4. Tutorial CVI

Introduction

CVI est un langage (langage C + librairies *CVI*) dédié à l'informatique industrielle. C'en est même un standard. Il possède de nombreuses bibliothèques pour l'informatique industrielle :

. Composants Temps Réel (timers ...)
. Bibliothèque d'Interface Utilisateur (graphiques, boîtes de dialogue, boutons ...)
. Librairie mathématique (traitement du signal, algèbre matricielle ...)
. Librairie de commande de périphériques (RS232, réseaux (TCPIP), instuments GPIB ...)
. etc ...

Structure d'un programme CVI

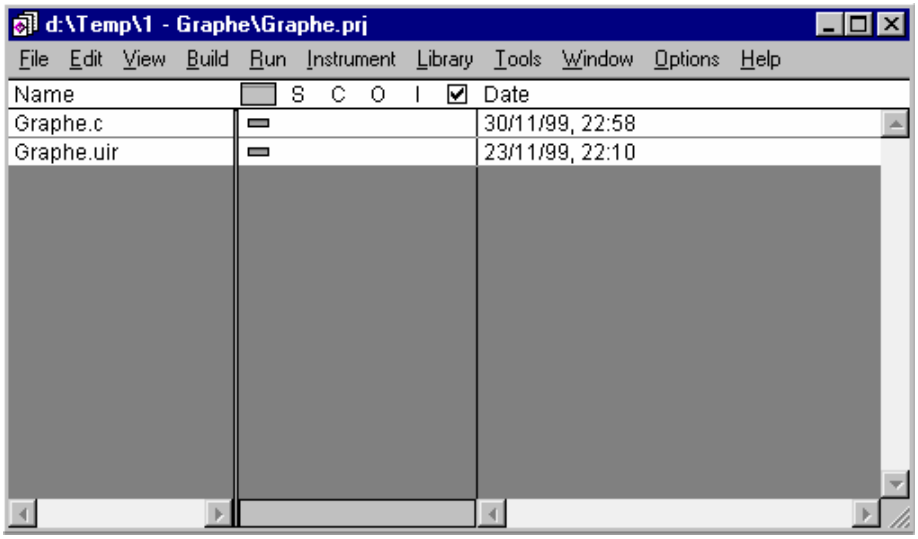
3 fichiers de base constituent un projet *CVI* dénommé pour l'exemple « *Graphe.prj* » :

1. un fichier d'interface graphique (objets : boutons, graphiques ...) *Graphe.uir* (code exécutable)

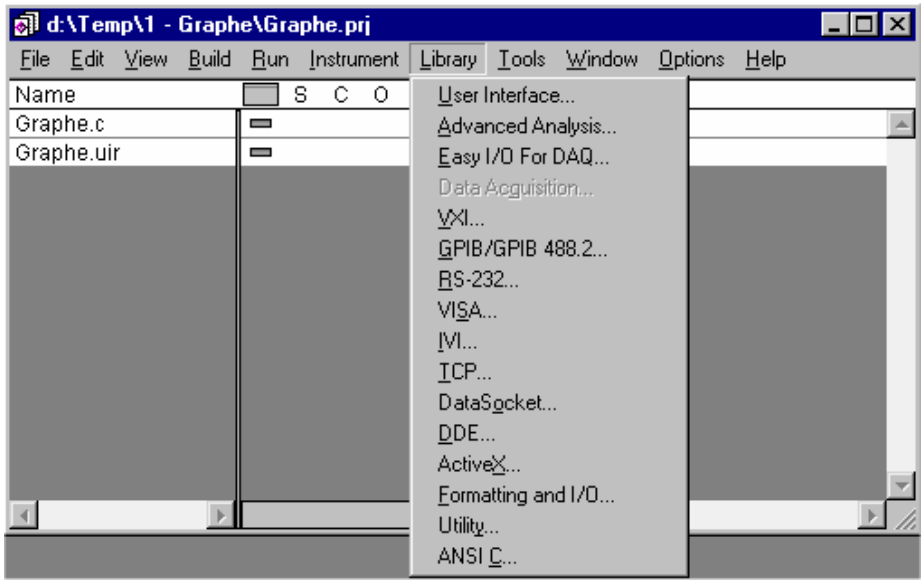
2. un fichier de prototypage (déclarations des objets, fonctions et variables de l'interface graphique) : *Graphe.h* (code source) généré par l'éditeur de l'interface graphique et à inclure dans le source *Graphe.c* gérant l'interface graphique

3. un fichier source en langage C (gestion de l'interface graphique, programme ...) *Graphe.c*

Les seuls fichiers à modifier étant Graphe.uir et Graphe.c, il est fortement recommandé de les inclure eux et eux seuls dans le projet Graphe.prj.



Des librairies de fonctions sont disponibles en bibliothèque (menu *Library*) :

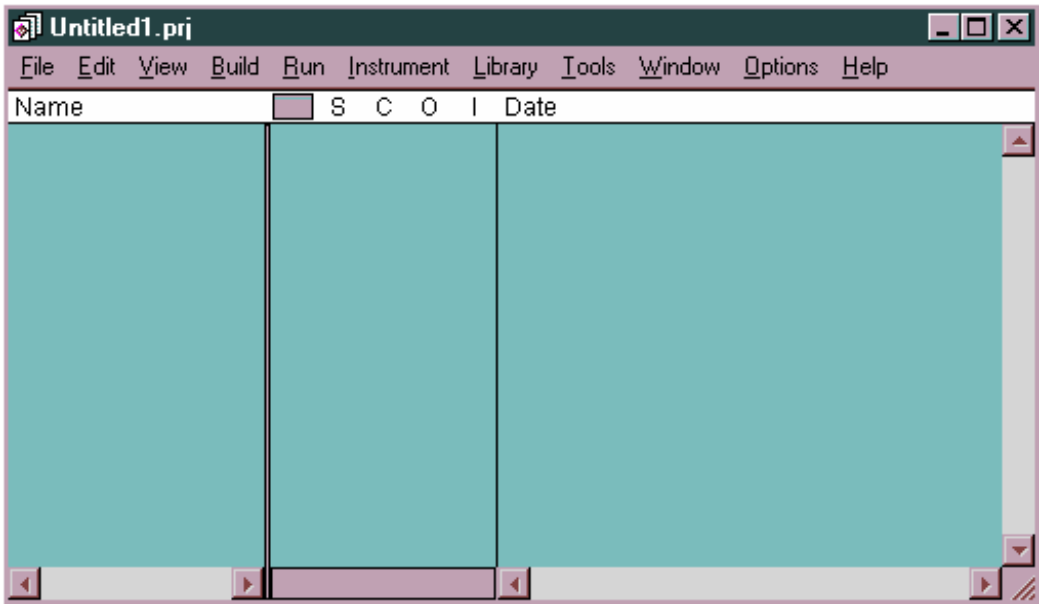

d:\Temp\1 - Graphe\Graphe.prj
File Edit View Build Run Instrument Library Tools Window Options Help
Name S C O
Graphe.c
Graphe.uir
User Interface...
Advanced Analysis...
Easy I/O For DAQ...
Data Acquisition...
VXI...
GPIB/GPIB 488.2...
RS-232...
VISA...
IVI...
TCP...
DataSocket...
DDE...
ActiveX...
Formatting and I/O...
Utility...
ANSI C...

Ecriture d'un programme CVI

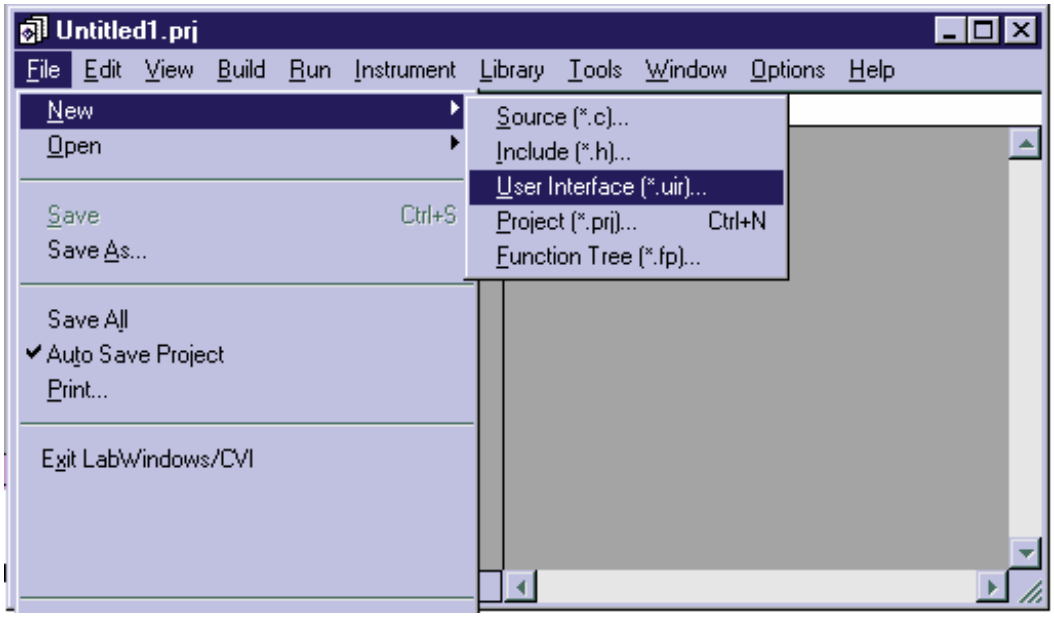
A titre d'exemple, nous allons réaliser un programme qui, après appui sur le bouton OK, trace le graphe d'un tableau (vecteur) de N points dans une fenêtre graphique. Un 2nd bouton (QUIT) autorise la sortie du programme.

PROGRAMME D'EXEMPLE :
GRAPHE D'UNE FONCTION (SINUS) DANS UNE FENETRE GRAPHIQUE

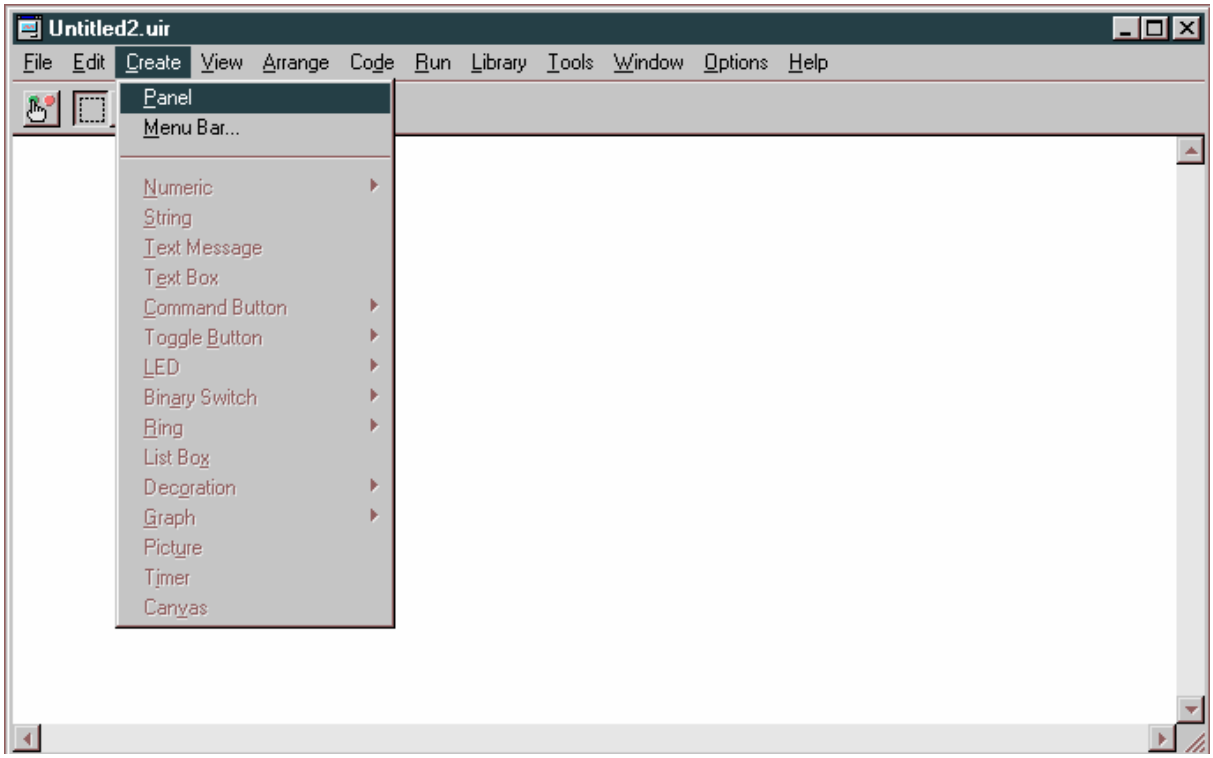
Etape 0. Lancement du programme CVI



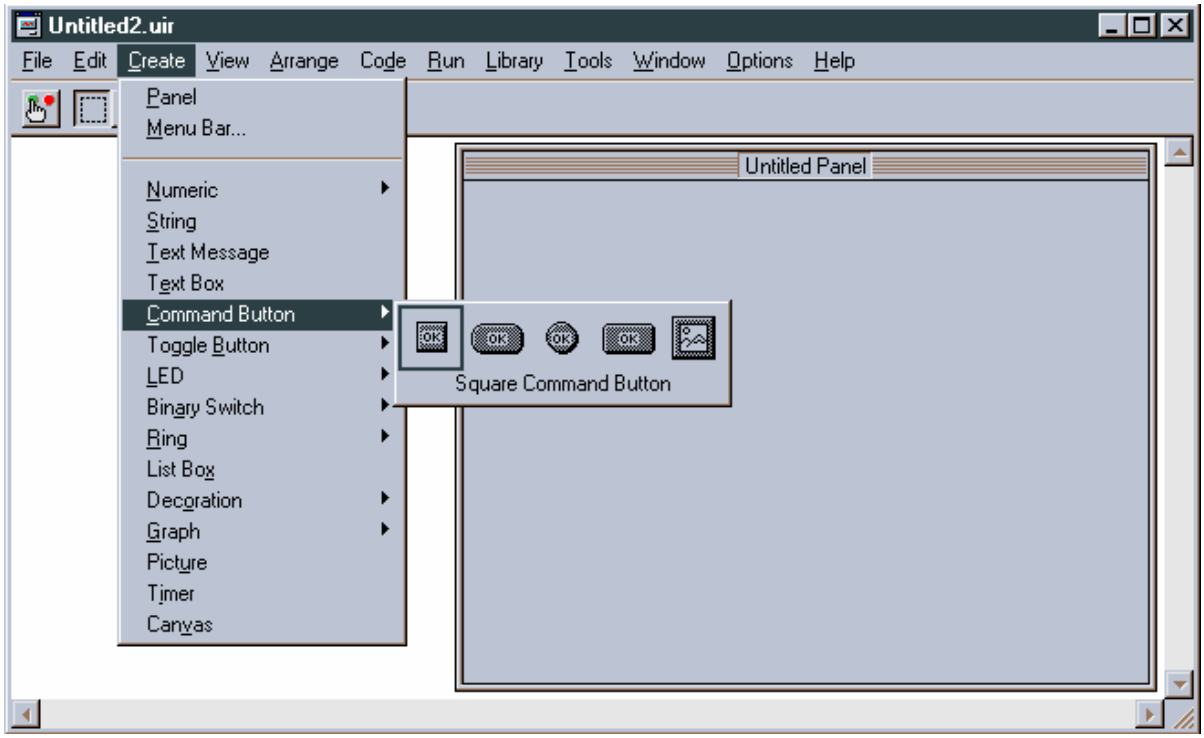
Etape 1. Création de l'interface graphique



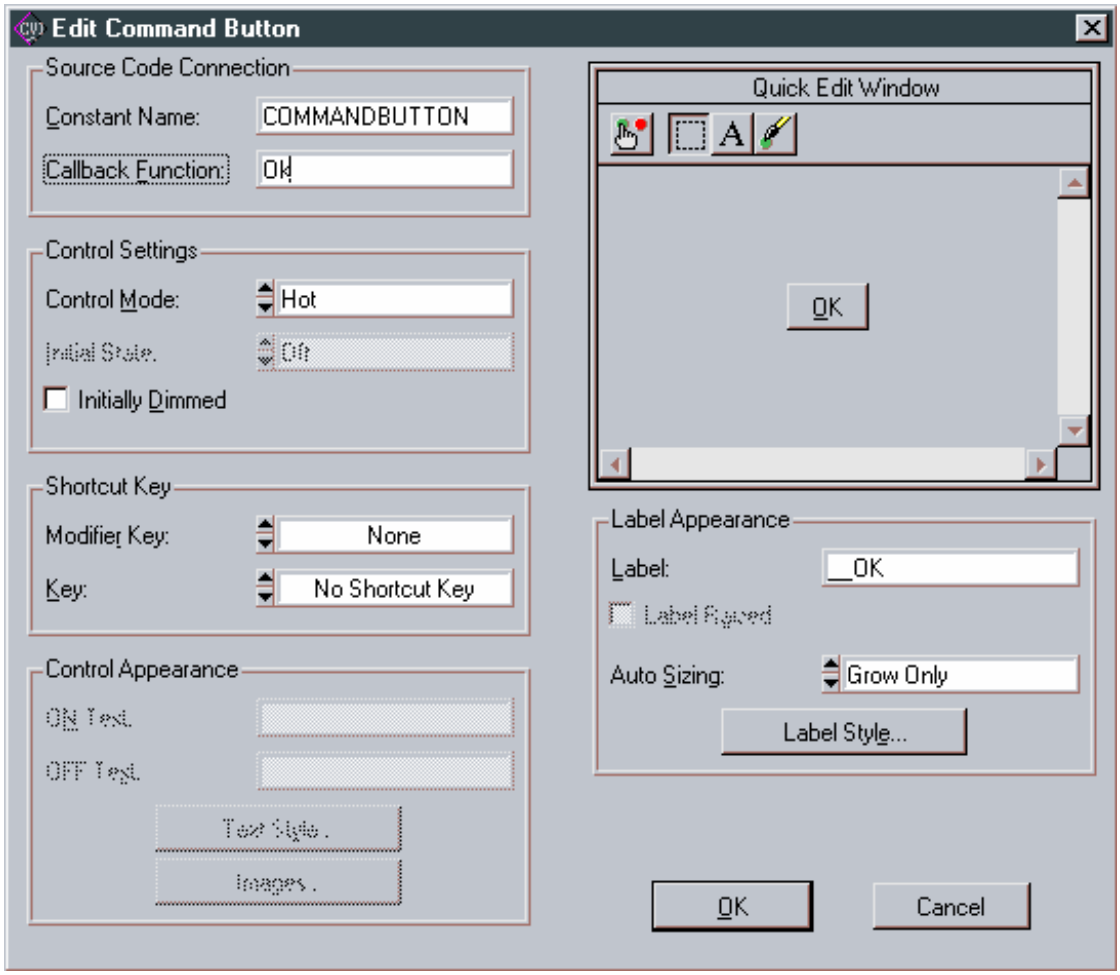
Création du panel (fenêtre) principal



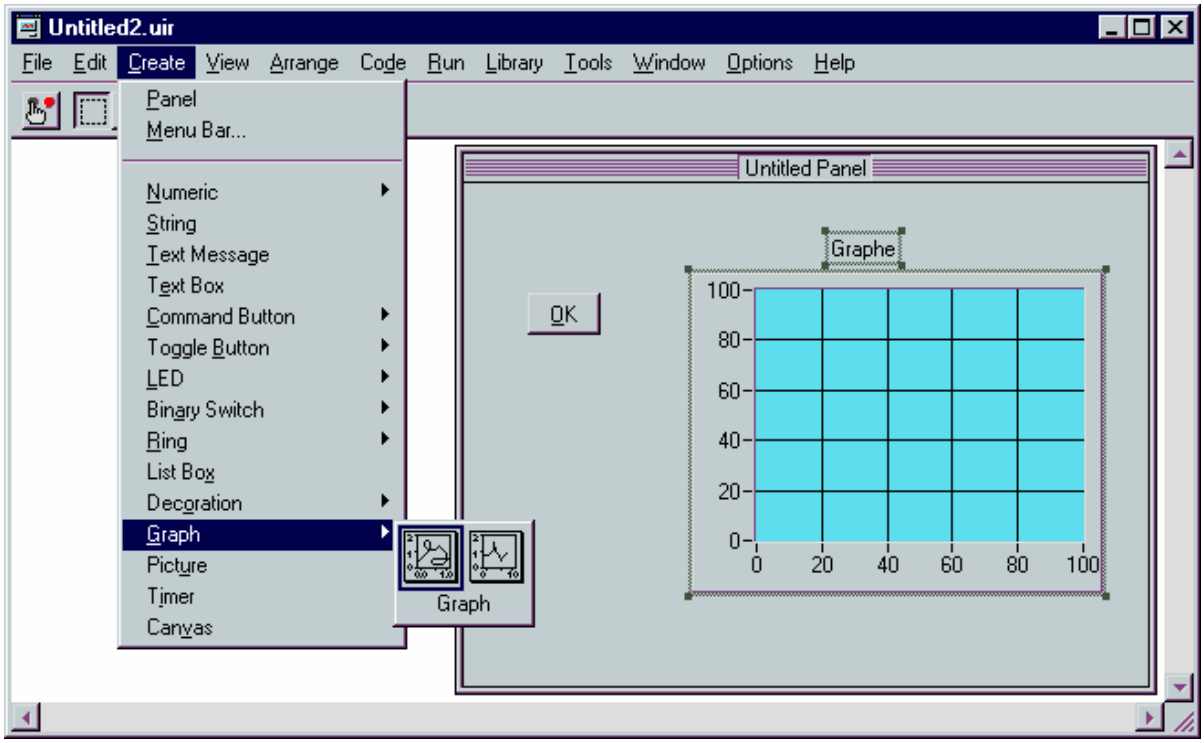
Création du bouton OK

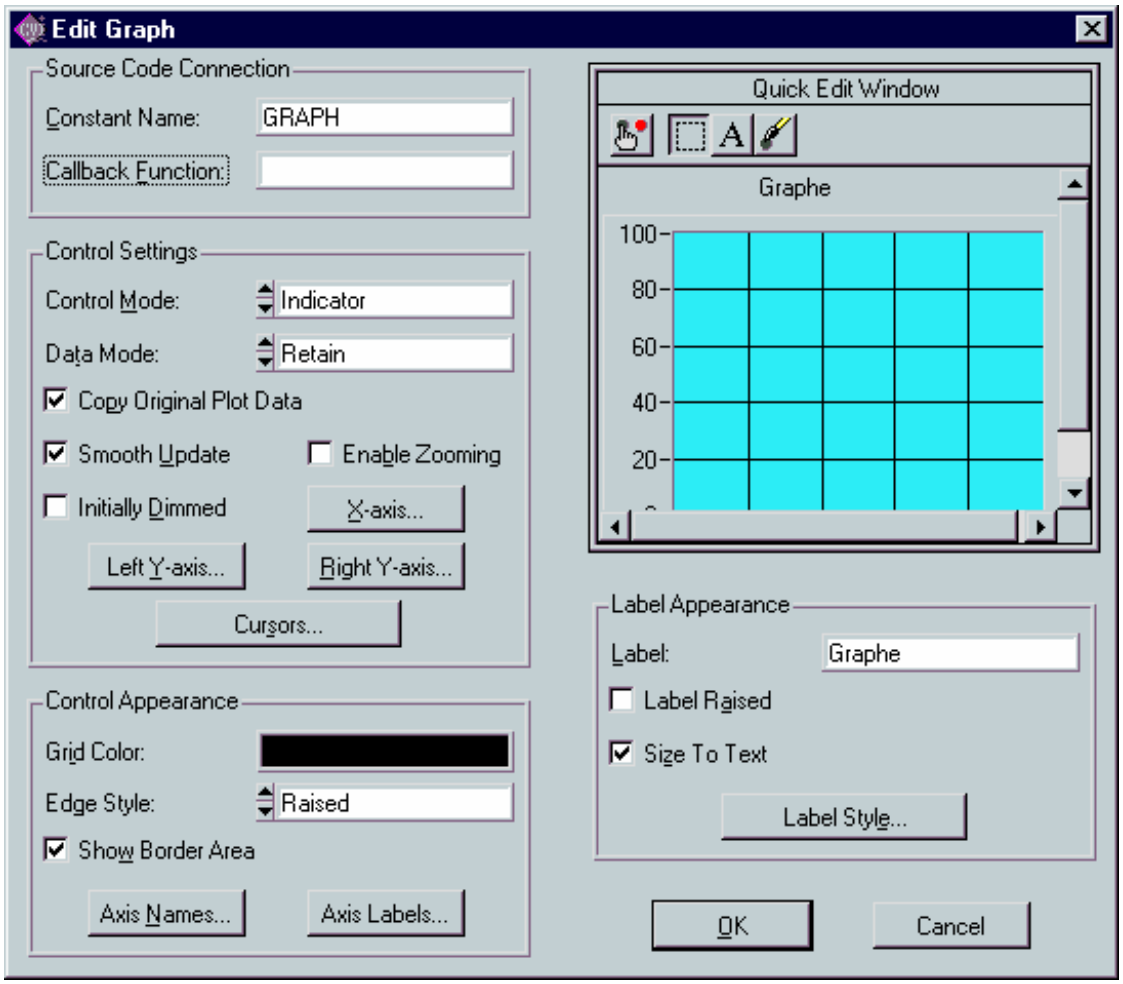


Création de la fonction callback Ok() associée au bouton OK

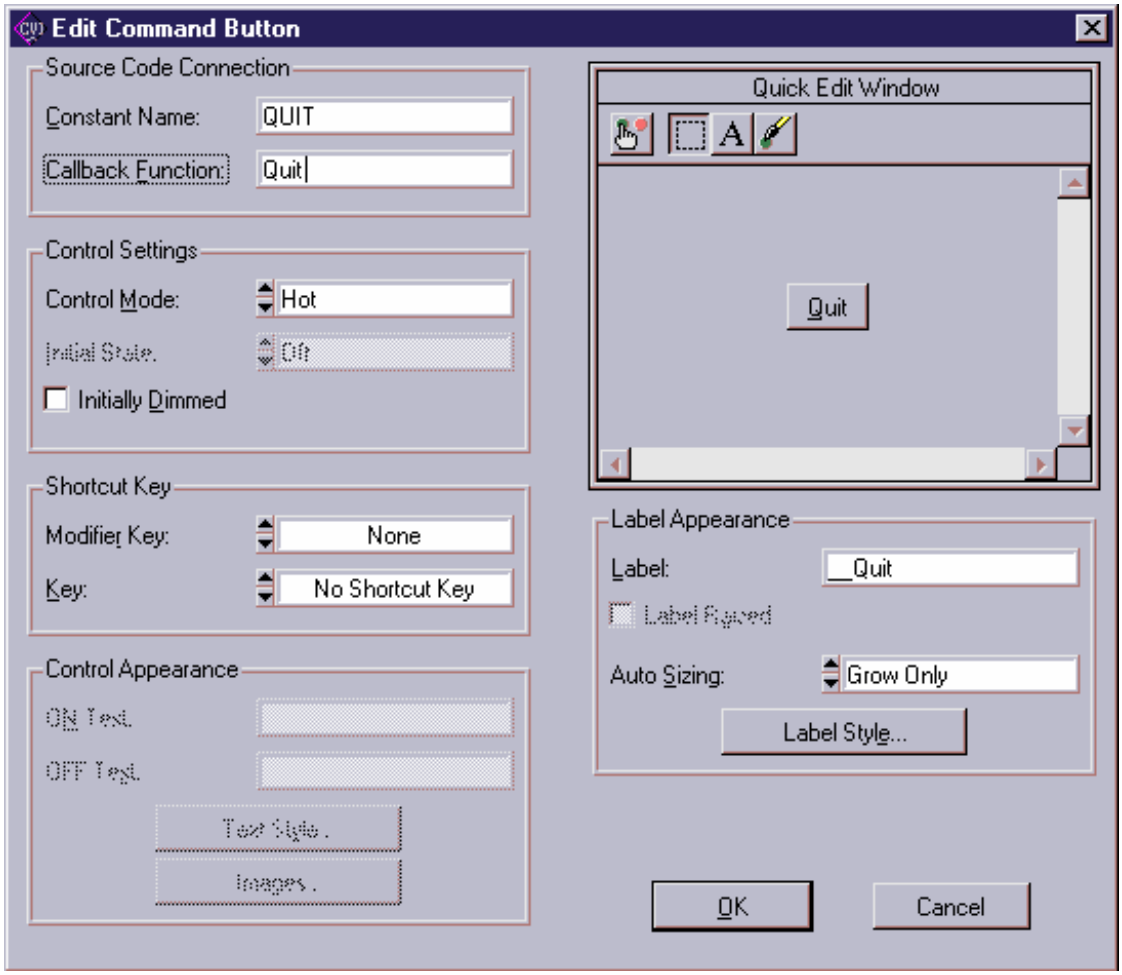


Création de l'objet graphe GRAPH

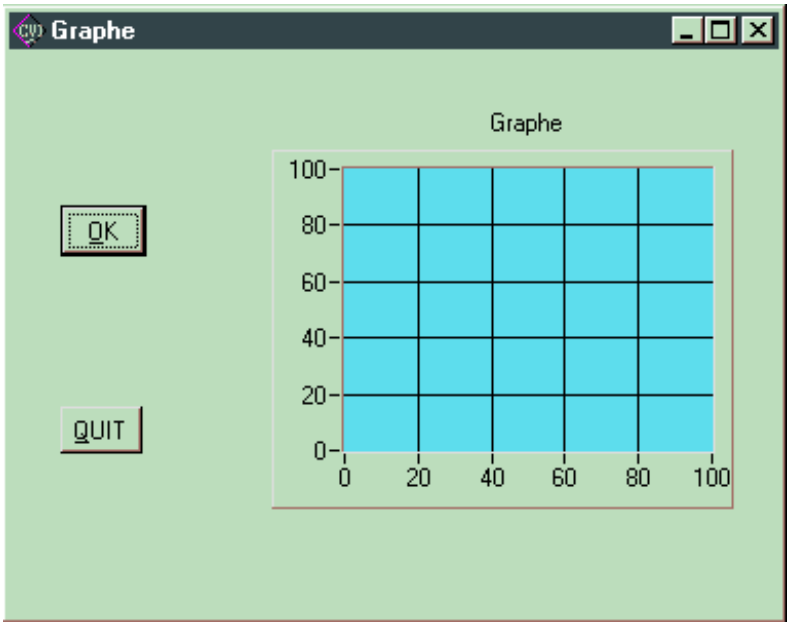




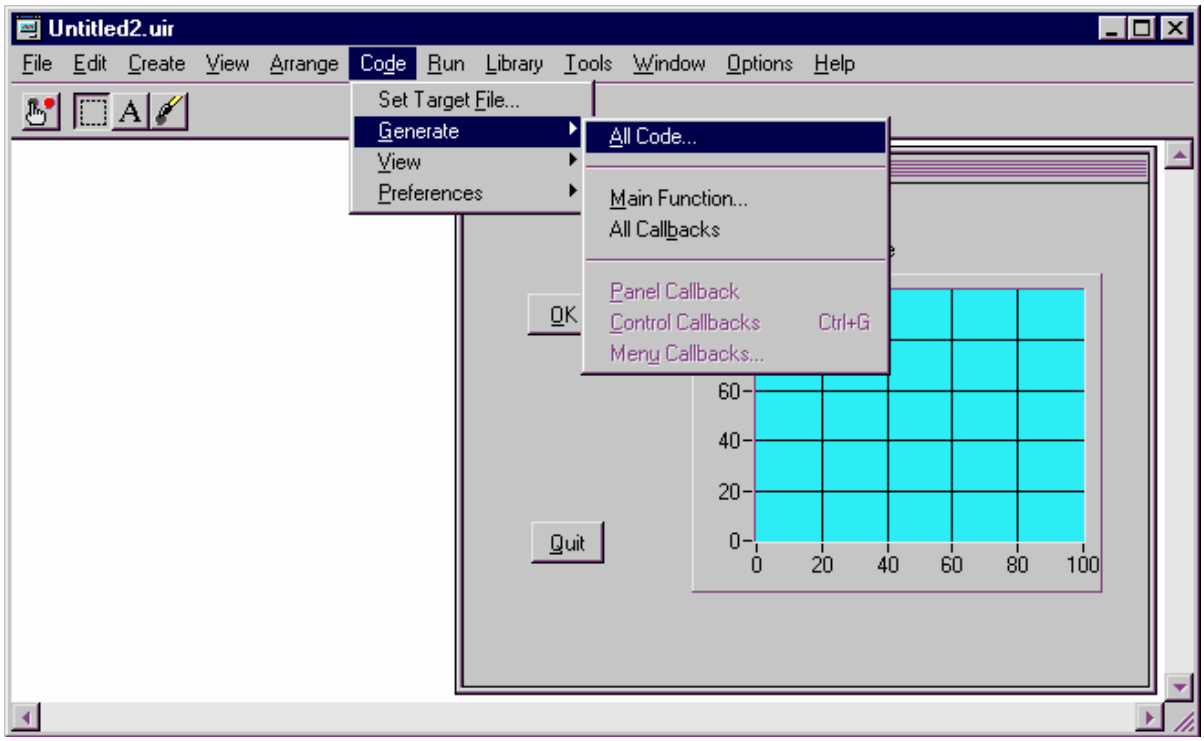
Création du bouton QUIT (fonction callback Quit() associée)



Interface terminée



Etape 2. Génération du code C



Les fichiers sont ensuite nommés *Graphe* et non plus « *untitled* ».

La fonction *QuitUserInterface()* est associée (mais cela peut être fait manuellement par la suite) au bouton *QUIT*.

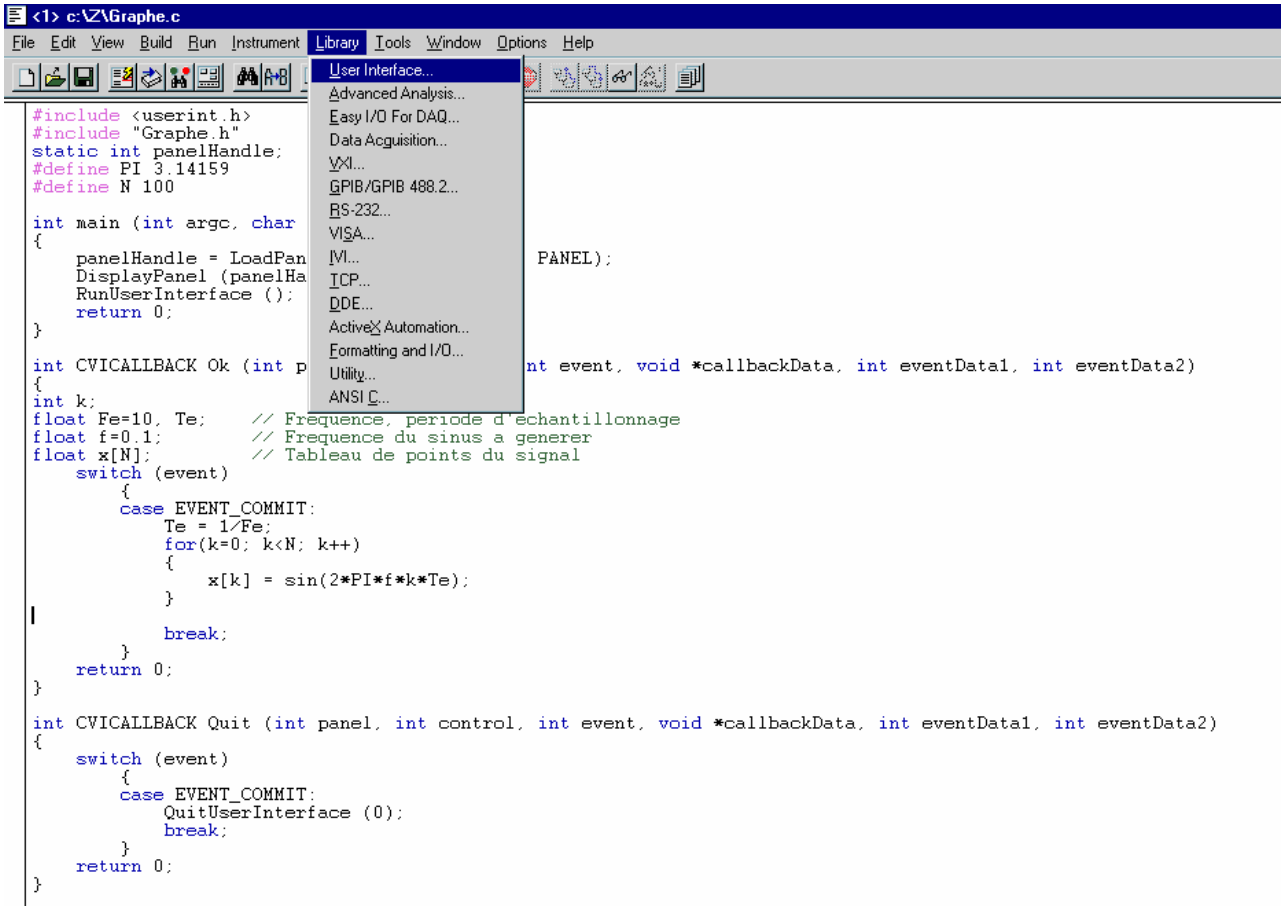
Fichier « Graphe.c » généré

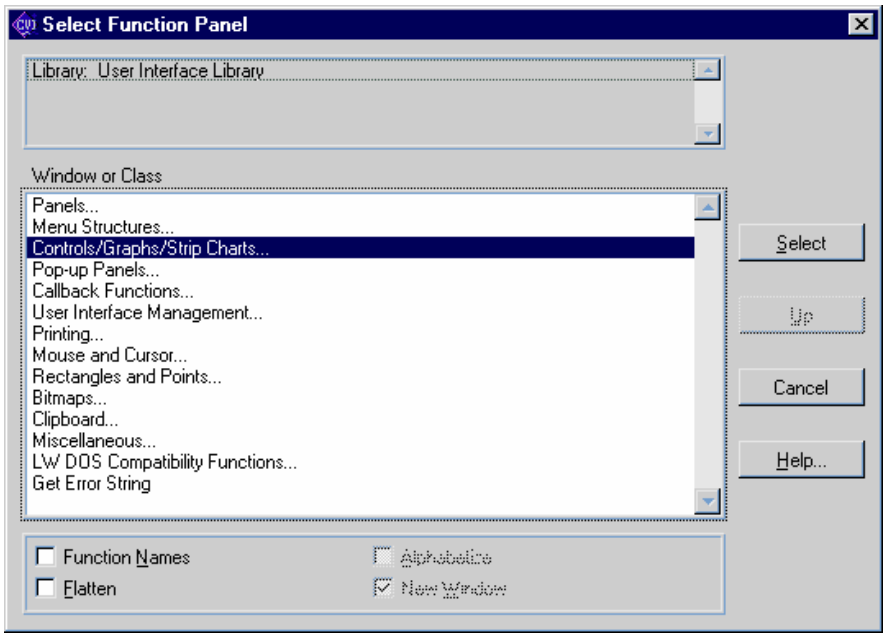
```
#include <userint.h>
#include "Graphe.h"
static int panelHandle;

int main (int argc, char *argv[])
{
    panelHandle = LoadPanel (0, "Graphe.uir", PANEL);
    DisplayPanel (panelHandle);
    RunUserInterface ();
    return 0;
}

int CVICALLBACK Ok (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
        {
        case EVENT_COMMIT:

            break;
        }
    return 0;
}

int CVICALLBACK Quit (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
        {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
        }
    return 0;
}
```

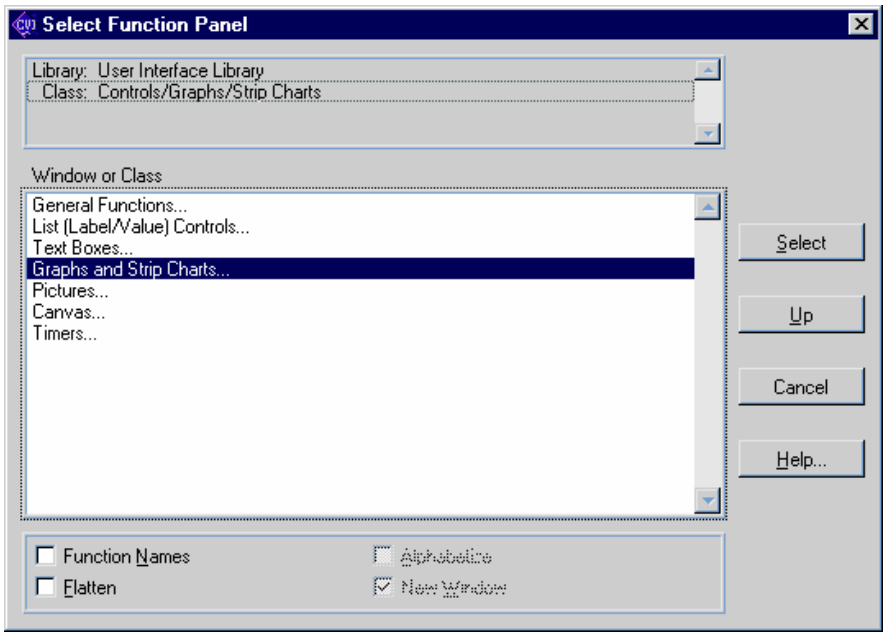
Etape 3. Insertion dans le code C généré de la création et de l'affichage du tableau de N points (sinus)

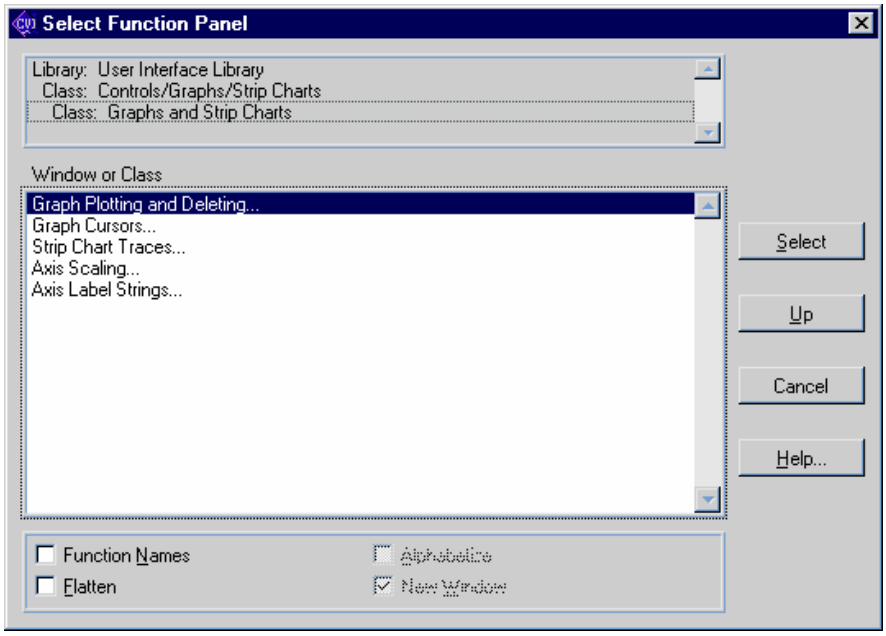
La fonction *PlotY()* d'affichage du tableau (comme la fonction *QuitUserInterface() de sortie de programme*) appartiennent aux librairies LabWindows/CVI, notamment la librairie *User Interface*.

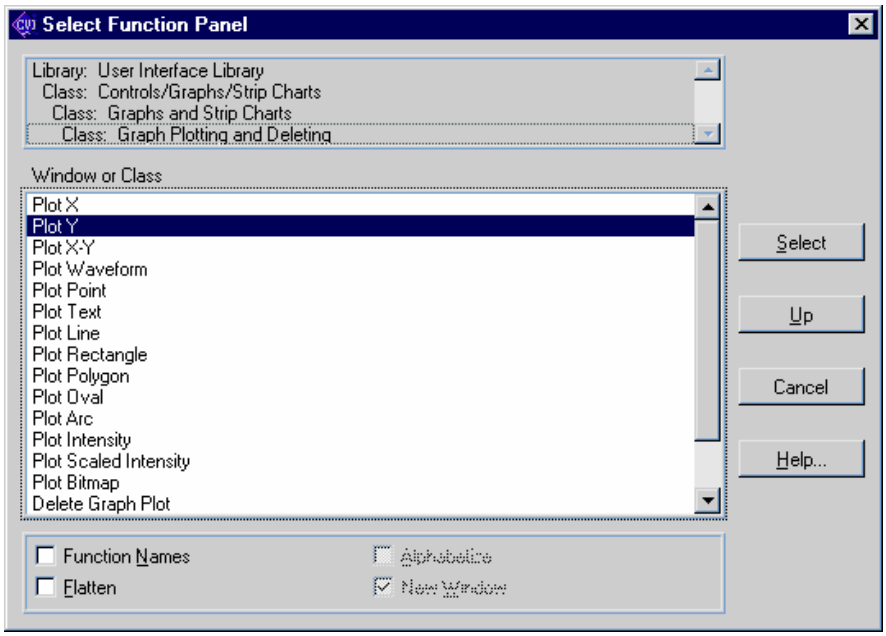
Par exemple, pour générer l'instruction d'appel de la fonction *PlotY()*, on place préalablement le curseur à l'endroit désiré, et on va chercher l'instruction *PlotY()* en librairie *User Interface* :

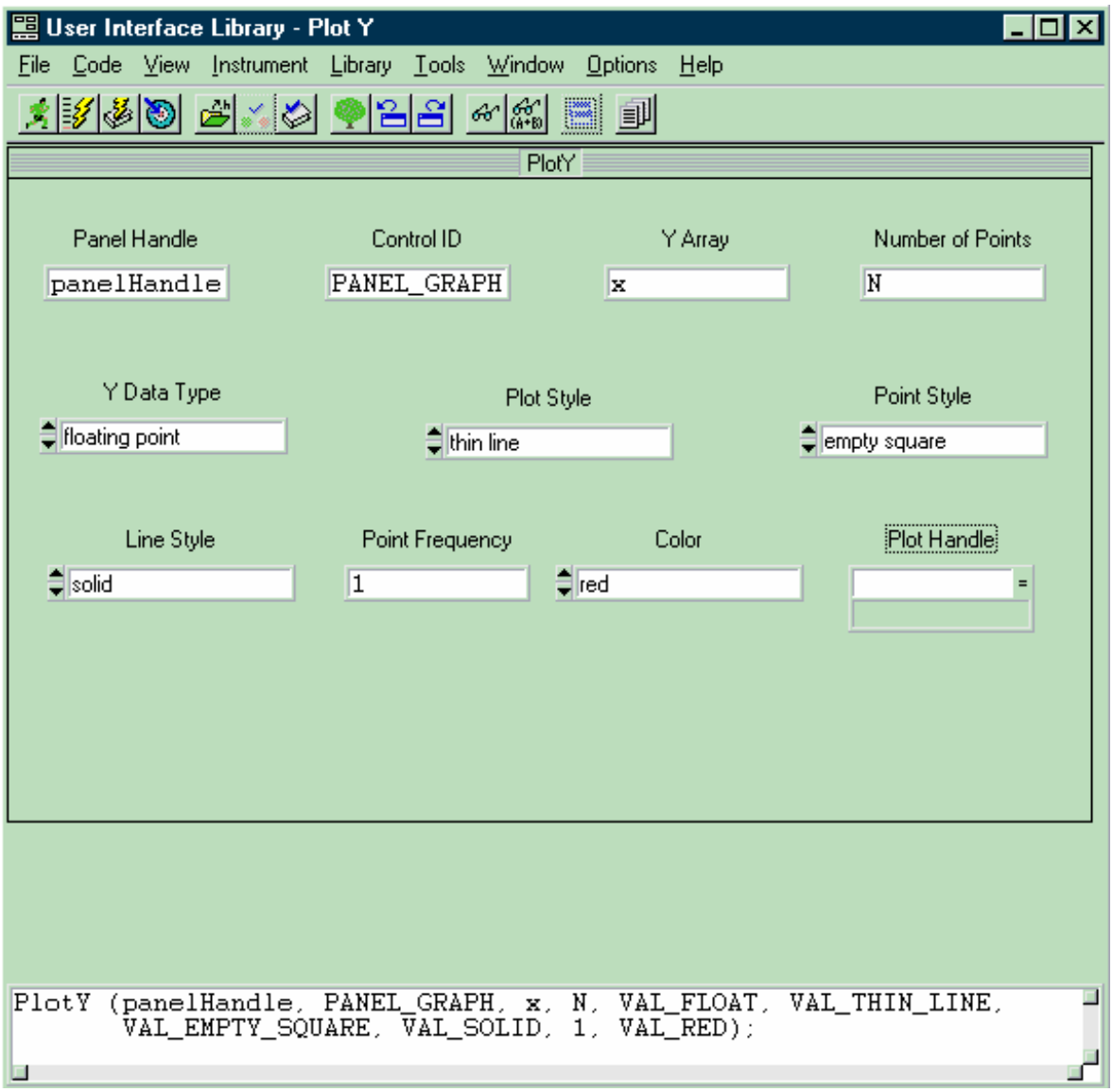


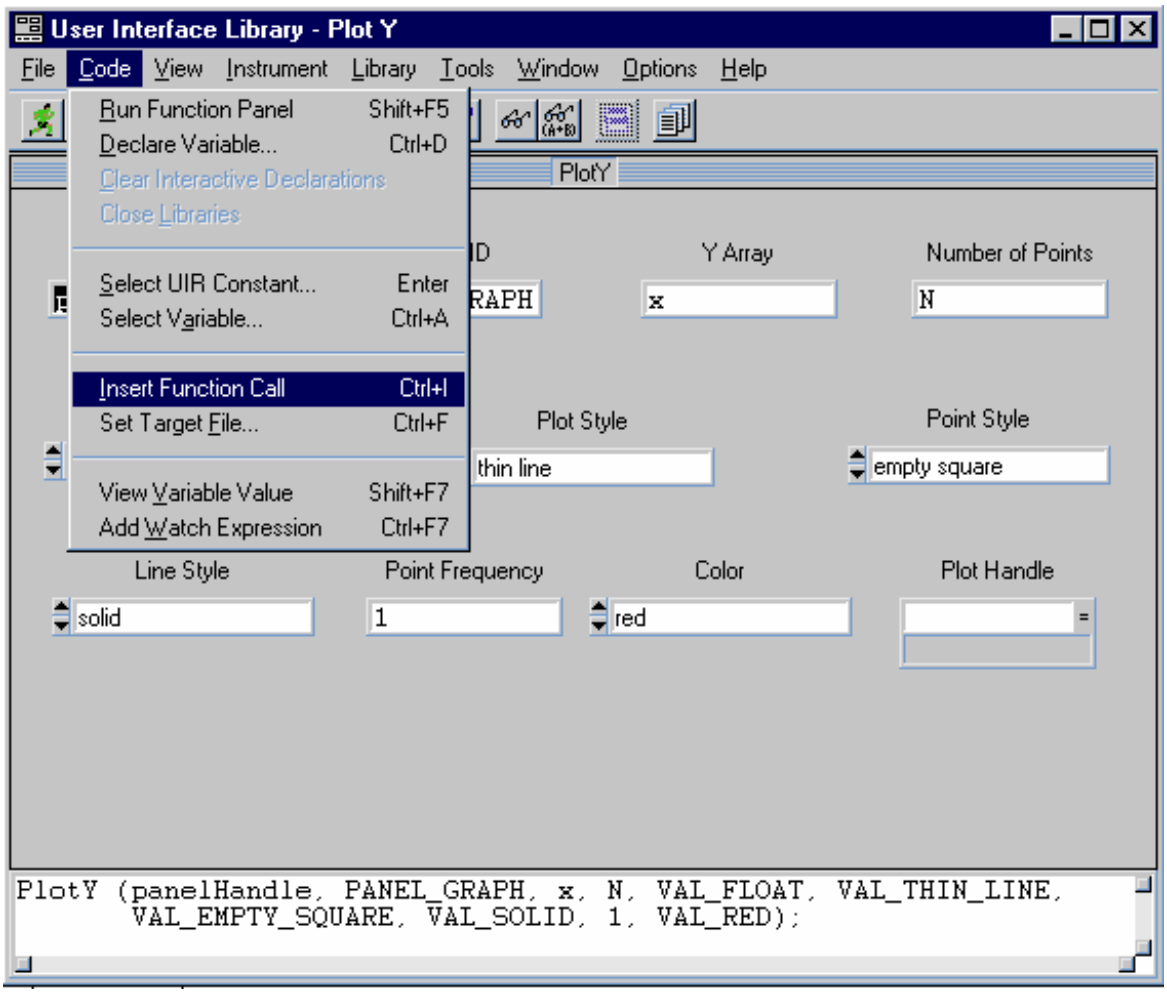



Select Function Panel
Library: User Interface Library
Class: Controls/Graphs/Strip Charts
Window or Class
General Functions...
List (Label/Value) Controls...
Text Boxes...
Graphs and Strip Charts...
Pictures...
Canvas...
Timers...
Select
Up
Cancel
Help...
Function Names
Flatten
Alphabetize
New Window


Select Function Panel
Library: User Interface Library
Class: Controls/Graphs/Strip Charts
Class: Graphs and Strip Charts
Window or Class
Graph Plotting and Deleting...
Graph Cursors...
Strip Chart Traces...
Axis Scaling...
Axis Label Strings...
Select
Up
Cancel
Help...
Function Names
Flatten
Alphabetize
New Window


Select Function Panel
Library: User Interface Library
Class: Controls/Graphs/Strip Charts
Class: Graphs and Strip Charts
Class: Graph Plotting and Deleting
Window or Class
Plot X
Plot Y
Plot X-Y
Plot Waveform
Plot Point
Plot Text
Plot Line
Plot Rectangle
Plot Polygon
Plot Oval
Plot Arc
Plot Intensity
Plot Scaled Intensity
Plot Bitmap
Delete Graph Plot
Select
Up
Cancel
Help...
Function Names
Flatten
Alphabetize
New Window

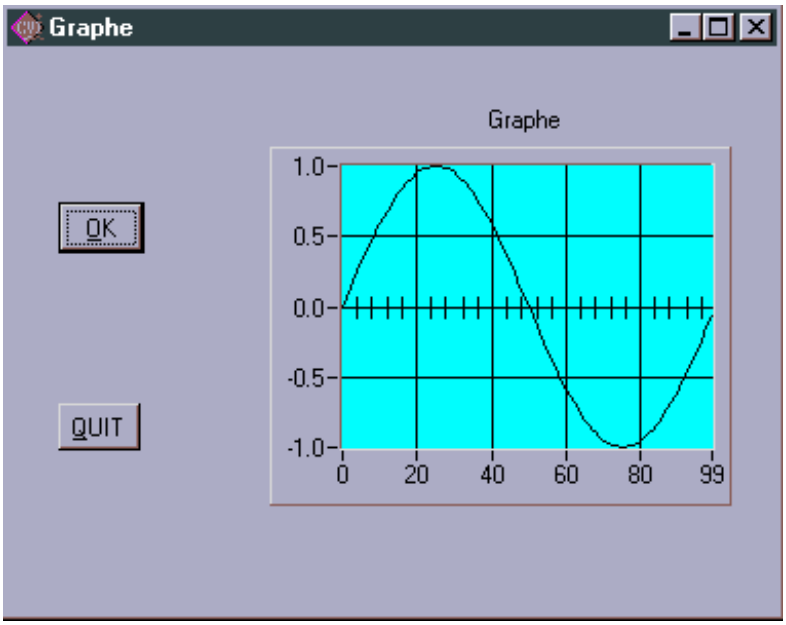

User Interface Library - Plot Y
File Code View Instrument Library Tools Window Options Help
PlotY
Panel Handle
panelHandle
Control ID
PANEL_GRAPH
Y Array
x
Number of Points
N
Y Data Type
floating point
Plot Style
thin line
Point Style
empty square
Line Style
solid
Point Frequency
1
Color
red
Plot Handle
PlotY (panelHandle, PANEL_GRAPH, x, N, VAL_FLOAT, VAL_THIN_LINE,
VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_RED);


User Interface Library - Plot Y
File Code View Instrument Library Tools Window Options Help
Run Function Panel Shift+F5
Declare Variable... Ctrl+D
Clear Interactive Declarations
Close Libraries
Select UIR Constant... Enter
Select Variable... Ctrl+A
Insert Function Call Ctrl+I
Set Target File... Ctrl+F
View Variable Value Shift+F7
Add Watch Expression Ctrl+F7
PlotY
Y Array
Number of Points
x
N
Plot Style
Point Style
thin line
empty square
Line Style
Point Frequency
Color
Plot Handle
solid
1
red
PlotY (panelHandle, PANEL_GRAPH, x, N, VAL_FLOAT, VAL_THIN_LINE,
VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_RED);

Fichier « Graphe.c » complété

```
#include <userint.h>
#include "Graphe.h"
static int panelHandle;
#define PI 3.14159
#define N 100

int main (int argc, char *argv[])
{
    panelHandle = LoadPanel (0, "Graphe.uir", PANEL);
    DisplayPanel (panelHandle);
    RunUserInterface ();
    return 0;
}

int CVICALLBACK Ok (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
int k;
float Fe=10, Te;// Frequence, periode d'echantillonnage
float f=0.1;            // Frequence du sinus a generer
float x[N];         // Tableau de points du signal
    switch (event)
        {
        case EVENT_COMMIT:
            Te = 1/Fe;
            for(k=0; k<N; k++)
            {
                x[k] = sin(2*PI*f*k*Te);
            }
            PlotY (panelHandle, PANEL_GRAPH, x, N, VAL_FLOAT,
                    VAL_THIN_LINE,VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_RED);
            break;
        }
    return 0;
}

int CVICALLBACK Quit (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
        {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
        }
    return 0;
}
```

Etape 4. Résultat d'exécution



2nd PROGRAMME D'EXEMPLE :
CLIGNOTANT PILOTE PAR UN TIMER REGLABLE