Programmation

LabVIEW®

Recueil de Travaux dirigés

Proposés par Michel Fiocchi

22 novembre 2002

Michel Fiocchi

Responsable de l'option Ingénierie des Systèmes Embarqués *fiocchi@ismea.fr*

ISMEA
Technopôle de Château Gombert
13451 Marseille Cedex 20 *www.ismea.fr*

Table des matières

Présentation

L'ensemble des travaux dirigés proposés ci-après permettent de balayer les notions nécessaires à la programmation graphique dans un environnement LabVIEW™.

Ils se présentent sous une forme commune:

- Une page d'introduction

Présente le thème du TD et les points abordés.
C'est une invitation à la consultation de l'aide et des exemples proposés en ligne.

- Un ou plusieurs exercices avec:
  - ✔ Une partie d'énoncé

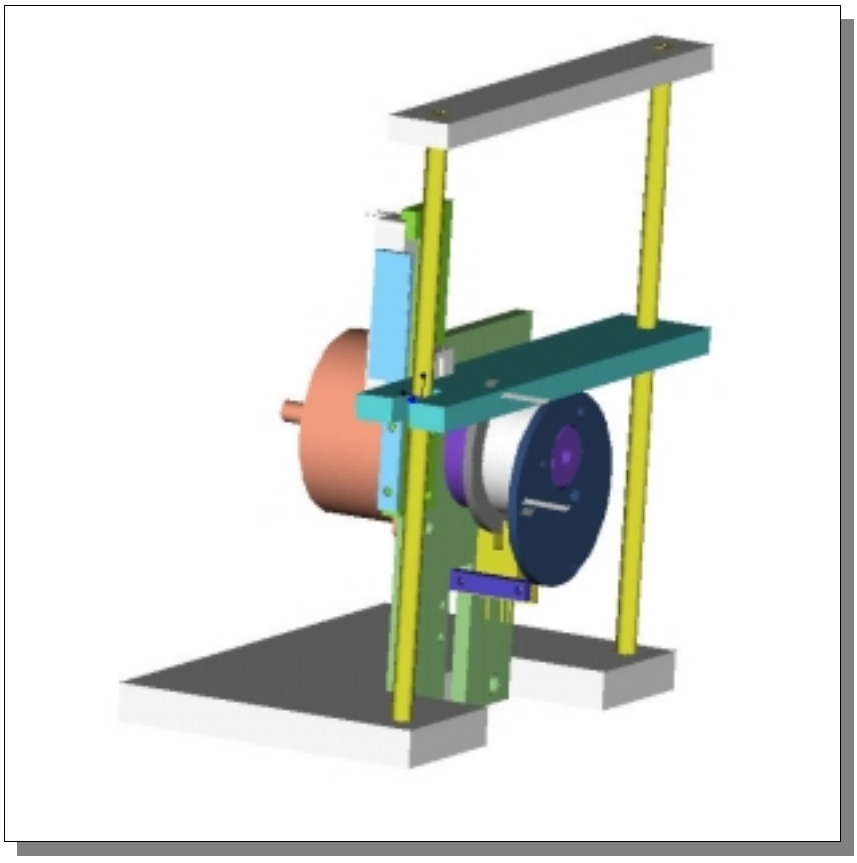
  - ✔ suivie d'une partie d'analyse

  - ✔ puis d'une partie solution .

Tous les exemples proposés ci-après sont issus des études de faisabilité ou de prototypage relatives à la réalisation du projet « Banc de mesure d'excentrique » proposé. Nous nous appliquerons plus particulièrement à l'étude d'un module permettant de tester et valider le fonctionnement de la partie matérielle du banc.

Le banc de mesure à réaliser est constitué d'une partie mécanique, d'une carte de contrôle/commande et d'un logiciel de gestion s'exécutant sur un ordinateur de pilotage.

La partie mécanique, décrite schématiquement ci-dessous, comprend un moteur assurant le déplacement de l'excentrique, le capteur de position angulaire associé et un capteur de déplacement mesurant la variation de distance à l'axe de rotation par rapport à une valeur de référence.



La carte de contrôle/commande assure les fonctions d'interface entre le logiciel de gestion d'une part et le groupe moteur / capteurs d'autre part. Elle communique avec l'ordinateur par le biais d'une liaison série de type RS232 et suivant un protocole maître/esclave à définir.

Bibliographie

COTTET F. - *LabVIEW, Programmation et application.*
Dunod, 2001

JOHNSON G. - *LabVIEW, Graphical Programming Practical Application in Instrumentation and Control.*
McGraw Hill, 1997

NATIONAL INSTRUMENTS. - *Initiation à LabVIEW.*
National Instruments,Sept 2000

NATIONAL INSTRUMENTS. - *Cours Basic I et II.*
ftp://ftp.ni.com/pub/training/

TD1: Création d'un VI

Dans le cadre du développement du logiciel de pilotage d'un banc de contrôle dimensionnel d'un excentrique, nous allons étudier un simulateur remplaçant la partie matérielle du projet. Pour ce premier exercice nous allons mettre en place les fonctionnalités de base de ce simulateur.

Nous allons introduire les notions de base sur:

- Les 3 éléments d'un VI
- Les Contrôles et Indicateurs
- Les outils d'édition
- Les sous-VI

Simul_0.vi

Exercice N° 1: Simulateur

A partir d'une commande de position angulaire (*P* exprimée en degrés par pas de 1.8°) le programme doit calculer la position atteinte (*A* valeur angulaire modulo 360°) et une mesure de la variation relative du rayon (ΔR exprimée en millimètre), donnée pour un excentrique de diamètre 100 mm et dont l'excentration (*E*) est de 10 mm avec un angle de déphasage (*D*) de 30°.

$$\Delta R = E \cdot \sin(P + D)$$

☑ **Analyser** le problème en terme de flux de données.

☑ **Mettre en place** les éléments de la face.

☑ **Enregistrer sous** le nom Simul_0.vi

☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Quotient & Reste
Calcule le quotient entier et le reste des entrées.

Sinus
Calcule le sinus de x, avec x en radians.

☑ Sauvegarder

Simul_0.vi

Variables d'entrées:

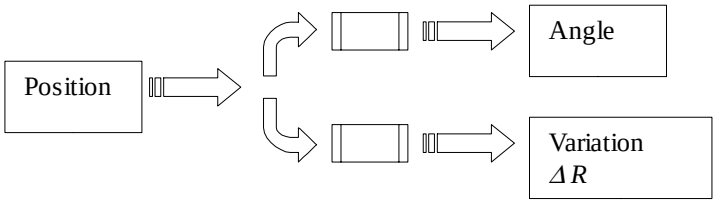
P		**Position angulaire à atteindre exprimée en degrés.**					
Sous titre		Position		*type*	double		
				apparence	standard		
défaut	0	*min*		*max*		*incr*	1.8

Variables de sortie:

A	**Valeur angulaire modulo 360°**		
Sous titre	Angle	*type*	double
		apparence	standard

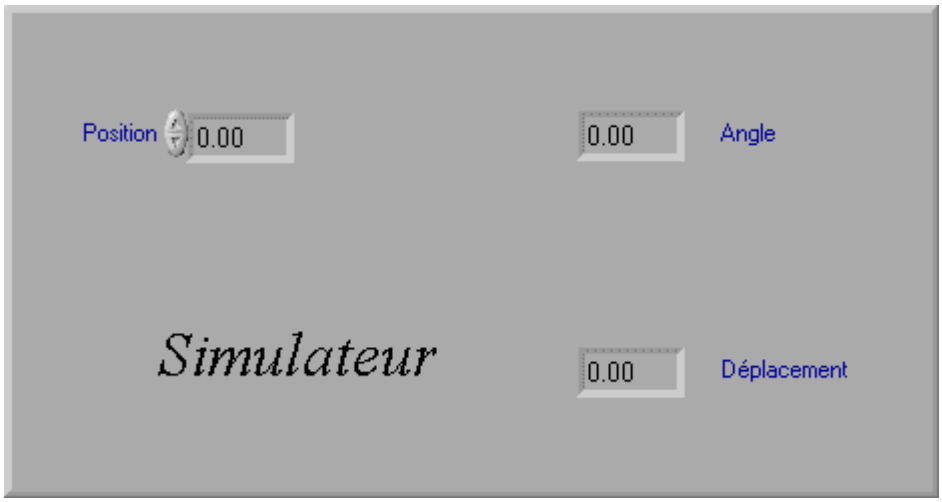
Δ R	**Variation relative du rayon exprimée en millimètres**		
Sous titre	Déplacement	*type*	double
		apparence	standard

Flux de donnée:



Ergonomie:

Pour des raisons d'uniformité nous adopterons un style pour tous nos VI.



Le cadre (élément de décoration) sert à délimiter la face avant (impression).
Seul les sous-titres sont visibles (times new roman /12 /bleu roi), les contrôles/indicateurs sont alignés par le haut et par la gauche.

Analyse

Deux calculs indépendants sont faits:

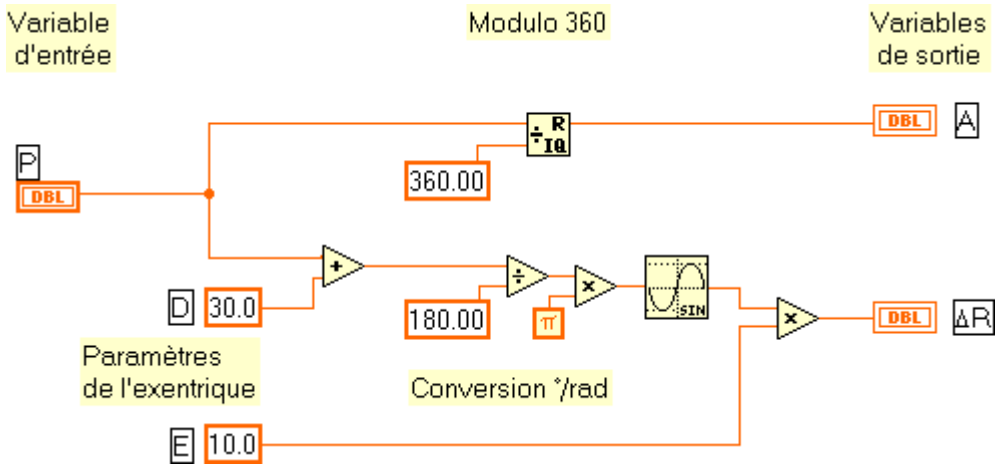
- ☑ **le calcule de l'angle modulo 360°.**
- ☑ **le calcule du déplacement suivant la formule:**

$$\Delta R = E \cdot \sin(P + D)$$

calcul de l'argument
conversion en radian
calcul du déplacement
E **(excentration) et** D **(déphasage) sont des paramètres constants.**

Simul_0.vi

Diagramme



➢Modifier si nécessaire les options de LabVIEW pour voir sur le diagramme des points aux jonctions de fils (outils>>options – diagramme).

➢Regrouper les blocs fonctionnels et aligner les variables, paramètres et constantes pour une meilleure lisibilité.

➢Documenter le diagramme.

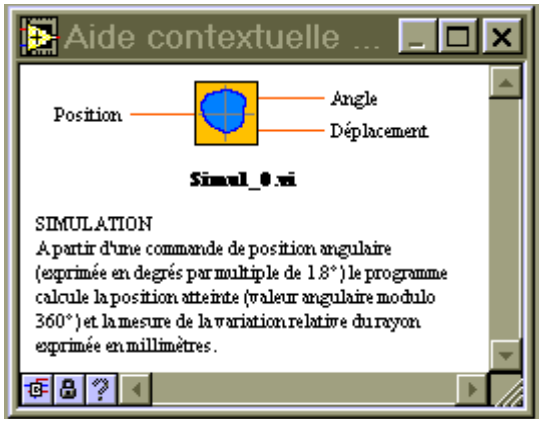
Simul_0.vi

Exercice N° 2: notion de sous-VI

Le module de simulation doit pouvoir être appelé par d'autres VI. Pour cela, il faut définir des *liaisons* entre les commandes et indicateurs de sa face avant (entrées et sorties de la fonction de simulation) et le VI appelant. C'est le rôle des connecteurs situés derrière l'icône de la face avant.

- ☑ **Dessiner l'icône**
- ☑ C**âbler** les connecteurs
- ☑ D**ocumenter**
- ☑ **Sauvegarder**

Résultat Obtenu:



TD2: Sous-VI

La carte de contrôle/commande du banc pilote le moteur pas à pas en fonction du nombre de pas (et non de l'angle) et assure la conversion de la mesure ΔR en valeurs numériques signées grâce à un convertisseur analogique numérique sur 12 bit.

Nous allons modifier le VI de simulation en conséquence.

Nous aborderons les points suivant:

- Appel de sous-VI
- Propriétés des objets
- Notions sur les attributs
- Fonctions de test
- Mise au point

TD2_1.vi

Exercice N° 1: Appel d'un sous-VI

Écrire un programme de simulation qui en appelant le VI simul_0.vi, accepte en entrée une commande sous forme de position exprimée en nombre de pas puis retourne l'angle atteint également exprimé en nombre de pas (200 pas par tour) et la mesure sous forme numérique.

Nous considérerons que la correspondance entre la valeur physique en millimètres et la valeur numérique retournée est linéaire entre -25mm et +25mm (2048 pour 25mm). Afin de donner plus de réalisme à notre simulateur, nous allons introduire une erreur de mesure aléatoire sur la mesure de déplacement de ±5 LSB (5 unités sur la gamme de mesure).

☑ **Analyser** avec soin le problème

☑ **Mettre en place** les éléments de la face avant d'un nouveau VI.

☑ **Enregistrer sous** le nom TD2_1.vi

☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Nœud d'expression
Utilisez le Nœud d'expression pour calculer des expressions ou des équations, qui contiennent une variable unique. Les Nœuds d'expression sont utiles lorsqu'une équation a seulement une variable mais est compliquée. Vous pouvez utiliser n'importe quel type de données lorsque vous créez un nœud d'expression.

☑ **Sauvegarder**.

TD2_1.vi

Variables d'entrées:

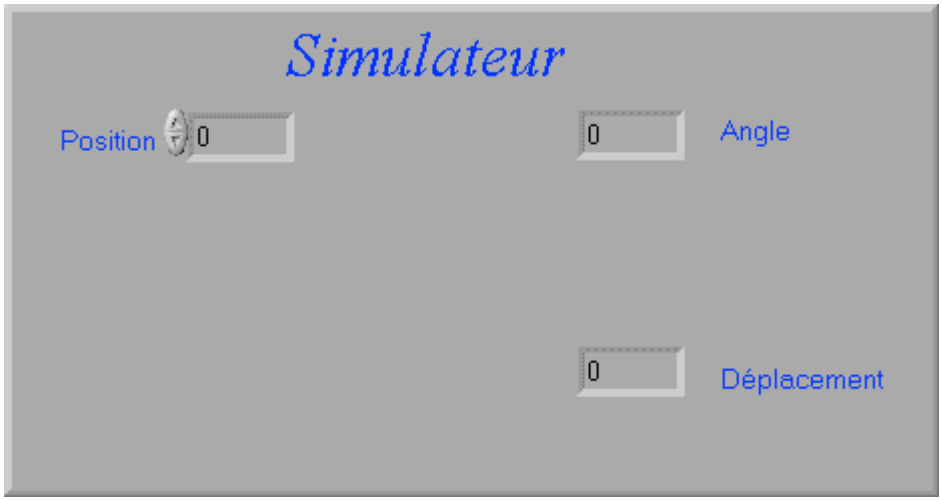
P		**Position angulaire à atteindre exprimée en pas**					
Sous titre		Position		*type*	Entier 16		
				apparence	standard		
défaut	0	*min*		*max*		*incr*	

Variables de sortie:

A	**Valeur angulaire en pas dans le tour**		
Sous titre	Angle	*type*	Entier 16
		apparence	standard

Δ R	**Variation relative du rayon sur 12 bit**		
Sous titre	Déplacement	*type*	Entier 16
		apparence	standard

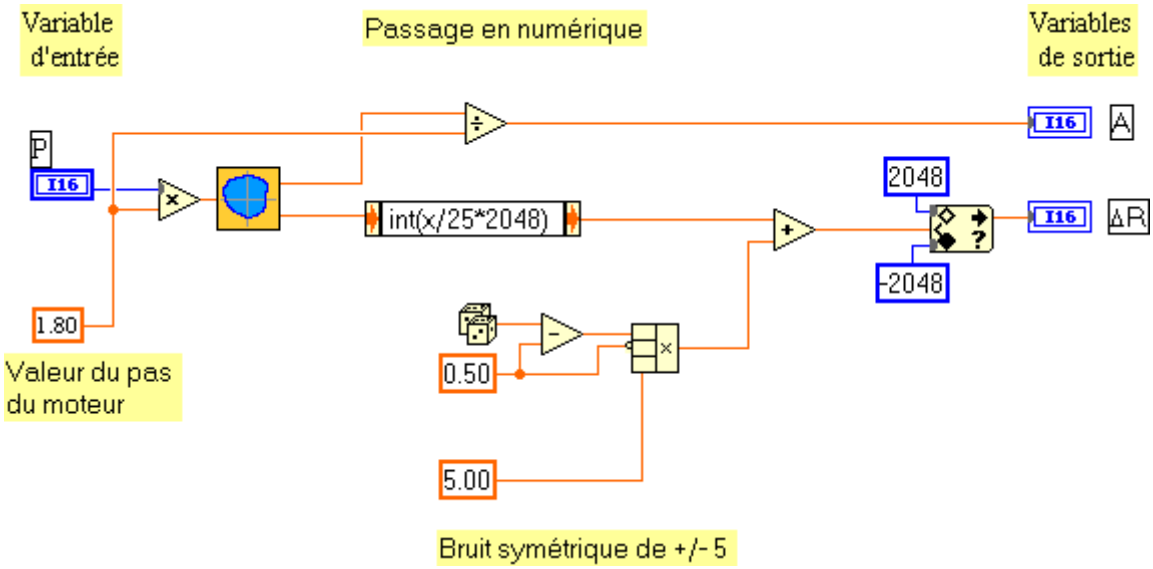
Ergonomie



- ☑ **La position angulaire en nombre de pas est convertie en angle (par multiplication par la valeur du pas 1,8°) pour donner la valeur de l'entrée** P **du sous-VI de simulation simul_0.vi.**
- ☑ **En sortie de simul_0.vi, la valeur de l'angle est convertie en nombre de pas (par division par 1,8°). Le résultat donne la valeur de la variable de sortie** A.
- ☑ **La sortie déplacement de simul_0.vi donne après conversion (règle de 3 ...../25*2048) et addition du bruit, la valeur de la variable de sortie** Δ R.
- ☑ **le bruit est généré à partir de la fonction Random [0-1] symétrisée à ±1 avant d'être mis à l'échelle.**
- ☑ **La valeur de sortie est bornée à ±2048**
  soit explicitement (correction)
  soit implicitement en bornant la valeur de l'indicateur

TD2_1.vi

Diagramme



Le lecteur remarquera les changements automatiques de type de données
I16 ⇨ DBL et DBL ⇨ I16

Exercice N° 2 : Création d'un sous-vi

Le détail de la réalisation du bruit symétrique ajouté à la mesure n'apporte rien à la compréhension du vi. Nous allons voir comment encapsuler le diagramme correspondant pour en faire un sous-vi.

- ☑ **Sélectionner** les fonctions et constantes devant être encapsulées.
- ☑ Dans le menu « Edition » **choisir** « Créer un sous vi ».
- ☑ **Editer** le nouveau vi, **nommer** les variables et **documenter**
- ☑ **Sauvegarder** sous le nom +bruit.vi.
- ☑ **Sauvegarder** le vi appelant sous le nom Simul_1.vi après avoir documenté l'aide.

+bruit.vi

Variables d'entrées:

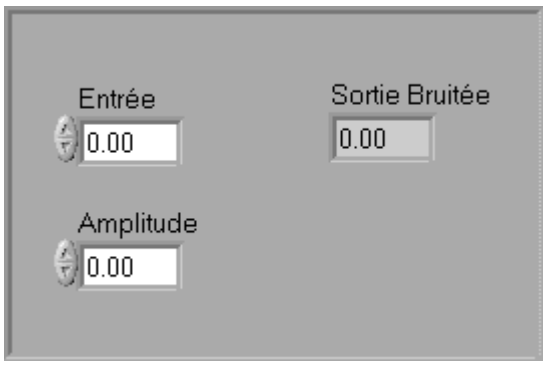
Sous titre				*type*			
				apparence			
défaut		*min*		*max*		*incr*	

Sous titre				*type*			
				apparence			
défaut		*min*		*max*		*incr*	

Variables de sortie:

Sous titre		*type*	
		apparence	

Ergonomie

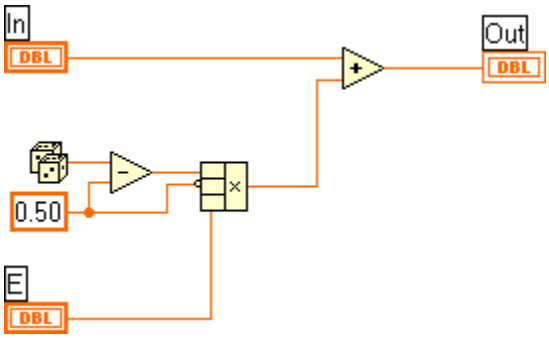


+bruit.vi

Additionne un bruit aléatoire entier dont l'amplitude est définie par l'entrée Amplitude

Cadre connecteur

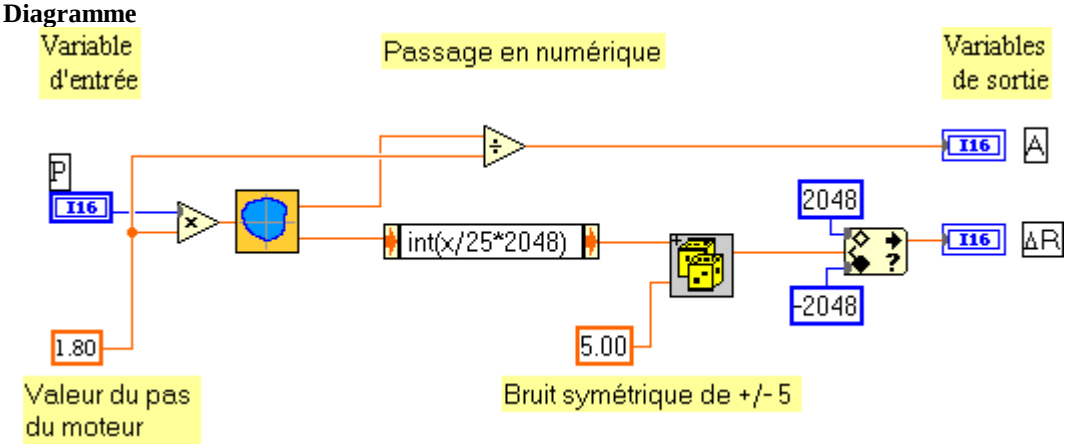
Diagramme



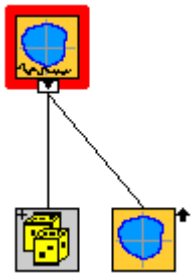
Simul_1.vi

Cadre connecteur

Diagramme



Position dans la hiérarchie



TD2_2.vi

Exercice N° 3: Mise au point

Écrire un programme de test qui en appelant le VI simul_1.vi, présente les mesures soit en valeur physique, soit en valeur numérique en fonction de l'état d'un sélecteur booléen en face avant.

- ☑ **Analyser** avec soin le problème
- ☑ **Mettre en place** les éléments de la face avant d'un nouveau VI.
- ☑ **Enregistrer sous** le nom TD2_2.vi
- ☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

  Sélectionner
  Retourne la valeur connectée à l'entrée t ou f, en fonction de la valeur de s. Si s est VRAI, cette fonction retourne la valeur connectée à t. Si s est FAUX, cette fonction retourne la valeur connectée à f.

- ☑ **Sauvegarder**.

TD2_2.vi

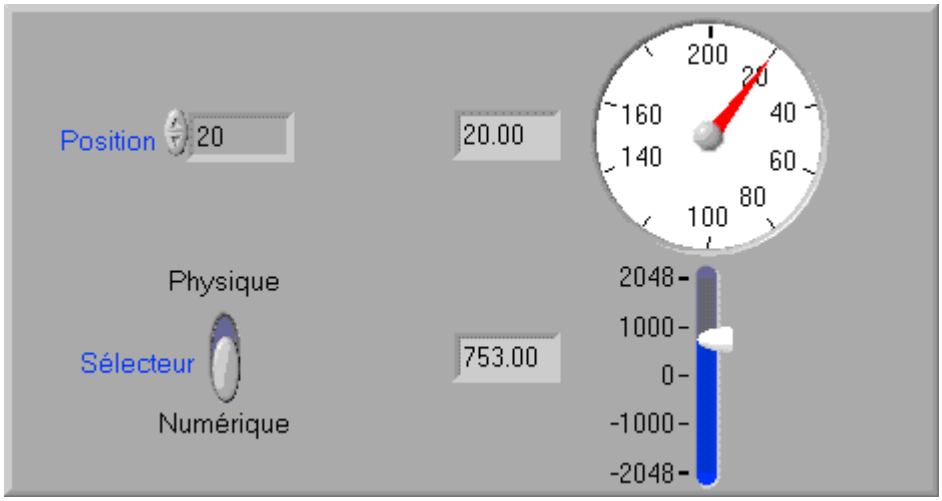
Variables Internes:

Sélecteur		**Choix du type d'affichage , mm (F) ou numérique (T)**					
Sous titre		Sélecteur		*type*	Boléen		
				apparence			
défaut	F	*min*		*max*		*incr*	

Glissière	**Valeur angulaire en pas dans le tour**		
Sous titre	Glissière	*type*	Double
		apparence	Glissière

Jauge	**Valeur angulaire en pas dans le tour**		
Sous titre	Jauge	*type*	Double
		apparence	Jauge

Ergonomie

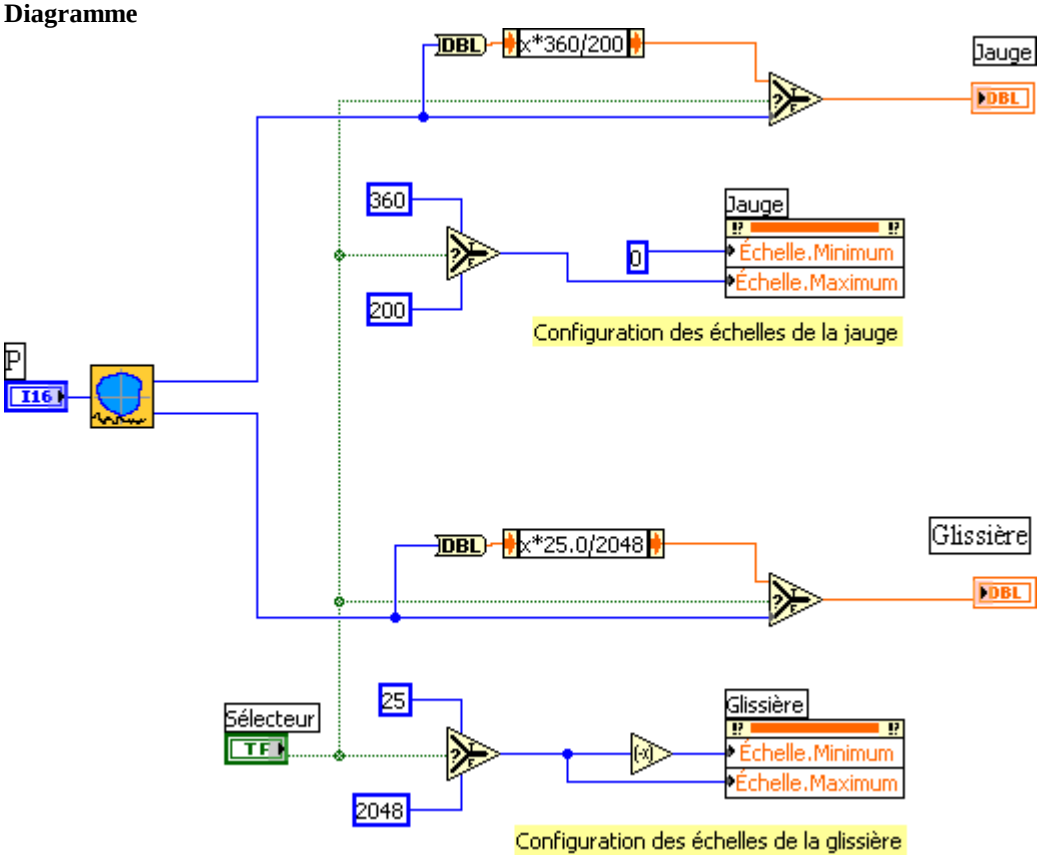


☑ **Les valeurs de chaque sortie de simul_1.vi sont affichées soit directement soit après conversion suivant la valeur de l'entrée** Sélecteur**.**

☑ **Suivant le type d'affichage les échelles des indicateurs doivent être reconfigurées.**

TD2_2.vi

Diagramme



TD3: Boucles et Graphes déroulants

Nous aborderons les points suivants:

- Présentation des graphes déroulants
- Structures de boucle For et While (les tunnels)
- Registres à décalage

Pour valider notre VI de simulation nous allons tracer l'allure du déplacement en fonction de la position demandée sur une plage relativement large. A chaque pas de calcul, le résultat est présenté sur un graphique de type oscilloscope.

TD3_1.vi

Exercice N° 1: Boucle FOR

En utilisant le VI de simulation Simul_1.vi, nous allons tracer les variations du rayon en fonction de la commande de position sur un tour (de 0° à 360°) pour un nombre de points fixé (compris entre 10 et 200). Les angles seront exprimés en degrés.

- ☑ **Analyser** avec soin le problème
- ☑ **Mettre en place** les éléments de la face avant d'un nouveau VI.
- ☑ **Enregistrer sous** le nom TD3_1.vi
- ☑ **Programmer** les opérations nécessaires. **Tester** puis **corriger** les problèmes de configuration d'axe.
- ☑ **Sauvegarder**.

TD3_1.vi

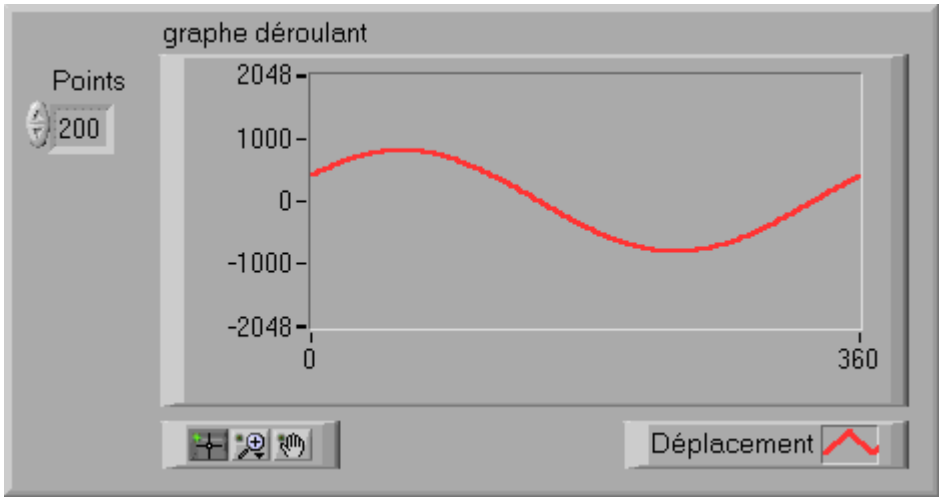
Variables d'entrées:

Nb		**Nombre de points par tour**					
Sous titre		Points		*type*	Entier 16 non signé		
				apparence	standard		
défaut	200	*min*	10	*max*	200	*incr*	1

Variables de sortie:

Graphe déroulant	**Variation de rayon (valeur numérique)**		
Sous titre	Graphe déroulant	*type*	Entier 16
		apparence	standard

Ergonomie



La boucle For fait un nombre d'itérations prédéterminé.

- ☑ **A chaque itération de la boucle, la variable** Position **à passer au sous vi** Simul_1.vi **est définie par :**

  $$Position = i * Ent(\frac{200}{Nb})$$

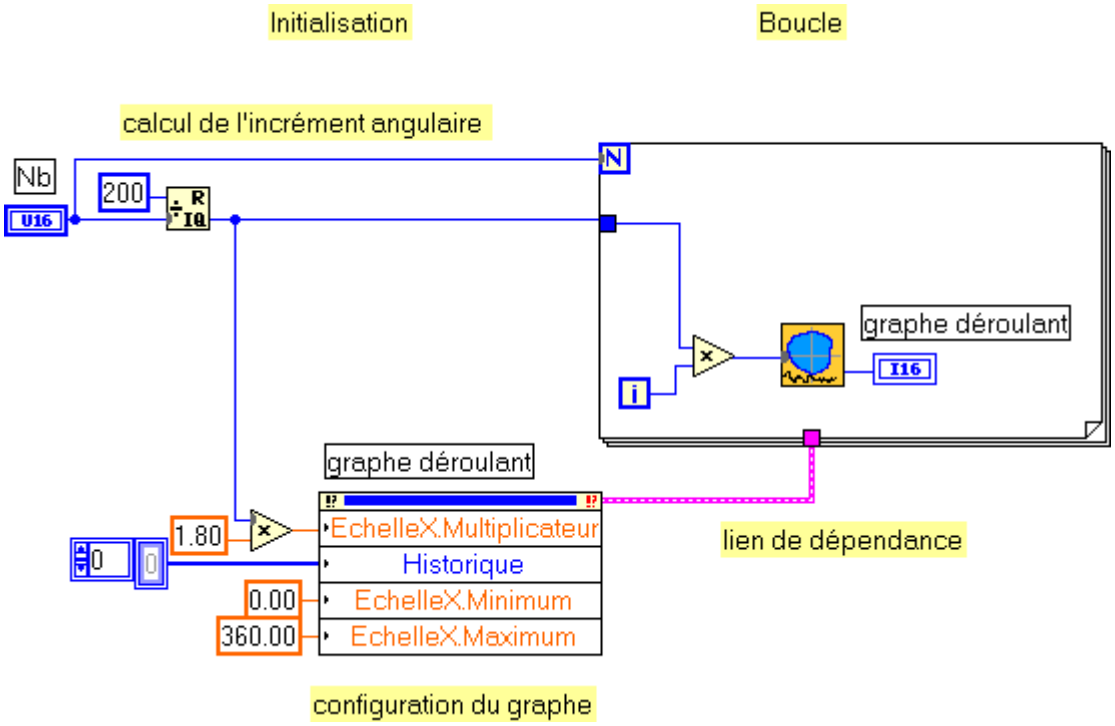
  ou i est l'indice de boucle.

  La variable de sortie Δ R **du sous vi** Simul_1.vi **est directement affichée sur le graphe déroulant .**
- ☑ **Le nombre d'itérations à faire est défini par la variable Nb.**

  L'incrément angulaire $Ent(\frac{200}{Nb})$ **est calculé en amont de la boucle.**
- ☑ **Le graphe est correctement initialisé:**
  effacement de la courbe
  configuration dynamique de l'échelle des x

TD3_1.vi

Diagramme



Le nœud de propriété de l'objet graphe déroulant permet de configurer dynamiquement l'échelle des x en précisant la valeur de l'incrément en x (en degrés) et les valeurs minimale et maximale de l'échelle.

La propriété Historique **permet, par un artifice , d'initialiser le graphe (ce serait une méthode) en forçant le tableau des valeurs à représenter à sa valeur initiale (structure vide).**

La structure d'erreur issue du nœud de propriété est reliée à un tunnel d'entrée de la boucle pour créer un lien de dépendance; tous les tunnels doivent être validés avant que la boucle puisse démarrer, nous nous assurons ainsi que le graphe est initialiser avant de tracer la courbe.

Les points grisés en entrée des fonctions (multiplication et Simul_1.vi) marque un changement de type de donnée.

TD3_2.vi

Exercice N° 2: Boucle WHILE

En utilisant le VI de simulation Simul_1.vi, nous allons tracer les variations du rayon en fonction de la commande de position entre une Position de départ et une Position d'arrivée pour un incrément angulaire inférieur ou égal à 15 pas. Les angles seront exprimés en degrés, l'incrément angulaire en pas.

TD3_2.vi

Variables d'entrées:

Pas		**Incrément angulaire (en pas)**					
Sous titre		Pas		*type*	Entier 16		
				apparence	standard		
défaut	1	*min*	1	*max*	15	*incr*	1

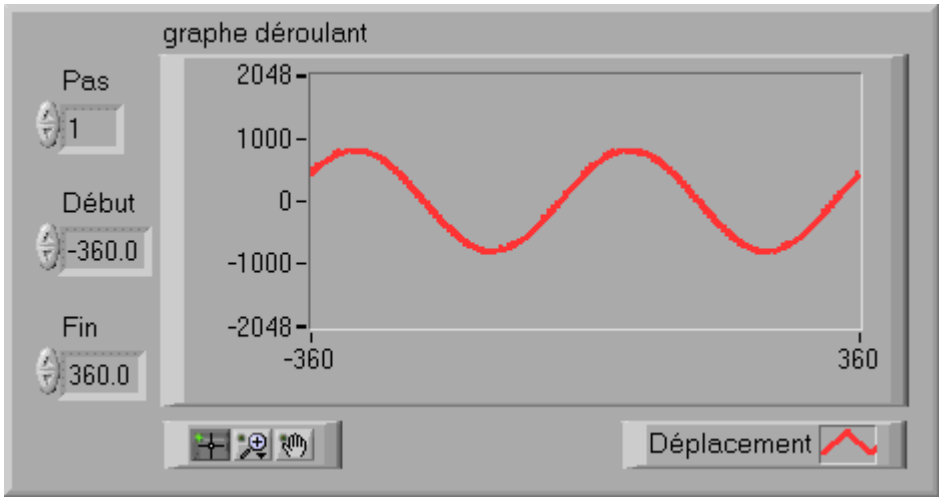
Début		**Position de début de mesure (en degrés)**					
Sous titre		Début		*type*	double		
				apparence	standard		
défaut	-360	*min*		*max*		*incr*	1.8

Fin		**Position de la dernière mesure (en degrés)**					
Sous titre		Fin		*type*	double		
				apparence	standard		
défaut	360	*min*		*max*		*incr*	1.8

Variables de sortie:

Graphe déroulant	**Variation de rayon (valeur numérique)**		
Sous titre	Graphe déroulant	*type*	Entier 16
		apparence	standard

Face-avant

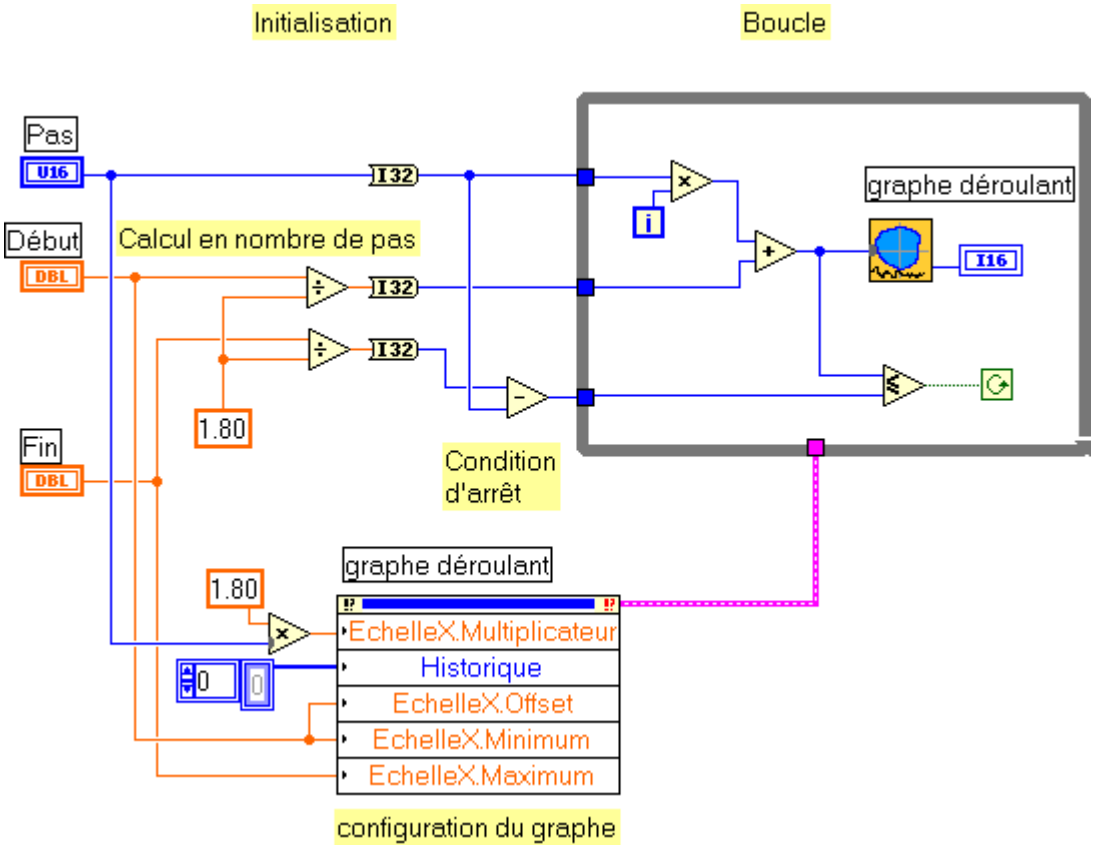


La boucle While itère depuis la position de début jusqu'à la position de fin. Les positions exprimées en degrés devront être au préalable converties en nombre de pas (.../1.8).

- ☑ **A chaque itération de la boucle, la variable** Position **à passer au sous vi** Simul_1.vi **est définie par :**
  $Position = Début + i * Pas$
  ou i est l'indice de boucle.
  La variable de sortie Δ R **du sous vi** Simul_1.vi **est directement affichée sur le graphe déroulant .**
- ☑ **La condition d'arrêt est atteinte dès que la position à venir (prochaine itération) dépasse la position de fin demandée. Le test darrêt étant fait en fin de boucle, la boucle itère tant que la position courante est inférieure ou égale à la valeur de fin diminuée de l'incrément angulaire (**Pas**).**
- ☑ **Le graphe est initialisé (nœud de propriétés)**
  effacer le graphe
  définir le début et la fin de l'axe de x
  définir l'incrément et le décalage de la courbe

TD3_2.vi

Diagramme



Les conversions en I32 assurent une compatibilité de type des opérateurs contenus dans la boucle; évite les conversions à chaque itération ce qui optimise le temps de boucle (non obligatoire).

Remarque:
L'entrée Position du vi Simul_1 ne correspond pas au type de donnée reçue

TD3_3.vi

Exercice N° 3: Registre à décalage

Dans cet exercice, le calcul de la position sera optimisé.

☑ **Modifier** le VI TD3_2 afin de calculer la position courante par la formule

$Position(i+1) \leftarrow Position(i) + Pas$

☑ **Sauvegarder** sous TD3_3.

TD3_3.vi

La boucle While itère depuis la position de début jusqu'à la position de fin. Les positions exprimées en degrés devront être au préalable converties en nombre de pas (.../1.8).

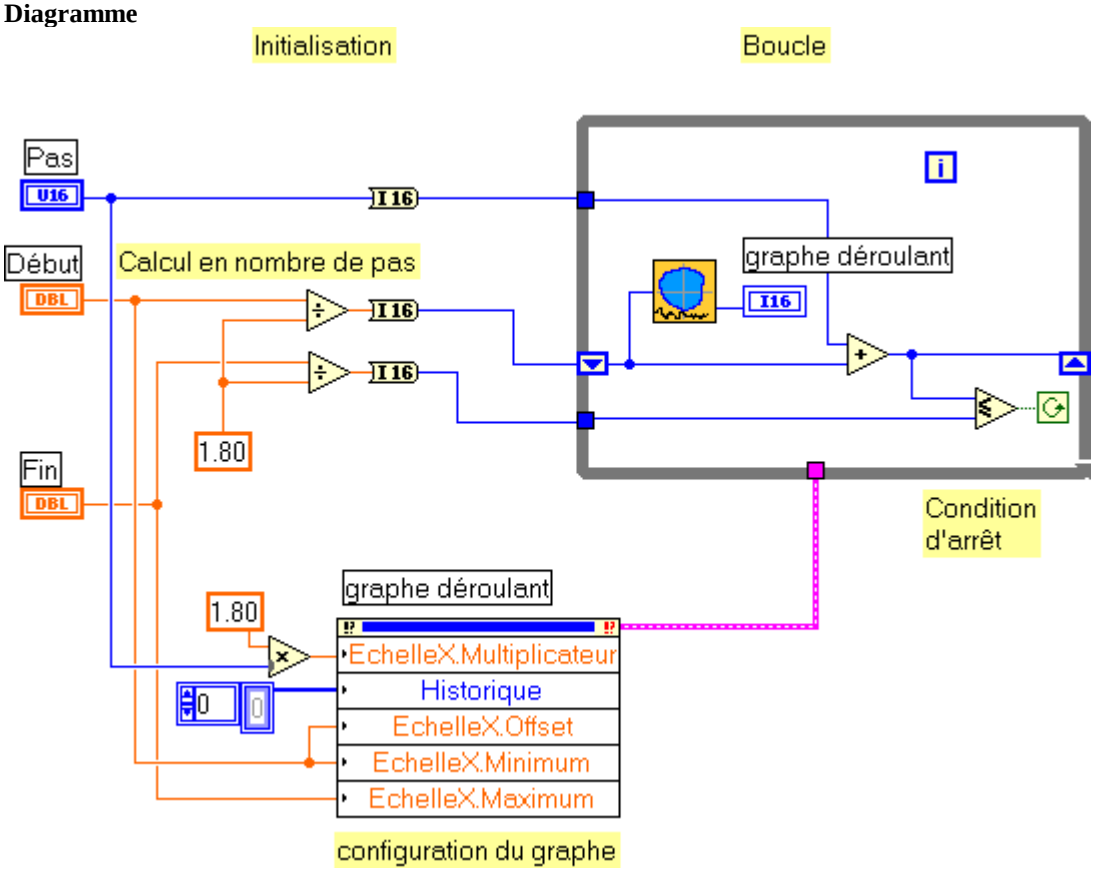
- ☑ **A chaque itération de la boucle, la variable** Position **à passer au sous vi** Simul_1.vi **est définie par :**

  $Position = Position_{-1} + Pas$

  La variable de sortie Δ R **du sous vi** Simul_1.vi **est directement affichée sur le graphe déroulant .**
- ☑ **La première boucle doit s'exécuter pour la position de départ. Il faut initialiser le registre avec cette valeur et calculer après l'appel du vi de simulation, la valeur de la position à venir.**
- ☑ **La condition d'arrêt est atteinte dès que la position à venir (prochaine itération) dépasse la position de fin demandée (tant qu'elle est inférieur ou égale ....)**
- ☑ **Le graphe est initialisé (nœud de propriétés)**
  - **effacer le graphe**
  - **définir le début et la fin de l'axe de x**
  - **définir l'incrément et le décalage de la courbe**

TD3_3.vi

Diagramme



TD3_4.vi

Exercice N° 4: Graphe Multicourbes

(***Complémentaire***)

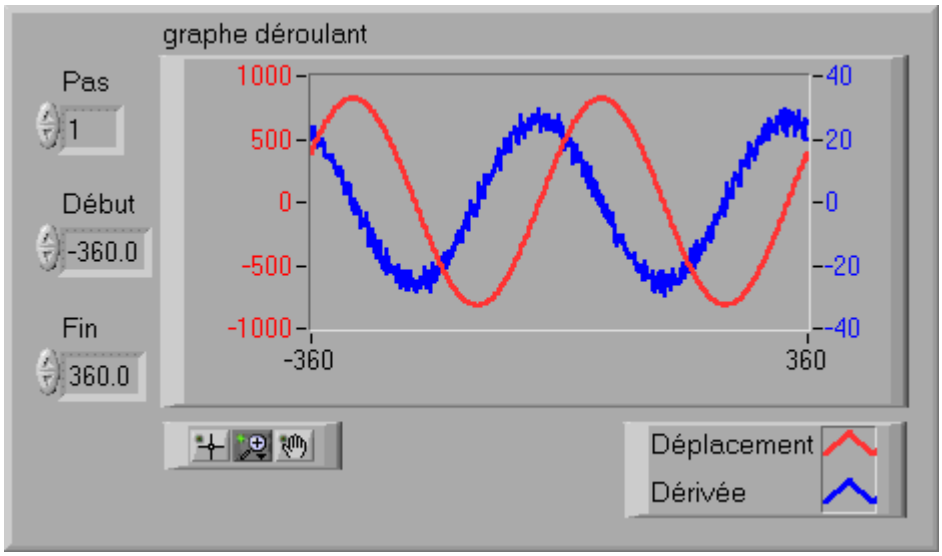
Pour mettre en évidence les erreurs de mesure, nous allons calculer et tracer la dérivée de la courbe de mesure. La dérivée sera évaluée en fonction du point précédent et du point suivant.

$$\dot{Y} = \frac{Y_{+1} - Y_{-1}}{(2 \cdot \Delta X)}$$

- ☑ **Reprendre** l'exercice précédent pour ajouter la seconde courbe sur le même graphe déroulant.
  Le lecteur consultera l'aide en ligne du graphe et l'exemple indiqué avant de faire l'analyse du problème.
- ☑ **Configurer** le graphe à l'éditeur pour avoir deux échelles pour les axes des Y, une pour la mesure, une pour la dérivée.

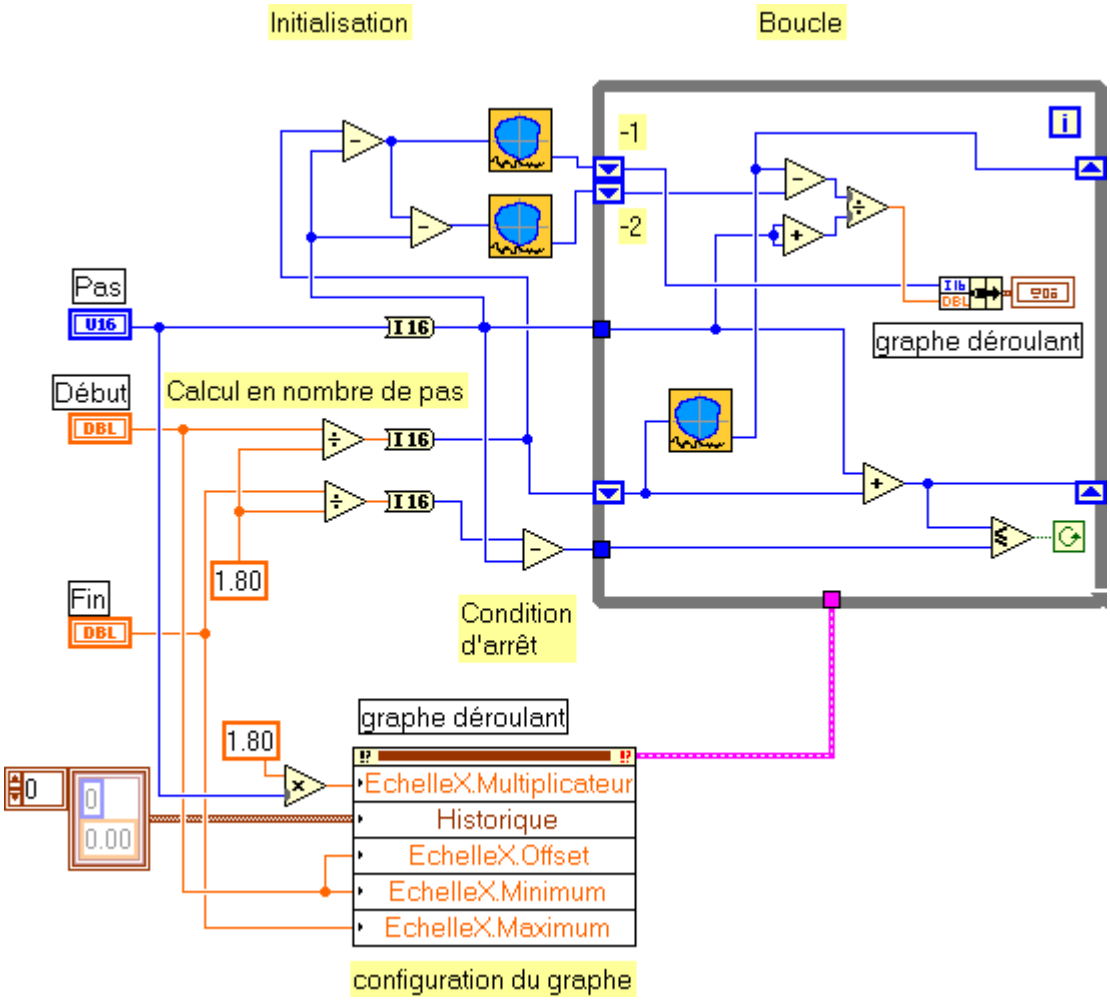
TD3_4.vi

Face-avant


graphe déroulant
Pas
1
Début
-360.0
Fin
360.0
1000
500
0
-500
-1000
40
20
0
-20
-40
-360
360
Déplacement
Dérivée

TD3_4.vi

Diagramme



Le lecture remarquera le changement de structure du graphe déroulant

Scalaire ⇨ Cluster

Simul_2.vi

Exercice N° 4: Registre non initialisé

Après une étude détaillée du fonctionnement du moteur, force est de constater qu'il est plus astucieux de passer en paramètre non pas la position à atteindre, mais le nombre de pas à exécuter. D'autre part, la rotation prend un temps proportionnel au nombre de pas, de l'ordre de 5 milli-secondes par pas.

- ☑ **Compléter** le vi Simul_1.vi pour prendre en compte ces deux remarques.
- ☑ **Prévoir** une entrée d'initialisation pour positionner le moteur pas à pas sur une position de référence (sortie Angle égale à 0).
- ☑ **Enregistrer** sous le nom Simul_2.vi.

Le vi obtenu sera considéré dans la suite des exercices comme le simulateur de la partie physique du banc de test.

Simul_2.vi

Variables d'entrées:

P		**Nombre de pas demandé**					
Sous titre		Incrément		*type*	Entier 16		
				apparence	standard		
défaut	0	*min*		*max*		*incr*	

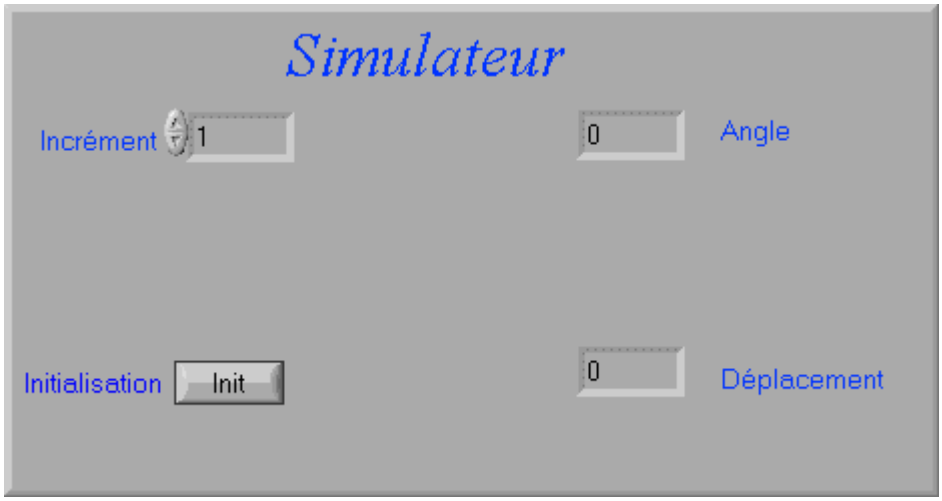
Initialisation		**Force la position à 0**					
Sous titre		Initialisation		*type*	Boléen		
				apparence	standard		
défaut	F	*min*		*max*		*incr*	

Variables de sortie:

A	**Valeur angulaire en pas dans le tour**		
Sous titre	Angle	*type*	Entier 16
		apparence	standard

Δ R	**Variation relative du rayon sur 12 bit**		
Sous titre	Déplacement	*type*	Entier 16
		apparence	standard

Ergonomie



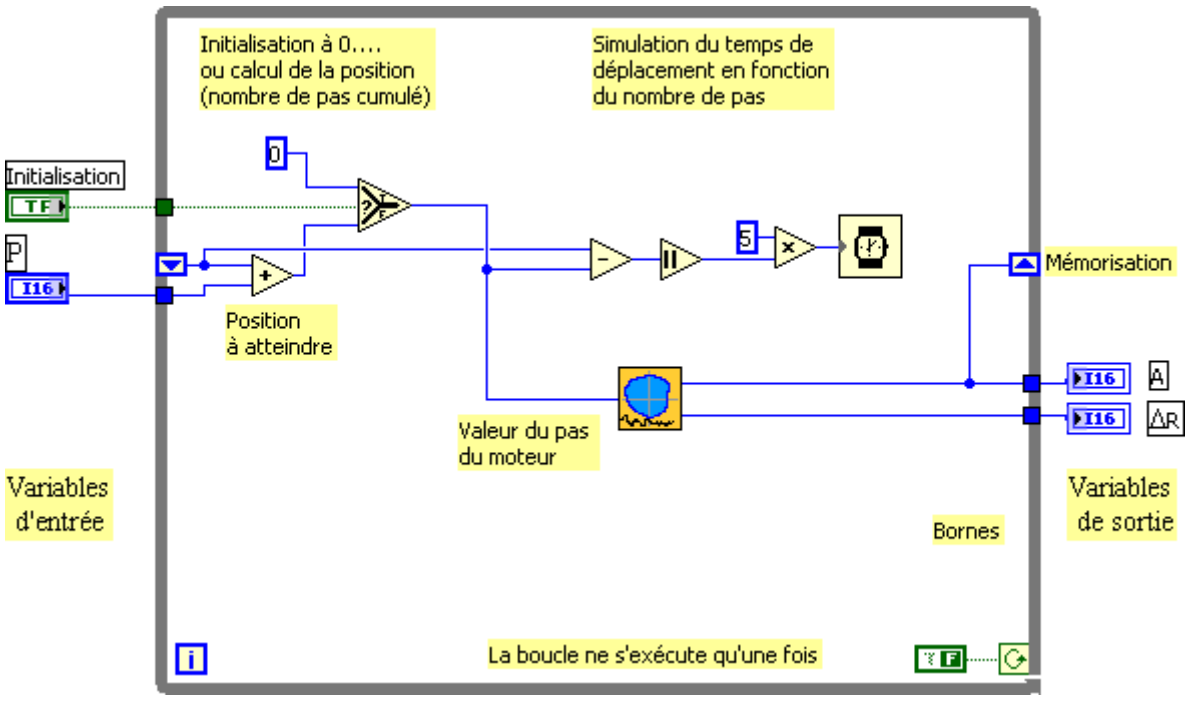
- ☑ **La position atteinte dans le tour doit être mémorisée par le vi (registre à décalage non initialisé)**
- ☑ **A chaque appel sans demande d'initalisation, la position est calculée en ajoutant le nombre de pas demandés à la valeur mémorisée.**
- ☑ **Pour un appel avec demande d'initialisation, la position est forcée à 0, le nombre de pas demandé est ignoré.**

Pour ajouter au réalisme, nous mettrons en place une temporisation égale à 5 ms par pas de déplacement à exécuter.

- ☑ **Calcul du nombre de pas à exécuter**
  différence entre la position à atteindre et la position mémorisée en valeur absolue.
- ☑ **Multiplication par 5 ms et appel de la fonction de temporisation**

Simul_2.vi

Diagramme



Cadre connecteur

TD4: Structure Condition

Pour tester la carte de pilotage du banc, il est nécessaire d'envoyer des commandes choisies par l'utilisateur et prises parmi le jeux de commandes comprises par la carte (Initalisation, Déplacement, Mesure, ...). Il faudra également analyser les réponses et les éventuelles erreurs. Nous allons dans les exercices proposés ci-après nous familiariser avec les problèmes de test ou de choix multiple.

Nous introduirons les notions de base sur:

- Les structures de condition
- Les contrôles et indicateurs de chaîne de caractères
- Listes déroulantes et Énumérations
- Les contrôles Boléen
- Les actions mécaniques
- Les tableaux et les manipulations de tableaux
- Les variables locales

TD4_1.vi

Exercice N° 1: Condition Vrai / Faux

(***Facultatif***)

Nombre de vi de la bibliothèque LabVIEW retournent en cas d'erreur une valeur -1 pour l'une des variables de sortie (généralement un index).

En utilisant le vi +Bruit.vi de telle sorte qu'il génère un nombre compris entre -1 et 7, écrire un vi qui, chaque fois qu'une erreur est signalée, demande à l'opérateur s'il veut poursuivre ou non l'exécution du vi.

☑ **Analyser** avec soin le problème du test.

☑ **Mettre en place** les éléments de la face avant d'un nouveau VI en visualisant le paramètre retourné +Bruit.vi et formaté en entier.

☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Boîte de dialogue 2 boutons

Affiche une boîte de dialogue qui contient un message et deux boutons. nom du bouton V et nom du bouton F sont les noms affichés sur les boutons de la boîte de dialogue.

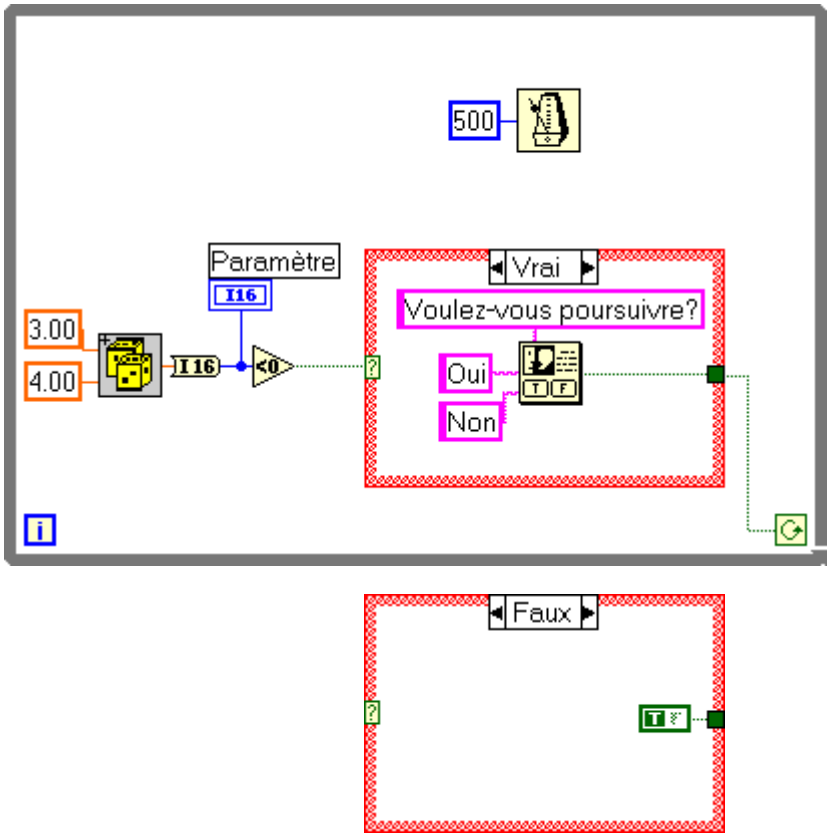
TD4_1.vi

Ergonomie

- ☑ **Le programme boucle jusqu'à la validation d'une erreur par l'opérateur (structure while).**
- ☑ **Le vi +**Bruit.vi **reçoit en entrée les paramètres 3 (**Entrée**) et 4 (**Amplitude**), sa sortie est convertie en entier pour donner** Paramètre**.**
- ☑ Paramètre **doit être testé, s'il est négatif (-1) il y a erreur.**
- ☑ **S'il n'y à pas d'erreur, la condition de boucle doit être positionnée à Vrais (**T**).**
  S'il y a erreur, le vi Boîte de dialogue 2 boutons.vi **est appelé, configuré pour demandez si le programme continu (une réponse affirmative donne une sortie à** T**) et la condition de boucle est activée par sa sortie.**

TD4_1.vi

Diagramme



La Temporisation permet de ralentir la boucle (½ seconde) afin de laisser à l'opérateur le temps de visualiser les valeurs affichées.

Cette boucle présente deux processus indépendants, la séquence de génération/test et la temporisation. Ils doivent être tous les deux terminés avant de passer à l'itération suivante.

Le lecteur remarquera le tunnel de sortie qui reçoit impérativement une valeur de chaque condition.

TD4_2.vi

Exercice N° 2: Menu déroulant

Pour cet exercice, nous allons à partir d'un contrôle de type menu déroulant, demander l'exécution de l'une des trois fonctions de base (Initalisation, Déplacement, Mesure). Chaque fonction consiste ici à écrire un message et dure respectivement 5 secondes, 2 secondes et 1 seconde. Un bouton Stop permet d'arrêter le programme.

☑ **Analyser** le problème

☑ **Mettre en place** les éléments de la face avant d'un nouveau VI .

☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Attendre (ms)
Attend pendant le nombre de millisecondes spécifié et retourne la valeur de l'horloge en millisecondes. 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.

TD4_2.vi

Variable d'entrées:

Menu déroulantes		Sélection de la fonction		
Sous titre		Menu déroulant	*type*	U16
			apparence	Menu déoulant
défaut	0	*Attente commande*		
	1	*Initialisation*		
	2	*Positionnement*		
	3	*Messure*		

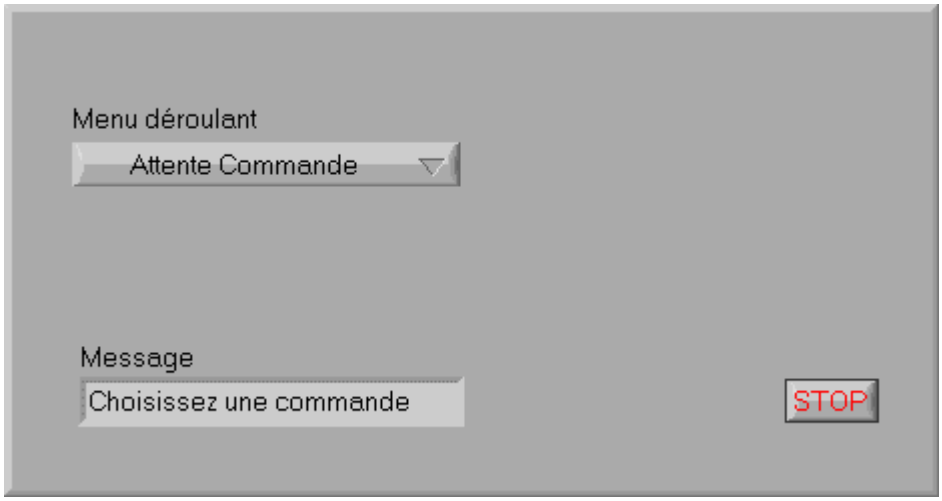
Variable de sortie:

Message		Indique la fonctionnalité exécutée		
Sous titre		message	*type*	chaine
			apparence	

Variable interne

stop				
Sous titre		stop	*type*	Boléen
			apparence	Bouton STOP
défaut	F	*Action Mécanique*	*Armement à l'appui*	

Face-avant



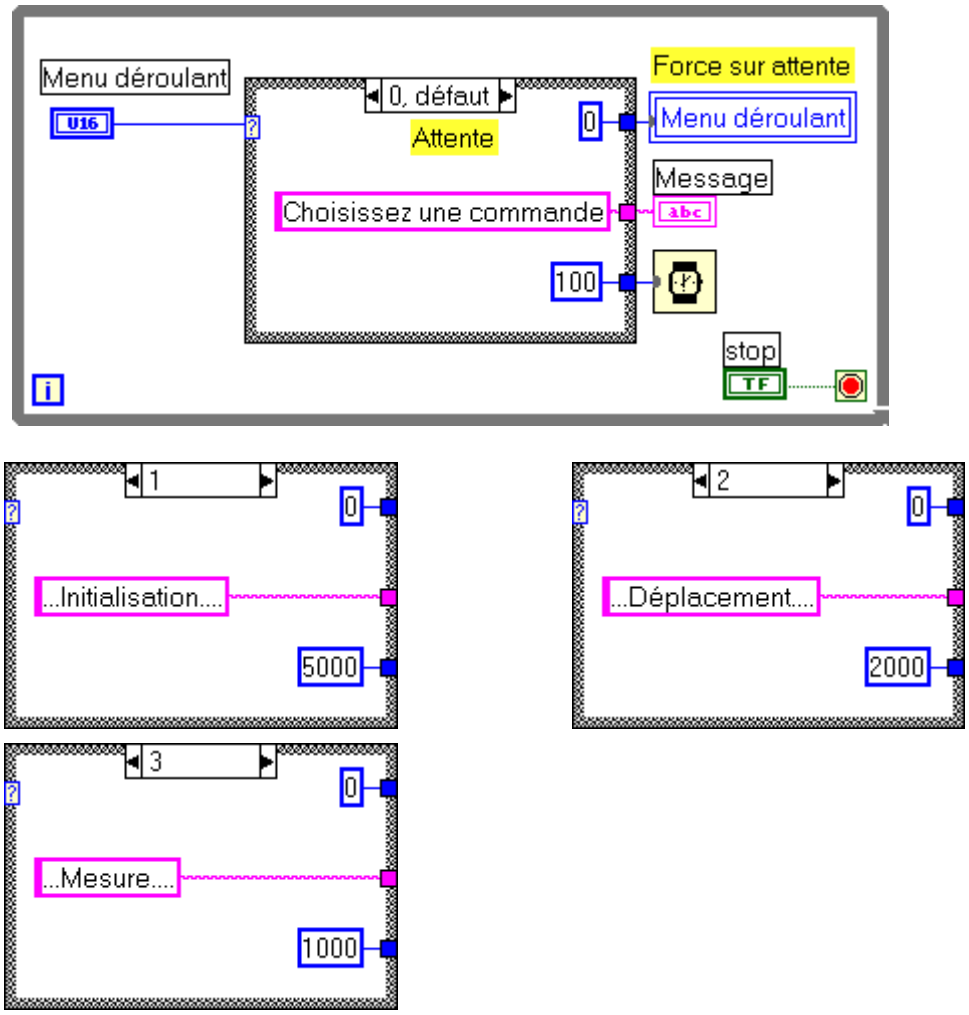
- ☑ **Le programme boucle jusqu'à l'arrêt par l'opérateur (bouton** STOP**).**
- ☑ **Le menu déroulant permet de sélectionner 1 état parmi 4, les trois fonctions et un état d'attente. Cet état est pris comme état par défaut.**
- ☑ **Pour chaque état, il doit être défini, un message et un temps. Ces valeurs sont passées à un afficheur et à une fonction de temporisation.**
- ☑ **En fin de boucle, le menu déroulant doit être forcé au cas par défaut.**

Remarque:

L'utilisation d'un menu déroulant ou d'une énumération comme contrôle d'entrée donne deux vi répondant à la même analyse mais avec une lisibilité du diagramme différente

TD4_21.vi

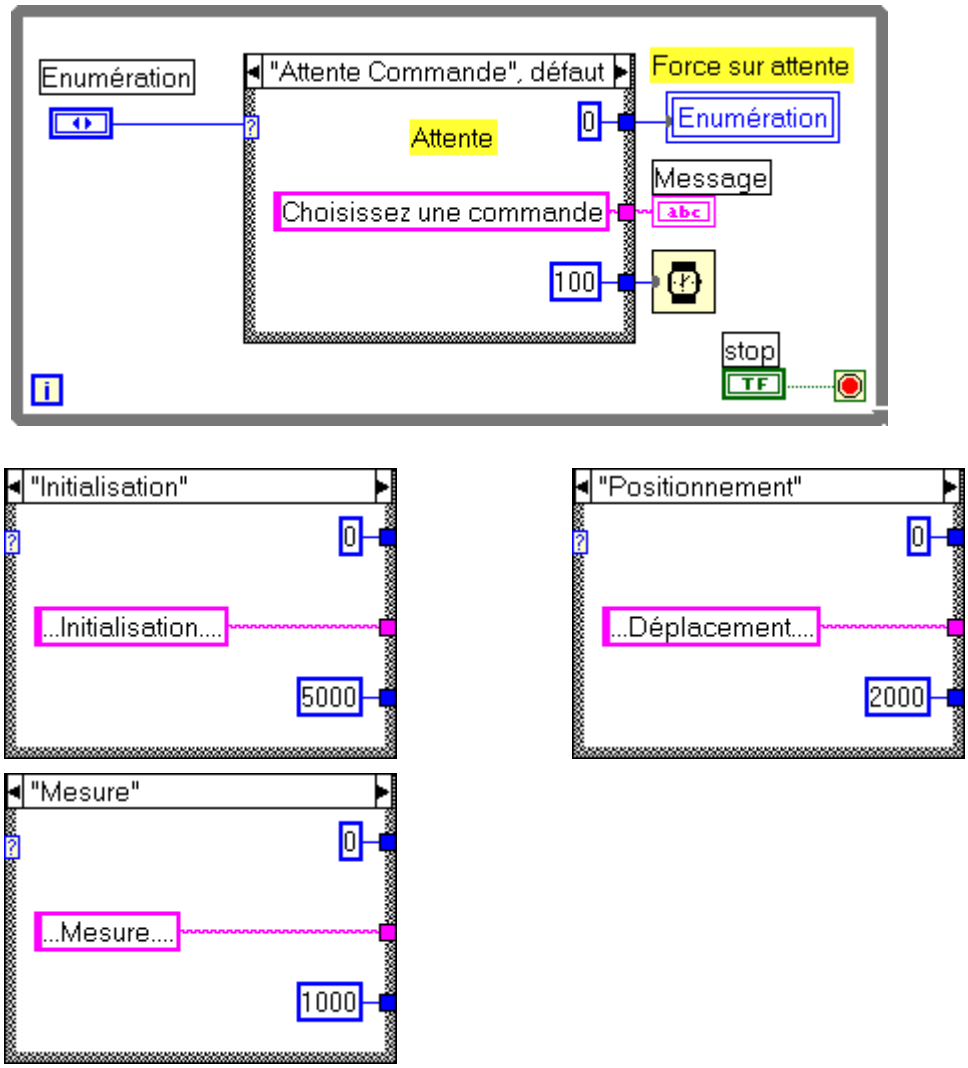
Diagramme



La temporisation de 100 ms pour le cas défaut permet d'adapter le temps de boucle au temps de réaction de l'opérateur et <u>surtout</u> de libérer du temps machine pour le système d'exploitation.

TD4_22.vi

Diagramme



L'utilisation de la variable locale permet de réinitialiser le contrôle d'entrée à la valeur 0, choisi comme valeur par défaut. Le même résultat pourrait être obtenu en forçant la propriété « valeur » du nœud de propriétés du contrôle.

TD4_3.vi

Exercice N° 3: Machine d'état.

Nous allons reprendre l'exercice précédent en modifiant l'ergonomie du vi; le menu est remplacé par un ensemble de boutons, un par fonction.

TD4_3.vi

Variable de sortie:

Message	Indique la fonctionnalité exécutée		
Sous titre	message	*type*	chaine
		apparence	

Variable interne

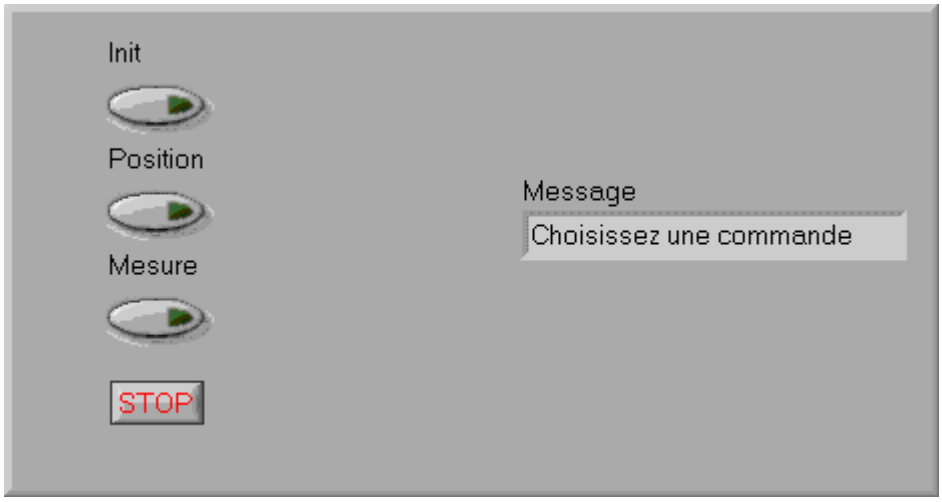
stop			
Sous titre		stop	*type*	Boléen
			apparence	Bouton STOP
défaut	F	*Action Mécanique*	*Armement à l'appui*	

Init			
Sous titre		Init	*type*	Boléen
			apparence	Bouton-poussoir
défaut	F	*Action Mécanique*	*Armement à l'appui*	

Positionnement			
Sous titre		Positionnement	*type*	Boléen
			apparence	Bouton-poussoir
défaut	F	*Action Mécanique*	*Armement à l'appui*	

Mesure			
Sous titre		Mesure	*type*	Boléen
			apparence	Bouton-poussoir
défaut	F	*Action Mécanique*	*Armement à l'appui*	

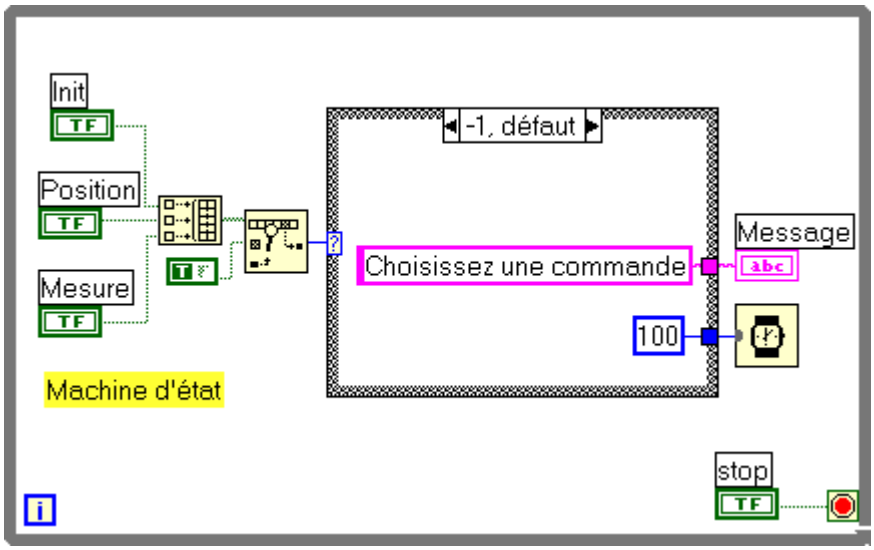
Face-avant

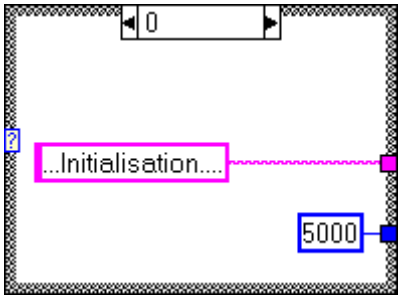


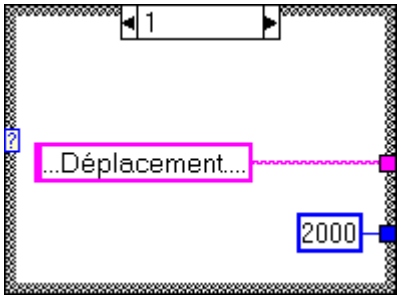
- ☑ **Le programme boucle jusqu'à l'arrêt par l'opérateur (bouton STOP).**
- ☑ **Un seul poussoir pouvant être actif simultanément, il suffit de chercher s'il y à un poussoir validé et lequel.**
- ☑ **Pour chaque état, il doit être défini, un message et un temps. Ces valeurs sont passées à un afficheur et à une fonction de temporisation.**
- ☑ **Si aucun poussoir n'est validé, le cas par défaut (attente) est exécuté. Sinon, le cas validé est exécuté**
- ☑ **Après lecture, les poussoirs prennent leur valeur par défaut (action mécanique de type armement)**

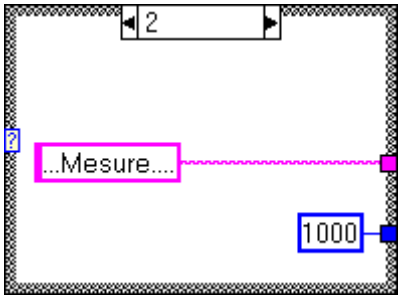
TD4_3.vi

Diagramme









Pour rechercher quel est le poussoir validé, les poussoirs sont regroupés dans une structure de tableau, puis le tableau analysé par une fonction de recherche de valeur (ici la valeur vrai). Si aucun élément correspond, la valeur de l'index est donnée à -1, sinon elle indique le numéro du premier élément correspondant. L'ordre dans le tableau définit une priorité de traitement.

Remarque: Le bouton stop aurait pu être intégré au tableau est traité par la structure condition.

TD4_4.vi

Exercice N° 4: Choix d'un cas parmi N

Pour réaliser la simulation de la carte de commande du banc de mesure, il faudra être capable de choisir l'exécution d'une commande à partir du message reçu.

Chaque commande est concrétisée dans ce message par une lettre (I ⇨Initalisation, P ⇨ Déplacement, M ⇨Mesure).

Nous allons réaliser un programme qui à partir de la saisie d'une lettre dans un contrôle de type chaîne de caractères exécute la fonction demandée si la lettre correspondante est saisie (en majuscule ou minuscule).

TD4_4.vi

Variable d'entrées:

Commande		**Lettre définissant la commande à exécuter**		
Sous titre		Commande	*type*	chaine
			apparence	
défaut	vide			

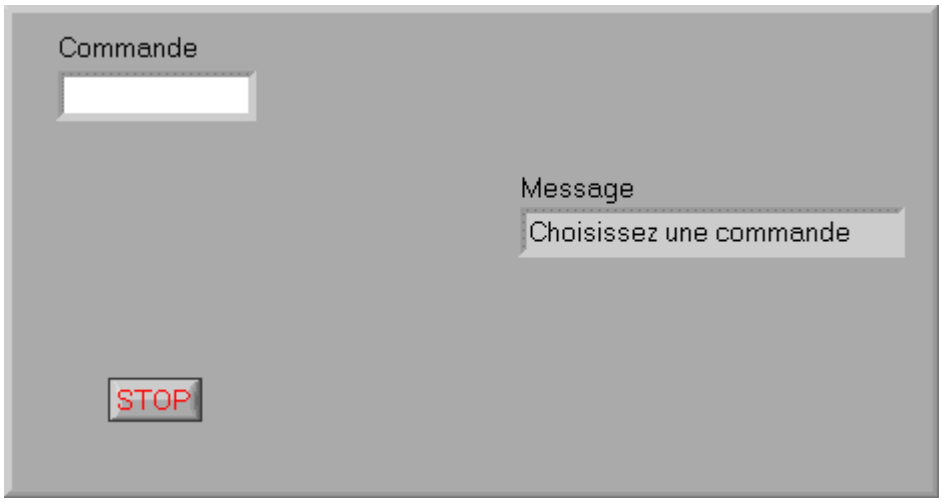
Variable de sortie:

Message	Indique la fonctionnalité exécutée		
Sous titre	message	*type*	chaine
		apparence	

Variable interne

stop				
Sous titre		stop	*type*	Boléen
			apparence	Bouton STOP
défaut	F	*Action Mécanique*	*Armement à l'appui*	

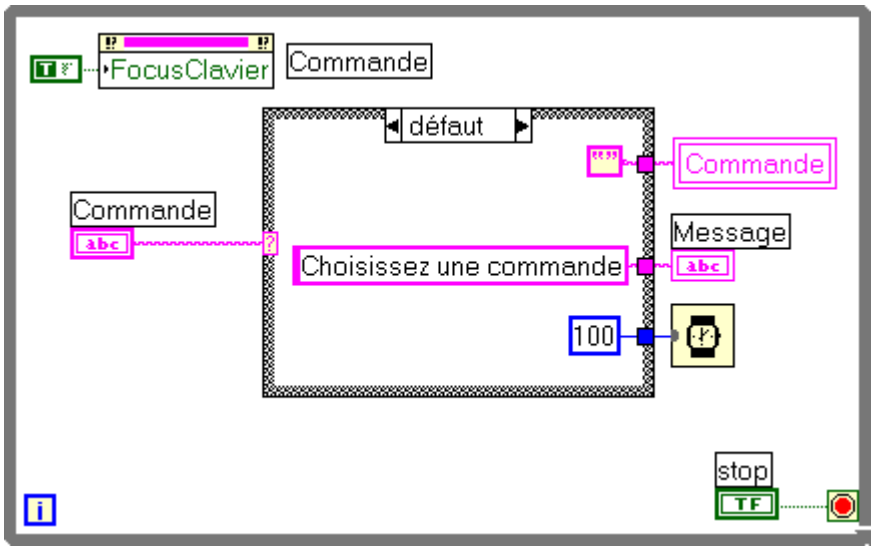
Face-avant

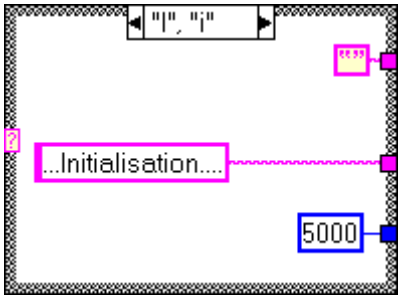


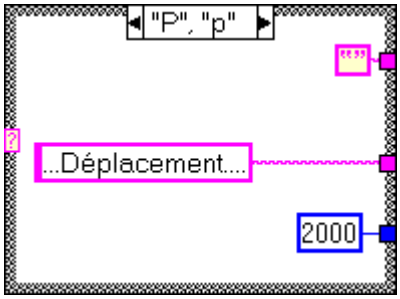
- ☑ **Le programme boucle jusqu'à l'arrêt par l'opérateur (bouton** STOP**).**
- ☑ **Pour chaque commande, il doit être défini, un message et un temps. Ces valeurs sont passées à un afficheur et à une fonction de temporisation.**
- ☑ **Toute lettre saisie est testée en respectant la casse, avec la lettre de chaque commande. S'il n'y a pas correspondance le cas par défaut (attente) est exécuté.**
- ☑ **En fin de boucle, le contrôle Commande doit être effacé (variable locale).**

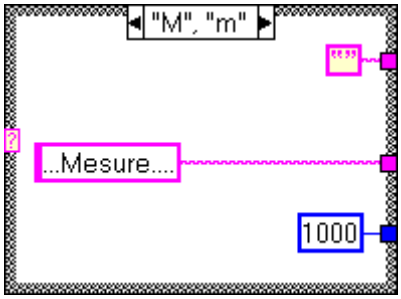
TD4_4.vi

Diagramme









La propriété Focus Clavier permet de forcer le curseur dans la zone d'écriture.

TD5: Chaînes de caractères

L'utilisation des chaînes de caractères est courante, saisie de paramètres, lecture de fichiers de configuration, sauvegarde de résultats, communication.... Au travers les exercices ci-après, vous découvrirez les fonctions élémentaires, la consultation des exemples vous apportera de précieux compléments.

Nous allons introduire les notions de base sur:

- Manipulation de chaînes
- Configuration de VI
- Mise en forme
- Notions sur les entrées/sorties série
- Introduction aux clusters

Dans ce TD, nous allons étudier la communication entre le calculateur et la carte de pilotage. Un premier exercice nous permettra de mettre en forme les commandes, un deuxième d'analyser une réponse et de gérer les erreurs, enfin un troisième intégrera les deux premiers en les faisant communiquer par le port série.

TD5_1.vi

Exercice N° 1: Mise en forme

Pour cet exercice, nous allons à partir par un ensemble de boutons poussoir créer la chaîne de commande correspondant à la fonction choisie (Initalisation, Déplacement ou Mesure). Ce vi s'inspire fortement de l'exercice n°3 du TD précédent, la boucle s'exécutant jusqu'à la validation d'un choix, la sélection du bouton Stop retournera une commande vide.

Le jeux de commandes retenu pour le pilotage du banc est un jeux de bas niveau correspondant au trois fonctions de base; mise en place initiale, avancer d'un ou plusieurs pas (1,8°), mesure du déplacement relatif.

Toutes les commandes sont codées en ASCII sur le même format, un délimiteur, **:** , une lettre majuscule définissant la commande, **C** , un paramètre optionnel et un caractère de fin de commande, le symbole Line Feed(\n)

:C<param>\n

Tous les nombres sont des entiers codés en ASCII

Tous les échanges se font sur le modèle maître/esclave. Le maître émet une commande puis attend la réponse de l'esclave. L'esclave attend une commande, l'exécute, met en forme la réponse et la retourne.

Initialisation	
Commande	Réponse
:I\n	:Iaaa\n ou aaa est un entier non signé sur 3 digit donnant la valeur de la mesure angulaire

Pas	
Commande	Réponse
:Pppp\n ou ppp est un entier non signé sur 3 digit donnant le nombre de pas à effectuer	:Paaa\n ou aaa est un entier non signé sur 3 digit donnant la valeur de la mesure angulaire

Mesure	
Commande	Réponse
:M\n	:M±mmmm\n ou mmmm est un entier signé sur 4 digit donnant la valeur de la mesure de déplacement

Remarque: pour les champs numériques, les chiffres non significatif sont remplacés par des zéro (0)

TD5_1.vi

Variable de sortie:

Commande	Indique la fonctionnalité exécutée		
Sous titre	message	*type*	chaine
		apparence	

Variable interne

Init			
Sous titre	Init	*type*	Boléen
		apparence	Bouton-poussoir
défaut F	*Action Mécanique*	*Armement à l'appui*	

Positionnement			
Sous titre	Positionnement	*type*	Boléen
		apparence	Bouton-poussoir
défaut F	*Action Mécanique*	*Armement à l'appui*	

Mesure			
Sous titre	Mesure	*type*	Boléen
		apparence	Bouton-poussoir
défaut F	*Action Mécanique*	*Armement à l'appui*	

Stop			
Sous titre	Stop	*type*	Boléen
		apparence	Bouton-poussoir
défaut F	*Action Mécanique*	*Armement à l'appui*	

Pas	**Incrément angulaire (en pas)**						
Sous titre	Pas			*type*	Entier 16		
				apparence	standard		
défaut 1	*min*	1		*max*	200	*incr*	1

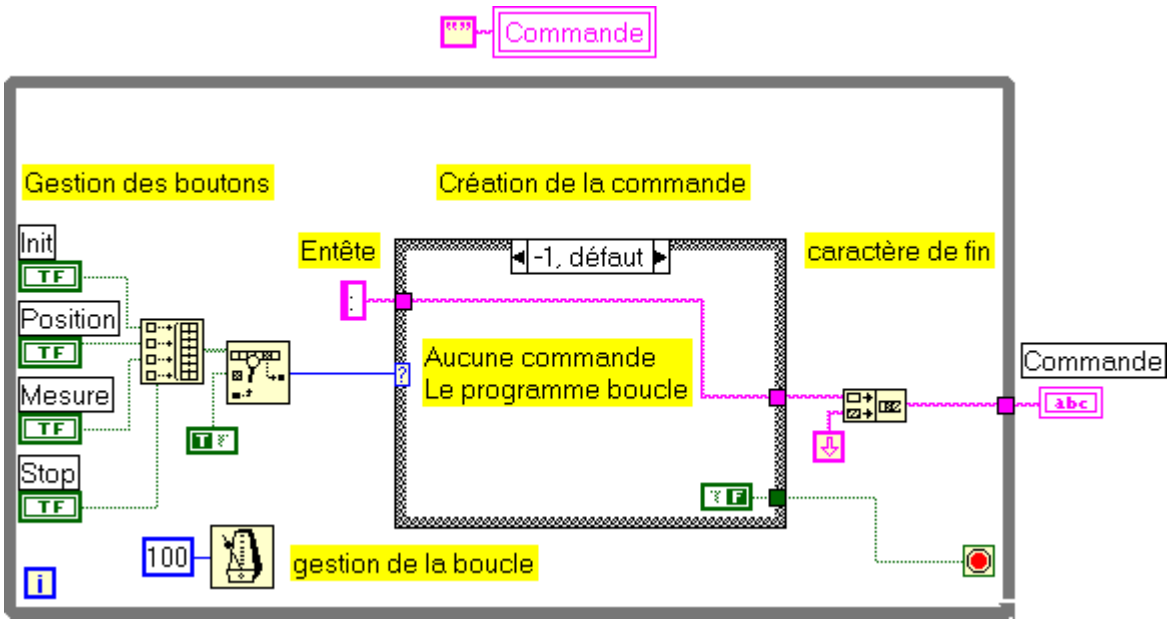
Ergonomie

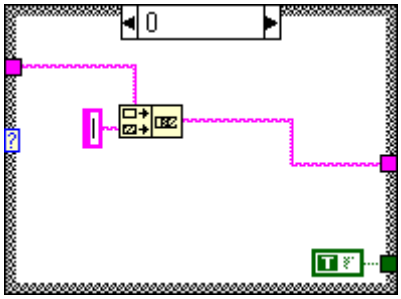
- ☑ **Le jeux de fonctions est géré par une machine d'état**
- ☑ **Dès qu'un bouton est validé, le vi retourne la chaîne de commande.**
- ☑ **Si le bouton Stop est activé, la chaîne de commande retournée est une chaîne vide.**
- ☑ **Les commandes sont obtenues par concaténation de l'entête commune (« : »), de la chaîne spécifique à chaque choix et du caractère de fin de commande (« \n »).**

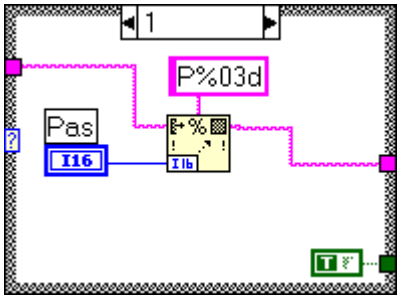
  Les commandes I **et** M **sont obtenues par simple concaténation, la commande** P **par mise en forme de la valeur du pas**

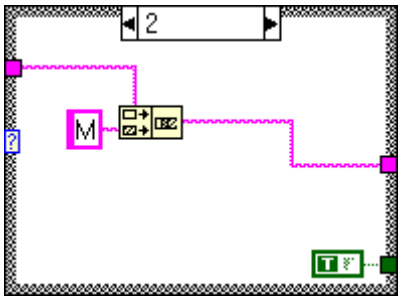
TD5_1.vi

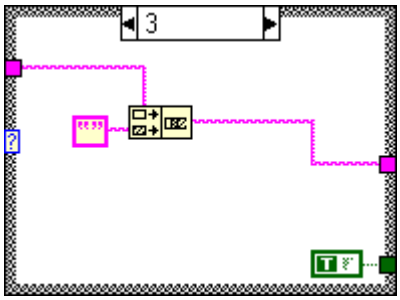
Diagramme











TD5_2.vi

Exercice N° 2: Analyse

La chaîne de commande reçue est analysée pour afficher un message indiquant soit la fonction demandée (Initialisation, Déplacement de n pas ou Mesure) soit la nature de l'erreur (format ou commande).

TD5_2.vi

Variables d'entrées:

Commande				
Sous titre		Commande	*type*	Chaîne de caractère
			apparence	
défaut	vide			

Variables de sortie:

Message			
Sous titre	Message	*type*	Chaîne de caractère
		apparence	

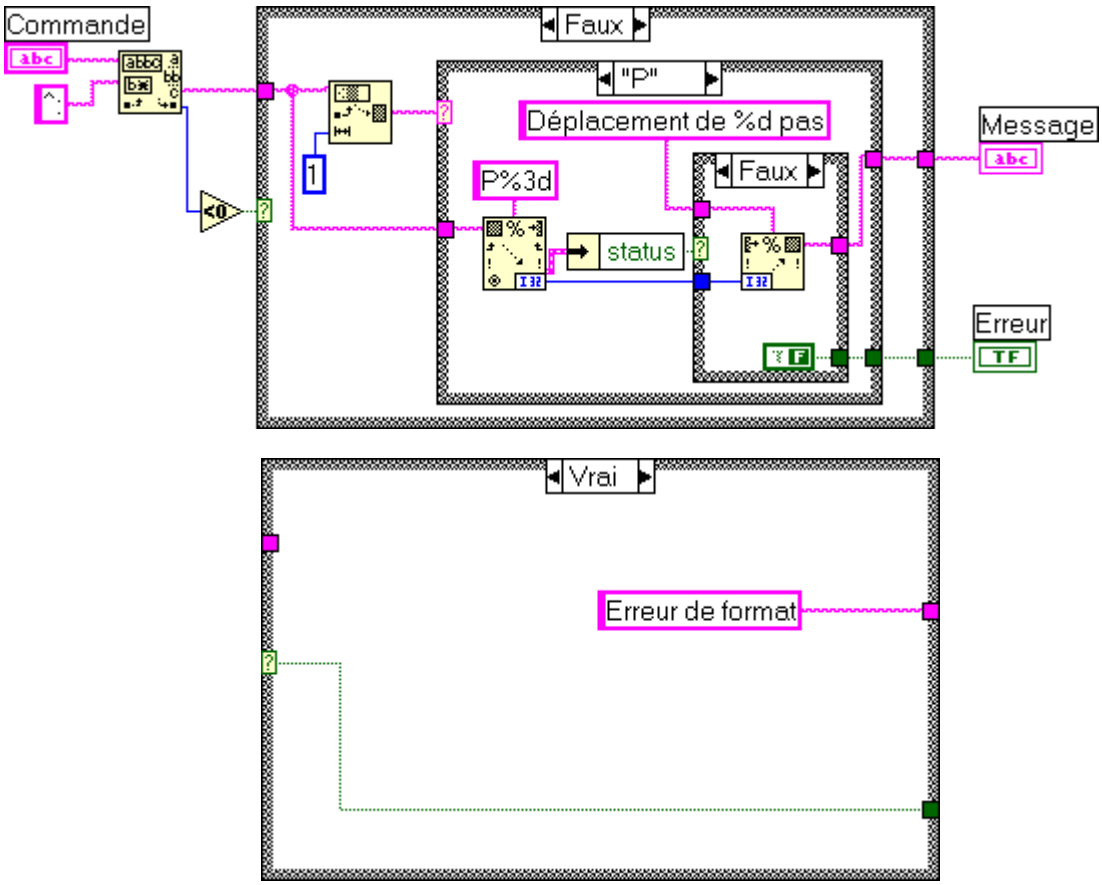
Erreur			
Sous titre	Erreur	*type*	Boléen
		apparence	Led carrée rouge

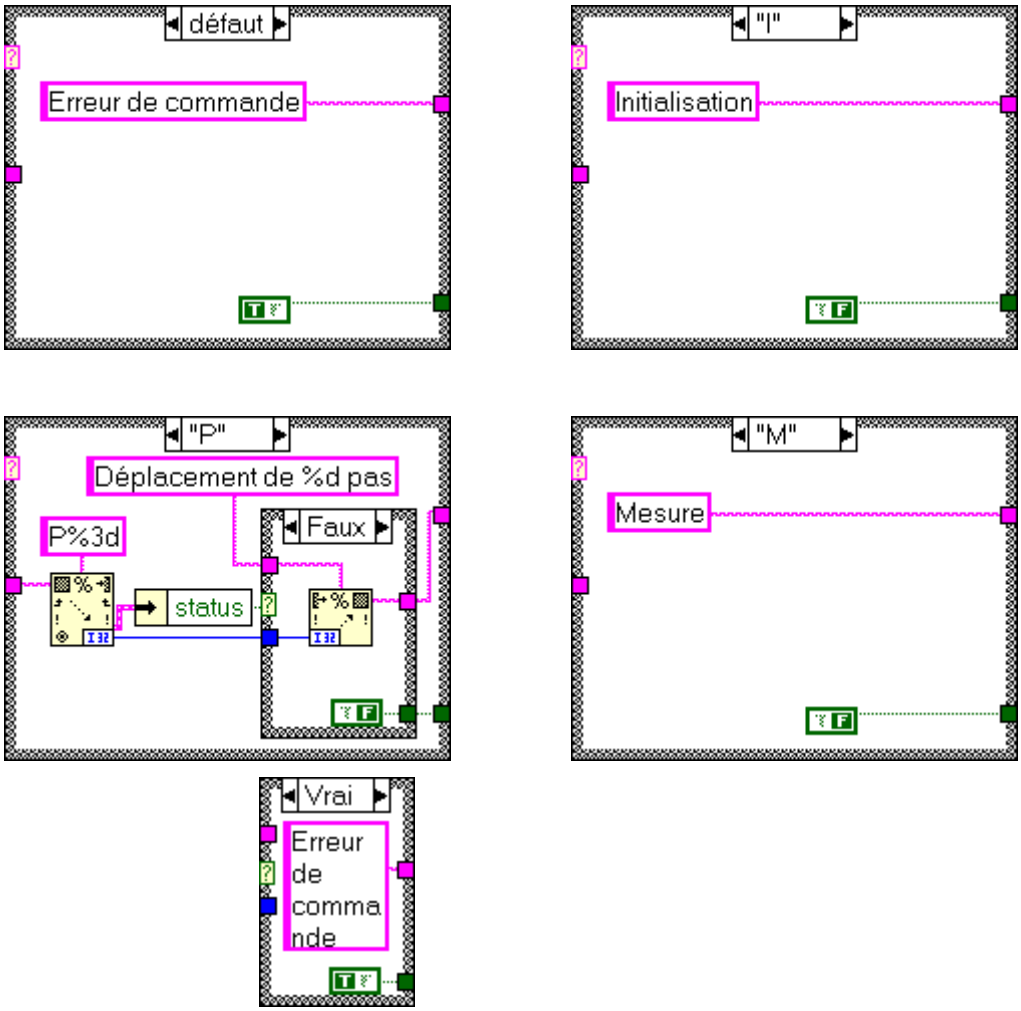
Face-avant

- ☑ Si le premier caractère n'est pas égal au caractère de début, ou si la commande ne contient pas le symbole Line Feed, le message devient « erreur de format » et la sortie d'erreur est positionnée
  sinon
  le premier caractère suivant est évalué
- ☑ Si ce caractère n'est pas une commande valide, le message devient « erreur de commande » et la sortie d'erreur est positionnée.
- ☑ Pour les commandes I et M le message est positionné en conséquence.
  Pour la commande P, la chaîne de commande est balayée pour extraire la valeur du pas. En cas d'erreur, une « erreur de commande » est spécifiée, sinon le message indiquant le nombre de pas de déplacement est mis en forme

TD5_2.vi

Diagramme





En sortie de la fonction de balayage, les éventuelles erreurs sont signalée dans une structure de type cluster. La variable booléenne de ce cluster est extraite pour tester la conformité au format recherché.

TD5_3.vi

Exercice N° 3: Communication série

Les commandes créées par TD5_1.vi sont transmises par le biais d'un port de communication au module d'analyse TD5_2.vi. Le port de communication série choisi devra être bouchonné (court-circuit des lignes TD et RD) pour permettre le dialogue.

A l'exécution, deux fenêtres sont ouvertes. Une fenêtre principale, permettant de choisir le port de communication, d'interrompre l'exécution, et d'afficher les messages d'interprétation avec les éventuelles erreurs. Une fenêtre secondaire affiche les boutons de choix de la fonction demandée.

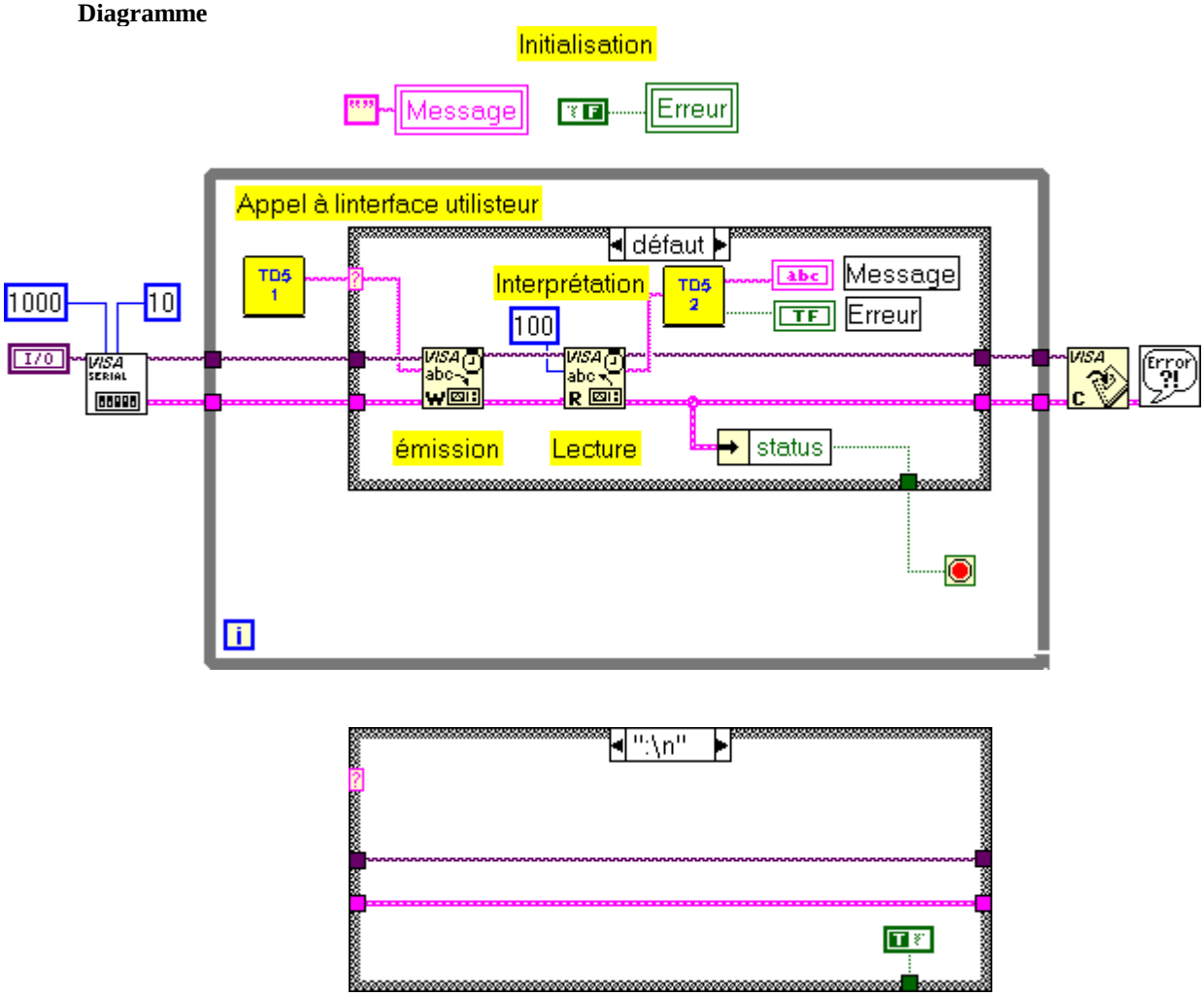
TD5_3.vi

Face-avant


Communication
nom de ressource VISA
COM1
STOP
Interpréteur de Commandes
Message
Erreur

TD5_3.vi

Diagramme



Si une erreur de communication apparaît, l'exécution est interrompue.
Le vi d'erreur permet d'interpréter le cluster d'erreur retourné par l'ensemble des vi de communication.

La durée du time-out (1 seconde) et la valeur de délimiteur de fin de commande (/n ou 0X0A ou 10**) sont définies à la configuration du port. La lecture du port se fait à concurrence de 100 caractères (impossible), de la détection du caractère de fin (/n)ou d'un délai trop long.**

Si une commande vide est retournée par TD5_1 **(bouton Stop) le vi est arrêter.**

Le nœud du sous-vi TD5_1 **est configuré pour s'ouvrir dès le chargement de l'application.**

TD6: Tableaux, clusters et graphes

Dans la suite de ce recueil de TD, nous allons étudier la gestion des mesures acquises; présentation, représentation graphique et archivage.

Après une acquisition simulée sur un tour à l'aide du VI *Simul_2.vi* nous allons présenter les résultats obtenus de diverses façons.

Nous aborderons les points suivant:

- Création d'un tableau par auto indexation
- Graphes X-Y
- Polymorphisme
- Manipulations et fonctions sur les tableaux
- Graphes
- Curseurs

TD6_1.vi

Exercice N° 1: Création d'un tableau

Nous allons présenter les résultats sous la forme d'un tableau de deux colonnes, la première contenant les angles, la seconde les déplacements (une structure identique à celle que retournerait une carte d'acquisition). La mesure se fait sur un tour, pour un nombre de points fixé (compris entre 10 et 200).

- ☑ **Reprendre** l'exercice 1 du TD3.
- ☑ **Mettre en place** les éléments de la face avant.
- ☑ **Programmer** en utilisant un tunnel avec auto-indexation.

TD6_1.vi

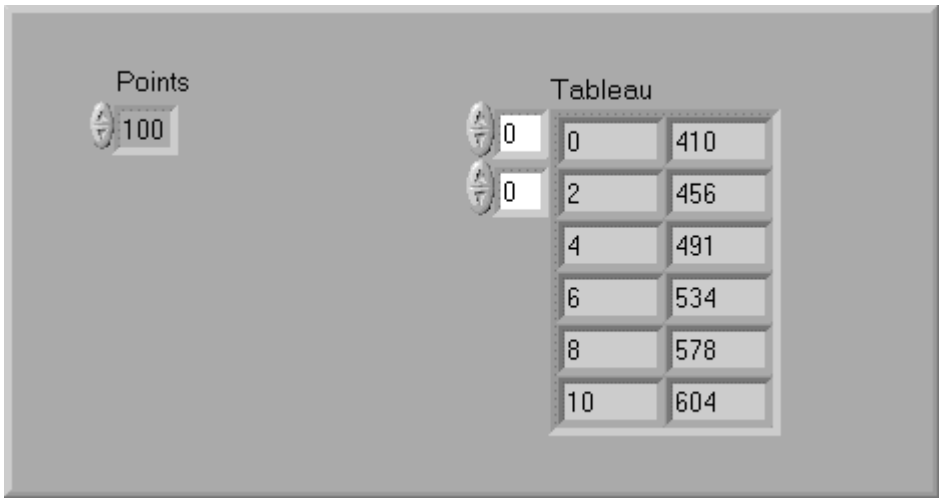
Variables d'entrées:

Nb		**Nombre de points par tour**					
Sous titre		Points		*type*	Entier 16		
				apparence	standard		
défaut	200	*min*	10	*max*	200	*incr*	1

Variables de sortie:

Tableau	**Tableau de mesure Position / Variation sur 2 colonnes**		
Sous titre	Tableau	*type*	Entier 16
		apparence	standard

Face-avant



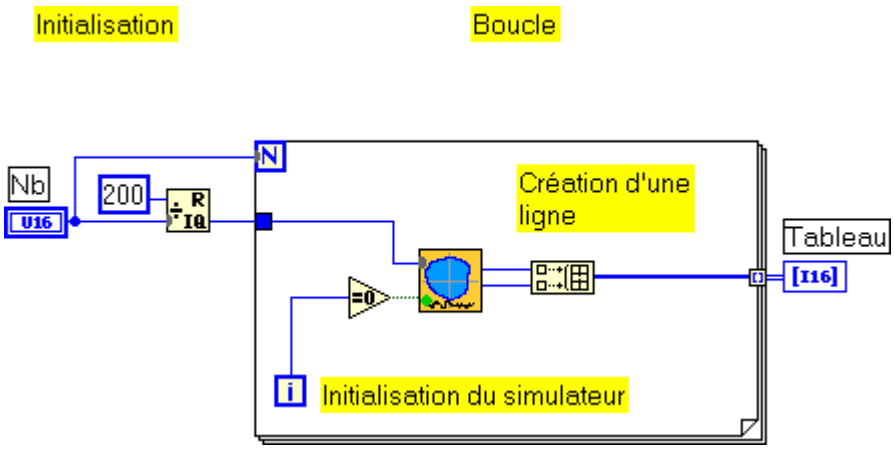
La boucle For fait un nombre d'itérations prédéterminé.

- ☑ **A chaque itération de la boucle, la variable** Pas **à passer au sous vi** Simul_2.vi. **La position dans le tour A et la mesure** Δ R **issues sous vi Simul_2.vi sont assemblées pour construire un tableau de 2 éléments. Un tunnel de sortie empile l'ensemble des lignes ainsi obtenues.**
- ☑ **A la première itération (i=0) le sous vi** Simul_2.vi **doit être initialisé en forçant l'entrée** Initialisation **à vrai**
- ☑ **Le nombre d'itérations à faire est défini par la variable Nb.**

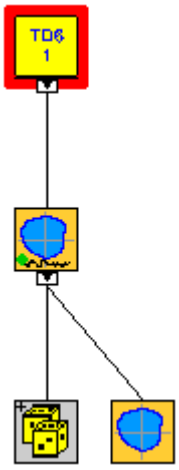
  L'incrément angulaire $I = Ent(\frac{200}{Nb})$ **est calculé en amont de la boucle.**
- ☑ **La structure contenue dans le tunnel de sortie est affichée dans la variable** Tableau**.**

TD6_1.vi

Diagramme



Position dans la hiérarchie



L'auto-indexation du tunnel de sortie doit être activée dans son menu contextuel.

TD6_2.vi

Exercice N° 2: Graphe X-Y

A partir du tableau obtenu précédemment, nous allons construire une représentation de l'excentrique sachant que la position de référence du capteur de déplacement est à 50 mm de l'axe de rotation.

☑ **Rappeler** le vi TD6_1et le **transformer** en sous-VI.

☑ **Mettre en place** les éléments de la face avant.

☑ A partir du tableau de mesure, **extraire** les vecteurs Angles et Déplacements, **calculer** le rayon et **tracer** le profil. **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Polaire vers cartésien 1D (uniquement dans la version avancée)
Convertit deux tableaux de coordonnées polaires représentées par les tableaux d'entrée d'amplitude et de phase en deux tableaux de coordonnées cartésiennes, selon les formules suivantes :

```
x = magnitude*cos(phase)
y = magnitude*sin(phase)
```

TD6_2.vi

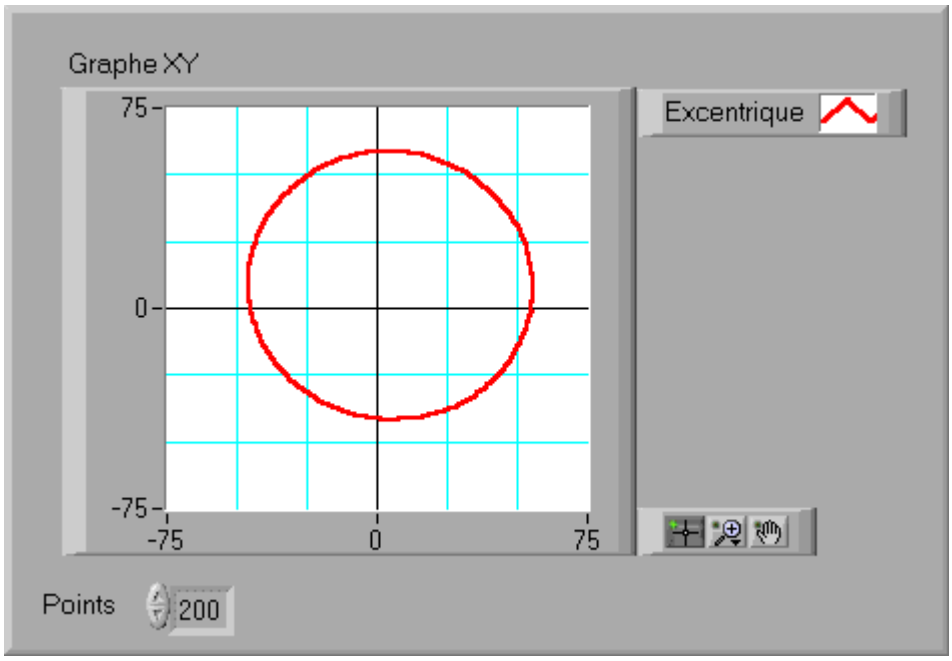
Variables d'entrées:

Nb		**Nombre de points par tour**					
Sous titre		Points		*type*	Entier 16		
				apparence	standard		
défaut	200	*min*	10	*max*	200	*incr*	1

Variables de sortie:

Graphe XY	**Profil de l'excentrique**		
Sous titre	Graphe XY	*Type*	cluster
		apparence	standard

Face-avant

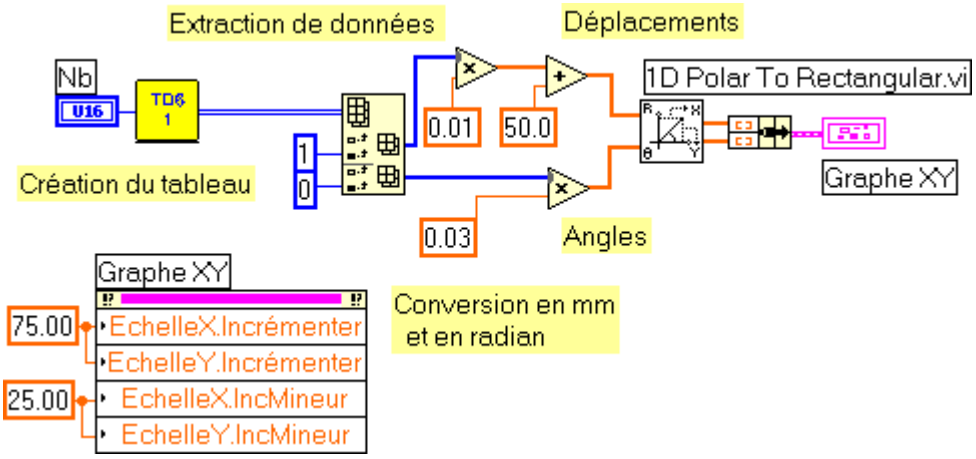


Le tableau de mesure donne des informations dans un repère polaire, la représentation que nous souhaitons se fait dans un repaire cartésien, il faudra donc changer de repère.

- ☑ **Le tableau de mesure est éclaté en deux vecteurs: le vecteur des variations de rayon et celui des angles.**
- ☑ **Chaque vecteur est converti en unité physique par multiplication par une constante (les angles en radians pour pouvoir faire les calculs de projection)**
- ☑ **le vecteur rayon est obtenu en ajoutant la position de référence du capteur de déplacement au vecteur des variations de rayon.**
- ☑ **Les coordonnées polaires obtenues sont converties en coordonnées cartésiennes puis assemblées pour former une courbe.**

TD6_2.vi

Diagramme



Remarques: d'autres solutions sont possibles pour les conversions.

La courbe est construite à partir d'un cluster contenant le tableau des abscisses et le tableau des ordonnées.

Le lecteur remarquera le polymorphisme des fonctions utilisées. Par exemple, l'addition accepte comme entrées un tableau et un scalaire et s'adapte automatiquement

TD6_3.vi

Exercice N° 3: Graphe

En utilisant le tableau retourné par le vi TD6_1.vi, construisons la représentation des variation de rayon sur un tour.

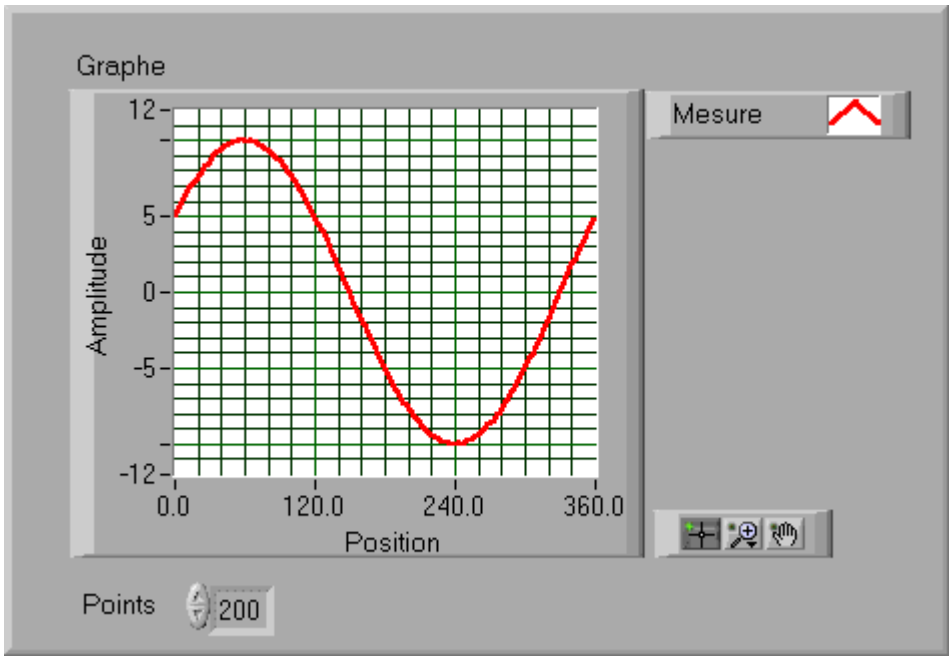
TD6_3.vi

Variables d'entrées:

Nb		**Nombre de points par tour**					
Sous titre		Points		*type*	Entier 16		
				apparence	standard		
défaut	200	*min*	10	*max*	200	*incr*	1

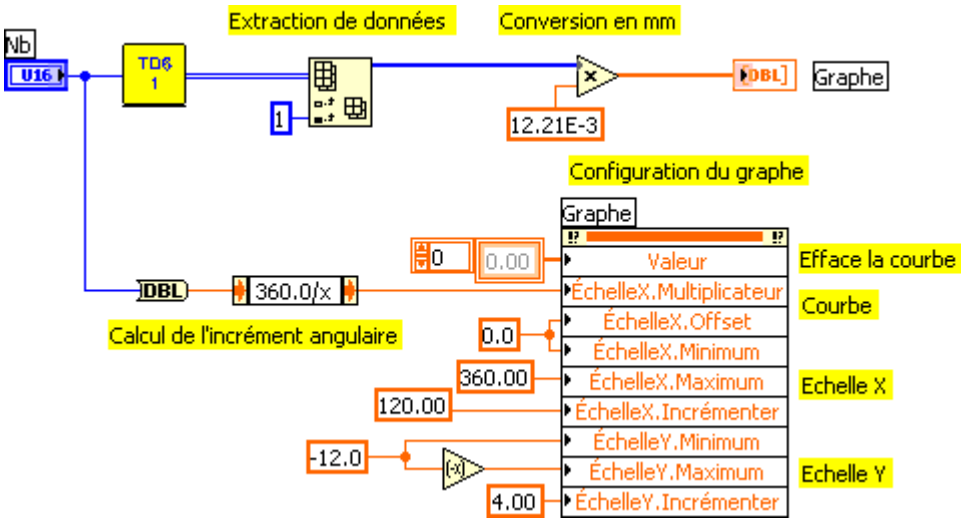
Variables de sortie:

Graphe	**Variation du rayon**		
Sous titre	Graphe	*type*	cluster
		apparence	standard


Graphe
12
5
0
-5
-12
Amplitude
0.0
120.0
240.0
360.0
Position
Mesure
Points
200

TD6_3.vi

Diagramme



La courbe est construite à partir d'un cluster $[X_0, \Delta X, Y]$

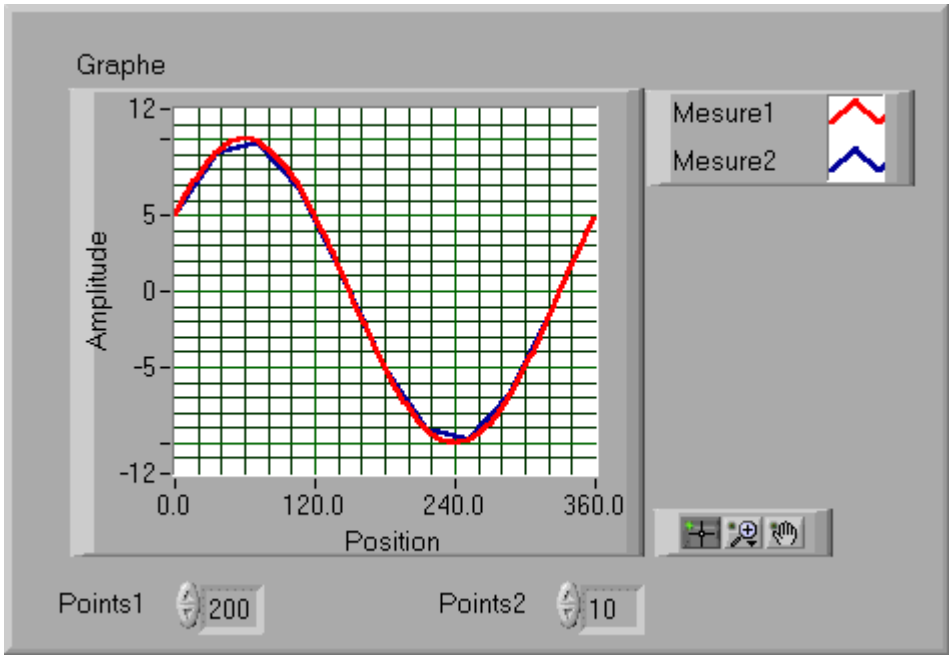
TD6_4.vi

Exercice N° 4: Graphe multicourbes

(Complémentaire)

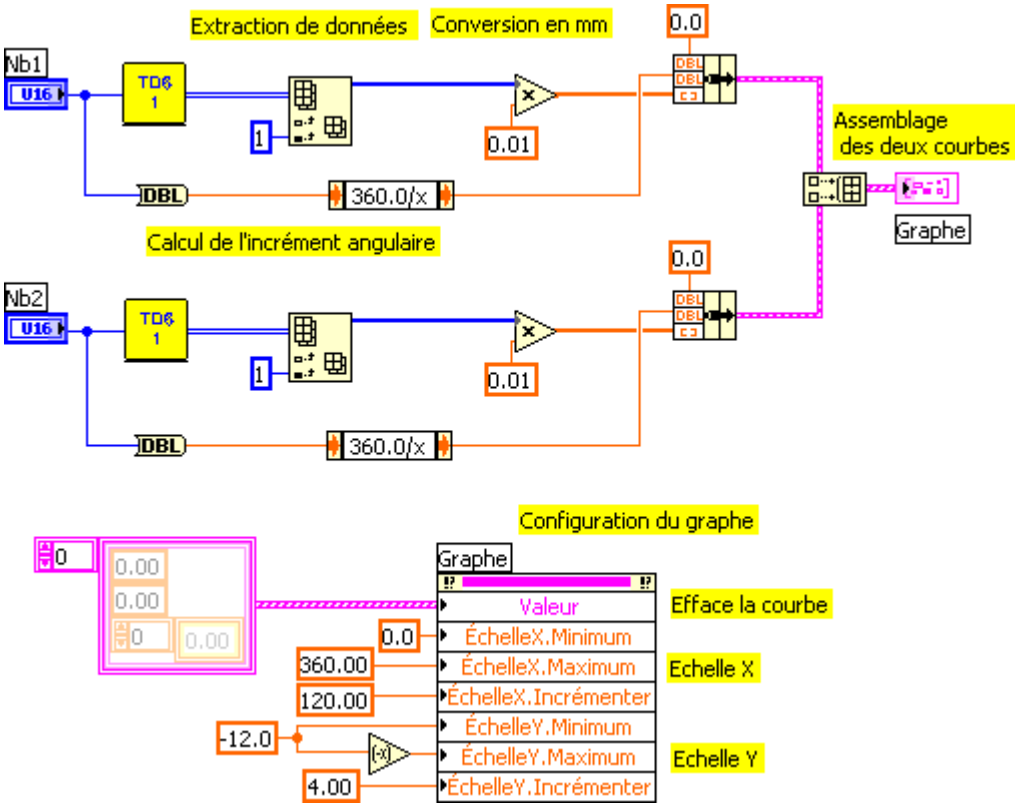
Pour observer l'impact du choix du nombre de points sur l'aspect de la courbe, vous allez représenter sur un même graphe, les courbes obtenues pour des pas différents

- ☑ **Reprendre** le vi TD4_3 et le **sauvegarder sous** TD4_4.vi
- ☑ **Dupliquer** la partie du diagramme correspondant à la construction d'une courbe.
- ☑ **Construire** le graphe multicourbes


Graphe
12
5
0
-5
-12
Amplitude
0.0
120.0
240.0
360.0
Position
Mesure1
Mesure2
Points1
200
Points2
10

TD6_4.vi

Diagramme



Les deux courbes sont assemblées dans un tableau de courbes. Le lecteur remarquera le changement de la structure d'initialisation.

TD6_5.vi

Exercice N° 5: Curseurs

(Complémentaire)

Nous allons finir le vi précédent en y ajoutant un curseur permettant de visualiser les coordonnées d'un point du graphe. Ce curseur pourra être attaché à l'une des courbes, et prendra sa couleur, ou être libre et visualisé en vert.

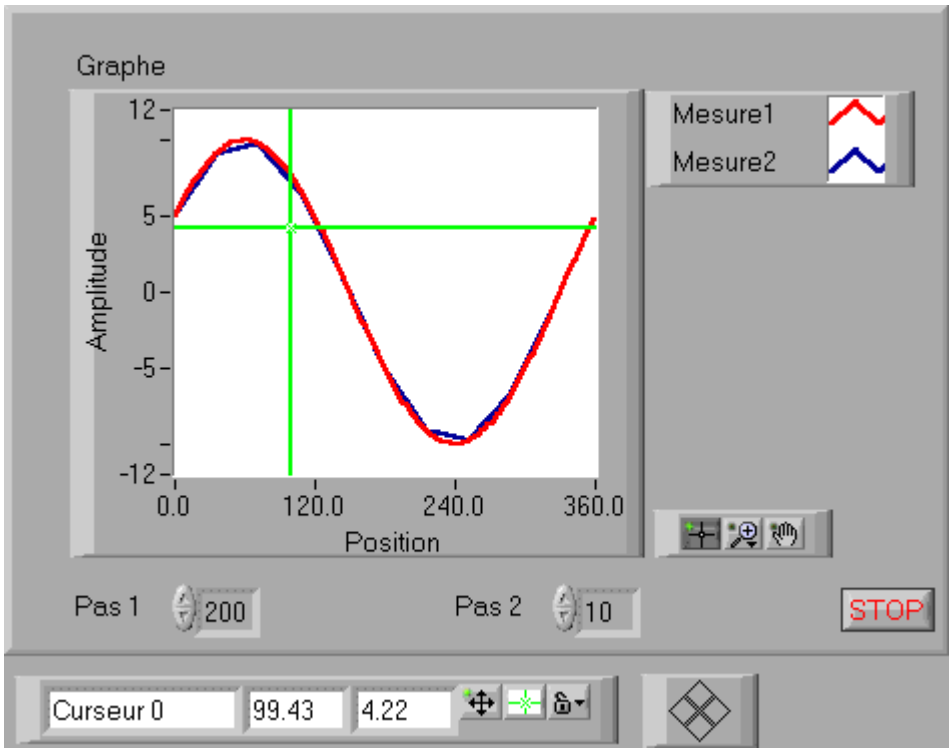
☑ **Reprendre** le vi TD6_4 et le **sauvegarder sous** TD6_5.vi.

☑ **Ajouter** un curseur et **modifier** le vi suivant l'analyse que vous avez faite du problème

TD6_5.vi

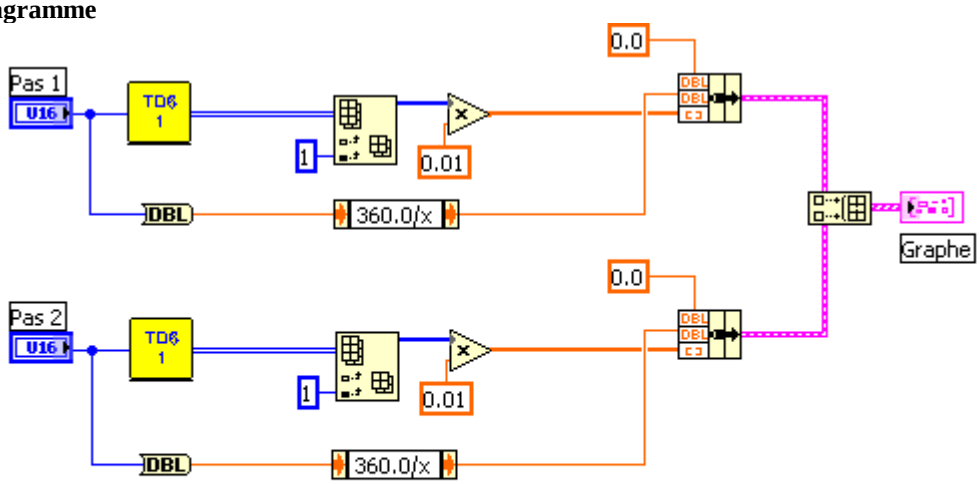
Analyse des modifications pour l'ajout du curseur

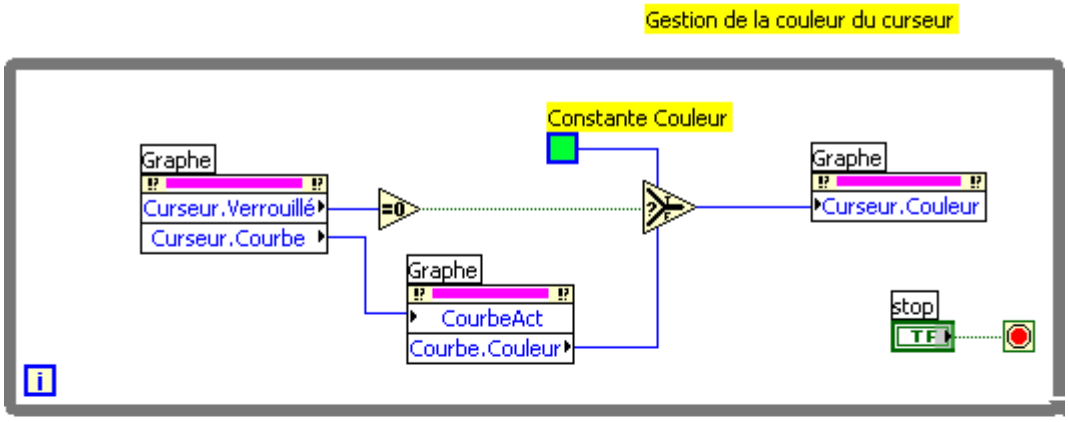
Face-avant

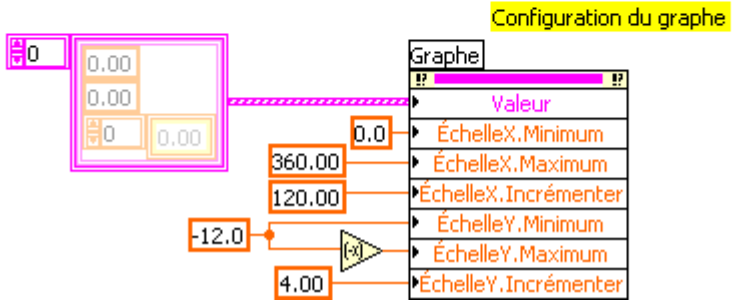


TD6_5.vi

Diagramme







La propriété Courbe.active permet d'adresser une des courbes afin de pouvoir accéder à ses paramètres.

TD7: Fichiers

Quel que soit le système développé, il est nécessaire de lire des données ou de sauvegarder des résultats dans un ou plusieurs fichiers. Nous allons voir comment enregistre et relire des données sous forme de texte tabulé, compatible avec toute suite bureautique.

De même, il est utile de pouvoir lire des données de configuration. Nous verrons comment lire un fichier contenant de telle données.

Nous aborderons les points suivants:

- Fonctions d'entrée/sortie fichiers de haut niveau
- Fonctions d'entrée/sortie fichiers de niveau intermédiaire
- Fichiers de configuration

TD7_1.vi

Exercice N° 1: Enregistrement.

En utilisant le programme écrit en premier exercice du TD6, TD6_1.vi, enregistrer le tableau de données obtenu dans le fichier de votre choix sous un format de type texte tabulé. Le nom du fichier choisi sera affiché afin de pouvoir l'ouvrir avec un tableur ou un éditeur de texte.

☑ **Ouvrir** un nouveau VI, **appeler** TD6_1.vi, **créer** les Entrées / Sorties nécessaires et l'**enregistrer** sous TD7_1.vi.

☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Écrire dans un fichier tableur
Convertit un tableau 2D ou 1D de nombres simple précision (SGL) en une chaîne de texte et écrit la chaîne dans un nouveau fichier standard ou ajoute la chaîne à un fichier existant. Vous pouvez choisir une option pour transposer les données. Le VI ouvre ou crée le fichier avant d'y écrire et le ferme ensuite. Vous pouvez utiliser ce VI pour créer un fichier texte lisible par la plupart des applications de type tableur. Ce VI requiert la fonction Tableau en chaîne au format tableur pour convertir les données.

TD7_1.vi

Variables d'entrées:

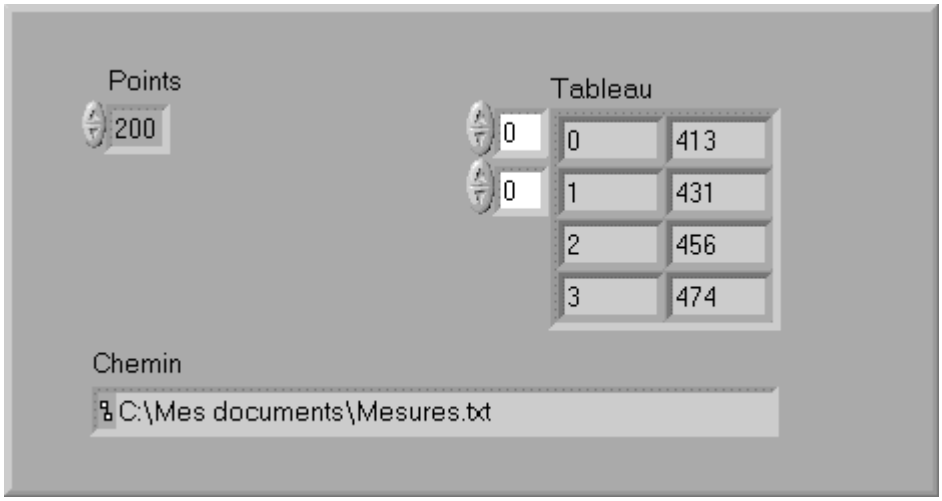
Nb		**Nombre de points par tour**					
Sous titre		Points		*type*	Entier 16		
				apparence	standard		
défaut	200	*min*	10	*max*	200	*incr*	1

Variables de sortie:

Tableau	**Tableau de mesure Position / Variation sur 2 colonnes**		
Sous titre	Tableau	*type*	Entier 16
		apparence	standard

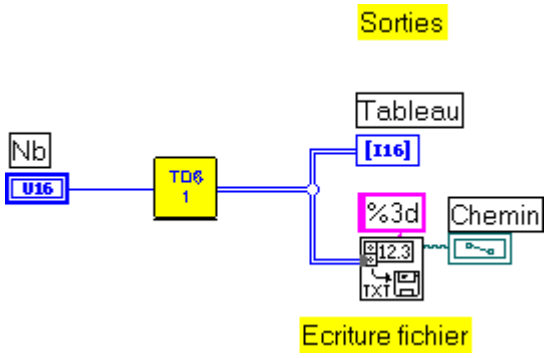
Chemin	**Position et nom du fichier de sauvegarde.**		
Sous titre	Chemin	*type*	path
		apparence	standard

Ergonomie

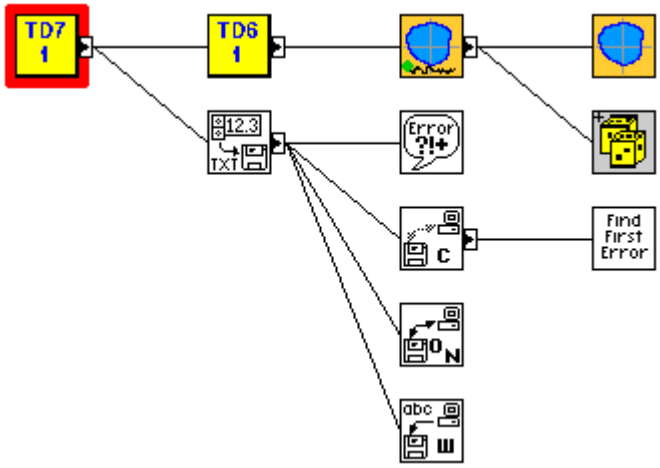

Points
200
Tableau
0
0
0
413
1
431
2
456
3
474
Chemin
C:\Mes documents\Mesures.txt

TD7_1.vi

Diagramme



Position dans la hiérarchie



La fonction par défaut ouvre l'explorateur de fichier pour choisir ou créer le fichier de sauvegarde et retourne le chemin de ce dernier.

Nous remarquons que la variable contenant cette information est d'un type spécifique.

Le format d'écriture des données est défini pour toutes les valeurs.

TD7_2.vi

Exercice N° 2: Enregistrement documenté.

Le fichier enregistré ne contient aucune information pour documenter les données. Nous allons le compléter par une entête contenant deux lignes, la première la date de mesure, la seconde le titre de chaque colonne.

☑ **Ouvrir** TD7_1.vi et l'**enregistrer** sous TD7_2.vi

☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Écrire des caractères dans un fichier
Écrit une chaîne de caractères dans un nouveau fichier standard ou ajoute la chaîne à un fichier existant. Le VI ouvre ou crée le fichier avant d'y écrire et le ferme ensuite.

Chaîne de date/heure
Convertit un nombre, indépendant du fuseau horaire, supposé être le nombre de secondes écoulées depuis le vendredi 1er janvier 1904 à midi (temps universel) en une chaîne de date/heure dans le fuseau horaire configuré pour votre ordinateur.

TD7_2.vi

Variables d'entrées:

Nb		**Nombre de points par tour**					
Sous titre		Points		*type*	Entier 16		
				apparence	standard		
défaut	200	*min*	10	*max*	200	*incr*	1

Variables de sortie:

Tableau	**Tableau de mesure Position / Variation sur 2 colonnes**		
Sous titre	Tableau	*type*	Entier 16
		apparence	standard

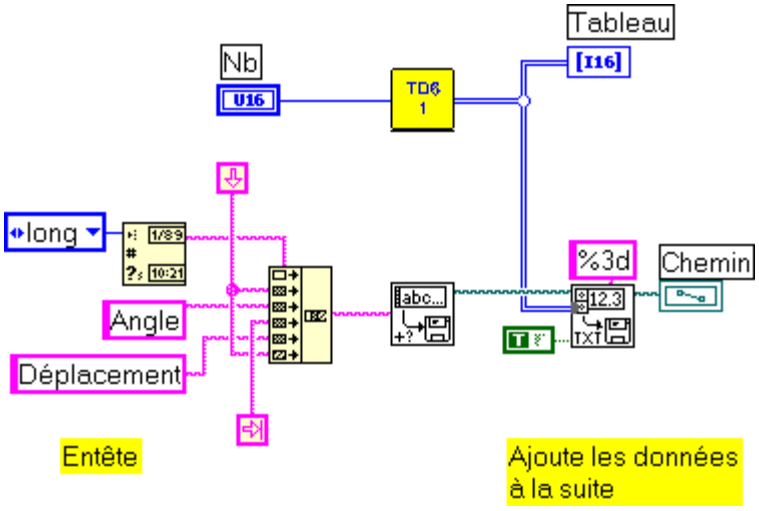
Chemin	**Position et nom du fichier de sauvegarde.**		
Sous titre	Chemin	*type*	path
		apparence	standard

Parallèlement à l'acquisition du tableau de mesure, l'entête est construite et écrite au début du fichier.

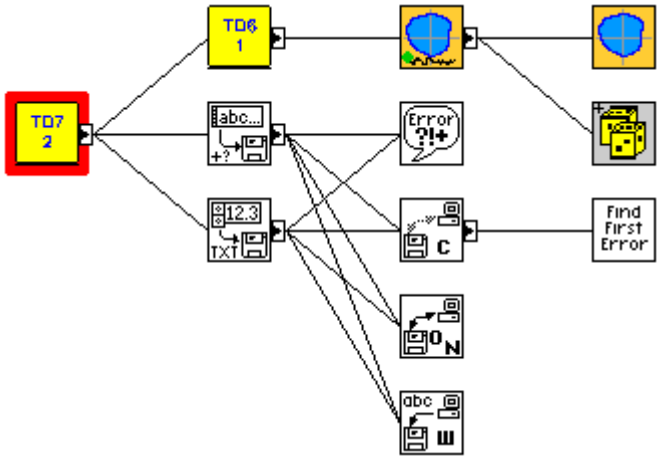
- ☑ **Parallèlement à l'acquisition du tableau de mesure, l'entête est construite par concaténation des éléments et écrite en début de fichier.**
  Elle comprend,
  la date au format long pour la première ligne (\n comme délimiteur de fin de ligne),
  la ligne de titres avec Angle **et** Déplacement **séparés par une tabulation (délimiteur de colonne)**
- ☑ **Le tableau de mesure acquis est écrit au format texte tabulé à la suite (entrée « ajouter au fichier? ») du même fichier (chaînage du nom de fichier)**

TD7_2.vi

Diagramme



Position dans la hiérarchie



Le lecteur remarquera le séquencement entre les deux fonctions d'écriture dans le fichier imposé par le flot de données; la sortie « nouveau chemin de fichier » du vi « Écrire des caractères dans un fichier » pilote l'entrée « chemin de fichier » du vi « Écrire dans un fichier tableur »

Les vi d'ouverture et de fermeture du fichier sont appelés plusieurs fois.

TD7_3.vi

Exercice N° 3: VI de niveau intermédiaire.

Dans le VI précédent, le fichier est ouvert puis fermé deux fois. Nous allons reprendre le VI de telle sorte que le fichier soit ouvert en début d'exécution, l'entête écrite, puis à chaque passage dans la boucle point par point, les données acquises enregistrées et enfin le fichier fermé.

Cette approche permet un contrôle plus fin de l'écriture sur le fichier. Le flux de données impose le séquencement des tâches.

☑ **Ouvrir** TD6_1.vi et l'**enregistrer** sous TD7_3.vi

☑ **Programmer** les opérations nécessaires en utilisant les fonctions listées ci-après:

Ouvrir/Créer/Remplacer le fichier
Ouvre un fichier existant, crée un nouveau fichier ou remplace un fichier existant en utilisant une boîte de dialogue de fichier de façon programmée ou interactive.

Écrire dans un fichier
Écrit les données dans le fichier désigné par refnum. L'écriture commence à l'emplacement désigné par mode pos et offset de pos pour un fichier standard et à la fin du fichier pour les fichiers journaux.

Formater dans un fichier
Convertit des arguments d'entrée en une chaîne résultante dont le format est déterminé par la chaîne de format.

Fermer un fichier
Ferme le refnum spécifié et renvoie le chemin du fichier associé au refnum.

Gestionnaire d'erreur simple
Détermine si une erreur est survenue. Il contient une base de données des codes d'erreur et de leur description, en fonction de laquelle il crée une description de l'erreur et affiche éventuellement une boîte de dialogue. Gestionnaire d'erreur simple appelle le gestionnaire d'erreur général et possède la même fonctionnalité de base que Gestionnaire d'erreur général, mais avec moins d'options.

TD7_3.vi

Variables d'entrées:

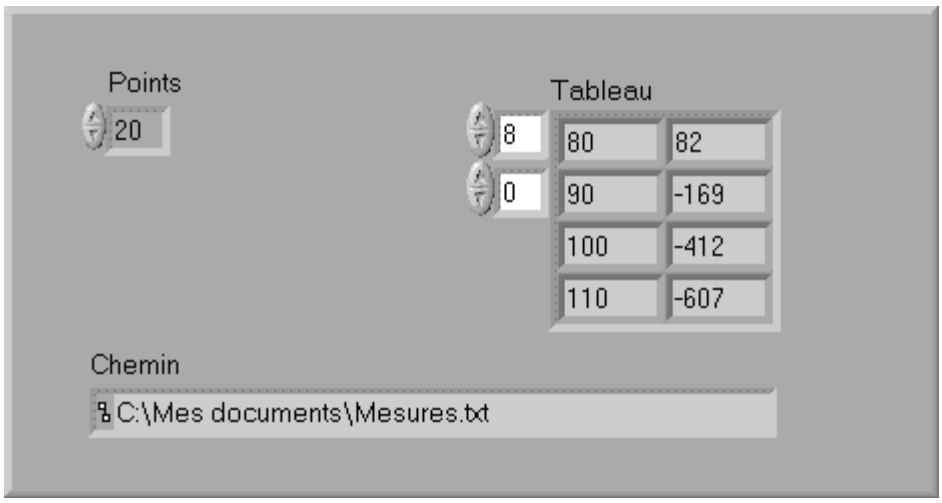
Nb		**Nombre de points par tour**					
Sous titre		Points		*type*	Entier 16		
				apparence	standard		
défaut	200	*min*	10	*max*	200	*incr*	1

Variables de sortie:

Tableau	**Tableau de mesure Position / Variation sur 2 colonnes**		
Sous titre	Tableau	*type*	Entier 16
		apparence	standard

Chemin	**Position et nom du fichier de sauvegarde.**		
Sous titre	Chemin	*type*	path
		apparence	standard

Ergonomie



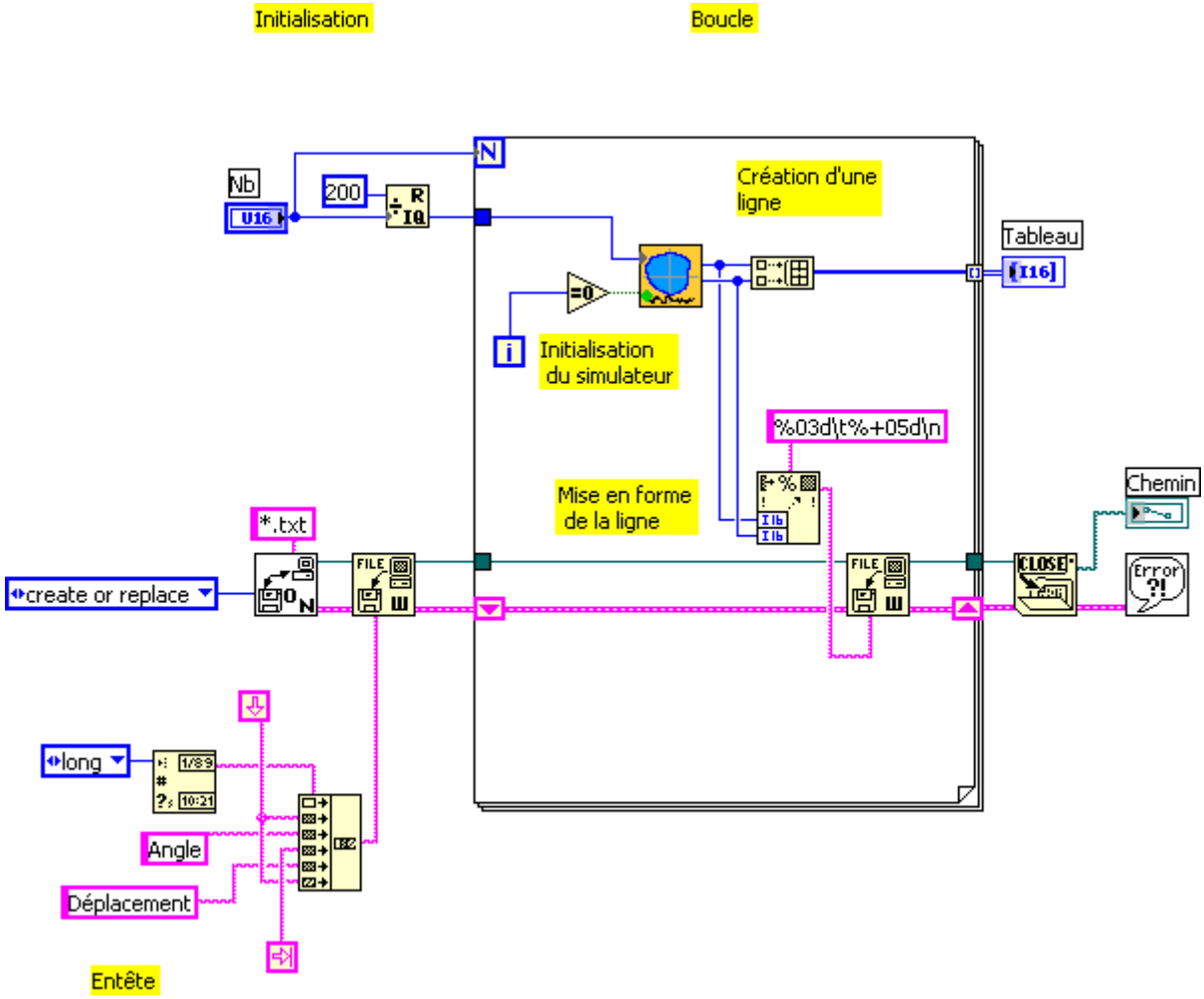
La boucle For fait un nombre d'itérations prédéterminé.

- ☑ **A chaque itération de la boucle, la variable** Pas **à passer au sous vi** Simul_2.vi. **La position dans le tour A et la mesure** Δ R **issues sous vi Simul_2.vi sont assemblées pour construire un tableau de 2 éléments. Un tunnel de sortie empile l'ensemble des lignes ainsi obtenues. Les mesures mises en forme (ligne tabulée) sont écrites à la suite du fichier ouvert, les erreurs éventuelles sont traitées afin d'interrompre l'écriture si nécessaire.**
- ☑ **A la première itération (i=0) le sous vi** Simul_2.vi **doit être initialisé en forçant l'entrée** Initialisation **à vrai**
- ☑ **Le nombre d'itérations à faire est défini par la variable Nb.**

  L'incrément angulaire $I = Ent(\frac{200}{NB})$ **est calculé en amont de la boucle.**
- ☑ **Le fichier est ouvert (créé ou remplacé) avant d'y écrire l'entête (voir analyse précédente).**
- ☑ **La structure contenue dans le tunnel de sortie est affichée dans la variable** Tableau**.**
- ☑ **Le fichier est fermé et les éventuelles erreurs reportées.**

TD7_3.vi

Diagramme



Le format d'écriture des données peut être défini de manière spécifique pour chaque valeur.

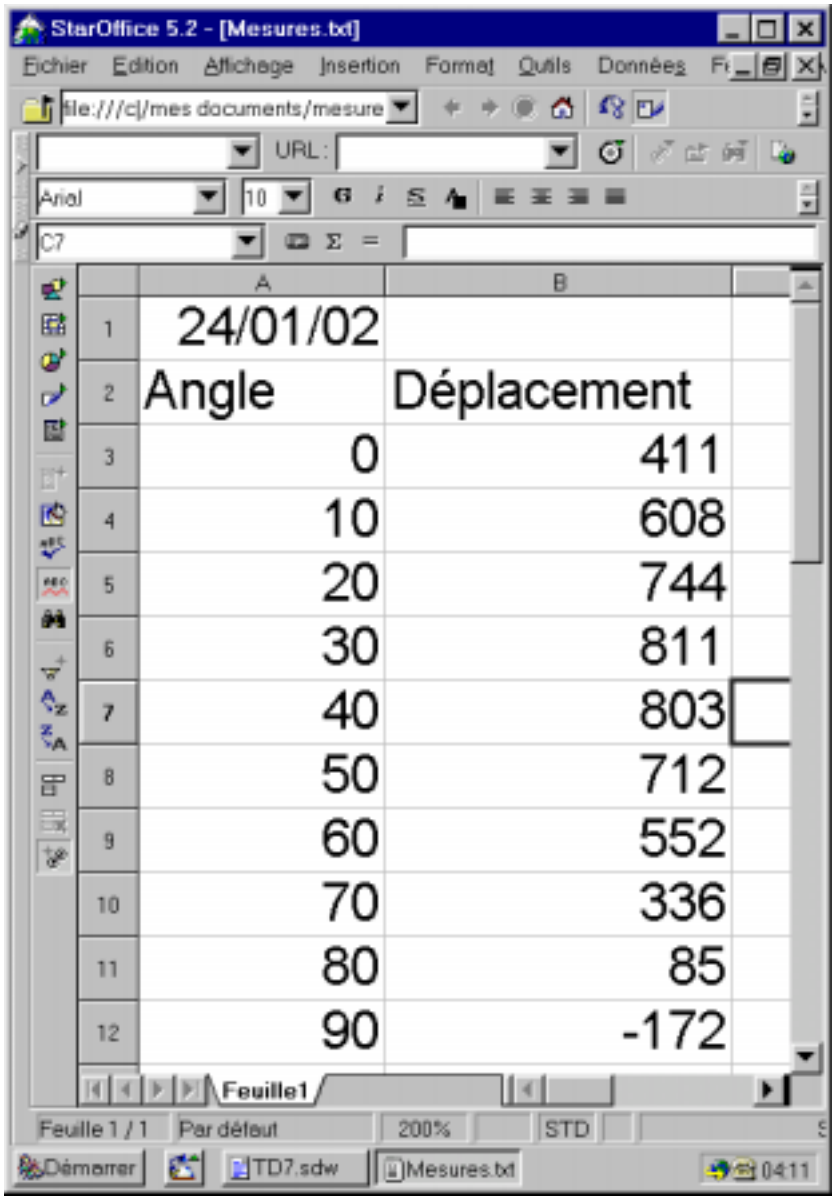
Le registre à décalage permet d'évaluer à chaque boucle l'éventuelle erreur apparue à la boucle précédente et interrompre ainsi le processus d'écriture le cas échéant. En son absence, se serait l'éventuelle erreur commise lors de l'écriture de l'entête qui serait testée à chaque itération.

TD7_4.vi

Exercice N° 4: Fichier texte.

Les fichiers créés à l'exercice précédent doivent être relus et les mesures présentées sous forme de tableau.

Pour mémoire, vous trouverez ci-contre l'un de ces fichiers ouvert avec un tableur.



	A	B
1	24/01/02	
2	Angle	Déplacement
3	0	411
4	10	608
5	20	744
6	30	811
7	40	803
8	50	712
9	60	552
10	70	336
11	80	85
12	90	-172

TD7_4.vi

Variables d'entrées:

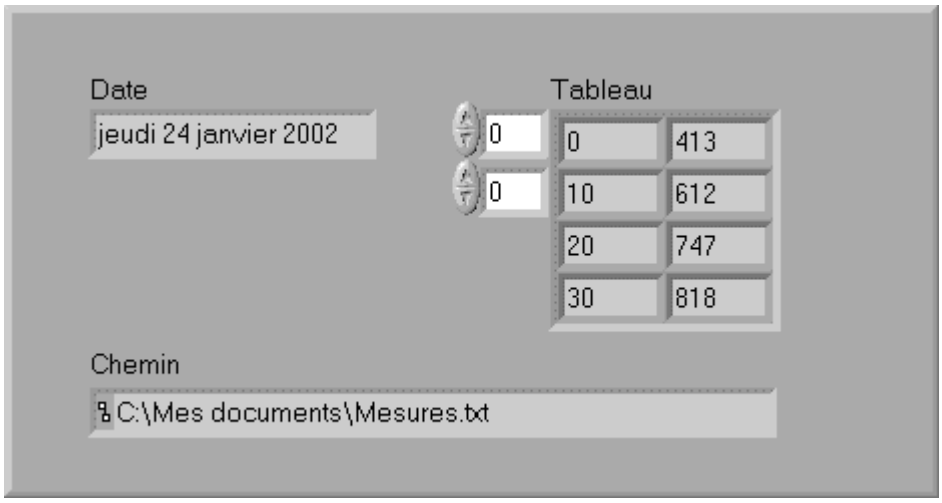
Variables de sortie:

Tableau	**Tableau de mesure Position / Variation sur 2 colonnes**		
Sous titre	Tableau	*type*	Entier 16
		apparence	standard

Chemin	**Position et nom du fichier lu.**		
Sous titre	Chemin	*type*	path
		apparence	standard

Date	**Position et nom du fichier de sauvegarde.**		
Sous titre	Date	*Type*	Chaîne de caractères
		apparence	standard

Ergonomie

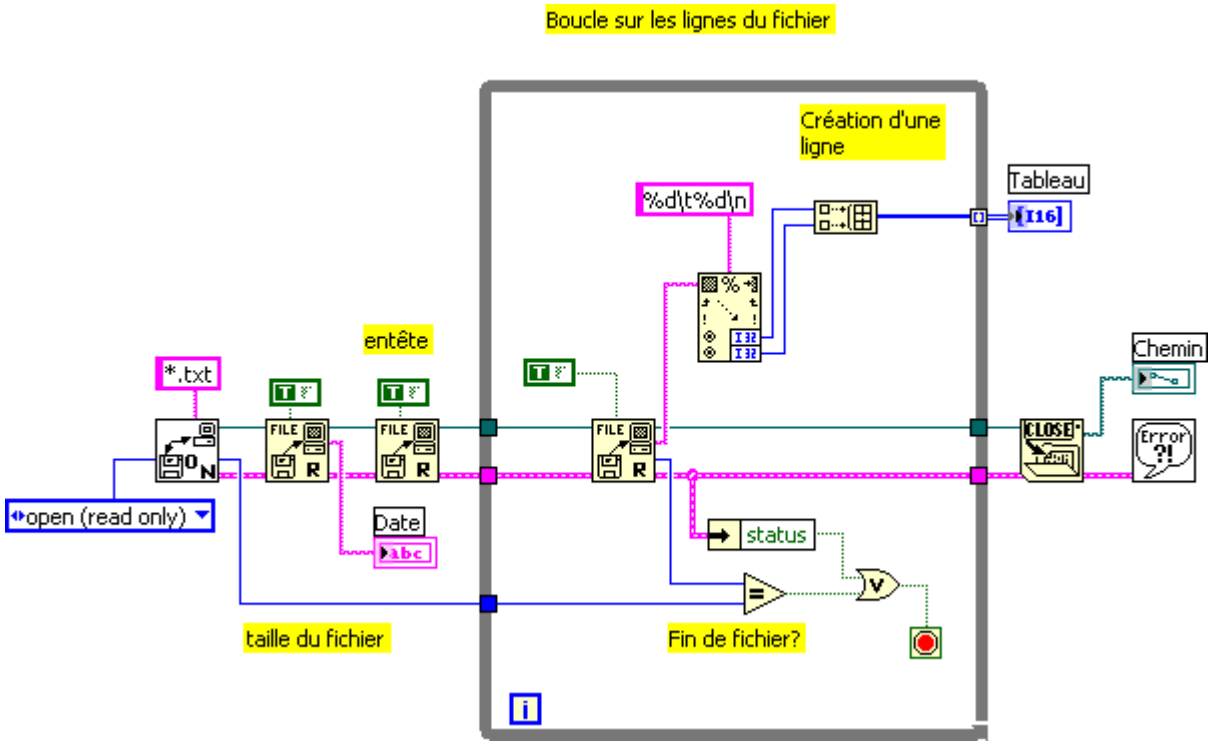


Les fichiers sont enregistrés sous forme de lignes de texte, deux pour l'entête suivies d'un nombre indéterminé pour les mesures. La forme générale du vi sera une boucle contrôlée par le nombre de lignes de données restant à lire (structure while).

- ☑ **A chaque itération de la boucle, une ligne du fichier est lue. La position dans le tour A et la mesure ΔR sont extraites et assemblées pour construire un tableau de 2 éléments.**
  Un tunnel de sortie empile l'ensemble des lignes ainsi obtenues.
- ☑ **La pointeur courant sur le fichier est comparé à la taille du fichier ouvert pour déterminer la condition d'arrêt (une ligne comprend au pons deux caractères). Une erreur de lecture arrête également la boucle.**
- ☑ **En début de programme le fichier est ouvert en lecture, la première ligne (date) est lue est sont contenu affiché. La deuxième ligne (titres des colonnes) est lue avant de commencer la boucle.**
  Le tableau est initialisé.
- ☑ **En fin de boucle, le fichier est fermé et les éventuelles erreurs reportées.**

TD7_4.vi

Diagramme



TD7_5.vi

Exercice N° 5: Fichier de configuration.

(Complémentaire)

Pour pouvoir avoir accès à un fichier de paramètre de façon aléatoire, LabVIEW supporte les fichiers de configuration au sens Windows du terme.

Nous allons écrire un VI qui récupère les paramètres de configuration du banc consignés dans un fichier TD7.ini écrit comme si dessous avec votre éditeur préféré.

```
Définition des paramètres du banc

[Copyright]
@= ESIM

[Moteur]
Pas= 200
Inc= 1.8

[Capteur]
a= 0.01220703125
b= 0.0
Réf= 50.0
```

TD7_5.vi

Variables de sortie:

Auteur	**Auteur du vi**		
Sous titre	Auteur	*type*	Chaîne de caractère
		apparence	standard

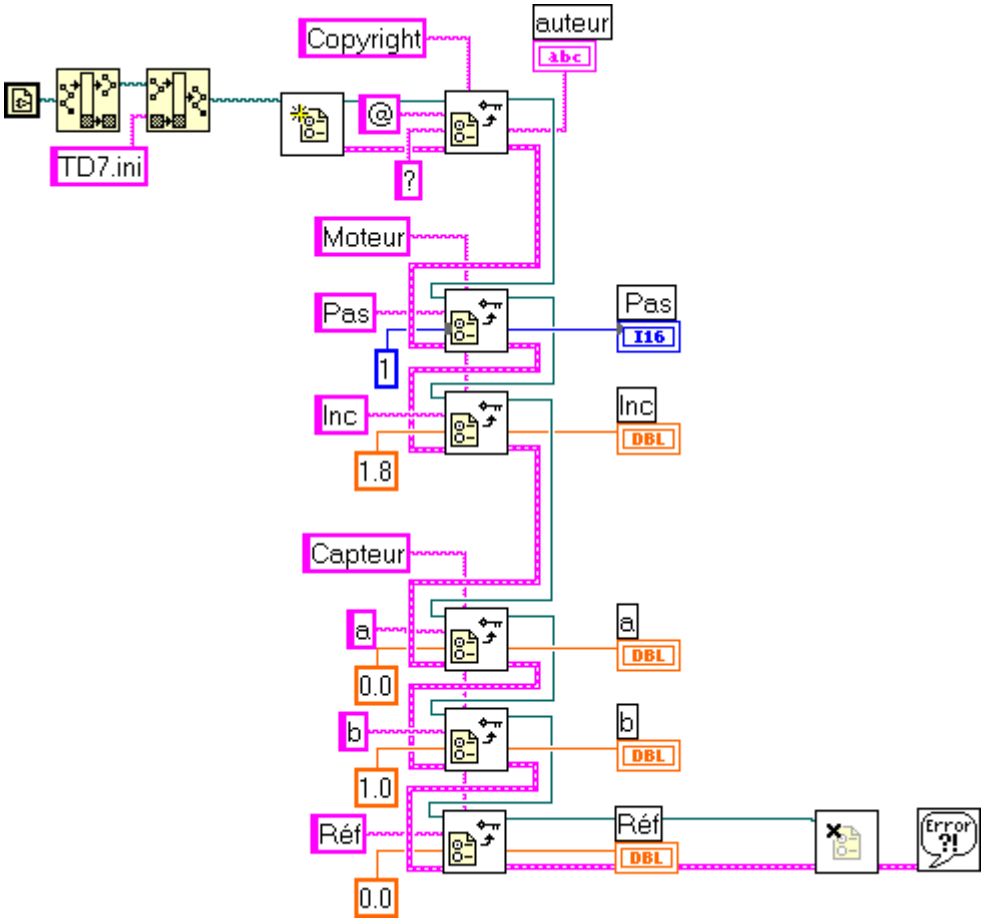
Sous titre		*type*	
		apparence	standard

Ergonomie

- ☑ **A partir du chemin du vi appelant (** TD7_5.vi **) le chemin du fichier de configuration est construit.**
- ☑ **Le fichier est ouvert**
- ☑ **Les paramètres souhaités sont pointés par le nom de la section et de la clef pour être lu.**
- ☑ **Le fichier est fermé et les éventuelles erreurs reportées.**

TD7_1.vi

Diagramme



TD8: Structures complémentaires

Sous la forme d'un travail de synthèse visant à faire le prototype d'un logiciel de contrôle d'excentrique, nous allons introduire trois structures complémentaires.

Nous aborderons les points suivant:

- Structure séquence
- Boite de calcul
- Boite à onglets

TD8_1.vi

Exercice N° 1: Structure séquence

Pour saisir les informations relatives à la pièce à tester, nous allons écrire un vi qui demande un nom de 8 lettres, puis une confirmation du diamètre et de la tolérance associée avant de construire la référence de la pièce comprenant le nom suivi de la valeur du diamètre (Excentrique-100) qui servira de nom au fichier de sauvegarde.

Ce vi pourrait être construit de plusieurs façons, nous utiliserons ici une structure séquence.

TD8_1.vi

Variables d'entrées:

Excentrique		**Nom de la pièce à tester**					
Sous titre		Excentrique		*type*	Chaîne de caractères		
				apparence	standard		
défaut		*min*		*max*		*incr*	

Diamètre		**Valeur du diamètre**					
Sous titre		Diamètre		*type*	I16		
				apparence	standard		
défaut	100	*min*	50	*max*	150	*incr*	10

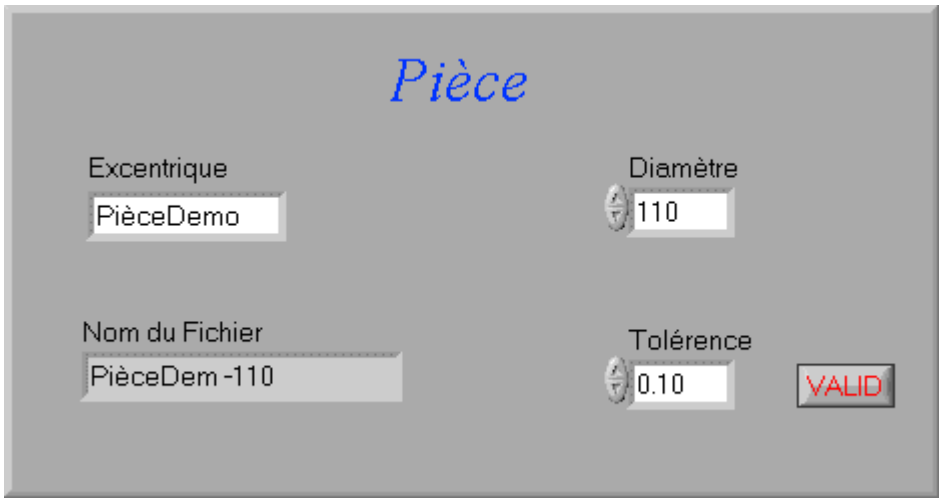
Tolérance		**Tolérance sur le diamètre**					
Sous titre		Tolérance		*type*	DBL		
				apparence	standard		
défaut	0.1	*min*	0.05	*max*	0.5	*incr*	0.01

Valid		**Bouton de validation de la saisie**					
Sous titre		Valid		*type*	Boléen		
				apparence	standard		
défaut	F	*min*		*max*		*incr*	

Variables de sortie:

fich	**Nom du fichier de sauvegarde**		
Sous titre	Nom du fichier	*type*	Chaîne de caractères
		apparence	standard

Face-avant

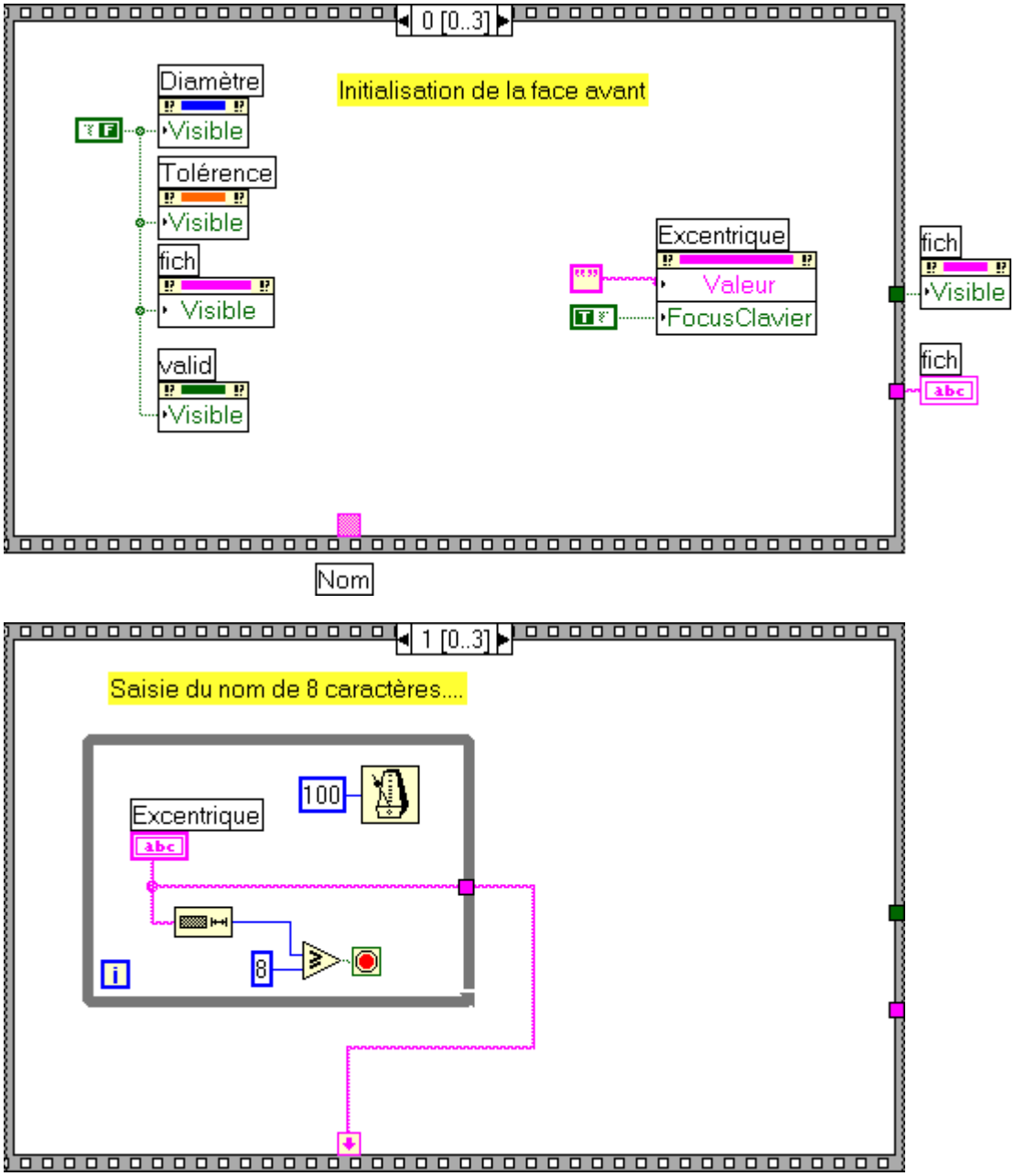


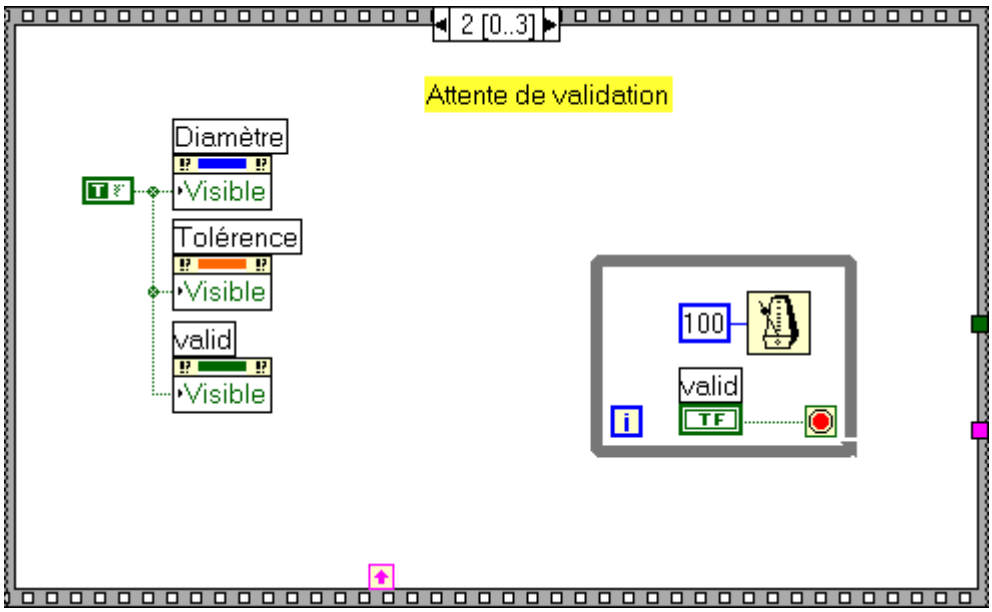
La structure séquence assure l'enchaînement des actions dans un ordre imposé

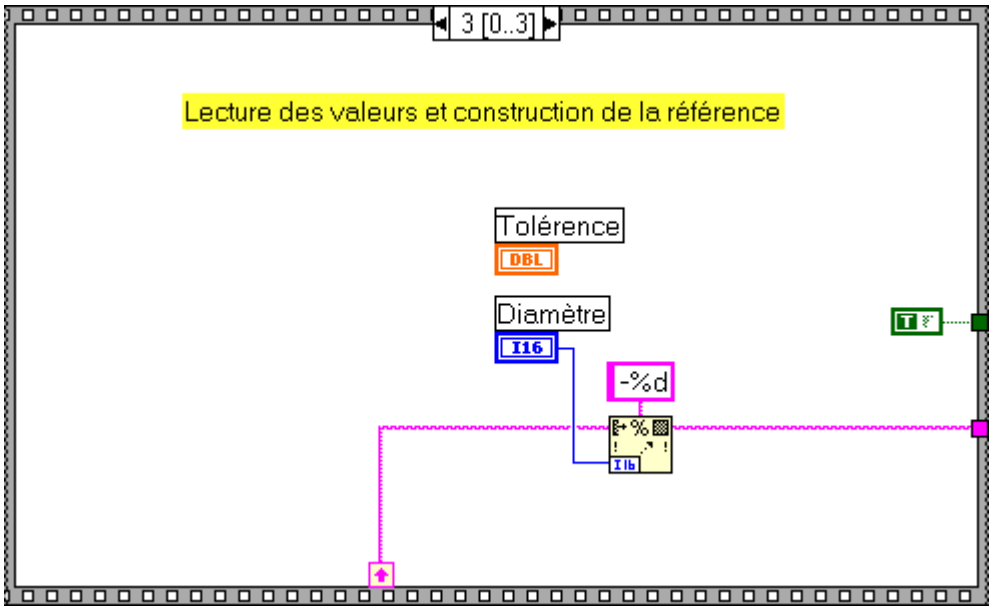
- ☑ **Initialisation de la face avant, seul le contrôle** Excentrique **est visible, le curseur est positionné dessus.**
- ☑ **Saisie de la Chaîne à concurrence de 8 caractères (boucle While).**
- ☑ **Les contrôles sont rendus visibles. On attend la validation(boucle While).**
- ☑ **Construction du nom de fichier et affichage.**

TD8_1.vi

Diagramme




2 [0..3]
Attente de validation
Diamètre
Visible
Tolérence
Visible
valid
Visible
100
valid
TF


3 [0..3]
Lecture des valeurs et construction de la référence
Tolérence
DBL
Diamètre
I16
-%d
I16

TD8_2.vi

Exercice N° 2: Boite de calcul

A partir des informations sur le diamètre, valeur et tolérance, et d'une mesure factice du diamètre, nous allons vérifier si la mesure est valide. Nous utiliserons une boite de calcul.

TD8_2.vi

Variables d'entrées:

Mesure		**Valeur mesuré du diamètre**					
Sous titre		Excentrique		*type*	DBL		
				apparence	standard		
défaut		*min*		*max*		*incr*	

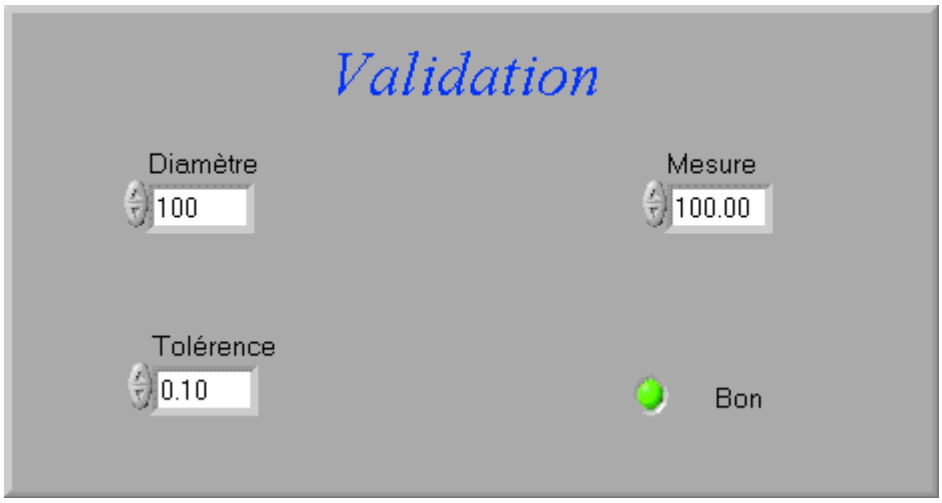
Diamètre		**Valeur du diamètre**					
Sous titre		Diamètre		*type*	I16		
				apparence	standard		
défaut	100	*min*	50	*max*	150	*incr*	10

Tolérance		**Tolérance sur le diamètre**					
Sous titre		Tolérance		*type*	DBL		
				apparence	standard		
défaut	0.1	*min*	0.05	*max*	0.5	*incr*	0.01

Variables de sortie:

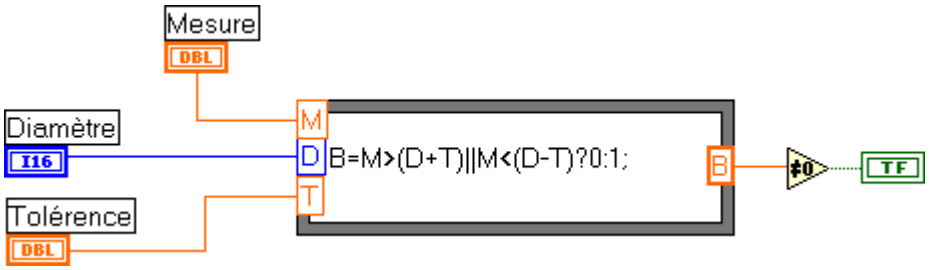
Valide	**Résultat du test de validité**		
Sous titre	Valide	*type*	Boléan
		apparence	Led Rouge/Verte

Face-avant



TD8_2.vi

Diagramme



Remarques: Syntaxe du C

TD8_3.vi

Exercice N° 3: Boite à onglets

Nous allons écrire l'ossature d'un vi de test d'excentrique qui devrais nous permettre de:

- ☑ Saisir l'identification d'une pièce.
- ☑ Lancer une séquence de mesure.
- ☑ Visualiser et vérifier la pièce.
- ☑ Sauvegarder les résultats.

Pour chaque pièce, il est possible de mesurer et visualiser plusieurs fois, seul le dernier contrôle est sauvegardé.

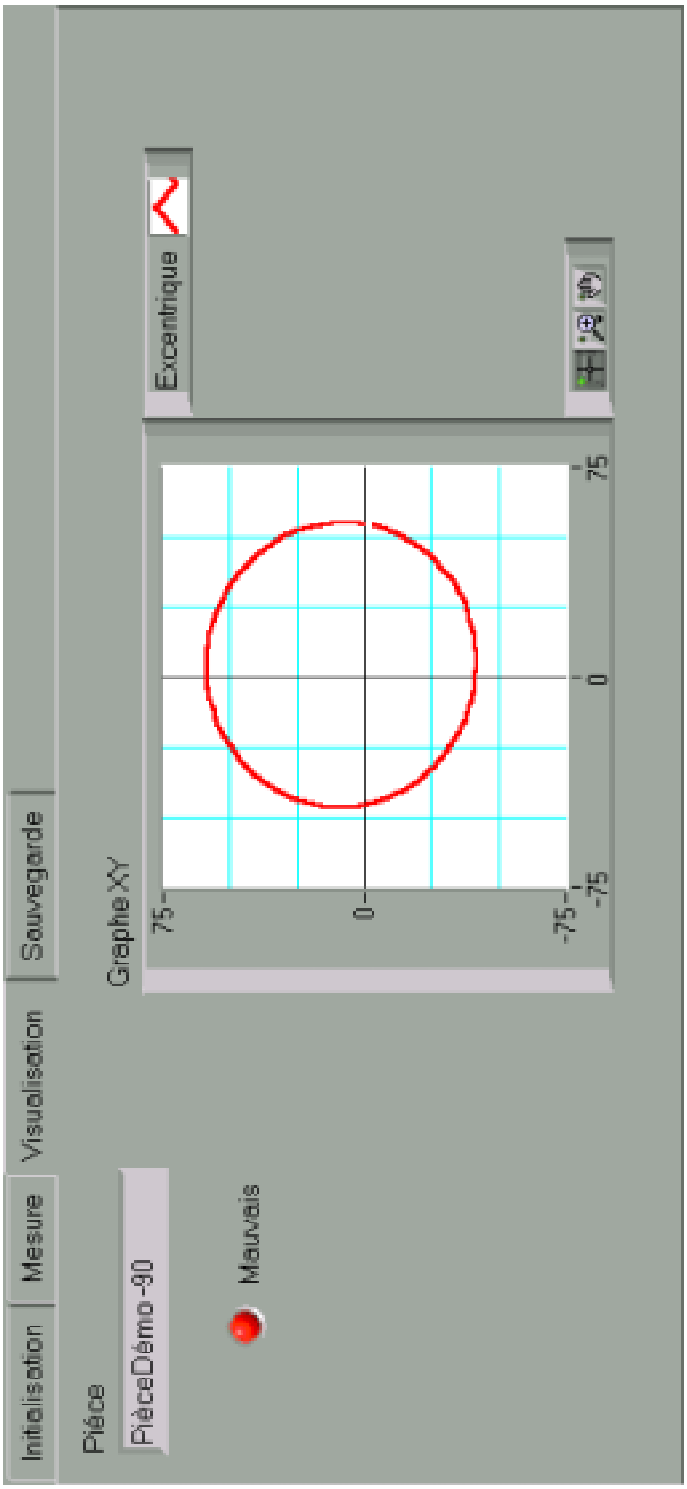
Chacune de ces phases du processus demande une interface spécifique. Nous pourrions faire appel à plusieurs sous-vi, chacun avec son IHM spécifique, ou utiliser une boite à onglets.

Vous vous appuierez sur les vi déjà écrits pour construire ce dernier.

TD8_3.vi

Variables

Face-avant

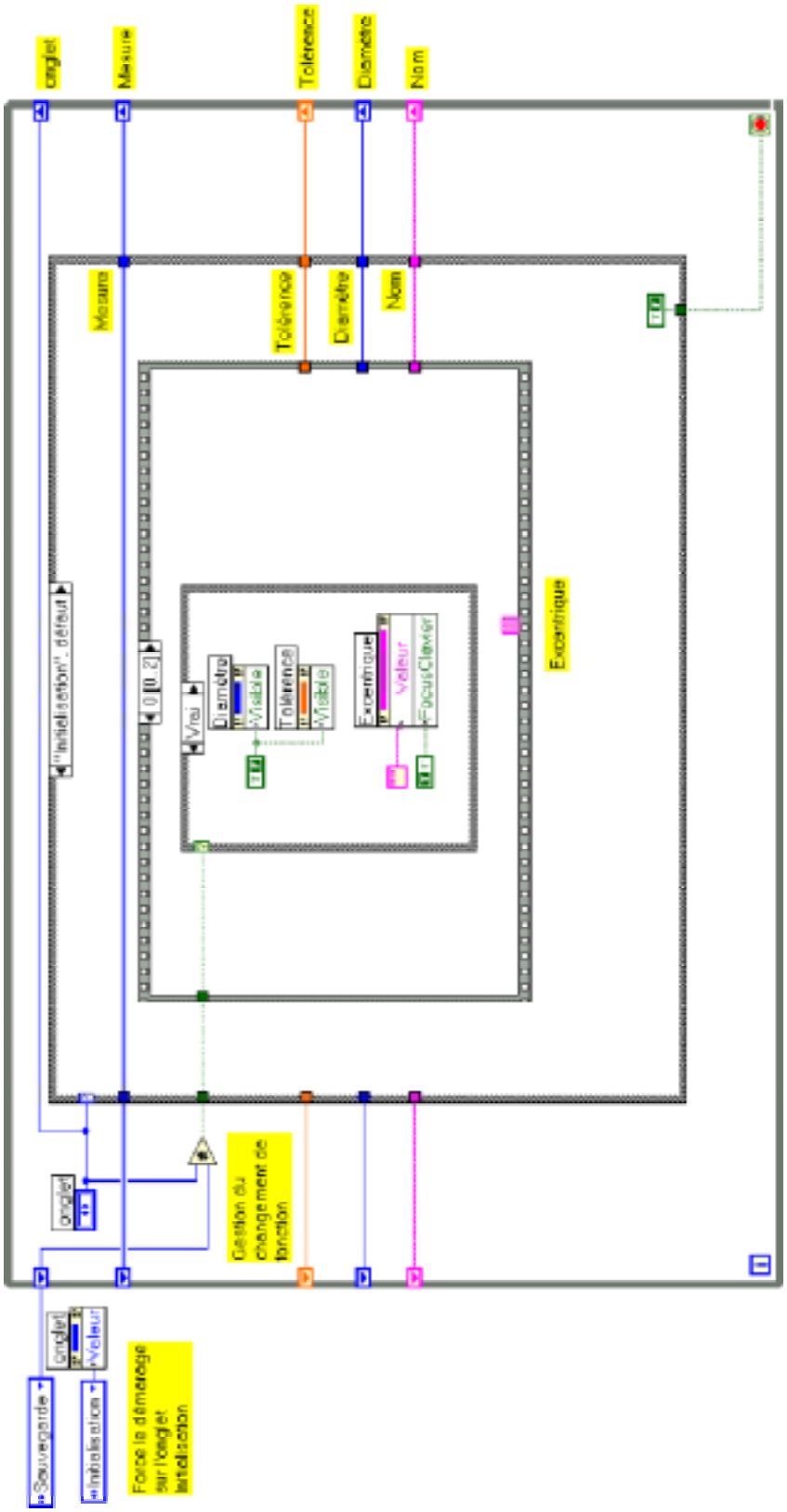

Initialisation
Mesure
Visualisation
Sauvegarde
Excentrique
Graphe XY
75
0
-75
Pièce
PièceDémo -90
Mauvais

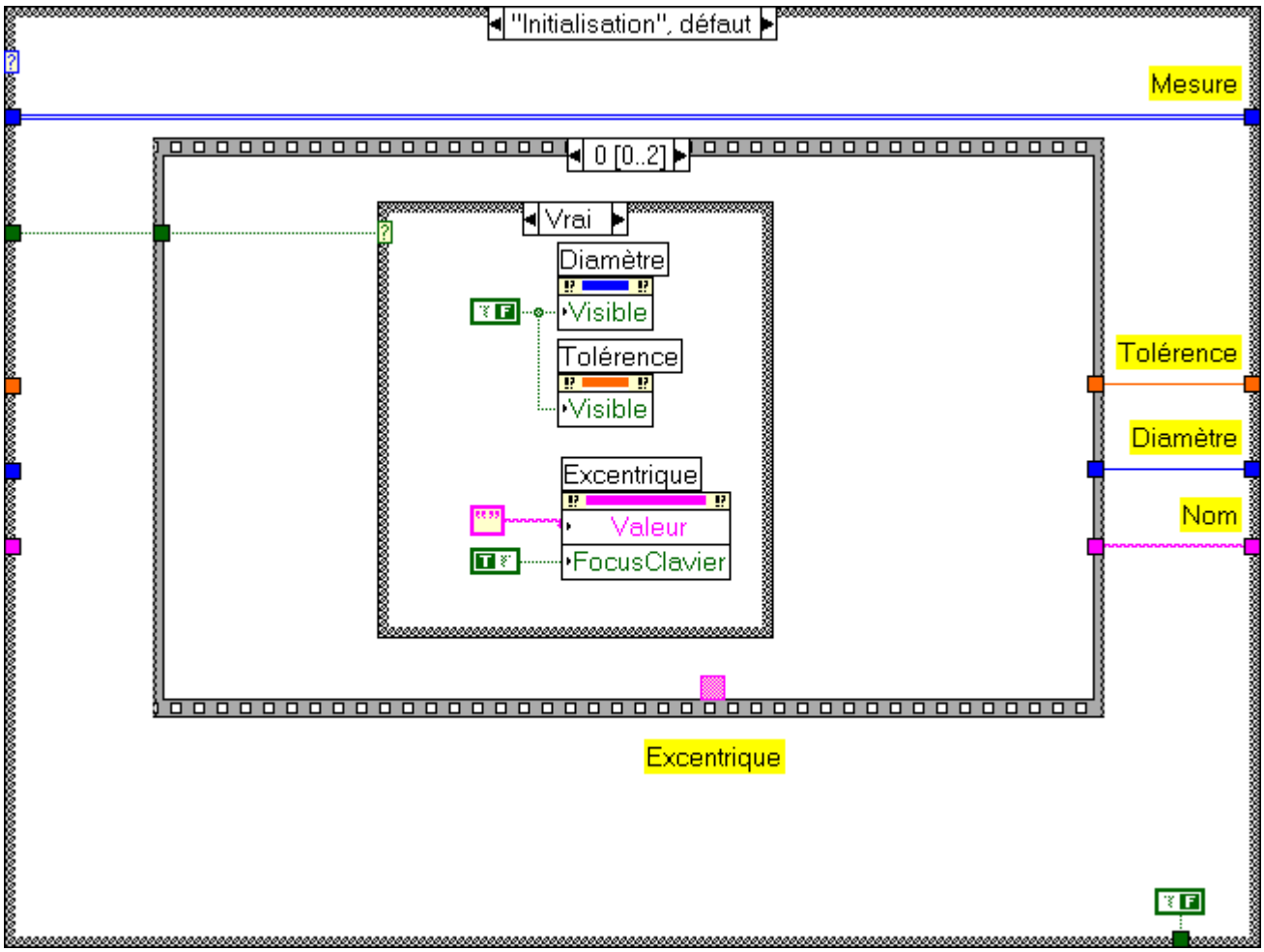
TD8_3.vi

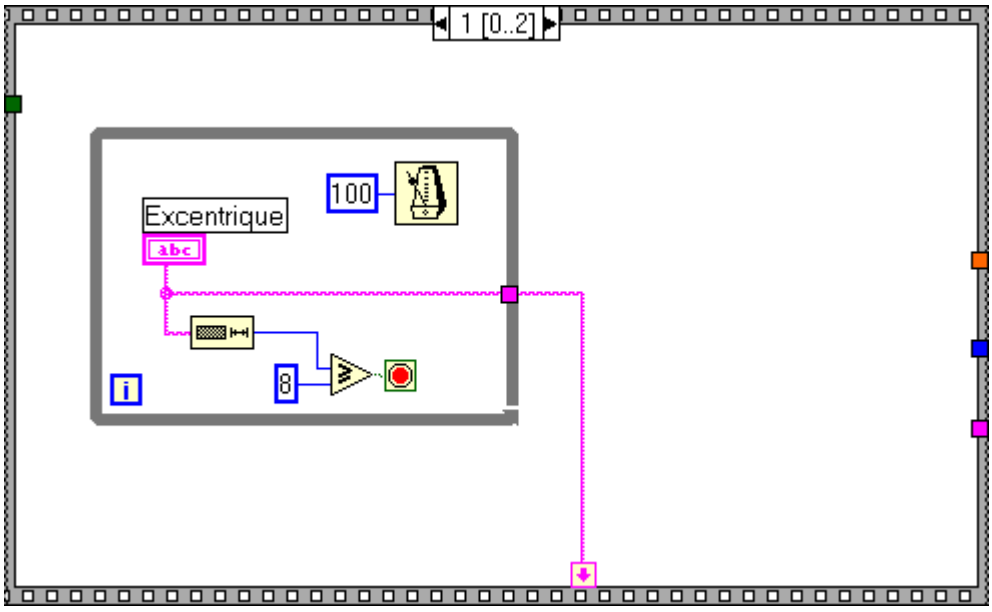
Diagramme – Structure générale

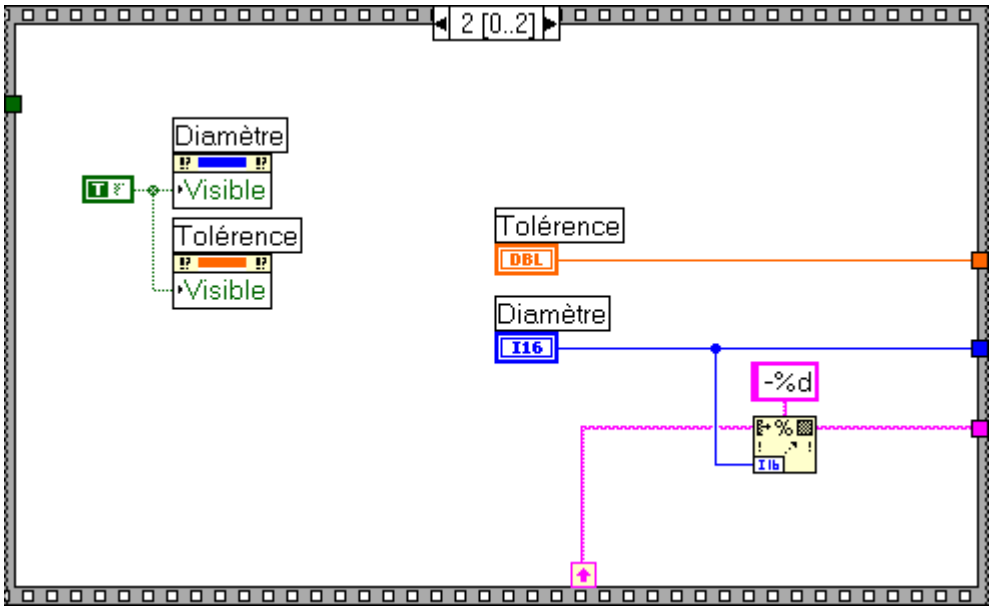
La variable onglet permet par le biais d'une structure condition d'associer à chaque partie de l' IHM un code spécifique .

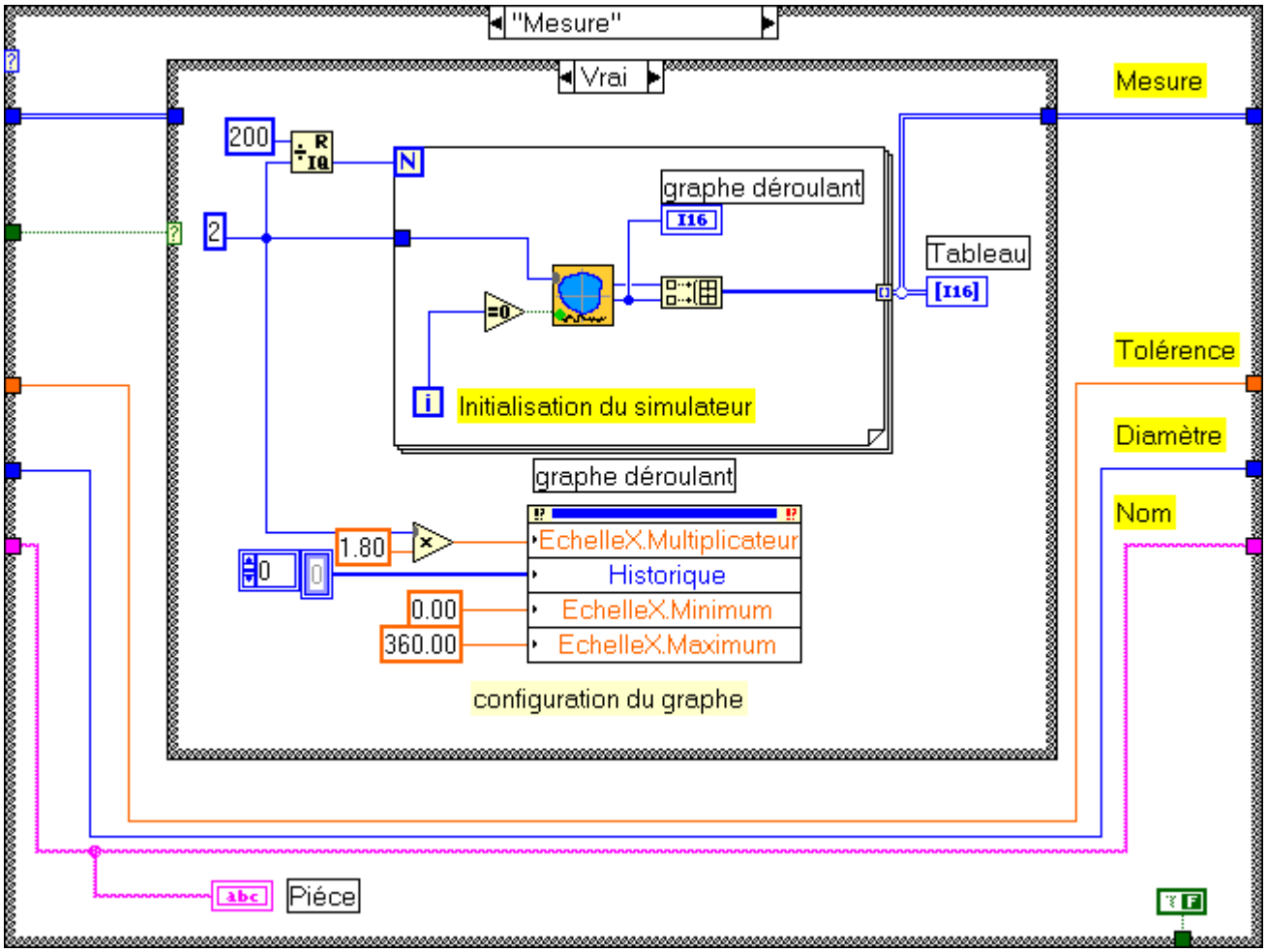
La boucle While, qui se termine après la sauvegarde des données, permet de mémoriser et de renvoyer la valeur des paramètres par le biais de registres à décalage, assurant ainsi le passage de valeur d'une partie de code à l'autre.

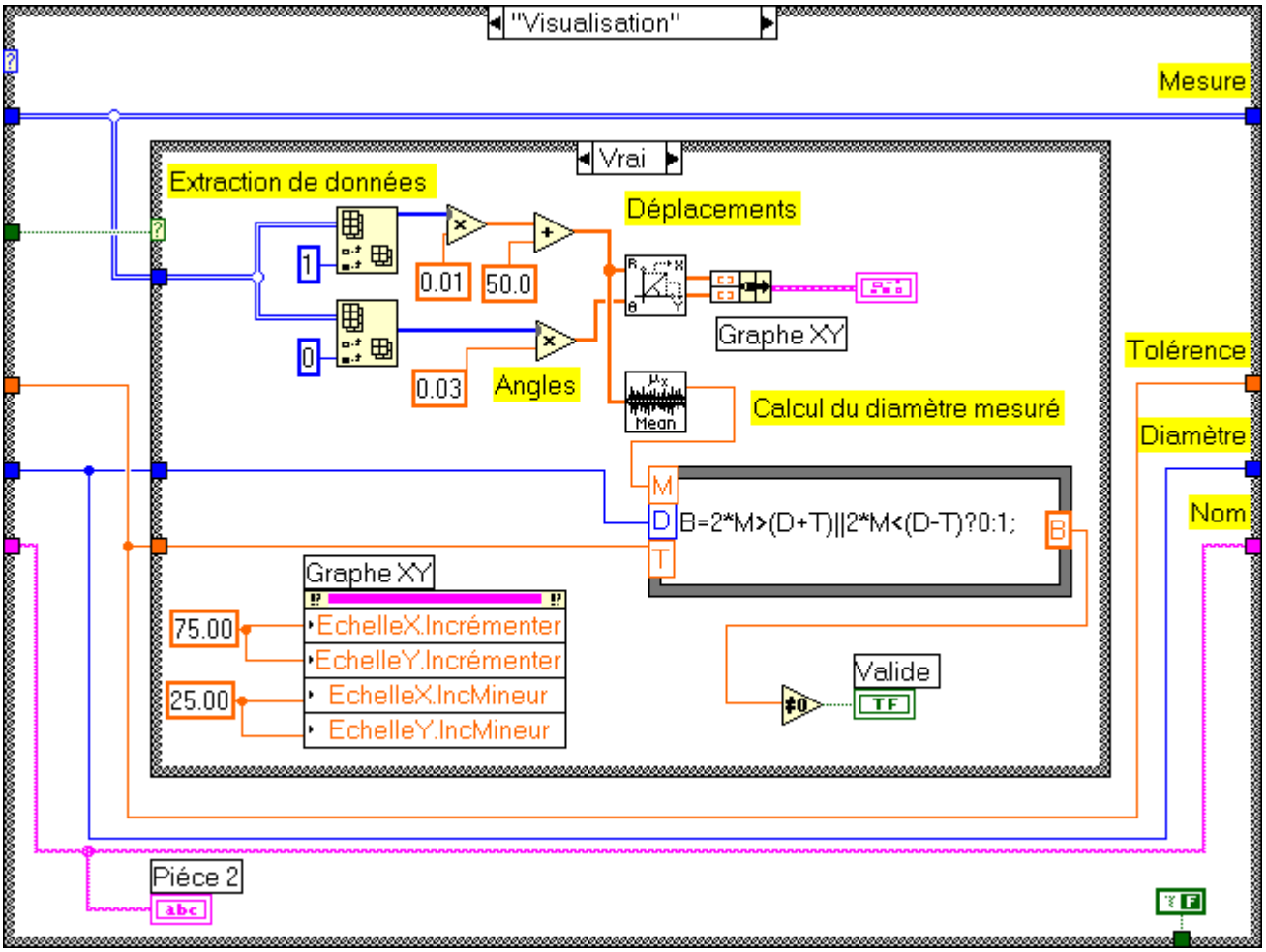

onglet
Mesure
Tolerance
Diamètre
Nom
Mesure
Tolérance
Diamètre
Nom
"Initialisation", défaut
0
[0..2]
Vrai
Diamètre
Visible
Tolérance
Visible
Excentrique
Valeur
FocusClavier
Excentrique
onglet
Gestion du changement de fonction
onglet
Valeur
Sauvegarde
Initialisation
Force le démarrage sur l'onglet Initialisation

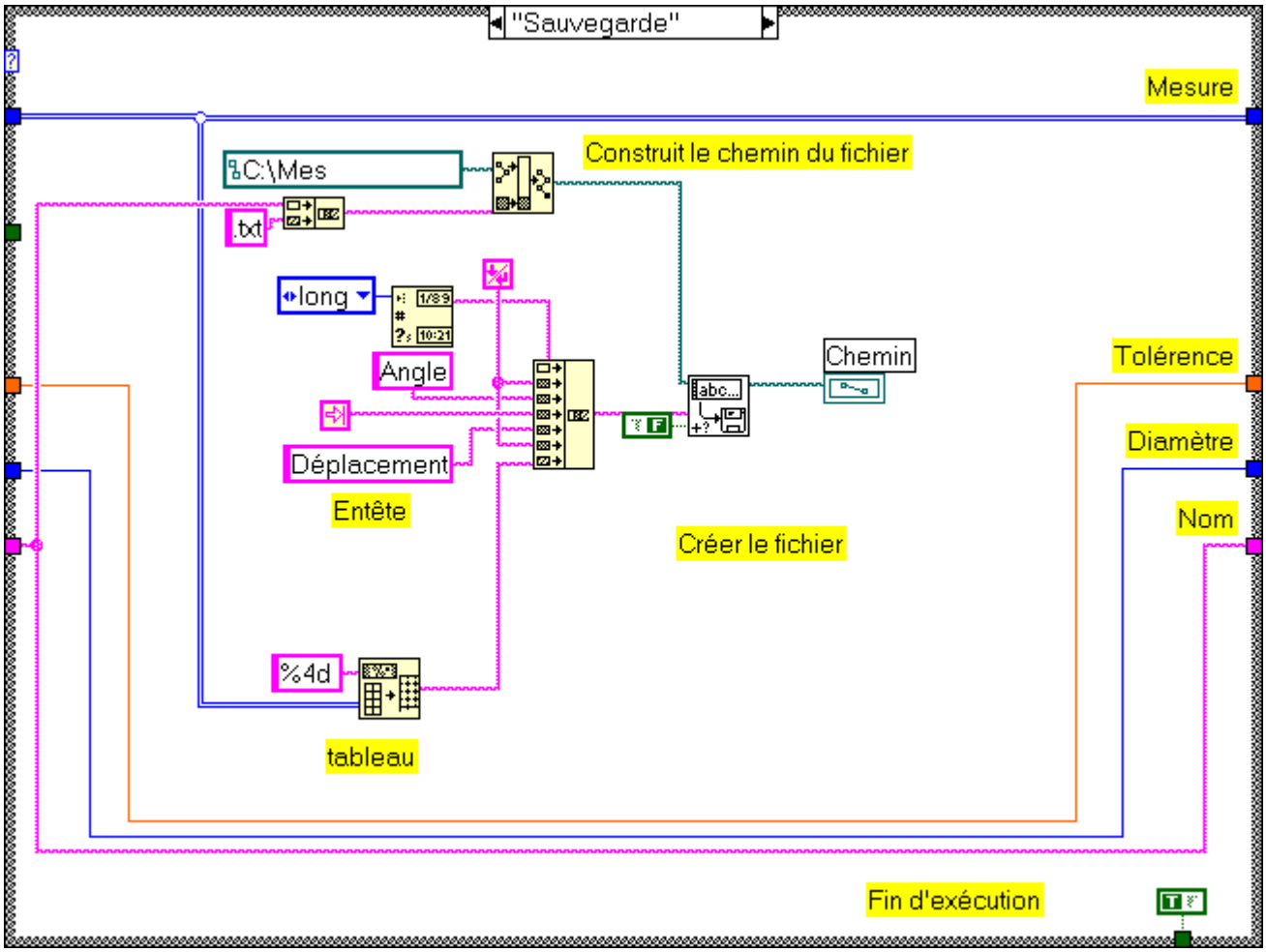

"Initialisation", défaut
Mesure
0 [0..2]
Vrai
Diamètre
Visible
Tolérence
Visible
Excentrique
Valeur
FocusClavier
Tolérence
Diamètre
Nom
Excentrique


1 [0..2]
Excentrique
abc
100
8
i


2 [0..2]
Diamètre
Visible
Tolérence
Visible
Tolérence
DBL
Diamètre
I16
-%d


"Mesure"
Vrai
Mesure
200
N
graphe déroulant
I16
2
Tableau
[I16]
=0
Initialisation du simulateur
Tolérence
Diamètre
Nom
graphe déroulant
1.80
0
EchelleX.Multiplicateur
Historique
0.00
EchelleX.Minimum
360.00
EchelleX.Maximum
configuration du graphe
abc
Piéce


"Visualisation"
Mesure
Vrai
Extraction de données
Déplacements
1
0
0.01
50.0
0.03
Angles
Graphe XY
Mean
Calcul du diamètre mesuré
Tolérence
Diamètre
Nom
M
D
T
B=2*M>(D+T)||2*M<(D-T)?0:1;
B
Graphe XY
75.00
25.00
EchelleX.Incrémenter
EchelleY.Incrémenter
EchelleX.IncMineur
EchelleY.IncMineur
Valide
TF
Piéce 2
abc


"Sauvegarde"
Mesure
C:\Mes
Construit le chemin du fichier
.txt
long
Angle
Déplacement
Entête
Chemin
Tolérence
Diamètre
Nom
Créer le fichier
%4d
tableau
Fin d'exécution