

Quand le langage C et LabView font bon ménage

Sabine DOUILLET Pôle Automatisation/Caractérisation
AlpeView 4 décembre 2014



2 exemples d'utilisations du C et de LabView dans le même projet:

1) Intégration d'un petit module LabView dans un programme en C (LabWindows/CVI)

Pilotage d'une expérience sur une plateforme d'optique

2) Réutiliser des fonctions déjà écrites en C dans un programme LabView

Pilotage d'expérience de diffraction X

1) Intégration d'un petit module LabView dans un programme en C (LabWindows/CVI)

```

if(VHeatlower>0)
    VHeatlower = VHeatlower-0.02;
else
    VHeatlower = VHeatlower+0.02;
SetVolages(VHeatlower, VBiaslower);
SetCtrlVal(panelHandle, MainPanel_VHeatingSetpoint, VHeatlower);
}
VbiasValue = 0;
SetVolages(VHeatingValue, VBiasValue);
SetCtrlVal(panelHandle, MainPanel_VHeatingSetpoint, VHeatingValue);
divisions = abs(VBiasValue/0.02);
for(i=0; i<divisions; i++)
{
    if(VBiaslower>0)
        VBiaslower = VBiaslower-0.02;
    else
        VBiaslower = VBiaslower+0.02;
    SetVolages(VHeatlower, VBiaslower);
    SetCtrlVal(panelHandle, MainPanel_VBiasSetpoint, VBiaslower);
}
VBiasValue = 0;
SetVolages(VHeatingValue, VBiasValue);
SetCtrlVal(panelHandle, MainPanel_VBiasSetpoint, VBiasValue);

//Reset Multimeter settings
return;
}

void IVMeasurement()
{
    IVMeasureActive = 1;

    GetCtrlVal(panelHandle, MainPanel_IVStartVoltage, &VStart);
    GetCtrlVal(panelHandle, MainPanel_IVStopVoltage, &Vstop);
    GetCtrlVal(panelHandle, MainPanel_IVNStep, &Points);
    GetCtrlVal(panelHandle, MainPanel_IVStepTime, &Vtime);

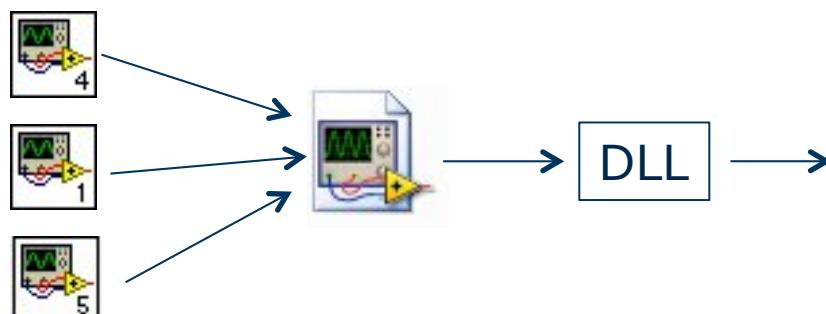
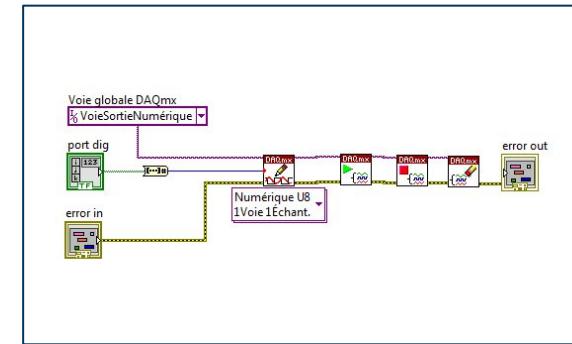
    VSpread = VStop-VStart;
    VoltsPerStep = VSpread/Points;
    VCurrent = VStart;
    IVCurrentStep = 0;
    SetVolages(VCurrent);
    printf("IVMeasure\n");
    SetCtrlAttribute(panelHandle, MainPanel_IVTimer, ATTR_INTERVAL, VTime);
    SetCtrlAttribute(panelHandle, MainPanel_IVTimer, ATTR_ENABLED, 1);

    // while (IVMeasureActive == 1)
    // {
    // return;
    // }

    void SetVolages(double VHeat, double VBias)
    {
        //calculate voltage levels for Lockin
        //transmit values

        int writestringlength = 0;
        char writestring[25];
    }
}

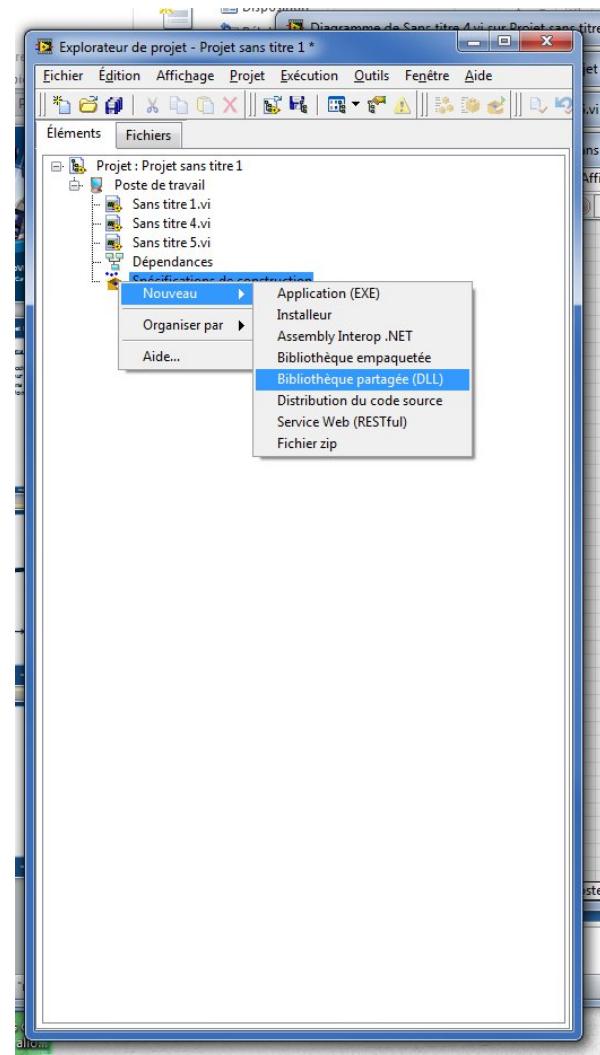
```



```

#include "nom de la dll.h"
.
.
.
Hinstlib=LoadLibrary("nom de la dll.dll")
.
.
.
ProcAddress=GetProcAddress(hinst,"nom du VI")
returnValue=(ProcAddress)(param1,param2)

```



```

if(VHeatlower>0)
    VHeatlower = VHeatlower-0.02;
else
    VHeatlower = VHeatlower+0.02;
SetVoltages(VHeatlower, VBiaslower);
SetCtrlVal(panelHandle, MainPanel_VHeatingSetpoint, VHeatlower);

VHeatingValue = 0;
SetVoltages(VHeatingValue, VBiasValue);
SetCtrlVal(panelHandle, MainPanel_VHeatingSetpoint, VHeatingValue);
divisions = abs(VBiasValue/0.02);
for(i=0; i<divisions; i++)
{
    if(VBiaslower>0)
        VBiaslower = VBiaslower-0.02;
    else
        VBiaslower = VBiaslower+0.02;
    SetVoltages(VHeatlower, VBiaslower);
    SetCtrlVal(panelHandle, MainPanel_VBiasSetpoint, VBiaslower);
}
VBiasValue = 0;
SetVoltages(VHeatingValue, VBiasValue);
SetCtrlVal(panelHandle, MainPanel_VBiasSetpoint, VBiasValue);

//Reset Multimeter settings
return;
}

void IVMeasurement()
{
    IVMeasureactive = 1;

    GetCtrlVal(panelHandle, MainPanel_IVStartVoltage, &VStart);
    GetCtrlVal(panelHandle, MainPanel_IVStopVoltage, &VStop);
    GetCtrlVal(panelHandle, MainPanel_IVNumSteps, &VPoints);
    GetCtrlVal(panelHandle, MainPanel_IVStepTime, &VTime);

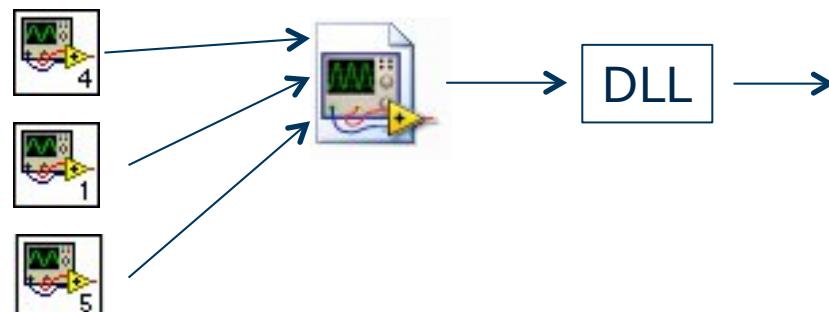
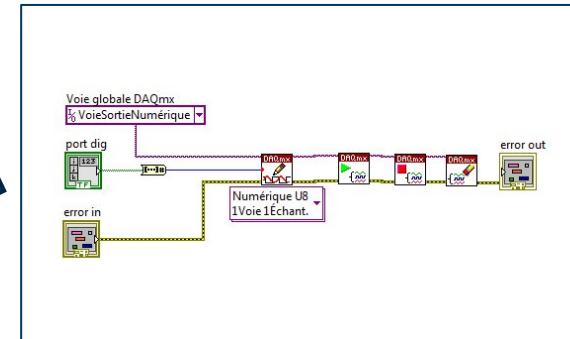
    VSpread = Vstop-Vstart;
    VoltsPerStep = VSpread/VPoints;
    VCurrentStep = 0;
    IVCurrentStep = 0;
    SetVolages(VStart, VCurrent);
    printf("IVmeasure \n");
    SetCtrlAttribute(panelHandle, MainPanel_IVTimer, ATTR_INTERVAL, VTime);
    SetCtrlAttribute(panelHandle, MainPanel_IVTimer, ATTR_ENABLED, 1);

//    while (IVMeasureactive == 1)
//    {
//        return;
//    }

void SetVoltages(double VHeat, double VBias)
{
    //calculate voltage levels for Lockin
    //transmit values

    int writestringlength = 0;
    char writestring[25];
}

```



#include "*nom de la dll.h*"

.

Hinstlib=LoadLibrary(""*nom de la dll.dll*"")

.

ProcAddress=GetProcAddress(hinst,'"*nom du VI*"')

returnValue=(ProcAddress)(param1,param2)

2) Réutiliser des fonctions déjà écrites en C dans un programme LabView

Pilotage d'expérience de diffraction X

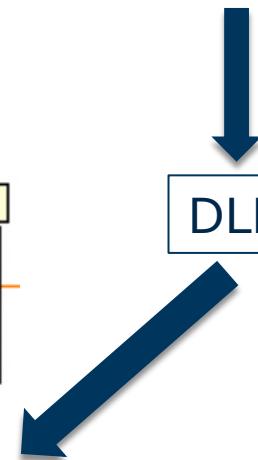
Programme en Pascal
sur plateforme MacOS
Écrit au début des années 90
≈25000 lignes
Ordinateur, OS et interface
électronique dépassés
Mais un grand nombre de
routines étaient réutilisables



Fonctions en langage C
intégrées dans un projet
LabWindows/CVI



DLL



Nouveau programme de pilotage développé en LabView
sous Windows intégrant de nombreux appels à la DLL

Conclusion

- Gain de temps
- Pas si compliqué que ça avec un tutorial
- Pas de connaissance approfondie en C requise
- Le passage par les DLL n'est pas la seule solution mais la plus simple à mon avis
- Solution pérenne malgré les successions de versions de Windows

Fonction LabView « Appeler une fonction d'une DLL »

