

Q. How to drive a DAQ card with LabWindows/CVI ?

A: Please follow these steps:

Step 1: Install the driver.

1. **Setup the UniDAQ driver.** The driver is located at:
 CD:\NAPDOS\PCI\UniDAQ\DLL\Driver\
<ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/driver/>
2. For detailed information about the UniDAQ driver installation, please refer to UniDAQ DLL Software Manual contained in:
 CD:\NAPDOS\PCI\UniDAQ\Manual\
<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/manual/>

Step 2: Install the hardware on PC.

1. Power off the PC.
2. Remove all covers from the computer.
3. Carefully insert the DAQ Card into slot.
4. Replace the PC covers.
5. Power on the PC.

Step 3: Get the ".h" and ".lib" files.

You can get the "UniDAQ.h" and "UniDAQ.lib" files from the VC demo. The VC demo is located at: CD:\NAPDOS\PCI\UniDAQ\DLL\Demo\
<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/>

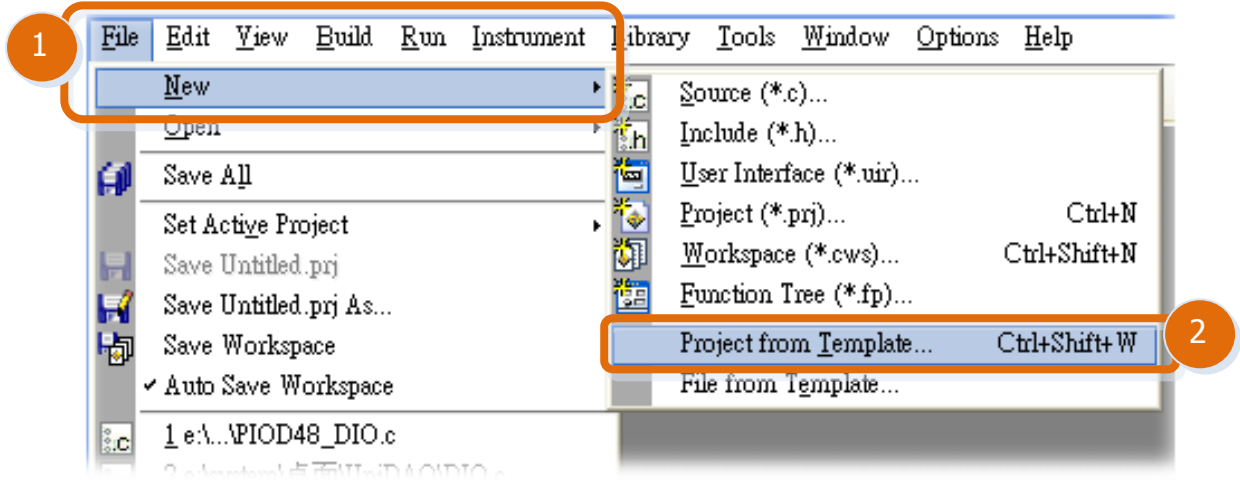


UniDAQ.h
C Header file
16 KB

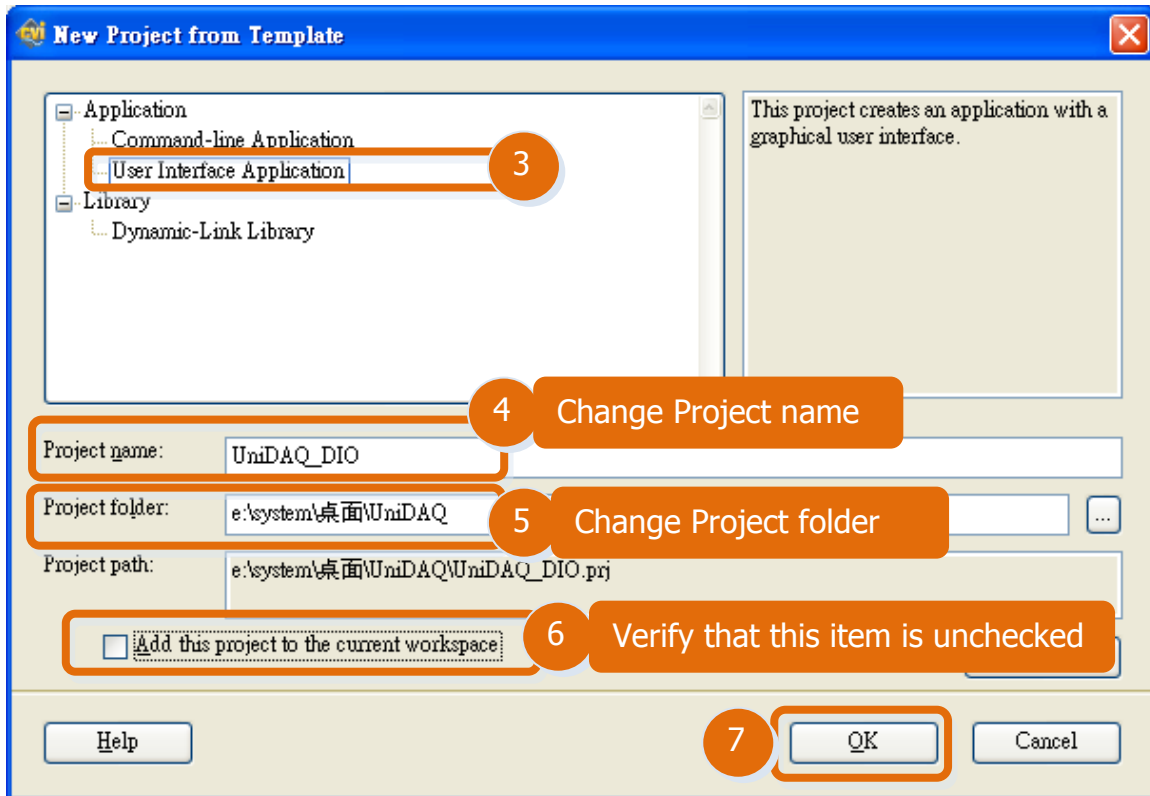


UniDAQ.lib
LIB 檔案
16 KB

Step 4: Launch LabWindows/CVI, select **File → **New** → **Project from Template** to open the “new Project from Template” dialog box.**

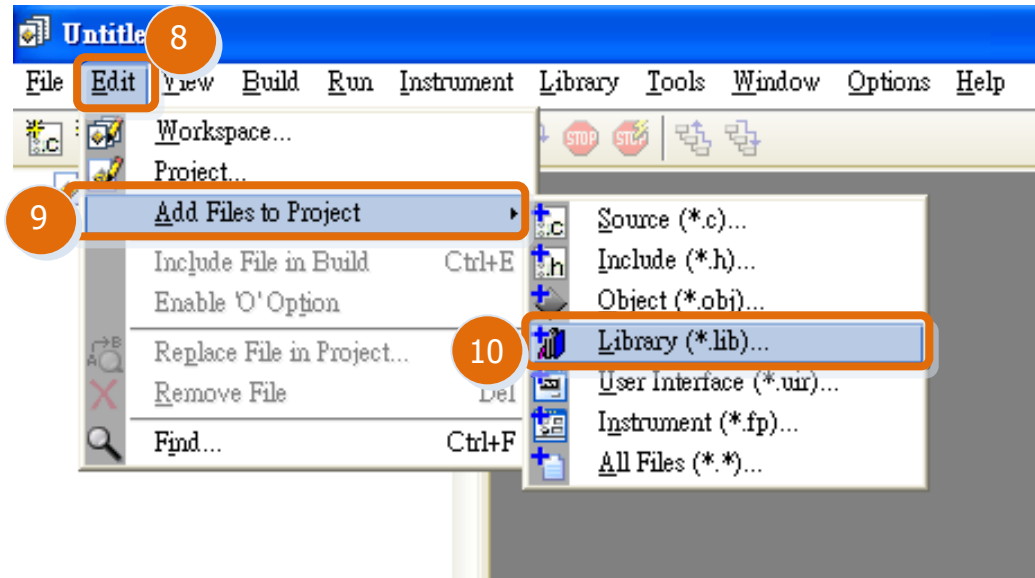


Step 5: Select the **User Interface Application item and click the OK button.**

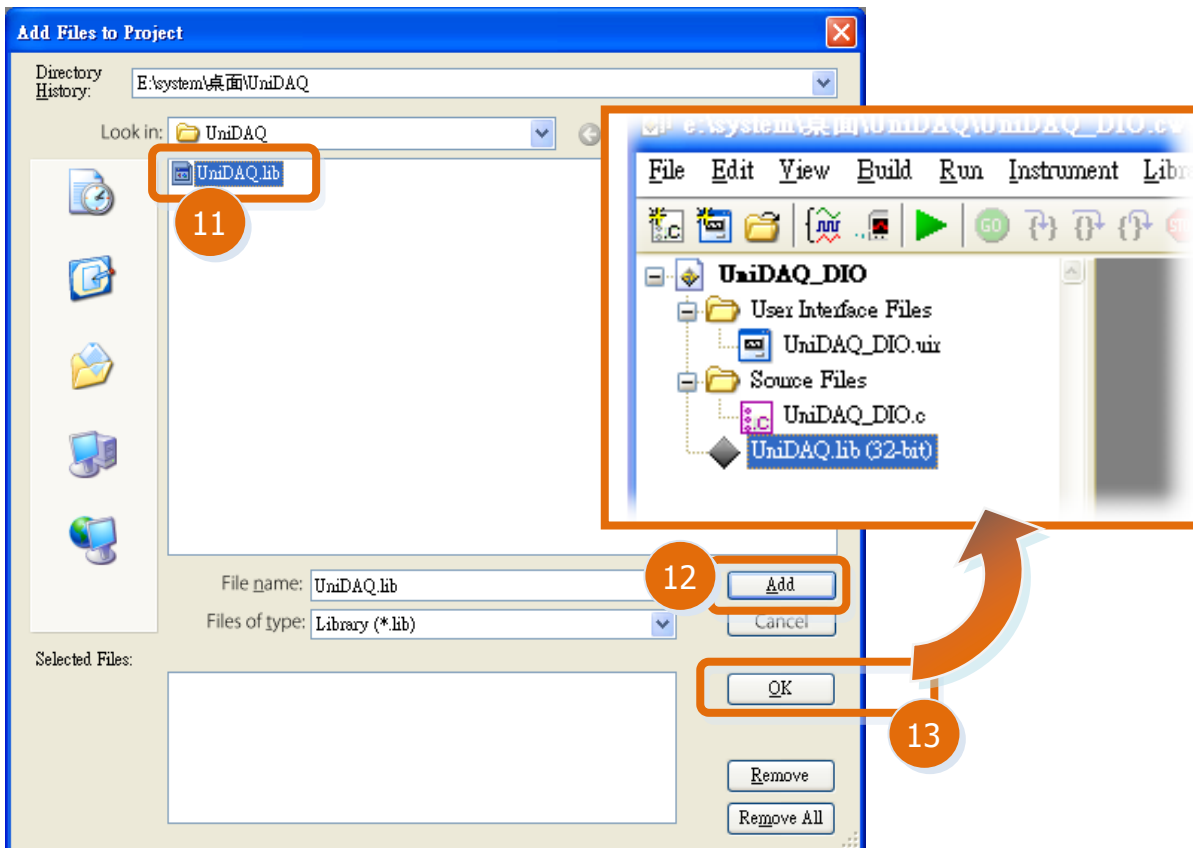


Step 6: Add the ".lib" file into the project.

1. In the LabWindows/CVI, select the **"Add Files to Project" → "Library (*.lib)"** item from the **"Edit"** menu.



2. Select the **"UniDAQ.lib"** file, click the **"Add"** button, and then click the **"OK"** button.



Step 7: Add header files by inserting the two lines "#include <Windows.h>" and "#include "UniDAQ.h"" at the beginning of the .c file.

The screenshot shows the UniDAQ software interface. On the left, a file explorer shows the project structure with 'UniDAQ_DIO.c' selected. An orange callout box with the number '14' and the text 'Double-Click ".c" file' points to this file. The main editor window displays the source code for 'UniDAQ_DIO.c'. An orange callout box with the number '15' highlights the 'Include files' section, where the following code has been added:

```
// Include files
#include <Windows.h> ✓
#include <cvirte.h>
#include <userint.h>
#include "UniDAQ_DIO.h"
#include "UniDAQ.h" ✓
```

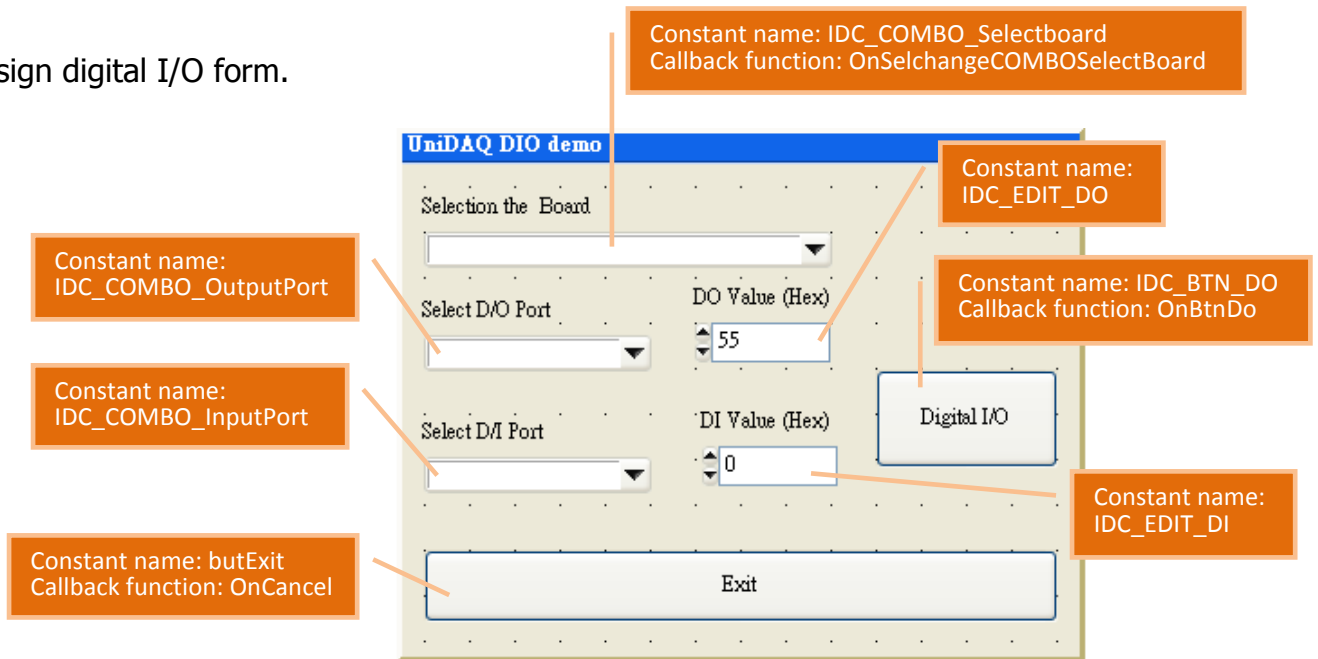
The rest of the code in the editor includes comments for Constants, Types, Static global variables, Static functions, Global variables, Global functions, and the main entry-point function:

```
/// Constants
//-----
// Types
//-----
// Static global variables
static int panelHandle;
//-----
// Static functions
//-----
// Global variables
//-----
// Global functions
/// HIFN The main entry-point function.
int main (int argc, char *argv[])
```

Step 8: Now you can start to design the user interface (.uir file) and call the UniDAQ DLL in your project.

For example: Write Digital Input and Digital Output application, as follows:

1. Design digital I/O form.



2. Write the code as below:

```
#include <Windows.h>
#include <cvirte.h>
#include <userint.h>
#include "UniDAQ_DIO.h"
#include "UniDAQ.h"

static int panelHandle;
WORD wRtn;
int wBoardNo;
int wBoardIndex;
WORD PortIndex;
IXUD_DEVICE_INFO sDevInfo[MAX_BOARD_NUMBER];
IXUD_CARD_INFO sCardInfo[MAX_BOARD_NUMBER];
char szModelName[20]="Unknow Device";

int main (int argc, char *argv[])
{
    int iCardNum =0;
    WORD wTotalBoards;

    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "UniDAQ_DIO.uir", PANEL)) < 0)
        return -1;
```

```

//Initial the resource and get total board number form driver
★ wRtn=Ixud_DriverInit(&wTotalBoards);
if (wRtn!=0) return wRtn;

ClearListCtrl(panelHandle, PANEL_IDC_COMBO_SelectBoard);
InsertListItem (panelHandle, PANEL_IDC_COMBO_SelectBoard, 0,"--Select Board--",0);

//Get DAQ Card Information
for(WORD wBoardIndex = 0; wBoardIndex<wTotalBoards ; wBoardIndex++)
{
★ wRtn = Ixud_GetCardInfo(wBoardIndex,&sDevInfo[wBoardIndex],&sCardInfo[wBoardIndex],szModelName);
InsertListItem (panelHandle, PANEL_IDC_COMBO_SelectBoard, -1, szModelName, wBoardIndex);
}

DisplayPanel (panelHandle);
RunUserInterface ();
DiscardPanel (panelHandle);
return 0;
}

int CVICALLBACK OnSelchangeCOMBOSelectBoard (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{
WORD wPortChNum;
char sPortNo[32];

GetCtrlIndex (panelHandle, PANEL_IDC_COMBO_SelectBoard, &wBoardNo);
wPortChNum = sCardInfo[wBoardNo].wDIOPortWidth;

switch (event)
{
case EVENT_COMMIT:

//Total D/O port
ClearListCtrl(panelHandle, PANEL_IDC_COMBO_OutputPort);

for (PortIndex = 0; PortIndex < (sCardInfo[wBoardNo].wDIOPorts +
sCardInfo[wBoardNo].wDOPorts) ;PortIndex++)
{
sprintf(sPortNo, "%02d", PortIndex);
InsertListItem (panelHandle, PANEL_IDC_COMBO_OutputPort, -1, sPortNo ,PortIndex);
}

//Total D/I port
ClearListCtrl(panelHandle, PANEL_IDC_COMBO_InputPort);

for (PortIndex = 0; PortIndex < (sCardInfo[wBoardNo].wDIOPorts +
sCardInfo[wBoardNo].wDIPorts) ;PortIndex++)
{
sprintf(sPortNo, "%02d", PortIndex);
InsertListItem (panelHandle, PANEL_IDC_COMBO_InputPort, -1, sPortNo, PortIndex);
}
break;
}
return 0;
}

```

```

int CVICALLBACK OnBtnDo (int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2)
{
    WORD wRtn;
    int wOutputPortNo;
    int wInputPortNo;
    DWORD dwDOVal;
    DWORD dwDIVal;

    GetCtrlIndex (panelHandle, PANEL_IDC_COMBO_SelectBoard, &wBoardNo);
    GetCtrlIndex (panelHandle, PANEL_IDC_COMBO_OutputPort, &wOutputPortNo);
    GetCtrlIndex (panelHandle, PANEL_IDC_COMBO_InputPort, &wInputPortNo);

    switch (event)
    {
        case EVENT_COMMIT:

            //Config the DIO Port
            if(sCardInfo[wBoardNo].wDIOPorts>0)
            ★ wRtn = Ixud_SetDIOModes32(wBoardNo, 1<<wOutputPortNo);

            //Write digital output port Value
            GetCtrlVal (panelHandle, PANEL_IDC_EDIT_DO, &dwDOVal);
            ★ wRtn = Ixud_WriteDO(wBoardNo,wOutputPortNo,dwDOVal);

            //Read digital input port Value
            ★ wRtn = Ixud_ReadDI(wBoardNo,wInputPortNo,&dwDIVal);
            SetCtrlVal (panelHandle, PANEL_IDC_EDIT_DI, dwDIVal);

            break;
    }
    return 0;
}

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

            //Release the resource form Driver
            ★ wRtn = Ixud_DriverClose();
            QuitUserInterface (0);
            break;
    }
    return 0;
}

```

★ For detailed information about the UniDAQ DLL, please refer to the UniDAQ DLL user manual. The manual is located at:
<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/manual/>