



Sciences et technologies

de l'Industrie et du développement durable

ET 24 : Modèle de comportement d'un système

Boucles de programmation avec Labview.

Sciences et Technologies de l'Industrie et du Développement Durable <i>Formation des enseignants</i>	
parcours : ET24	Modèle de comportement d'un système
Durée : 3 h	
Objectif : Etre capable de modéliser des comportements de type WHILE, FOR, Séquences, conditions.	
Pré-requis : Les bases de Labview – Le VI	
Bases théoriques : Structures élémentaires de programmation	
Outil : Labview	
Support :	
Modalités : Activité sous forme de TD	
Synthèse et validation : Être capable de recréer en autonomie les modèles proposés.	



Sciences et technologies

de l'Industrie et du développement durable

Sommaire

1	Introduction.	3
2	La boucle WHILE.	4
2.1	<i>Propriétés élémentaires.</i>	4
2.2	<i>Registre à décalage.</i>	4
3	La boucle FOR.	6
3.1	<i>Propriétés élémentaires.</i>	6
3.2	<i>Le registre à décalage.</i>	6
4	La boucle de séquence.	8
5	Boucle condition.	11
5.1	<i>Boucle condition testant un booléen.</i>	11
5.2	<i>Boucle condition testant une grandeur.</i>	12
6	Détection de fronts.	15



Sciences et technologies de l'Industrie et du développement durable

1 Introduction.

Tout comme le C ou d'autres langages, Labview est un langage de programmation.

La différence majeure avec le C est que Labview est un langage de programmation graphique.

Cependant, les structures élémentaires de raisonnement logique sont identiques. Ce document a pour objectif de présenter comment ces structures élémentaires sont programmées avec Labview.



Sciences et technologies de l'Industrie et du développement durable

2 La boucle WHILE.

2.1 Propriétés élémentaires.

La boucle WHILE est une boucle classique dans Labview. La condition de fin a déjà été étudiée précédemment.

Il est possible d'utiliser l'information du nombre d'itérations de la

boucle au moyen de : 

2.2 Registre à décalage.

Vous allez maintenant créer une petite application permettant de faire défiler des LED à une fréquence voulue.

Pour cela, placez sur la face avant :

- un bouton rotatif gradué de 0 à 1000 ;
- quatre LED ;
- un bouton stop.

Placez sur le diagramme :

- une boucle WHILE ;
- une fonction attendre (ms) à partir de *Programmation / Informations temporelles / attendre (ms)* ;
- quatre constantes booléennes à l'extérieur de la boucle WHILE à partir de *Programmation / Booléen*.

Reliez le bouton rotatif à la fonction attendre, ainsi que le bouton stop à la condition fin de la boucle while.

Créez un registre à décalage au moyen d'un clic droit sur la boucle puis *Ajouter un registre à décalage*.

A l'aide de la souris, agrandissez les entrées du registre à décalage, à gauche, de manière à ce qu'elles soient au nombre de 4.

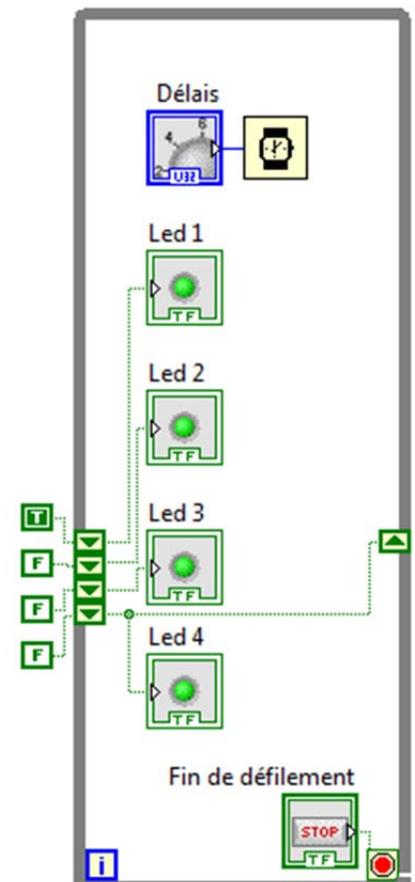
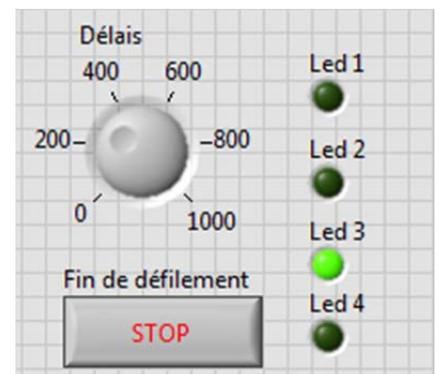
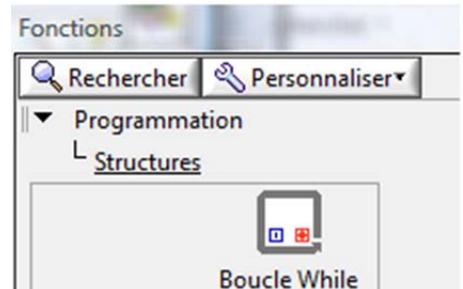
Reliez les constantes booléennes aux entrées du registre comme indiqué. Réglez la première à True, les autres à False par un clic.

A l'initialisation du VI, les entrées du registre, à gauche, s'initialisent avec les valeurs données hors de la boucle while.

Après l'initialisation, les valeurs d'initialisation n'ont plus aucun rôle dans le fonctionnement du registre.

Dans notre cas :

- L'entrée de Led 1 s'initialise à 1
- Les entrées de Led 2,3 et 4 s'initialisent à 0.

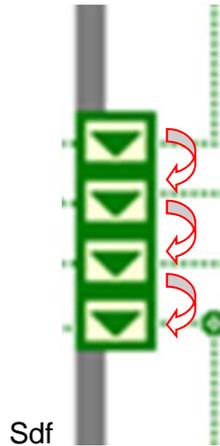




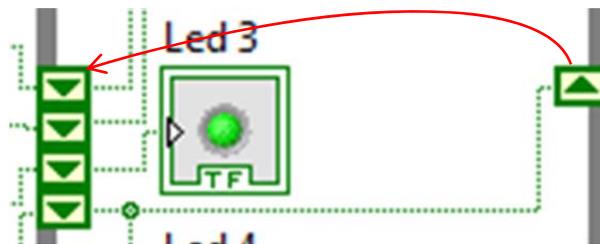
Sciences et technologies de l'Industrie et du développement durable

Par la suite, à la fin de chaque itération, il se passe deux évènements.

- Les entrées du registre à décalage se décalent d'un cran vers le bas.



- La première entrée du registre à décalage est affectée de la valeur de sortie du registre :



Reliez la dernière entrée du registre à décalage à la sortie comme indiqué sur la page d'avant.

Exécutez le VI et admirez le résultat !



Sciences et technologies de l'Industrie et du développement durable

3 La boucle FOR.

3.1 Propriétés élémentaires.

Dans Labview, la boucle FOR exécute ce que l'on attend d'elle dans une structure de programmation simple. A savoir : exécuter une boucle N fois.

Une des applications les plus simples que l'on peut écrire avec la boucle FOR est d'afficher à tout moment le nombre de boucles exécutées.

Le nombre de boucles programmées est câblé en haut à gauche devant le **N**.

L'index d'exécution de boucles est câblé derrière le **i**. Cet index commence à 0 et fini à N-1.

La condition d'arrêt de boucle n'est pas affichée par défaut. Il est possible de l'ajouter au moyen d'un clic droit sur le bord de la boucle, puis en cochant *Terminal de condition*.

Dans ce diagramme, une horloge a été rajoutée afin d'être en mesure de visualiser le déroulement de la séquence.

Câblez cette boucle FOR et observez le résultat.

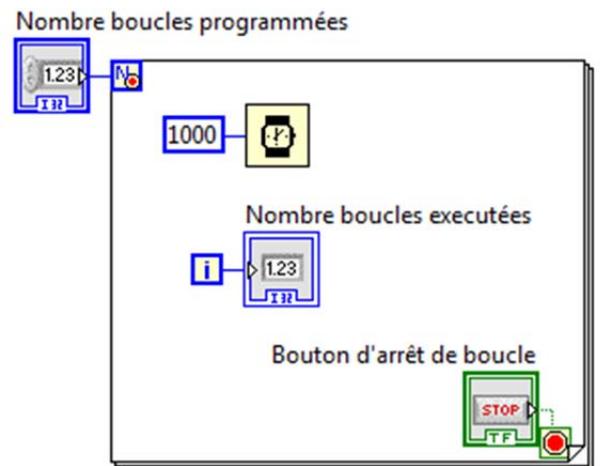
3.2 Le registre à décalage.

Il est aussi possible de rajouter des registres à décalage pour une boucle FOR. Pour illustrer cette fonctionnalité, imaginons que l'on ait besoin de remplir un tableau de 9 lignes avec des nombres aléatoires. Pour les besoins d'un test par exemple.

Insérez une boucle FOR sur un nouveau VI. Ajoutez lui une horloge d'attente de 1000 ms pour que l'on puisse visualiser ce qui se passe, puis un registre à décalage. Pour l'instant, vous obtenez la figure ci-contre sur votre diagramme.

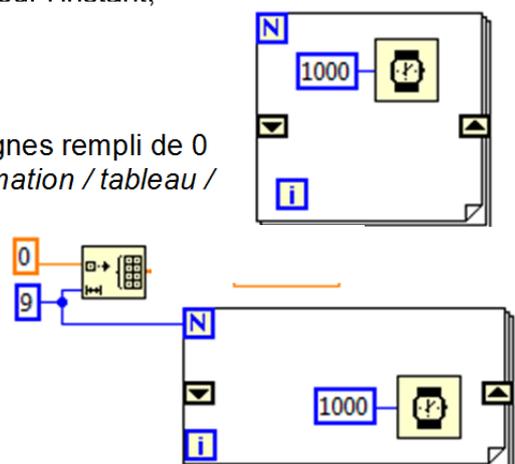
En dehors de la boucle, déclarez la création d'un tableau de 9 lignes rempli de 0 en dehors de la boucle. Pour cela, il utilise la fonction *Programmation / tableau / initialiser*. Reliez le nombre de lignes du tableau au nombre d'itérations à effectuer.

Vous obtenez le diagramme ci-contre.



Nb de boucle à exécuter

- Éléments visibles
- Aide
- Exemples
- Description et info-bulle...
- Point d'arrêt
- Palette Structures
- Redimensionnement automatique
- Exclure du nettoyage du diagramme
- Terminal de condition





Sciences et technologies de l'Industrie et du développement durable

Servez-vous du tableau que vous venez de créer pour initialiser le registre à décalage.

Afin de remplir le tableau avec des nombres aléatoires, placez une fonction de remplissage de tableau dans la boucle FOR. Cette fonction se trouve dans *Programmation / Tableau / Remplacer une portion d'un tableau*.

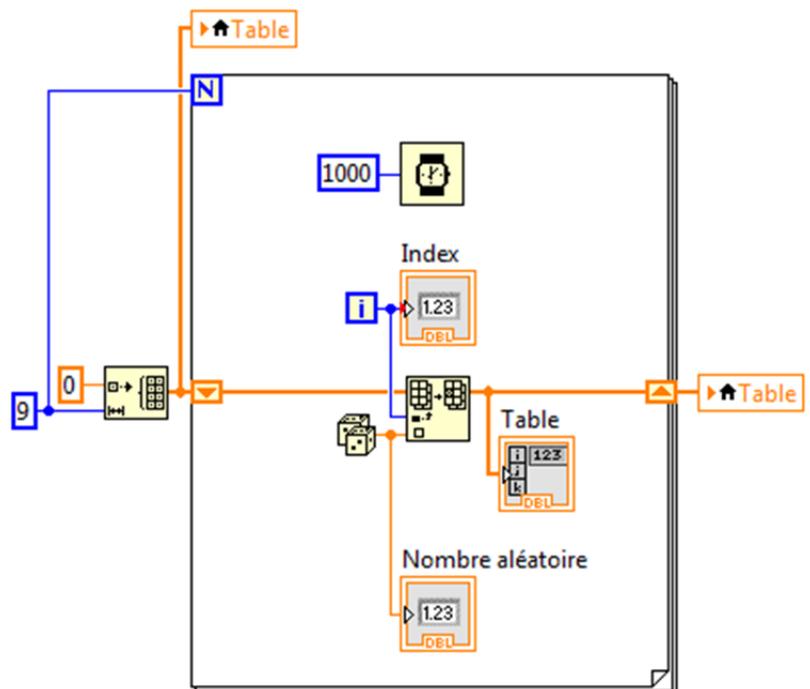
Servez-vous de l'index de la boucle pour indexer la ligne à modifier.

Utilisez la fonction de nombres aléatoires pour remplir les lignes du tableau. Cette fonction se trouve dans *Programmation / Numérique / Nombre aléatoire*.

Placez des indicateurs permettant de visualiser les grandeurs manipulées.

Seulement après l'étape précédente, stockez le tableau créé dans une variable locale. Pour cela, utiliser la fonction *Programmation / Structures / Variable locale*, puis faire un clic droit sur la variable locale pour choisir la variable à associer (tableau).

Stockez le tableau modifié dans la variable locale Table située à l'extérieur de la boucle FOR. Pour cela, procédez comme précédemment. Mais il faut préciser en plus que l'accès à la variable est effectué en écriture : clic gauche sur la variable, puis *Changer en écriture*.



Vous pouvez obtenir un diagramme du type de celui ci-dessus.

Dans la face avant, il est nécessaire d'agrandir le tableau avec les poignées de redimensionnement afin de visualiser toutes les lignes du tableau.

Lancez l'exécution du VI afin de visualiser les résultats.

Table	Index	Nombre aléatoire
0	8	0,35899€
0,69155€		
0,01948€		
0,13016€		
0,42350€		
0,11544€		
0,42555€		
0,31971€		
0,95770€		
0,35899€		



Sciences et technologies de l'Industrie et du développement durable

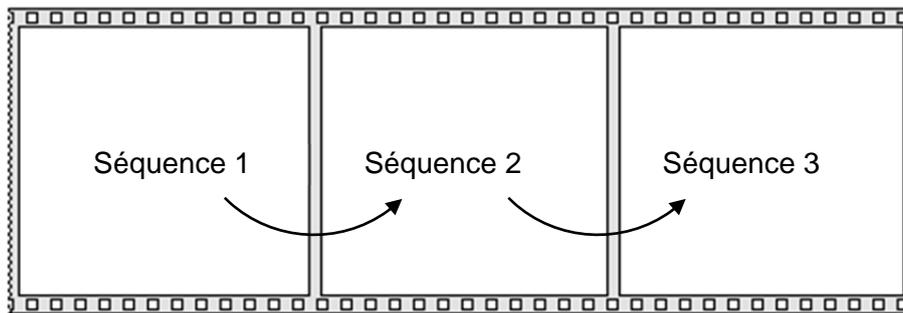
4 La boucle de séquence.

Il peut être intéressant de séquencer un diagramme, ne serait-ce que pour améliorer sa lisibilité du diagramme.

Il serait pertinent de développer un exemple spécifiquement pour cette nouvelle notion. Mais dans un souci d'efficacité, nous allons réécrire l'exemple ci-dessus en séparant le fonctionnement en trois étapes explicites :

- initialisation du tableau ;
- écriture du tableau ;
- lecture du tableau.

Pour cela, il faut utiliser une boucle de séquence.



Créez une boucle de « séquence déroulée » sur un nouveau VI. Ajoutez tout de suite deux séquences par un clic droit sur le bord, puis *Ajouter une séquence après*.

Dans la première séquence, créez un tableau de 9 lignes rempli de 0. Ce tableau est stocké dans une variable locale, et visualisé dans un VI. Vous devez obtenir :



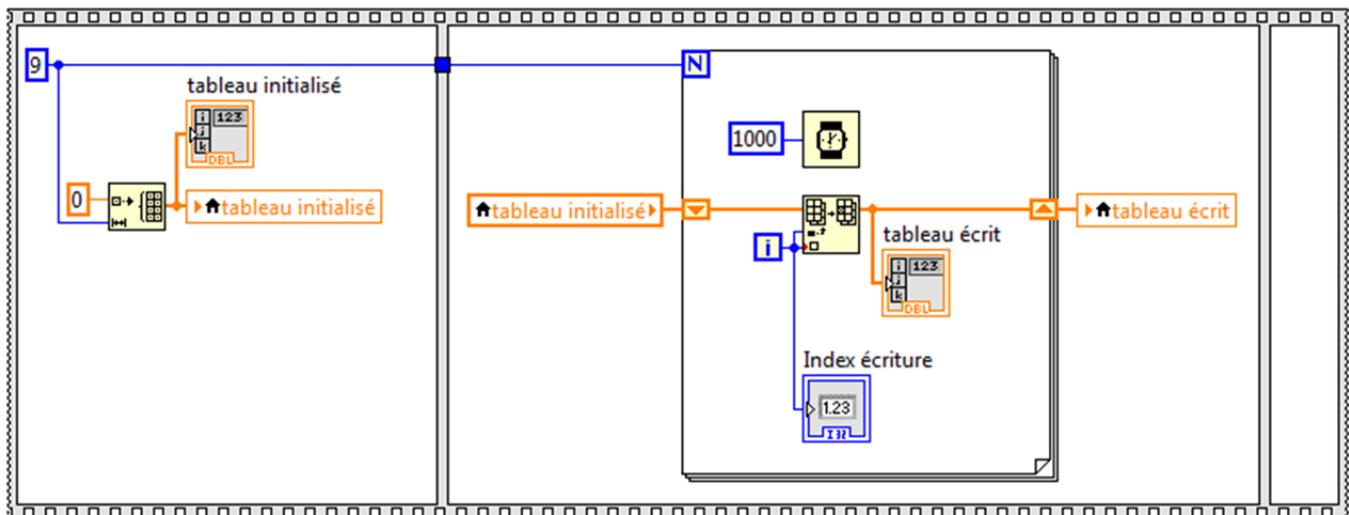


Sciences et technologies de l'Industrie et du développement durable

La seconde séquence traite de l'écriture du tableau. Dans cette seconde séquence :

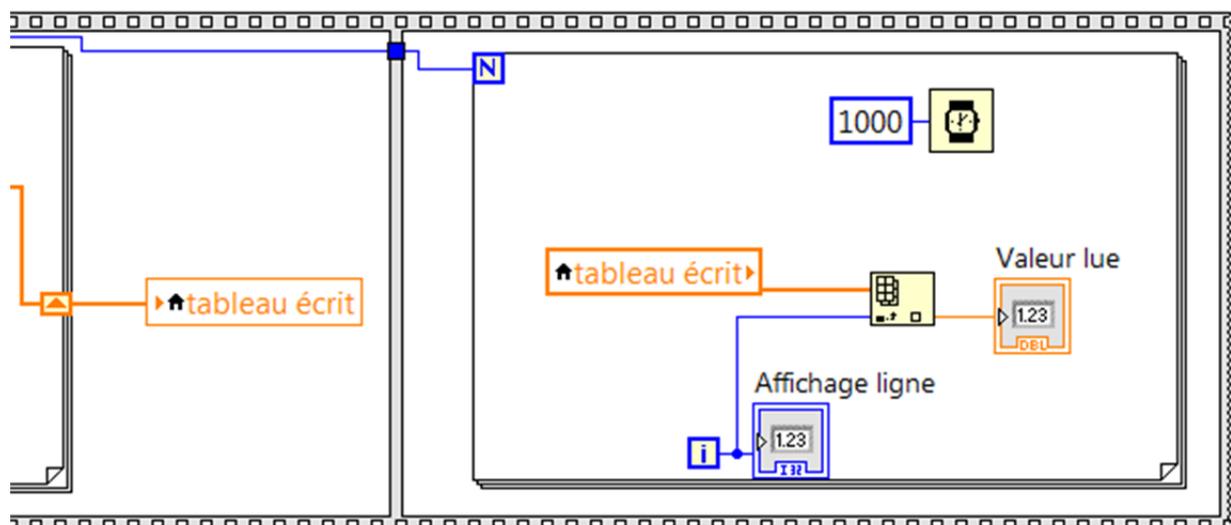
- Placez une boucle FOR en câblant N avec le nombre de lignes du tableau.
- Placer un registre à décalage.
- Placez une variable locale *Tableau initialisée*. Cette variable devant être lue, il faut faire un clic droit dessus, puis sélectionner *Changer en lecture*.
- Traitez l'écriture du tableau comme précédemment. Cette fois-ci, il est possible d'écrire le numéro de ligne dans la ligne.
- Placez un indicateur *Tableau écrit*, et un indicateur *Index écriture*.
- Ecrivez le résultat dans la variable locale Tableau écrit.

Le résultat doit fonctionner de la même manière que le diagramme ci-dessous :



La troisième séquence traite de la lecture du tableau. Complétez-la de la manière suivante :

- Placez la variable *Tableau écrit* en lecture.
- Placez une boucle FOR en câblant le nombre d'itérations et en la retardant.
- Pour lire le tableau, utilisez la fonction Indexer, qui se trouve dans *Programmation / Tableau / Indexer*.
- Placez un indicateur *Index Lecture* pour la ligne lue et *Valeur Lue* pour la lecture.





Sciences et technologies de l'Industrie et du développement durable

La face avant devrait fonctionner comme celle-ci-dessous :

tableau initialisé		tableau écrit			
0	0	0	0,544929	8	Index écriture
	0		0,689198		
	0		0,571343		
	0		0,559499		
	0		0,301057	Index lecture	Valeur lue
	0		0,537057	8	0,635908
	0		0,812373		
	0		0,38639		
	0		0,635908		

L'avantage de cette méthode est qu'elle sépare les processus d'initialisation, d'écriture et de lecture.



Sciences et technologies de l'Industrie et du développement durable

5 Boucle condition.

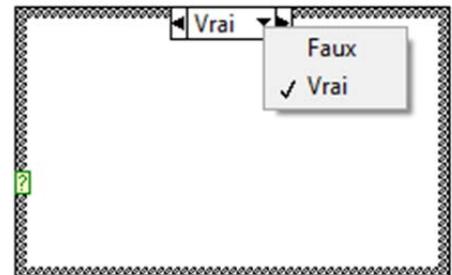
Un programme est souvent amené à prendre des décisions. Il est toujours possible de câbler le processus de décision avec des éléments de logique combinatoire ou séquentielle. Mais Labview offre une possibilité d'écrire cela à un niveau permettant une relecture plus facile du programme : la boucle condition.

5.1 Boucle condition testant un booléen.

Le premier exemple pour initier à la boucle condition sera trivial : si une valeur binaire est vraie, une Led s'allume, si elle est fausse, la Led s'éteint.

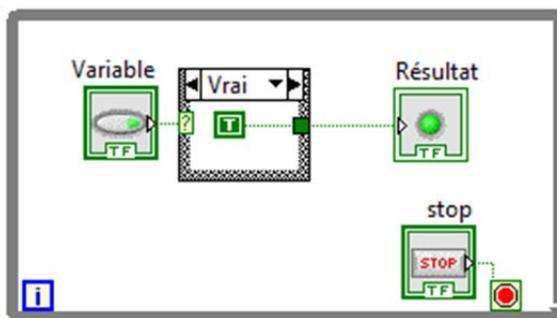
Ouvrez un nouveau VI et insérez-y une boucle condition se trouvant à *Programmation / Structures / Condition*.

Remarquez que la boucle attend le câblage d'une variable à tester à gauche. La boucle présente un menu préparé Vrai ou Faux.

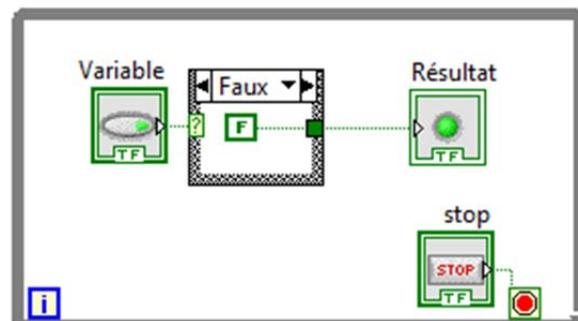


Placez une commande en amont du test. Placez un indicateur de résultat. Placez une boucle WHILE autour. Placez une commande de fin de WHILE.

Dans le cas VRAI, reliez une constante TRUE au résultat. Dans le cas FAUX, reliez une contante FALSE au résultat.



Cas « Vrai »



Cas « Faux »

Organisez la face avant et testez-la :



Il peut paraître curieux d'utiliser une boucle condition pour un résultat aussi simple, mais le traitement à l'intérieur de la boucle condition peut être beaucoup plus complexe.



Sciences et technologies de l'Industrie et du développement durable

5.2 Boucle condition testant une grandeur.

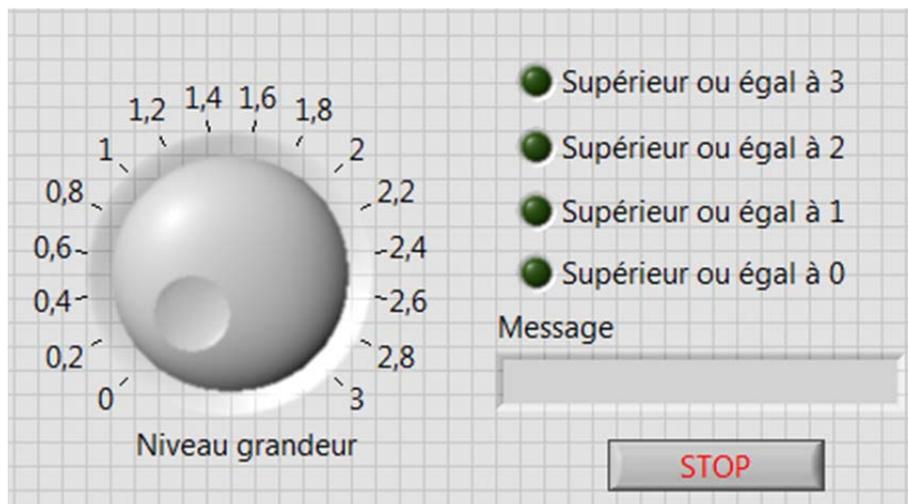
Il est possible de tester d'autres valeurs au moyen de la boucle condition, comme les entiers ou les chaînes de caractère.

Le cas développé ici présente la surveillance d'une grandeur physique pouvant comporter des niveaux de 0 à 4.

Sur une nouvelle face avant, placez :

- un bouton rotatif dont l'échelle est étendue de 0 à 3 ;
- quatre LED indiquant un niveau supérieur à 0, 1, 2, 3 ;
- un indicateur chaîne de caractères ;
- un bouton stop.

La face avant peut ressembler à l'exemple ci-dessous :



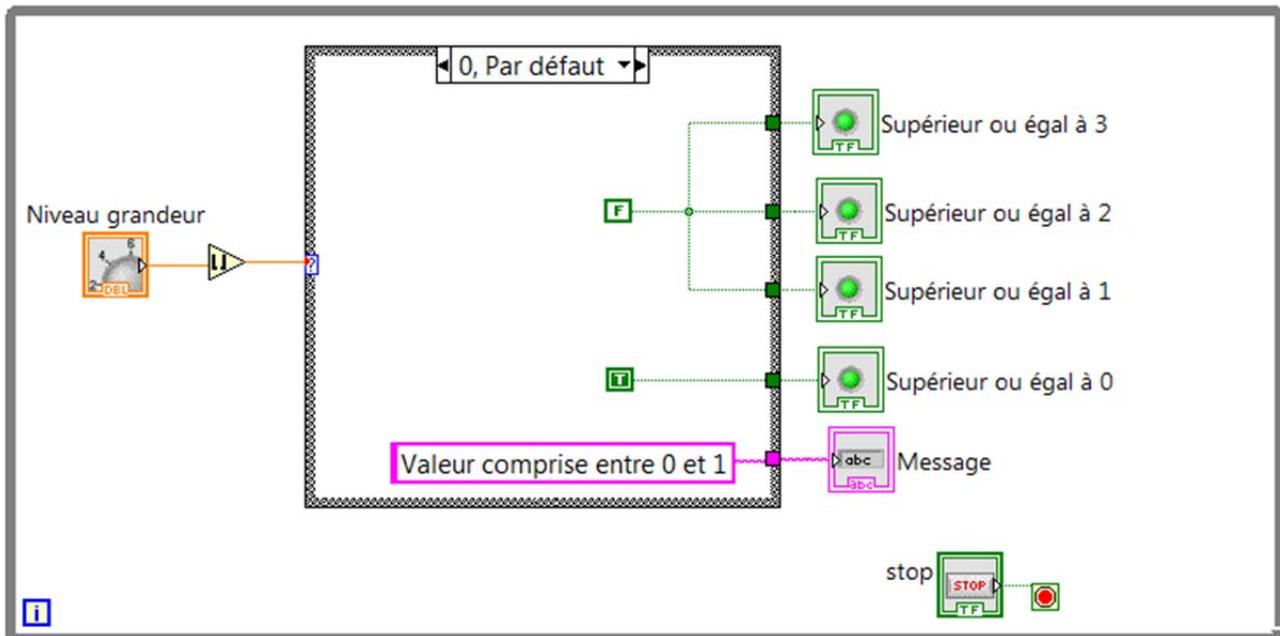


Sciences et technologies de l'Industrie et du développement durable

Le diagramme ci-dessous doit tester le niveau de la grandeur physique mesurée. (Dans un cas réel, le bouton rotatif serait remplacé par l'entrée d'une carte d'acquisition).

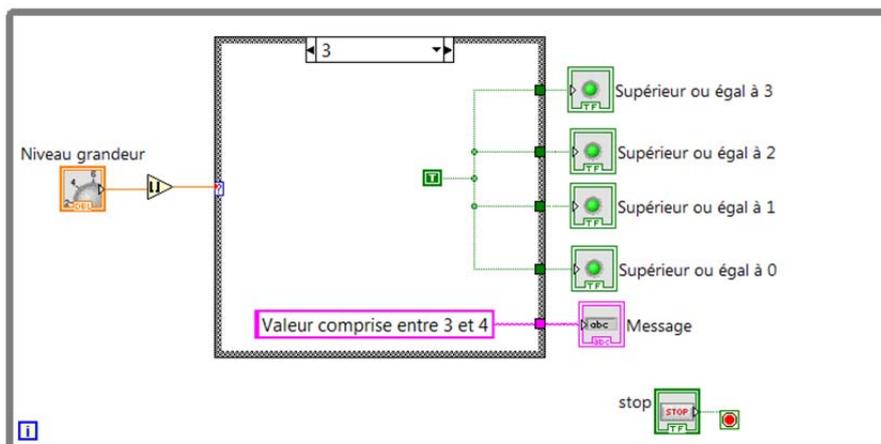
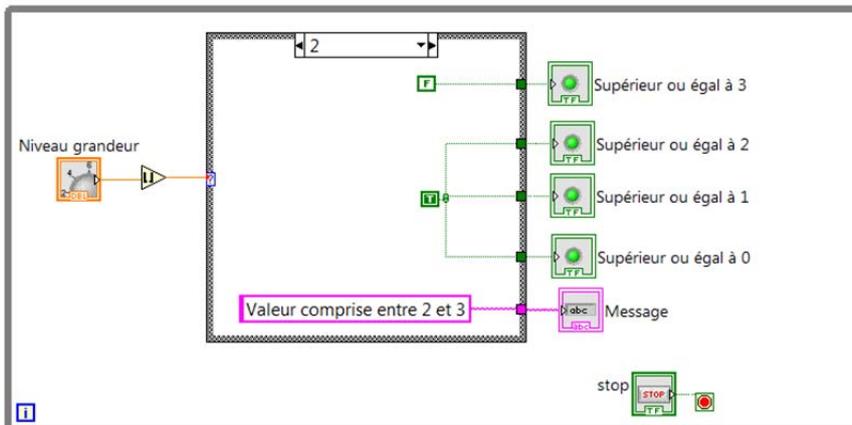
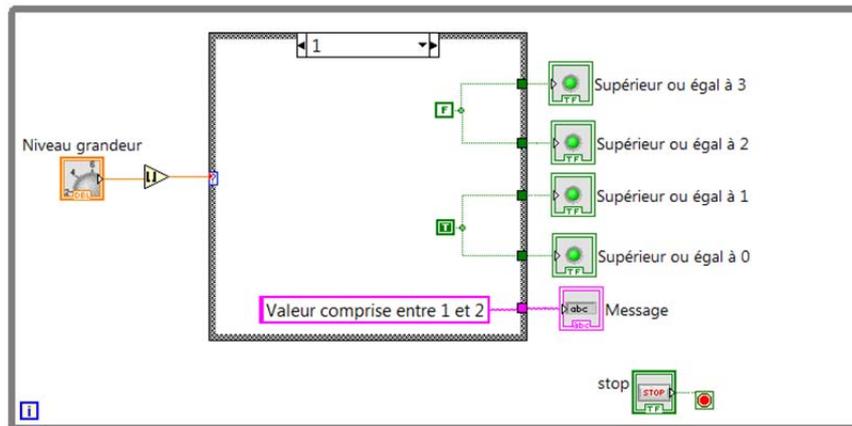
Pour cela :

- Brancher le bouton rotatif à l'entrée de la boucle condition en arrondissant à l'entier inférieur. Le bloc concerné se trouve dans *Programmation / Numérique / Arrondir vers -l'infini*.
- Pour le cas « 0 » : placer quatre constantes booléennes dans la boucle condition. Les régler comme indiqué ci-contre et les câbler aux Led. Placer de même la chaîne de caractères (*Programmation / chaîne / constante chaîne*) et la câbler au message.
- Pour les cas 1, 2 et 3, câbler comme indiqué dans les diagrammes suivants.
- Placer une boucle WHILE et câbler le bouton STOP.





Sciences et technologies de l'Industrie et du développement durable



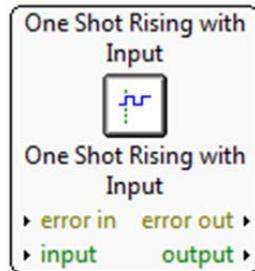
Testez le fonctionnement du VI.



Sciences et technologies de l'Industrie et du développement durable

6 Détection de fronts.

Les fonctions de détection de fronts se trouvent à *Real-Time / Fonction bloc / Edge detect*.



Exercice : Une grandeur physique à mesurer peut varier entre 0 et 100.

- Afficher un message d'avertissement persistant lorsque la grandeur passe au-dessus de 80.
- Afficher un message d'avertissement persistant lorsque la grandeur passe au-dessous de 50.
- Ne plus afficher les messages lorsque l'on appuie sur un bouton « Acquiescement des avertissements ».