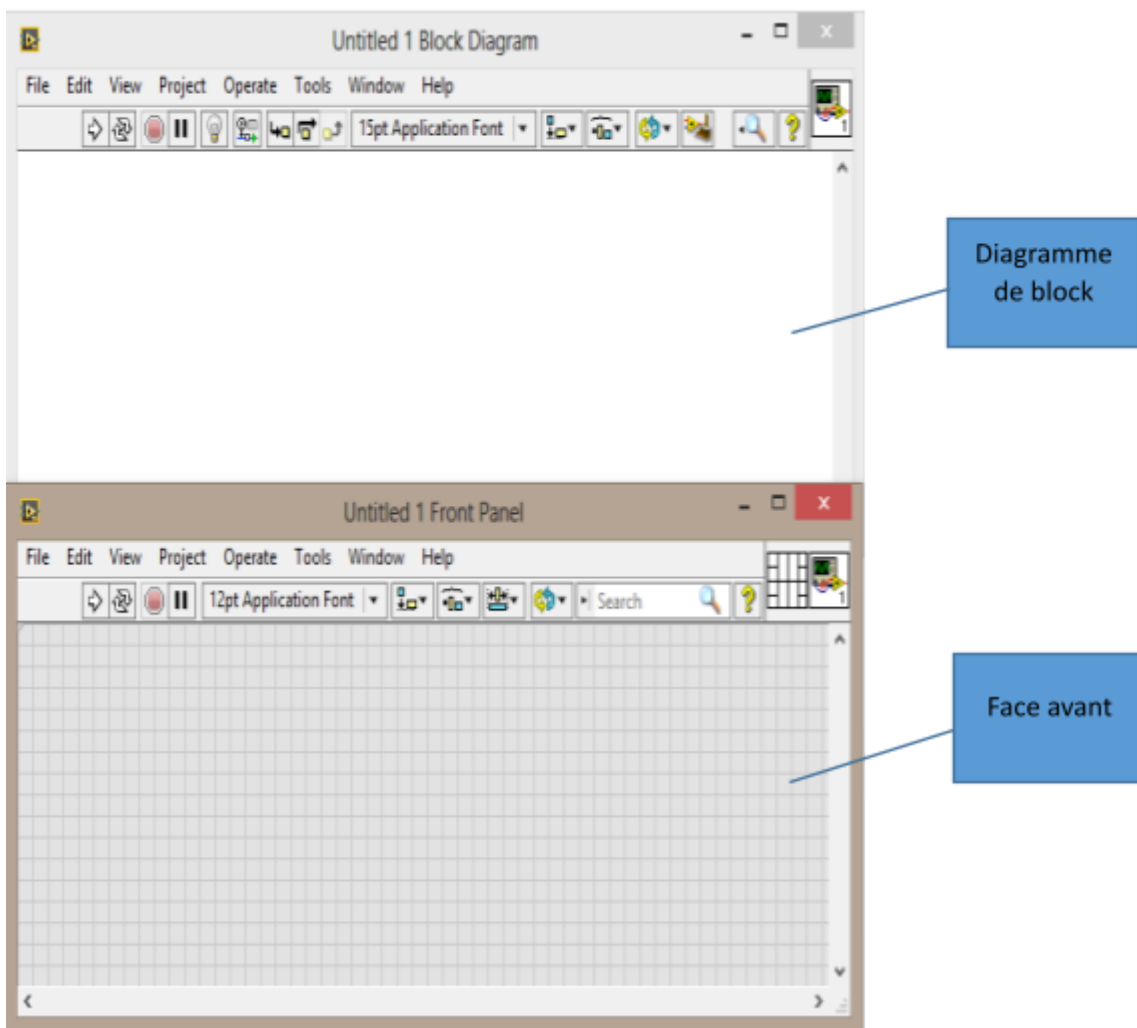
II. Interface de labview



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

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

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



TP 1 : Création d'un VI simple Fonctions mathématiques de base

Dans ce premier TP nous allons commencer par des programmes simples de base composés des entrées (les contrôles) et des sorties (les indicateurs) avec des opérations arithmétiques prédéfinies autant que fonctions à savoir : addition, soustraction, multiplication, division et la racine carrée.

L'addition

On suit les étapes suivantes :

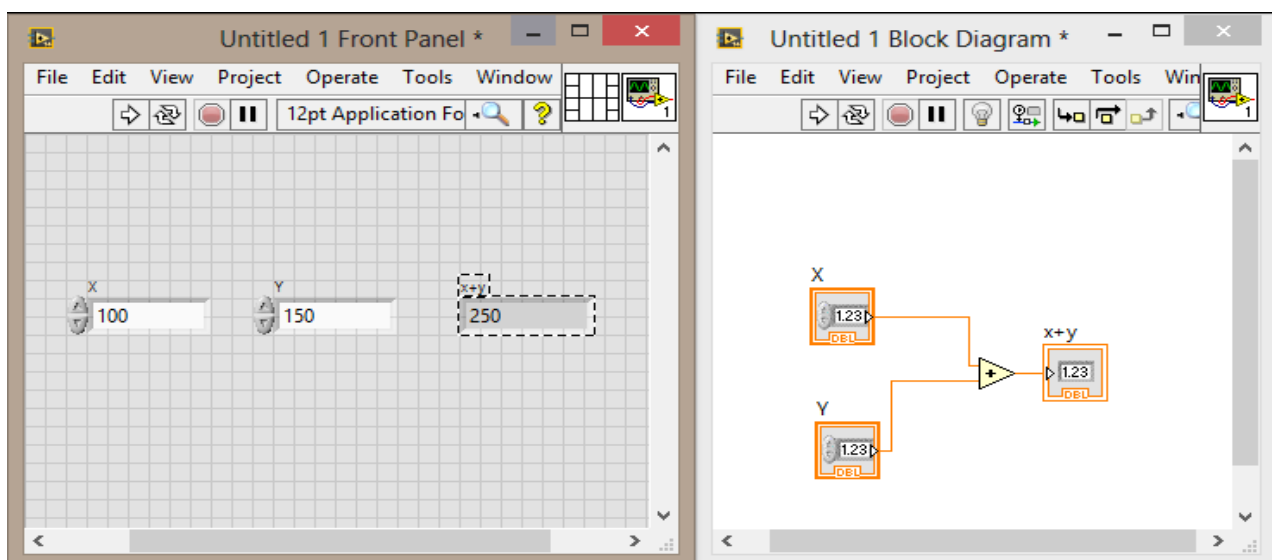
Face avant :

- bouton droit et on choisit commande numérique (2 fois)
- bouton droit et on choisit indicateur numérique

Diagramme :

- bouton droit □ numérique □ additionner.
- Puis on relie entre les différentes commandes

- Ensuite on exécute le programme en cliquant sur :



La soustraction

On suit les étapes suivantes :

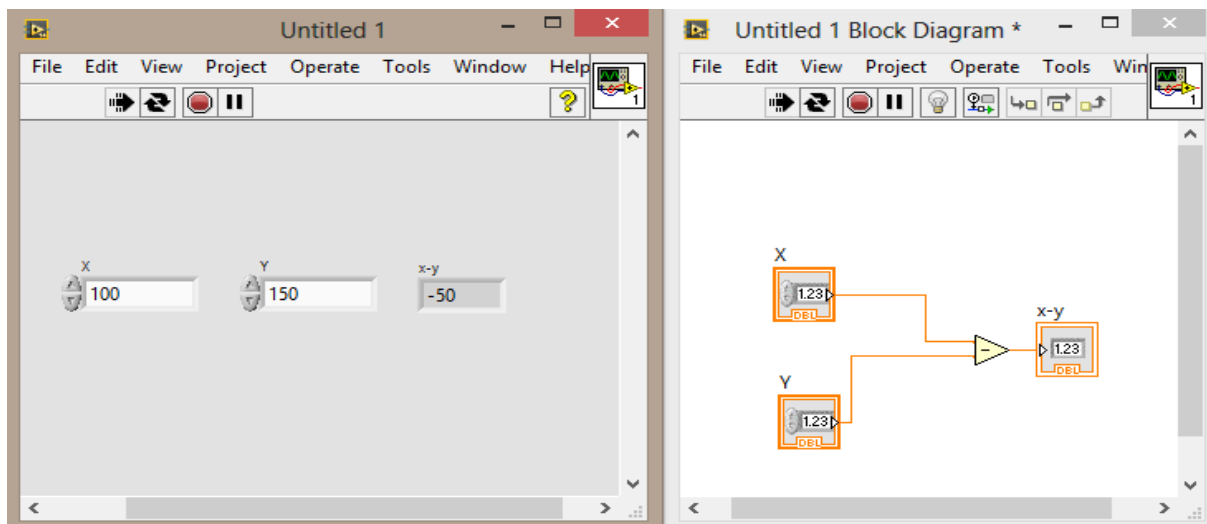
Face avant :

- bouton droit et on choisit commande numérique (2 fois)
- bouton droit et on choisit indicateur numérique

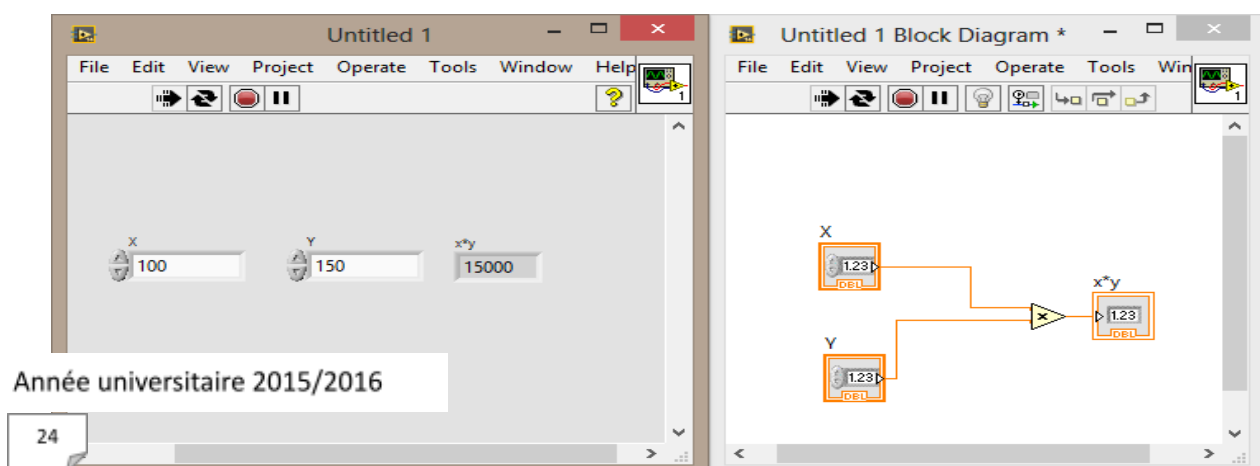
Diagramme :

- bouton droit ☐ numérique ☐ soustraire.
- Puis on relie entre les différentes commandes

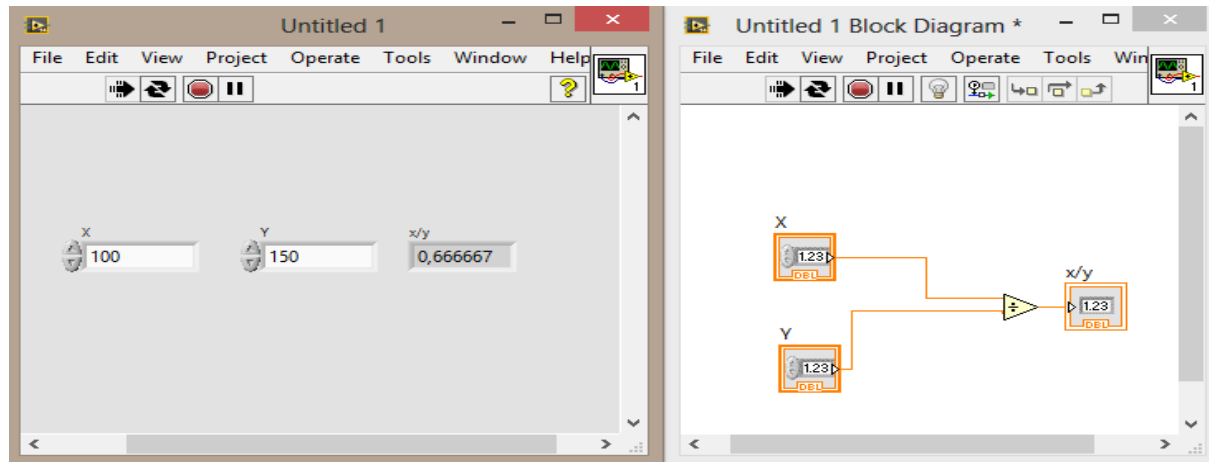
- Ensuite on exécute le programme en cliquant sur : 



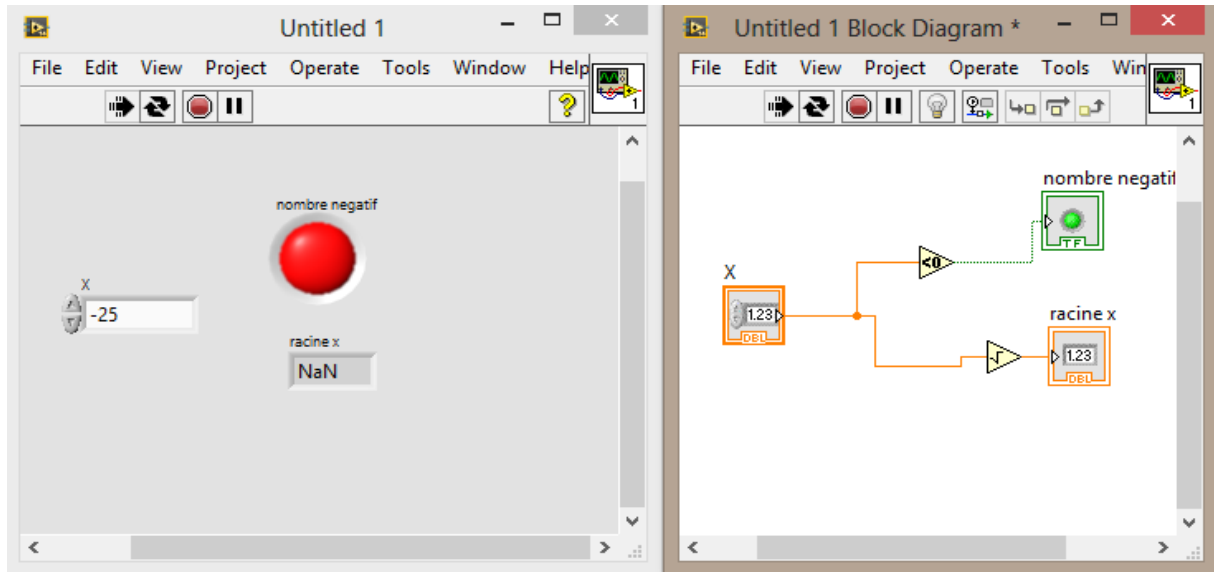
La multiplication



La division



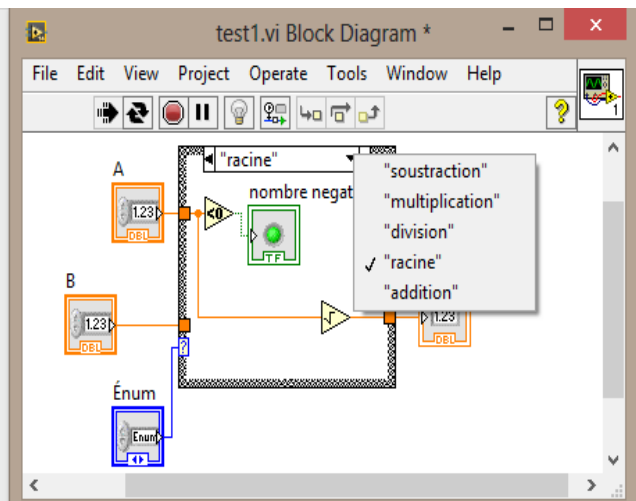
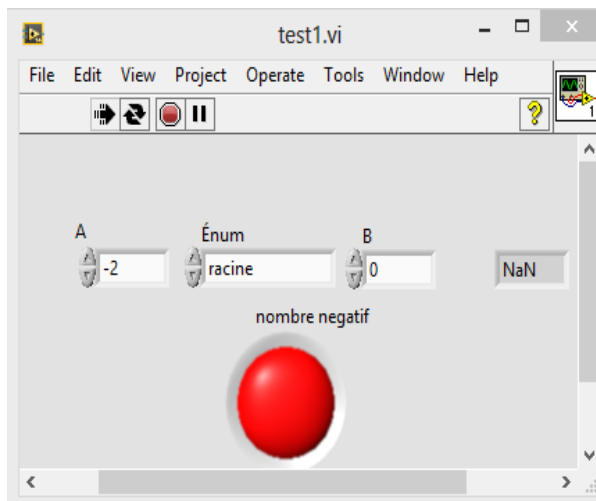
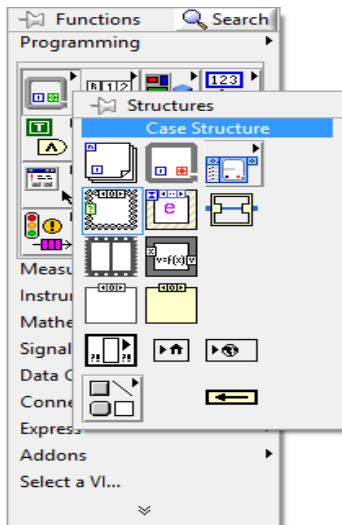
La racine : nous allons faire un test sur la valeur souhaiter pour voir si elle est négative
si c'est vrai alors une LED va s'allumer et indiquer que le nombre est négative.



Structure de choix

La structure de choix correspond à la structure « if...then...else » utilisée dans les langages textuels. Elle comporte un sélecteur de choix dans sa partie centrale supérieur (le « case » du C) et une entrée de sélection (le « Switch » du C) en forme de point d'interrogation. Les flèches de part et d'autre du sélecteur de choix, permettent de faire défiler les divers cas possibles. Le code de chaque cas est écrit dans la case correspondante, une seule case est visible à la fois à l'écran, elles sont détaillées à l'impression. Par défaut, l'entrée de sélection est booléenne, il est cependant possible d'y connecter des entiers, des énumérations (collection d'entiers et de labels disponibles dans Commande»Menu déroulant & enum) et des chaînes. Le sélecteur de choix s'adapte automatiquement au type d'entrée connectée.

Dans ce cas-là, nous allons assembler toutes les opérations précédentes dans un seul programme ainsi le choix de l'opération sera fait par l'utilisateur en ajoutant un énumérateur.

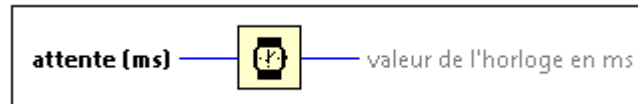


TP 2 : Opération et test avec les LEDs

Dans cette partie nous allons réaliser trois programmes :

1. 4 LEDs qui clignotent avec 4 fréquences différentes
2. 4 LEDs qui réaliseront un chenaillât avec une fréquence désiré par l'utilisateur
3. Des LEDs qui indiqueront le type d'un nombre soit hexadécimal, octal, ...

Dans toute cette partie, nous allons utiliser la fonction ATTENDRE

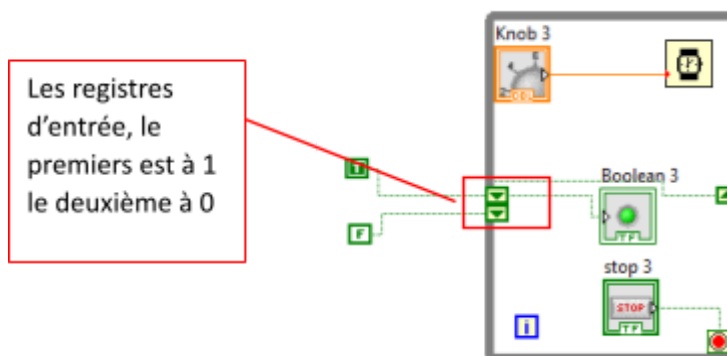


Elle attend le nombre de millisecondes spécifié et renvoie la valeur de l'horloge en millisecondes. Si vous câblez la valeur 0 à l'entrée **attente (ms)**, le thread actif cédera le contrôle au processeur.

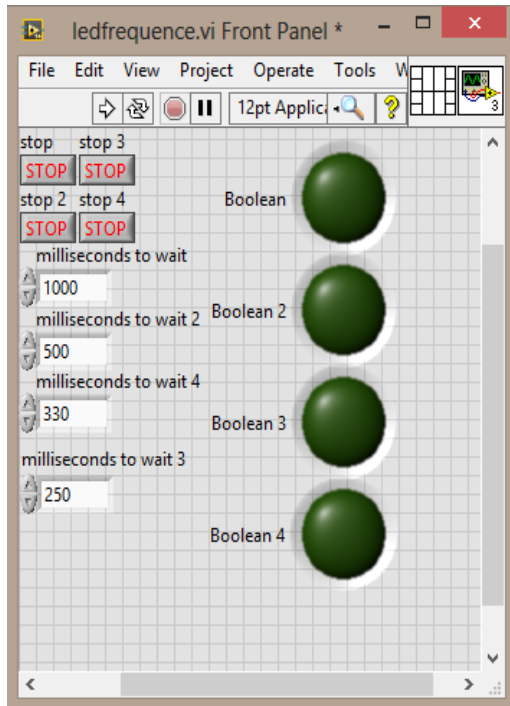
La fonction Attendre (ms) exécute des appels de système asynchrones, mais les nœuds fonctionnent de manière synchrone. Elle attend donc que le temps spécifié se soit écoulé pour terminer l'exécution.

Manipulation 1

On va utiliser une boucle WHILE, et on ajoutera deux registres déroulants à l'entrée et on va les initier à 1 et 0. Le registre à la sortie sert à basculer entre les valeurs des deux registres d'entrée.



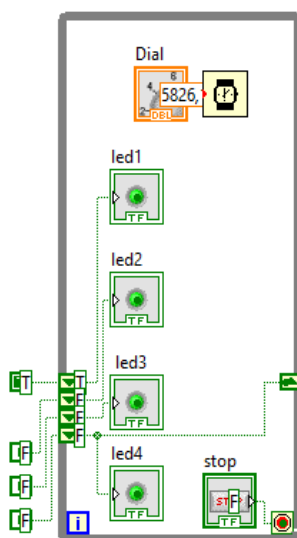
On va utiliser le même programme précédent 4 fois mais avec des temporisations différentes 1Hz, 2Hz, 3Hz et 4Hz. Nous allons avoir une face avant comme suit :



Sachant que la temporisation sur Labview fonctionne en (ms), donc pour avoir une fréquence de 2Hz il faut diviser a par 2 pour avoir le temps en milliseconde qui est 500 ms de même pour les 3 autres leds.

Manipulation 2

La réalisation du chenillard est simple, il fonctionne avec le même principe qui est de basculer les valeurs des registres d'entrée.

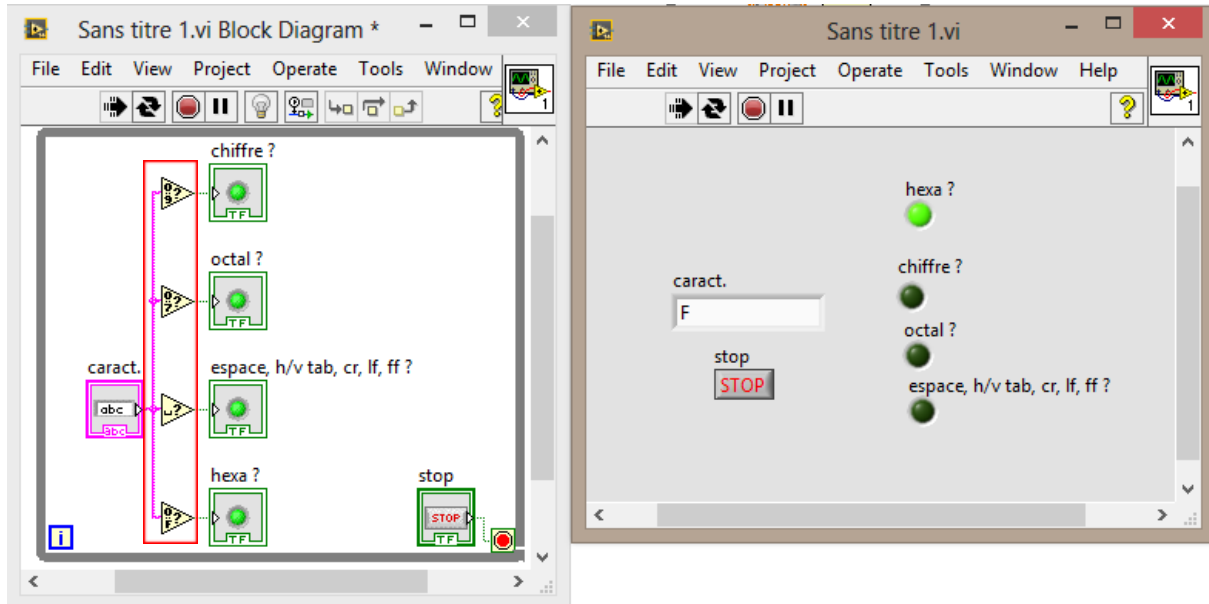


On va utiliser la simulation détaillé pour voir ce qui ce passe.

Dans la première itération on remarque que les 4 registres sont initiés à 1 0 0 0 donc c'est la led1 qui est allumé pendant la période désiré. L'itération qui suit la led2 sera allumée donc les registres seront à 0 1 0 0 et ainsi de suite.

Manipulation 3

Cette manipulation a pour but d'indiquer si un nombre est en hexa, octal, binaire ou bien espace, nous allons utiliser les testeurs encadrés en rouge dans la figure suivante



Le F est un nombre hexadécimal comme indiquer dans le programme précédent.



TP3 : Calcule matricielle a l'aide de LaBVIEW

Dans cette partie, l'objectif est de créer une interface sur labview permettant de faire les opérations d'addition et de multiplication entre deux matrices (grâce à un menu déroulant).

N.B : Pour la 1^{ère} matrice, calculer son déterminant et dès que celui-ci est égale à zéro, une led s'allumera pour le signaler.

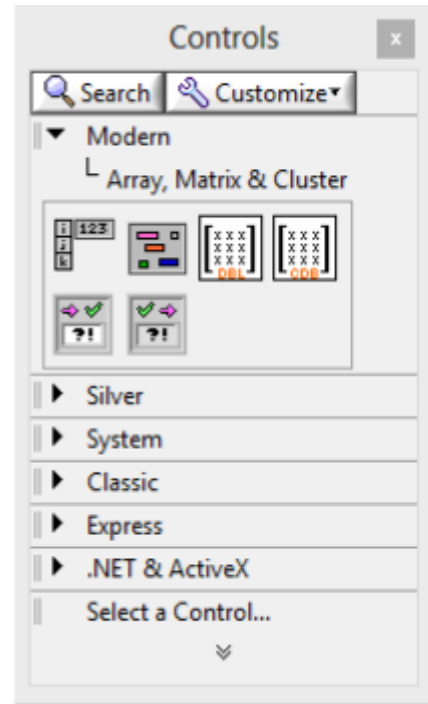
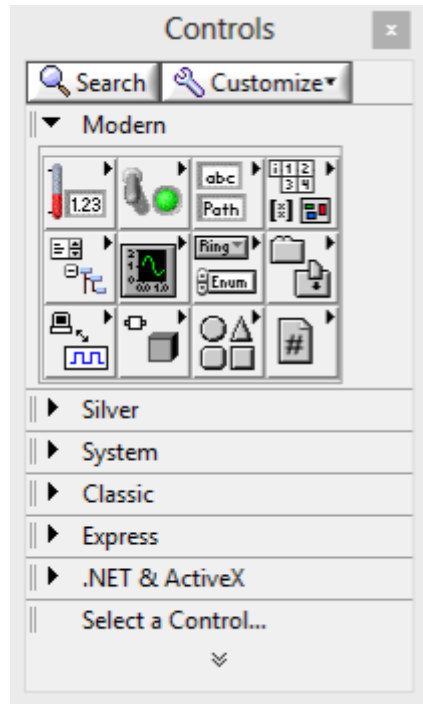
PROCEDURE DE TRAVAIL

Nous allons diviser notre travail en plusieurs parties (programme pour l'addition, programme pour la multiplication, calcul du déterminant et led) ainsi tous les assembler dans une boucle de condition afin de créer un menu permettant de les défiler.

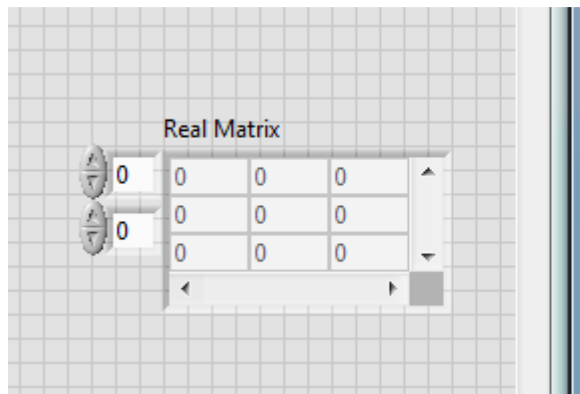
- Partie 1 : addition de deux matrices

Pour créer une matrice :

- Clic droit sur la face-avant de labview : apparition de la fenêtre CONTROLS
- Cliquer sur ARRAY, MATRIX, CLUSTER
- Sélectionner Matrix et poser là sur la face-avant de labview ; l'équivalent apparaîtra dans la partie diagramme de block.



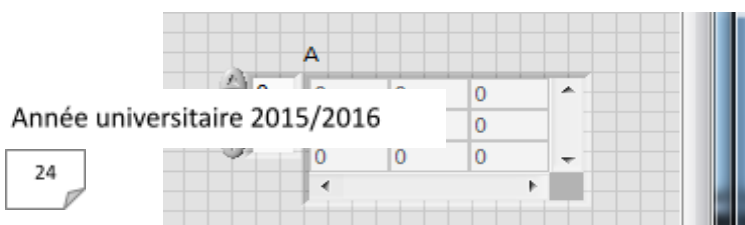
Matrice sur la face-avant et son équivalent sur le diagramme de block



Nous aurons besoin de 4 matrices, les deux matrices sur lesquelles nous allons effectuer les opérations et deux autres pour l'affichage de l'addition et la multiplication

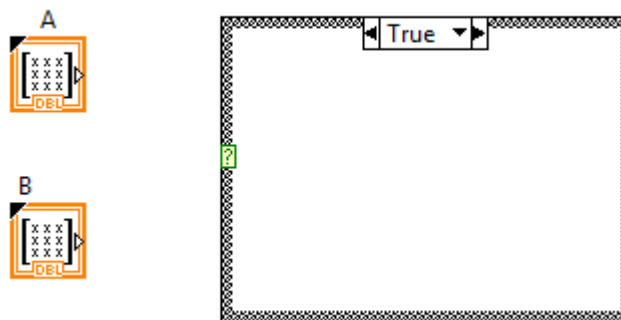
Etapes pour réaliser l'addition de deux matrices

1. Créer deux matrices



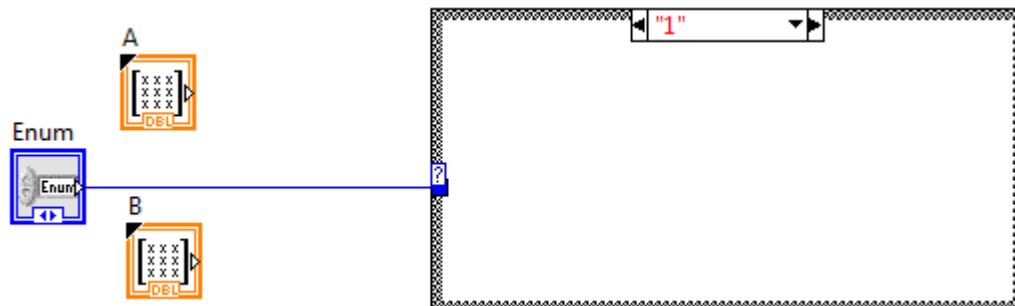
2. Créer une structure condition, en procédant de la manière suivante :

- Clic droit dans diagramme de block
- Cliquer sur structures
- Clic sur case structure, puis le poser sur le diagramme de block



Créer un menu défilant grâce à la fonction ENUM

- Clic droit dans diagramme de block
- Cliquer sur NUMERIC, puis le poser dans le diagramme de block
- Clic droit sur ENUM puis cliquer sur « change to control »
- Puis le relier au « case structure »



Pour insérer les noms des opérations dans le nom puis dans le diagramme voici la procédure à suivre :

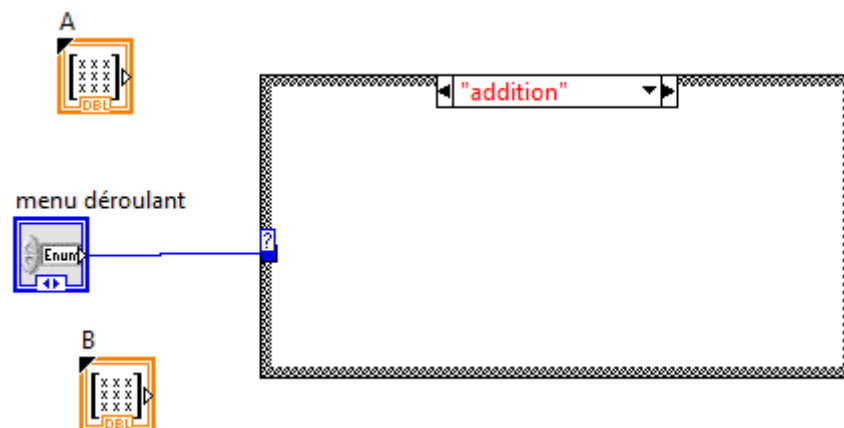
- Clic droit sur ENUM
- Cliquer sur Properties
- Cliquer sur Edit items
- Cliquer sur insert : pour insérer les opérations dans le menu
- Puis OK

Maintenant nous venons de créer un menu dans la face-avant de LabVIEW qui nous permettra de défiler entre les cas de la boucle condition.

Mais avant tout, il faudrait aussi définir les mêmes opérations dans la boucle ; pour ce faire il faudrait juste changer les labels de la structure (c'est-à-dire 1 et 0) par les labels de vos opérations cas vrai et faux par les noms « addition pour 1 et multiplication pour 0 » ;

En cas de plus de deux opérations, il suffit de faire clic droit sur la boucle et cliquer sur « **add case after** ».

Donc on obtient :

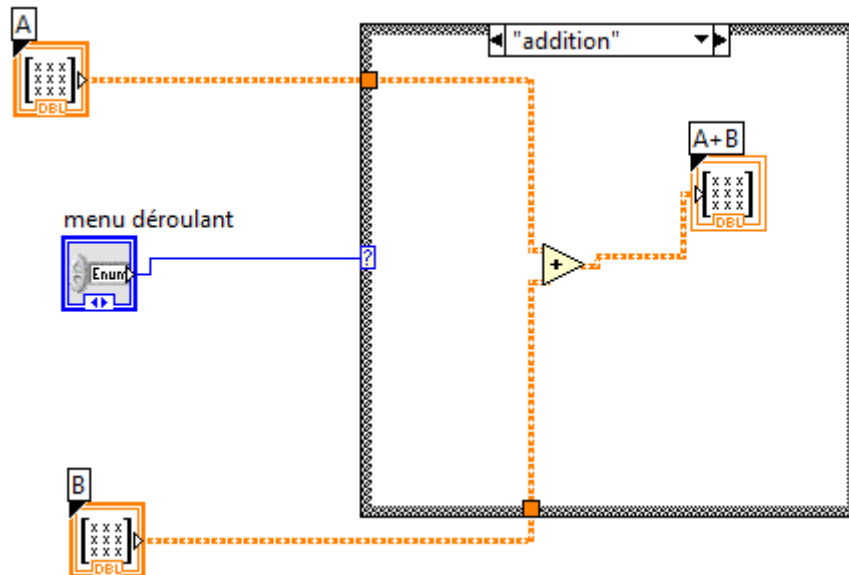


Plaçons maintenant notre opération  à l'intérieur et aussi une matrice de réception de l'opération effectuée

Pour insérer l'addition  :

- clic droit dans block diagram
- cliquer sur NUMERIC
- Cliquer sur « add »

Ensuite faire les liaisons, et on obtient



• Partie 2 : multiplication de deux matrices

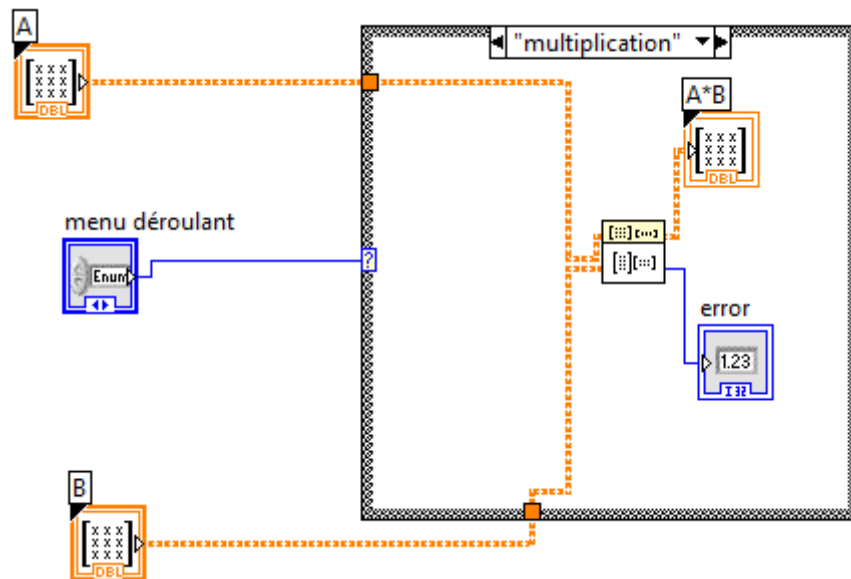
Dans la structure condition, passons du côté « multiplication » pour insérer la multiplication et une matrice de réception du produit.

Insertion de la multiplication :



- Clic droit dans le block diagram
- Cliquer sur mathematics
- Cliquer sur linear algebra
- Cliquer sur AxB

On obtient finalement



Pour la touche error , il faudrait faire un clic droit sur la sortie de la multiplication ,puis cliquer sur « create indicator »

- Partie 3 : inverse de la 1^{ère} matrice et allumage d'une led

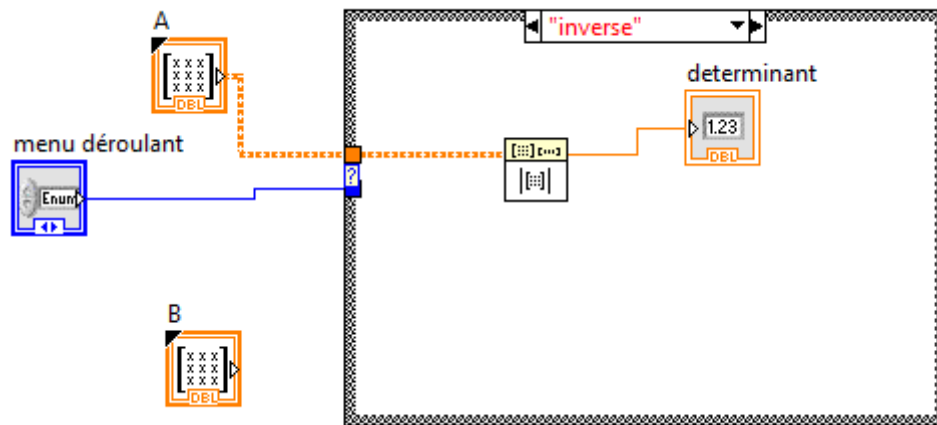
Même principe dans cette partie, insérons le calcul de déterminant et aussi un indicateur pour affiché le résultat ; et aussi une led pour vérification du résultat.

Pour insérer le calcul de déterminant :

- Clic droit dans le block diagram
- Cliquer sur mathematics
- Cliquer sur linear algebra
- Cliquer sur déterminant

Faire faire les liaisons

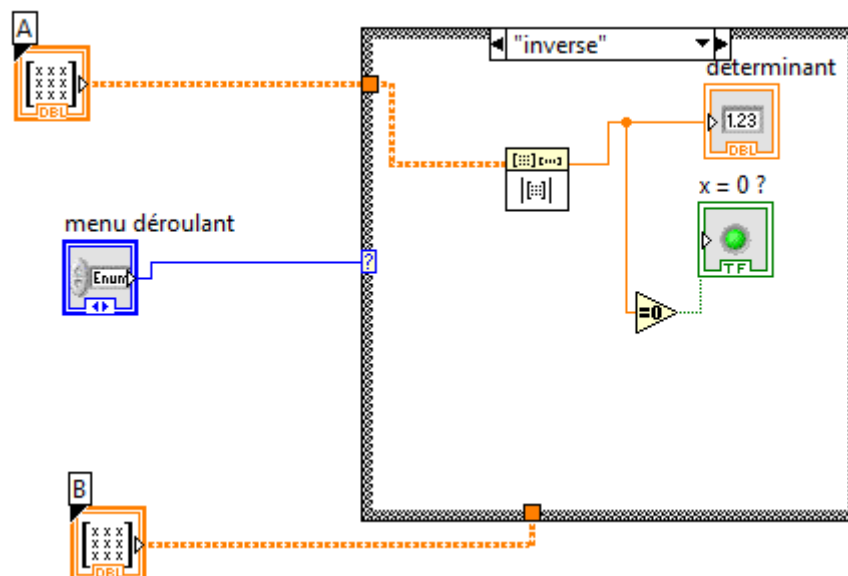
Puis clic droit sur la sortie de l'icône déterminant, ensuite cliquer sur « create indicator »



Puis pour la vérification du résultat par la led , Insérons tout d'abord une icône de vérification si le résultat est nul :

- Clic droit sur comparaison
- Cliquer sur « equal to 0 ? »
- Clic droit sur la sortie du comparateur
- Cliquer sur « create indicator »

Ainsi on obtient



INTERFACE HOMME MACHINE(FACE-AVANT)

menu déroulant
multiplication

A

0	0	0	0
0	0	0	0
0	0	0	0

A*B

0	0	0	0
0	0	0	0
0	0	0	0

déterminant
0,00E+0

x = 0 ?

B

0	0	0	0
0	0	0	0
0	0	0	0

A+B

0	0	0	0
0	0	0	0
0	0	0	0

Vérification de quelques résultats

menu déroulant
inverse

A

0	1	0	3
0	3	0	0
0	7	0	15

B

0	6	2	7
0	0	12	0
0	4	0	1

A*B

0	18	2	10
0	18	6	21
0	102	14	64

déterminant
0,00E+0

x = 0 ?

A+B

0	7	2	10
0	3	12	0
0	11	0	16

TP4 : Générateur de fonctions

Durant cette partie, nous allons créer des générateurs de signaux. C'est dernier sont un signal sinusoïdale un signal carré et un autre triangulaire, tout en utilisant les propriétés et fonctions de LabVIEW. Afin de réaliser cette manip on a trouvé deux méthodes à utiliser l'une est prédéfinie par labview mais l'autre est à programme soit même.

A l'aide des fonctions prédéfinies sur LabVIEW

Dans LABVIEW il existe des générateurs de fonctions définies dans le programme, ils génèrent des signaux carrés, sinusoïdaux et bien d'autre



: Générateurs de fonction sinusoïdale

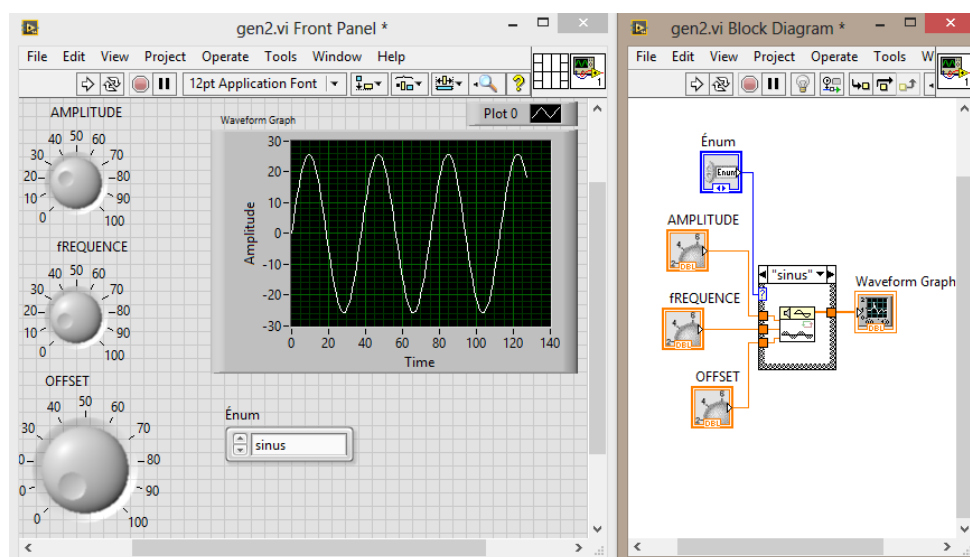


: Générateurs de fonction triangulaire

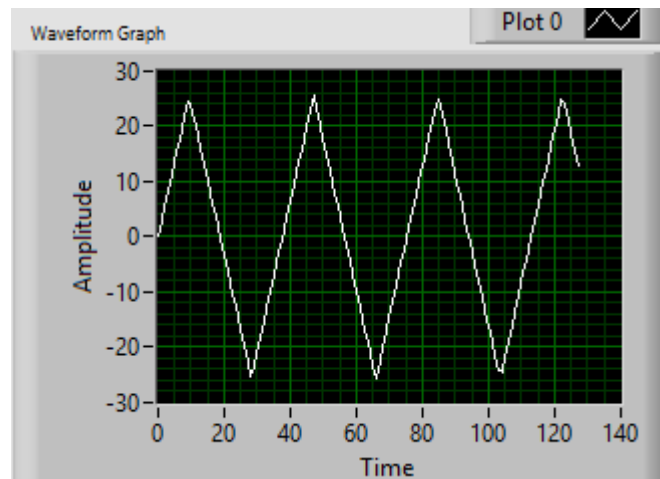


: Générateurs de fonction carrée

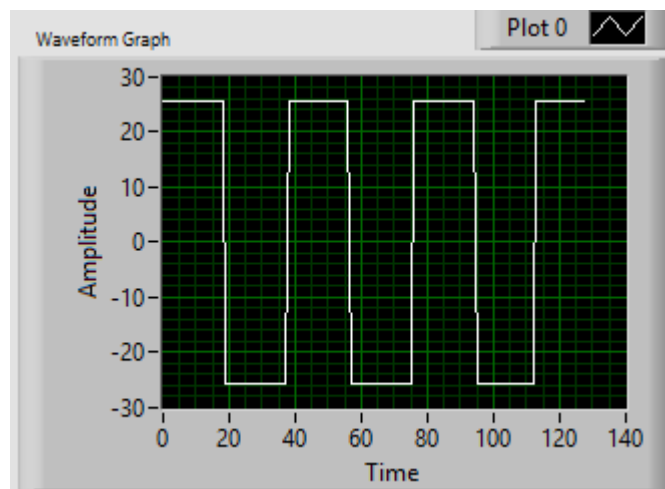
Nous allons ajouter les trois générateurs dans une boucle condition pour dérouler entre les trois



Triangulaire

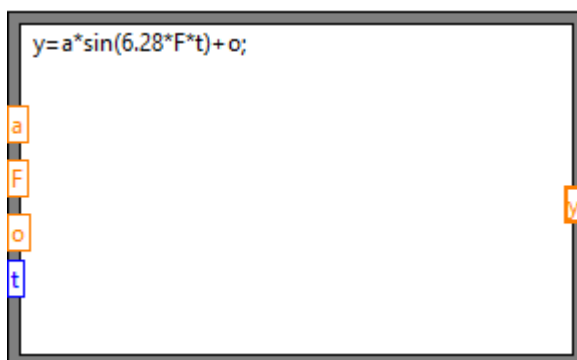


Carrée

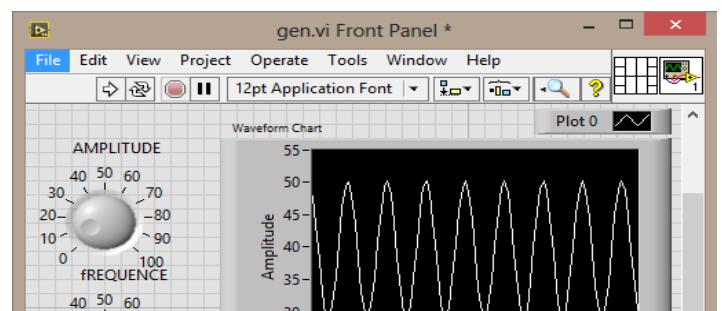
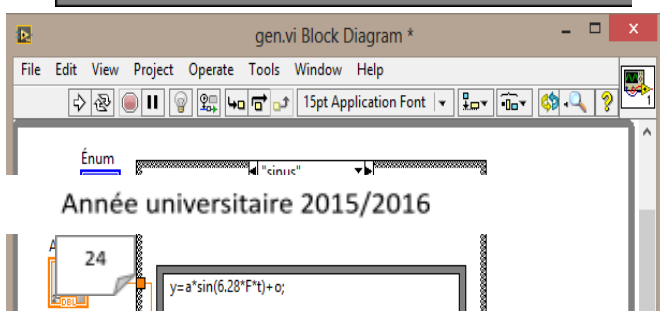


L'utilisation d'une boîte de calcul

Boîte de calcul : Évalue les formules et expressions mathématiques du diagramme dont la syntaxe est similaire au langage C. Cet outil va nous permettre de générer des fonctions plus complexes telles que la fonction carrée ou triangulaire.



On ajoutera des entrées A : l'amplitude, F : la fréquence, t : la variable et aussi o : l'offset





La face avant et le diagramme de block du générateur de fonction en utilisant la boîte de calcul.

Conclusion