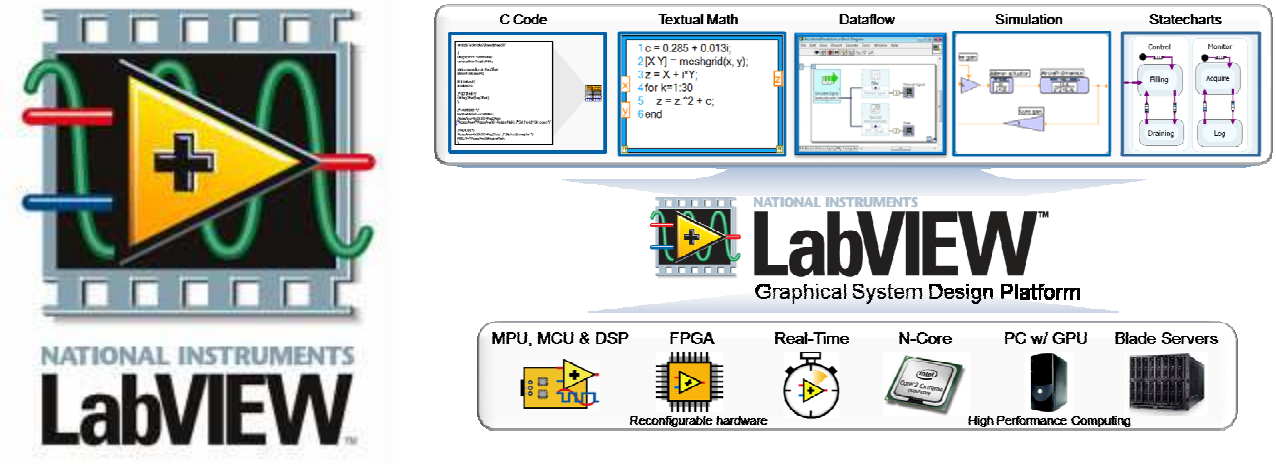
PROGRAMMATION EN « G »



Objectifs de l'activité pratique :

Comprendre les avantages de la programmation graphique
Présentation du logiciel LabVIEW
Créer un VI

Support d'activité :

Logiciel : LabVIEW
Boitier NI-USB 6009 et câble USB
Une diode DEL rouge standard, une résistance de 330 Ω, une résistance de 10 kΩ et une thermistance (CTN)
Micro-cordons et platine d'essai
Ce document aux formats papier et PDF en couleur

OBSERVATIONS

NOTE : / 20

DOCUMENTS RÉPONSES

NOMS : _____/_____/_____/_____/_____

GROUPE : _____ **DATE :** _____

AVANTAGES DE LA PROGRAMMATION GRAPHIQUE

Depuis plus de 20 ans, des millions d'ingénieurs et de scientifiques utilisent le logiciel NI LabVIEW pour développer des applications sophistiquées de test, de mesure et de contrôle. LabVIEW fournit une gamme de fonctionnalités et d'outils, allant des assistants interactifs jusqu'aux interfaces configurables par l'utilisateur, mais il se distingue des autres par son langage de programmation graphique standard (appelé **G**), le compilateur intégré qui lui est associé, son éditeur de liens et ses outils de mise au point.

Pour mieux comprendre les avantages de la programmation graphique de LabVIEW, voici un rappel de quelques faits marquants de l'histoire du premier langage de programmation de haut niveau. Au début de l'ère de l'ordinateur moderne dans les années 50, une petite équipe d'ingénieurs chez IBM décida de créer un nouveau langage, qui serait plus pratique pour programmer l'énorme ordinateur IBM 704 (le super-ordinateur de l'époque), qui remplacerait l'assembleur de bas niveau et deviendrait le langage le plus moderne de son temps. C'est ainsi que le FORTRAN (FORmula TRANslator est un langage de programmation utilisé principalement en calcul scientifique) vit le jour. C'était un langage de programmation plus facilement lisible dont le but principal était d'accélérer le processus de développement.

Dans les premiers temps, les ingénieurs étaient sceptiques et ne croyaient pas que cette nouvelle méthode pourrait faire mieux que les programmes en assembleur faits à la main. Toutefois, les programmes créés en FORTRAN ont rapidement prouvé qu'ils s'exécutaient presque aussi efficacement que ceux écrits en assembleur. Dans la foulée, le FORTRAN réduisait le nombre d'instructions de programmation nécessaires par un facteur de 20, et c'est une des raisons pour laquelle il est souvent considéré comme le premier langage de programmation de haut niveau. Logiquement, le FORTRAN fut vite accepté par la communauté scientifique et exerce toujours une certaine influence.

Cinquante ans plus tard, il existe toujours des leçons à tirer de cette anecdote. Tout d'abord, les ingénieurs n'ont pas cessé de rechercher des façons plus simples et plus rapides de résoudre leurs problèmes par le biais de la programmation. Ensuite, les langages de programmation choisis par les ingénieurs pour traduire leurs tâches ont eu tendance à évoluer vers des niveaux plus abstraits. Ces leçons expliquent en grande partie l'immense popularité et l'adoption généralisée du G depuis son introduction en 1986. Le G représente un langage de programmation de niveau extrêmement élevé qui a pour objectif d'augmenter la productivité de ses utilisateurs tout en exécutant les programmes à quasiment la même vitesse que les langages de niveaux inférieurs tels que le FORTRAN et le C++.

PRÉSENTATION DE LABVIEW

LabVIEW se différencie de la plupart des langages de programmation courants par deux caractéristiques essentielles. Premièrement, la programmation en G s'effectue en câblant des icônes graphiques sur un diagramme, lui-même ensuite compilé directement en code machine pour permettre aux processeurs de l'ordinateur de l'exécuter. Bien qu'il soit représenté graphiquement et non textuellement, le G obéit aux mêmes concepts de programmation que la plupart des langages traditionnels. Par exemple, le G inclut tous les éléments de construction standards, tels que types de données, boucles, gestion d'événements, variables, récursivité et programmation orientée objet.

La seconde différence réside dans le fait que le code G développé avec LabVIEW s'exécute en fonction des règles de flux de données au lieu d'une approche basée sur des procédures plus classiques (en d'autres termes, une série de commandes séquentielles à exécuter), comme c'est souvent le cas avec la plupart des langages de programmation textuels tels que le C et le C++. Un langage de programmation par flux de données comme le G (ou Agilent VEE, Microsoft Visual et Apple Quartz Composer) met les données en valeur en tant que concept clé derrière chaque programme. Dans ce modèle de programmation, ce sont les données qui déterminent l'ordre d'exécution du flux de données. C'est le flux de données entre les nœuds dans le programme, et non pas les lignes séquentielles de texte, qui détermine l'ordre d'exécution.

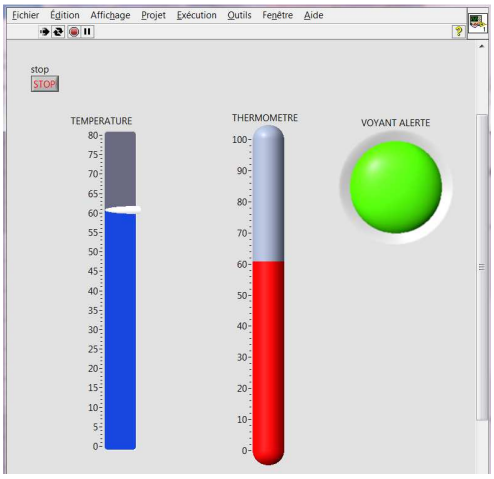
Cette distinction peut paraître insignifiante au premier abord, mais son impact est primordial car elle attire l'attention du développeur sur les chemins de données entre les différentes parties du programme. Dans un programme LabVIEW, les nœuds (c'est-à-dire les fonctions, structures comme les boucles, sous-programmes, etc.) possèdent des entrées, traitent des données et produisent des signaux en sortie. Dès que les entrées d'un nœud donné reçoivent des données valides, ce nœud s'exécute selon sa logique, produit des données en sortie et transfère les données au nœud suivant dans le chemin du flux de données. Un nœud qui reçoit des données d'un autre nœud ne s'exécute que lorsque le premier nœud a terminé son exécution.

Le code en G permet souvent aux ingénieurs et scientifiques de comprendre aisément et rapidement car ils ont souvent l'habitude de visualiser, et même de produire, des modèles de procédures et de tâches sous forme de diagrammes et d'organigrammes (qui obéissent aux règles des flux de données). De plus, comme les langages de programmation par flux de données exigent de baser la structure de votre programme sur le flux de données, vous êtes amené(e) à penser en fonction du problème à résoudre. Par exemple, un programme typique en G est susceptible d'acquérir plusieurs voies contenant des données de température, puis de transférer ces données vers une fonction d'analyse et enfin, d'écrire les données analysées sur disque. Dans l'ensemble, les flux de données et les étapes de ce programme sont faciles à comprendre à l'aide d'un diagramme LabVIEW.

Tous les programmes de LabVIEW sont appelés **V**irtual **I**nstrument (**VI**), ils se composent d'une face avant et d'un diagramme.

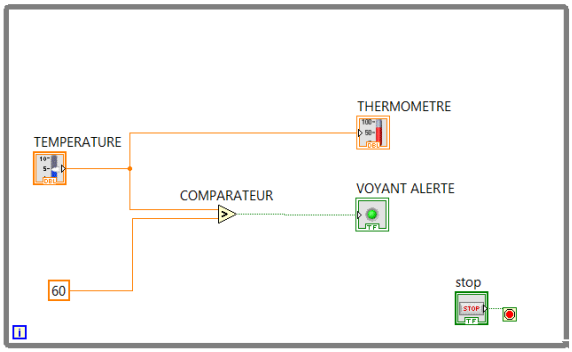
LA FACE AVANT

La face avant est une interface graphique qui réceptionne les données entrées par l'utilisateur et affiche celles fournies en sortie par le programme. Cette face avant contient des commandes ou des indicateurs.



LE DIAGRAMME OU FENÊTRE DE PROGRAMMATION

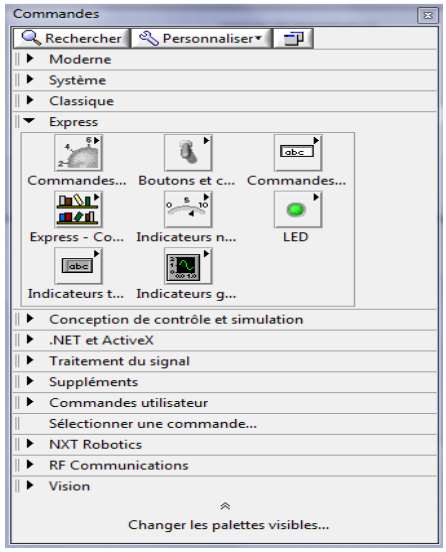
Le diagramme contient le code source graphique du VI. On programme le VI pour contrôler et remplir les fonctions sur les entrées et sorties créées sur la face avant. Le diagramme peut contenir des fonctions et des structures issues des bibliothèques de LabVIEW. Il peut contenir des terminaux associés à des commandes et à des indicateurs de la face avant.
Les deux fenêtres sont interactives.



LES PALETTES

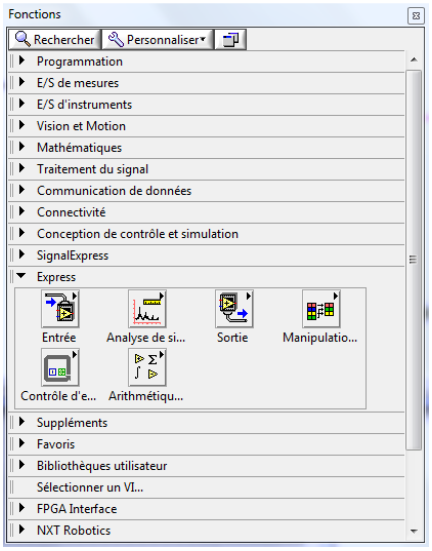
La palette des commandes :

La palette des commandes ne s'utilise que dans la face avant. Elle contient les commandes et les indicateurs qui servent à créer l'interface utilisateur.



La palette outils :

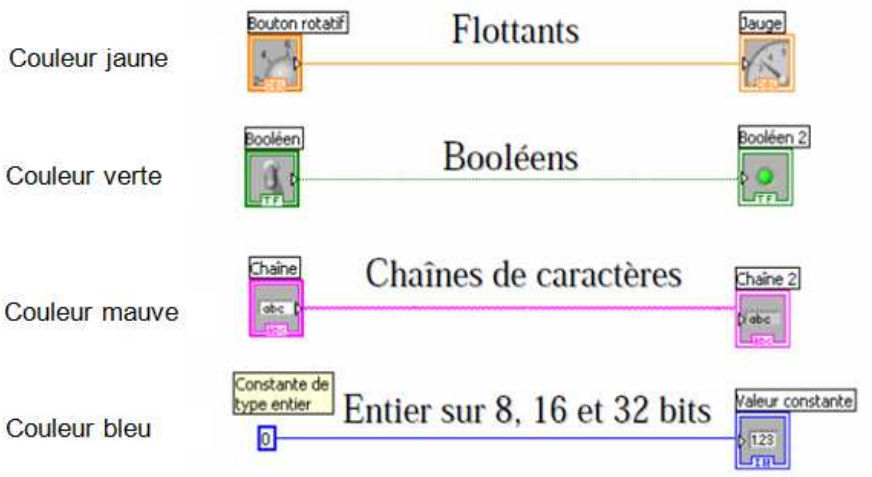
Elle s'utilise aussi bien dans la face avant que dans le diagramme. Elle contient les outils nécessaires pour éditer et mettre au point les objets du diagramme et de la face avant.



La palette des fonctions :

La palette des fonctions ne s'utilise que dans le diagramme. Elle contient les objets qui servent à programmer le VI (opération mathématique, gérer les entrées/sorties d'une carte…)

LES COULEURS



LA BARRE D'OUTILS

LES TOUCHES DE RACCOURCIS

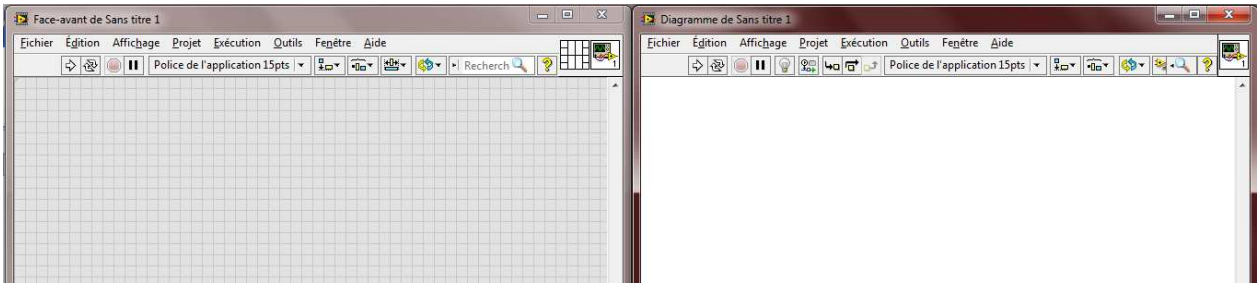
Ctrl T	Juxtapose la fenêtre de face-avant et celle du diagramme
Ctrl B	Supprime tous les fils de liaison brisés d'un VI
Ctrl E	Affiche la fenêtre de la face-avant ou du diagramme
Ctrl U	Redirige tous les fils de liaison existants et réorganise automatiquement tous les objets sur le diagramme

CRÉATION D'UN VI

INTRODUCTION

On veut réaliser un VI qui affichera une détection d'incendie. L'élévation de température sera simulée par une glissière à curseur verticale. La température sera affichée par un thermomètre. Un voyant s'allumera dès que la température dépassera 60℃.

Ouvrir LabVIEW (sur le bureau, dossier : Programmation et réseaux) > VI vide.
Ouvrir la face avant (fenêtre grise) et le diagramme (fenêtre blanche).
Organiser les fenêtres comme ci-dessous.



Enregistrer le projet dans votre dossier personnel sous le nom : ***Programmation_en_G_votre NOM\VI1***.

Remarques :

Programmation_en_G_votre NOM est un dossier à créer.
Penser à sauvegarder régulièrement votre travail.

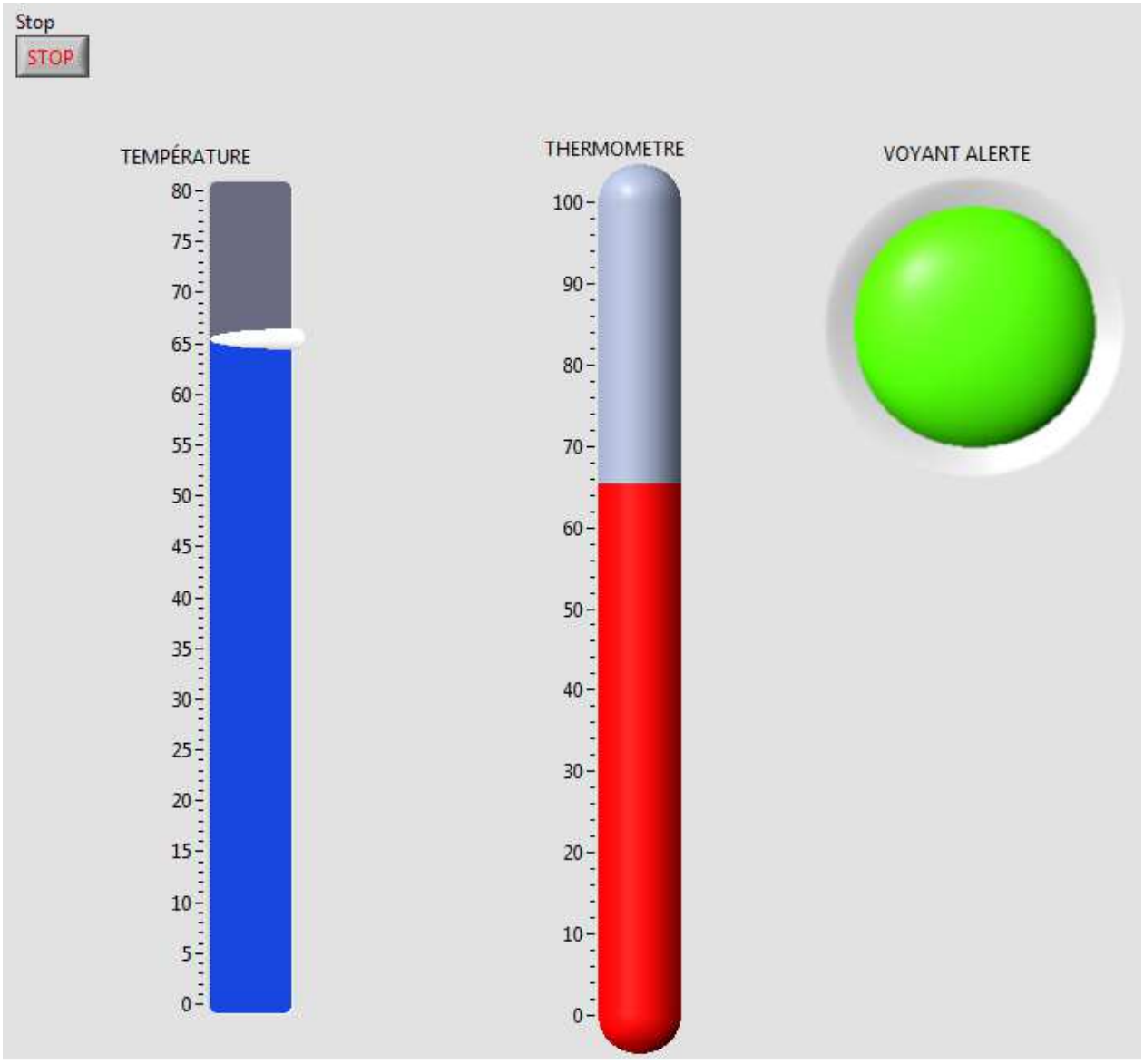
LA FACE AVANT

Voir la face avant page suivante.

Ouvrir la palette des commandes : **Affichage > palette des commandes**.

Cliquer sur : **Express > Commandes numériques** pour placer la glissière à curseur verticale.
Cliquer sur : **Express > Indicateurs numériques** pour placer le thermomètre.
Cliquer sur : **Express > Boutons et commutateurs** pour placer le bouton stop.
Cliquer sur : **Classique > Booléen classique** pour placer le voyant d'alerte.

La face avant :



LE DIAGRAMME

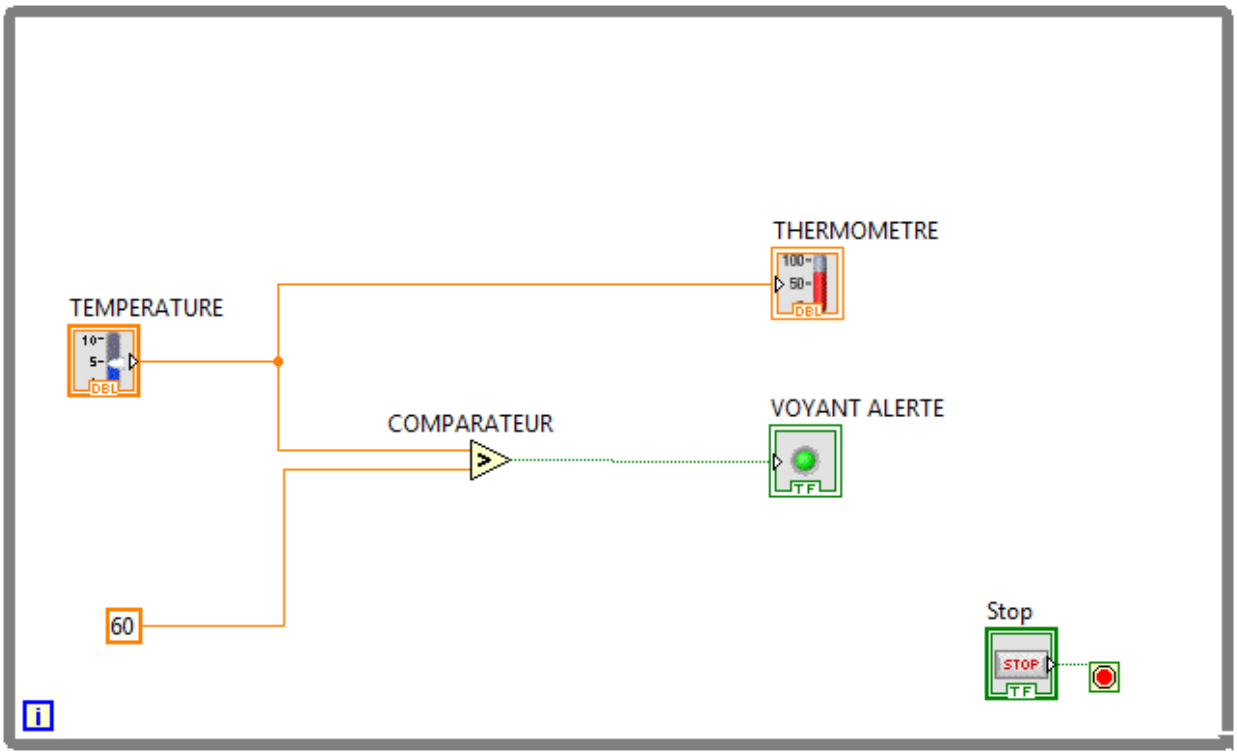
Voir le diagramme page suivante.

Ouvrir la palette des commandes : **Affichage > palette des fonctions**.

Cliquer sur : **Express > Arithmétique et comparaison > Comparaison > placer le comparateur Supérieur ?**
Cliquer sur **: Express > Contrôle d'exécution > Boucle while (à élargir pour englober tous les icônes à l'écran)**.
Cliquer sur **: Mathématiques > numérique > cnste DBL**.

Relier les différents blocs par des fils à l'aide de l'outil bobine : Affichage > Palette d'outils > Connecter les terminaux.

Le diagramme :



Cliquer sur **: Exécuter**.

Tester votre programme.

APPELER LE PROFESSEUR POUR VALIDER LE FONCTIONNEMENT

Question 1 :

À votre avis, quel est l'intérêt d'avoir une boucle while dans le programme ? Que se passerait-il si l'on supprimé la boucle while dans le diagramme ?

……

……

……

……

……

……

……

Question 2 :

*On souhaite ajouter dans le programme précédent une deuxième détection (« **GEL** ») si la température descend en dessous de 0°C.*
Modifier et sauvegarder le programme précédent dans votre dossier personnel sous le nom : ***Programmation_en_G_votre NOM\VI2***.

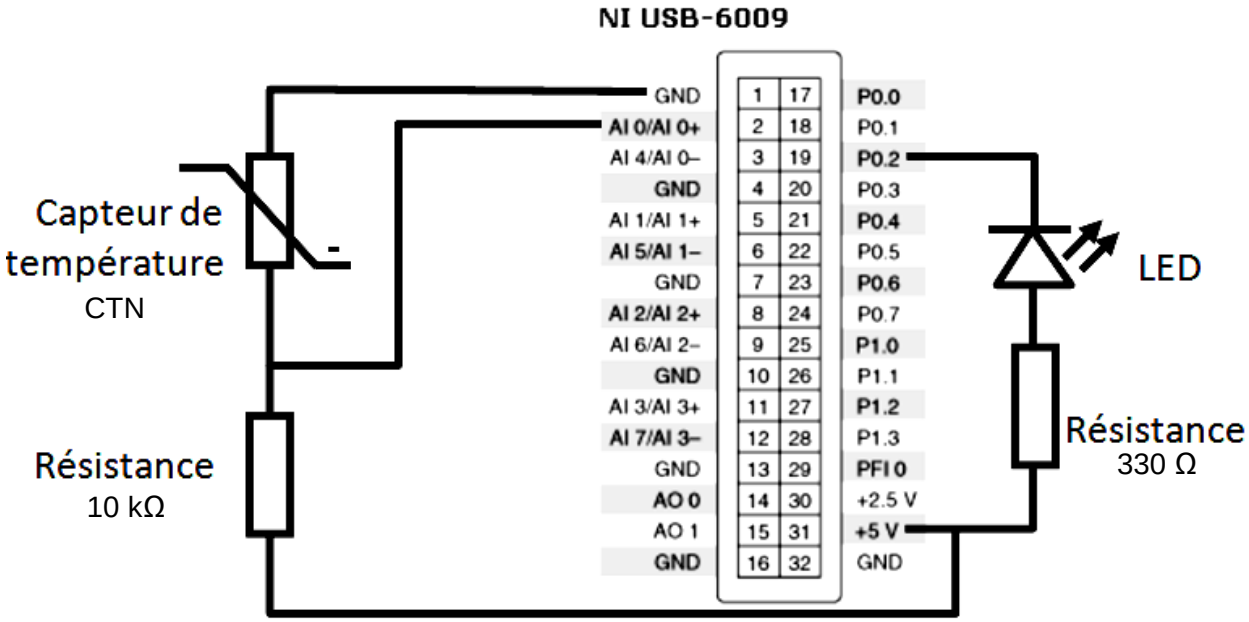
Exécuter et tester votre programme.

APPELER LE PROFESSEUR POUR VALIDER LE FONCTIONNEMENT

RÉALISATION D'UNE ALARME

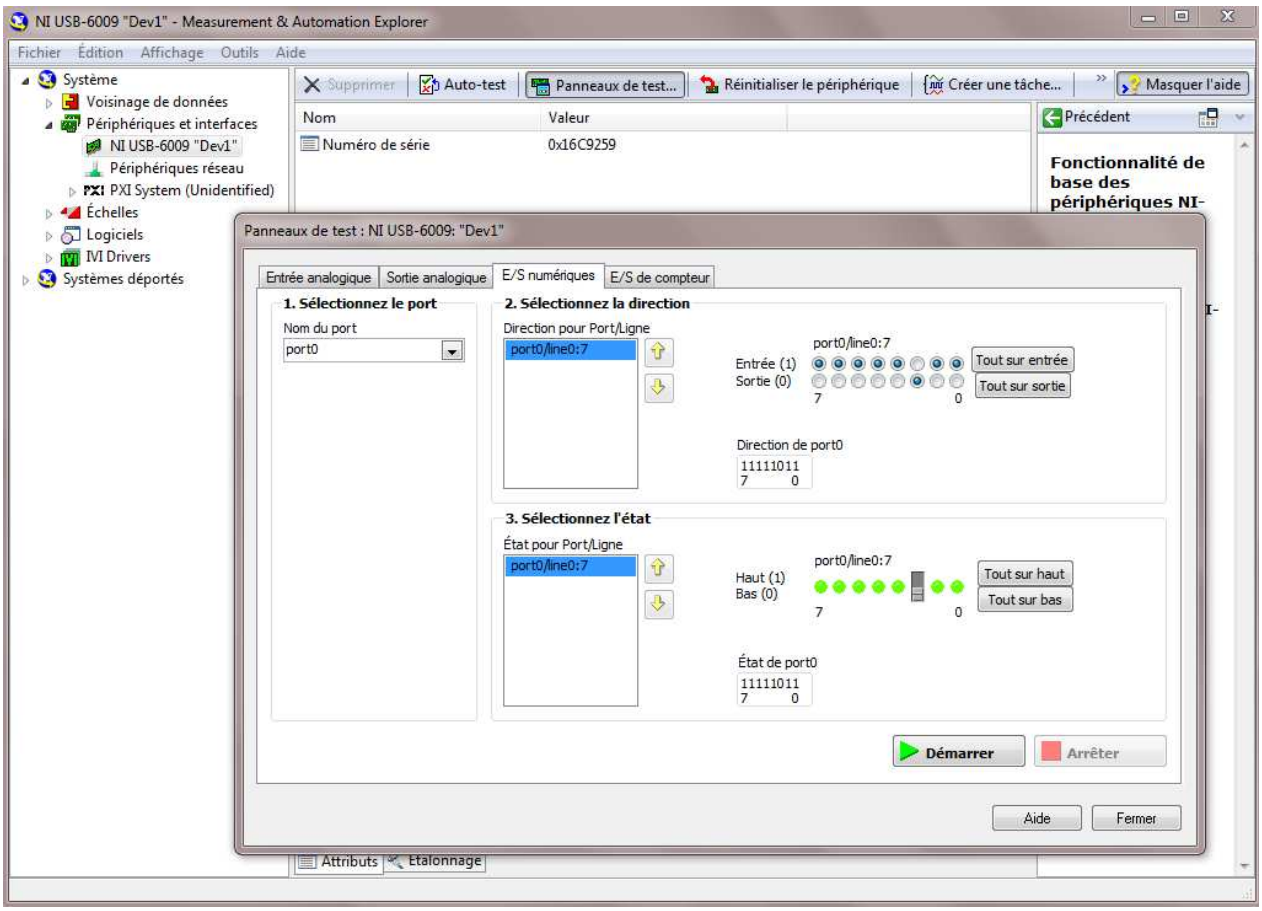
Question 3 :

Effectuer les câblages et connexions du boitier NI-USB 6009 comme ci-dessous.



APPELER LE PROFESSEUR POUR VALIDER

*Connecter le boitier NI-USB 6009 au PC puis lancer MAX (**Outil>Measurement & Automation explorer**).*



Cliquer sur Périphériques et interfaces (après avoir laissé à MAX le temps de les chercher).

Cliquer sur le NI USB-6009 qui doit être activé (vous visualiserez aussi toutes les interfaces précédemment connectées au PC même si elles sont désactivées).

L'onglet Brochage du périphérique permet de visualiser son brochage.

L'onglet Panneaux de test permet de tester le fonctionnement du montage.

Paramétrer comme indiqué sur la capture d'écran ci-dessus (P02 en sortie et à l'état bas), cliquer sur démarrer et vérifier que la DEL soit éteinte.

Vérifier que la DEL s'allume en configurant à l'état haut.

APPELER LE PROFESSEUR POUR VALIDER

Enregistrer le projet dans votre dossier personnel sous le nom : ***Programmation_en_G_votre NOM\VI3***.

Créer la face avant et le diagramme en vous aidant des informations ci-après……

Face avant :

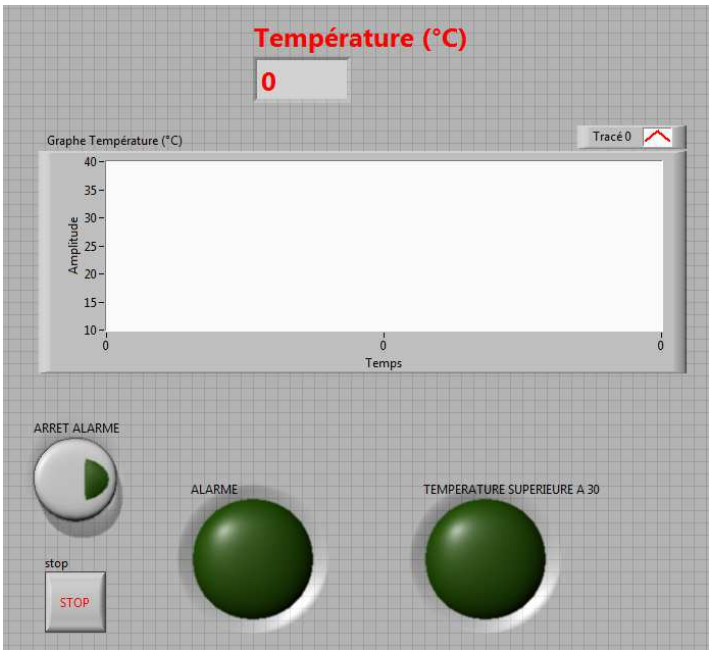
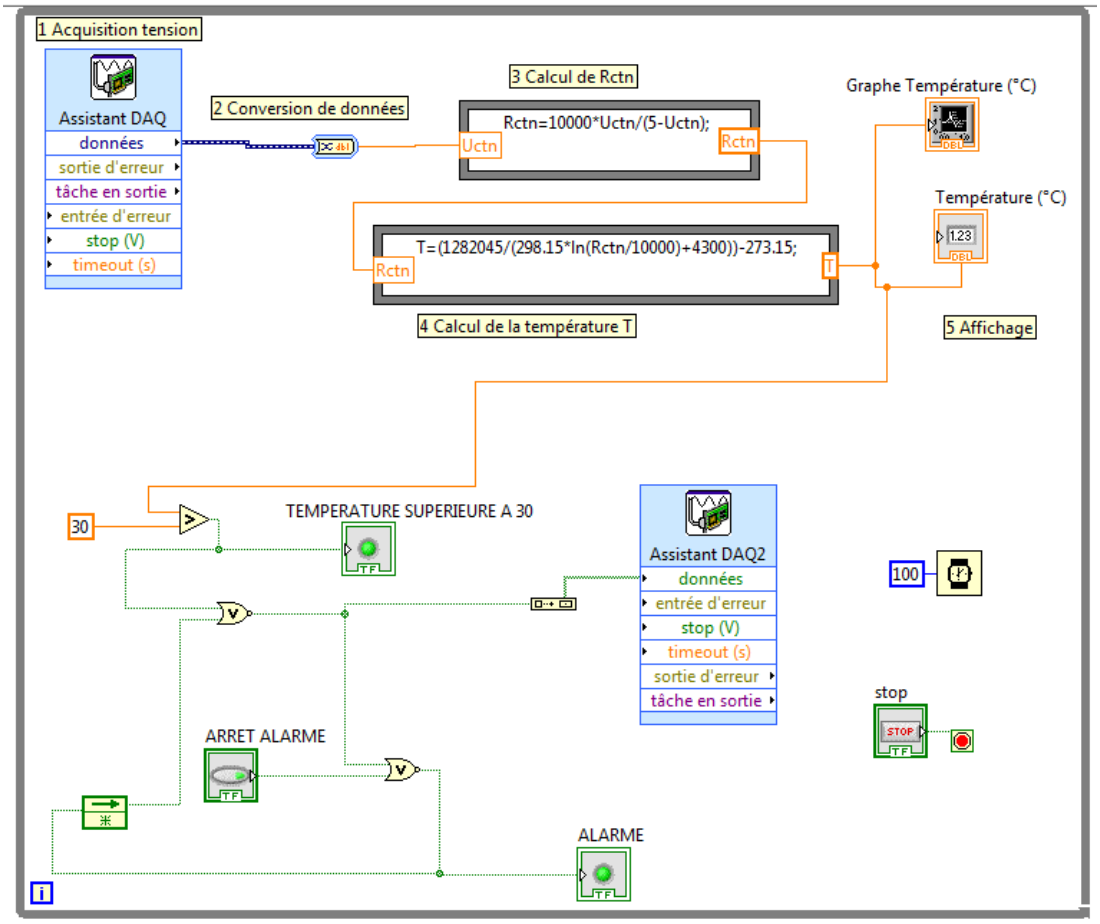


Diagramme :



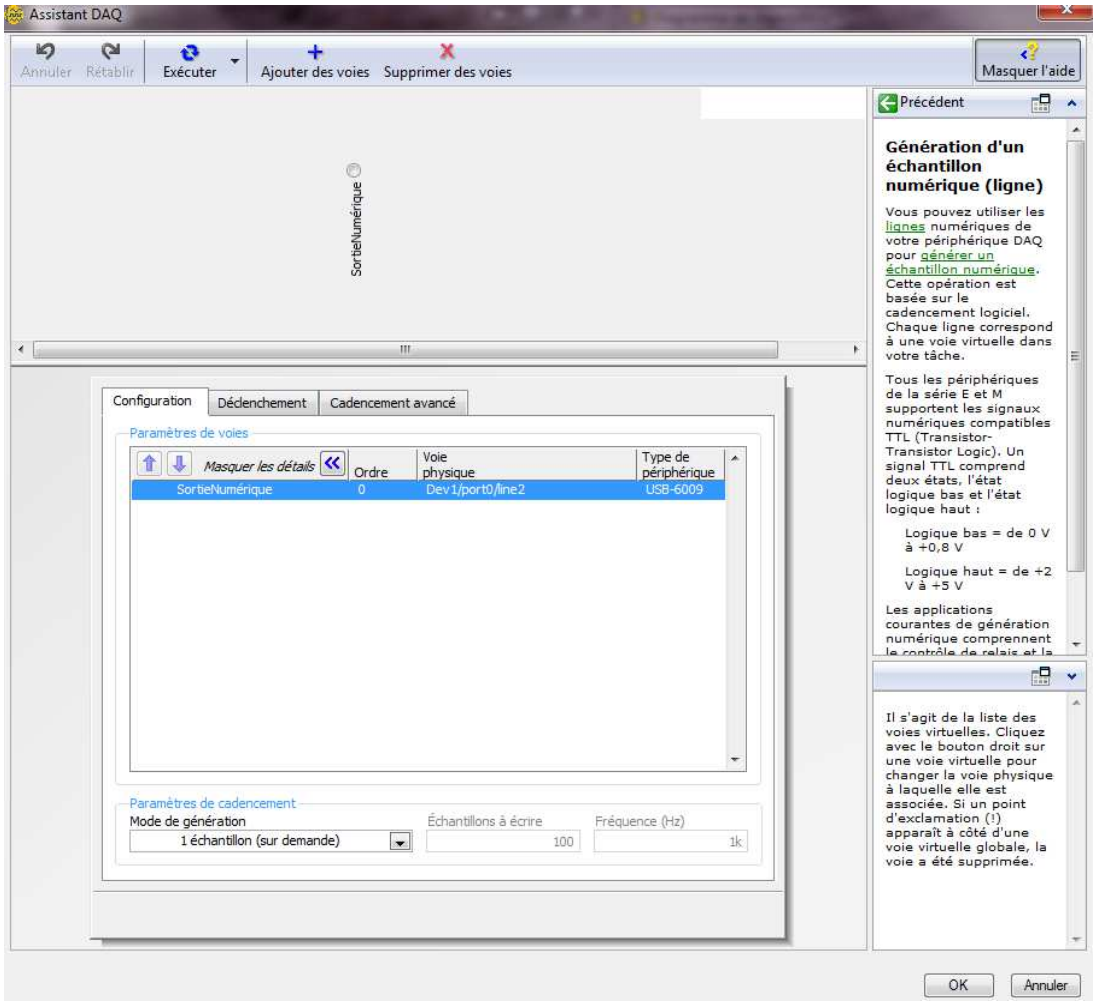
Se placer sur l'entrée de la condition de boucle puis clic droit > **Créer commande** permet de créer le bouton STOP.

Acquisition de l'information (tension) : Assistant DAQ

Conversion de données : Palette Fonctions > Express > Manipulations > DDT-- >
Type de données résultant : Scalaire unique

Calcul de Rctn : Palette > Programmation > Structures > Boîte de calcul

Double cliquer sur l'assistant DAQ : Acquérir des signaux > ai0 puis configurer comme suit :



Lorsque tout est connecté, exécuter et tester le VI.

APPELER LE PROFESSEUR POUR VALIDER LE FONCTIONNEMENT

Glisser une copie de votre dossier « ***Programmation_en_G_votre NOM*** »contenant les VIs dans le dossier ***Travail/Spécialité SIN***.