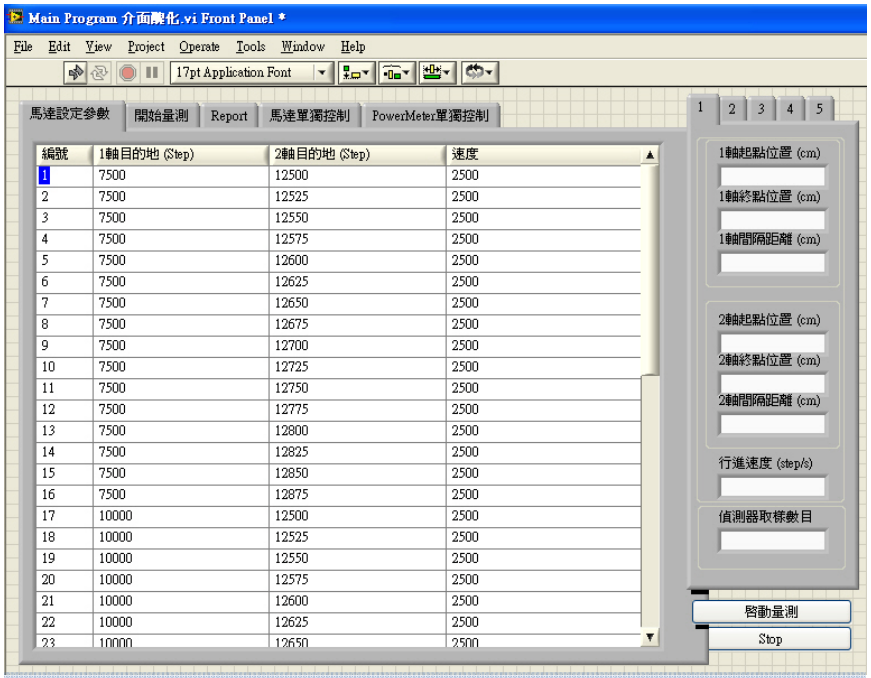
# 單元10：
# 人機介面設計技術

主題：

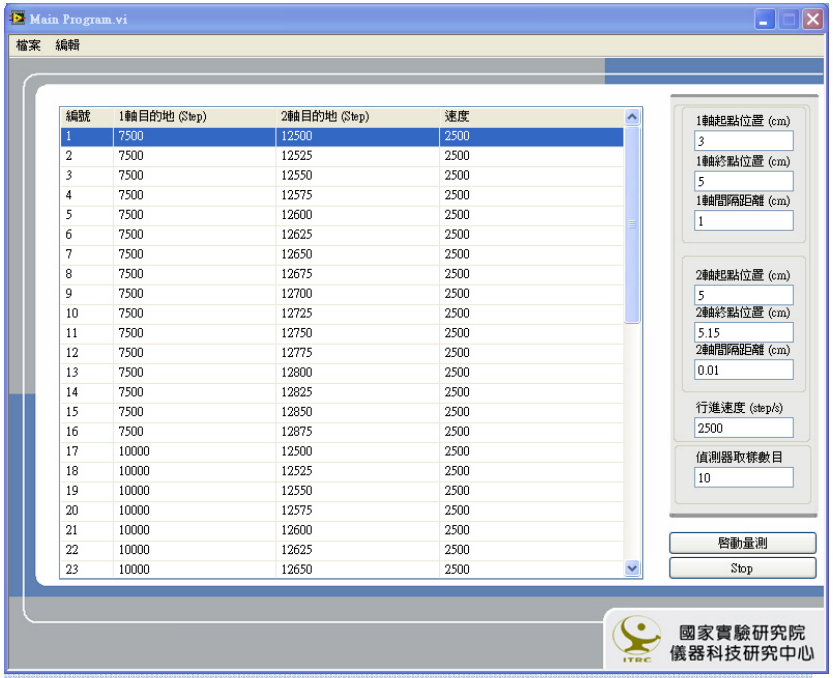
a. 基礎使用者人機介面議題
b. 適當的接線導引
c. 關於程式方塊圖註解
d. 錯誤處理技術
e. 序列結構
f. LabVIEW run-time 目錄

# 人機介面的設計

- 適當運用各種技巧可以改善人機介面，讓使用者更容易操作你的儀控程式



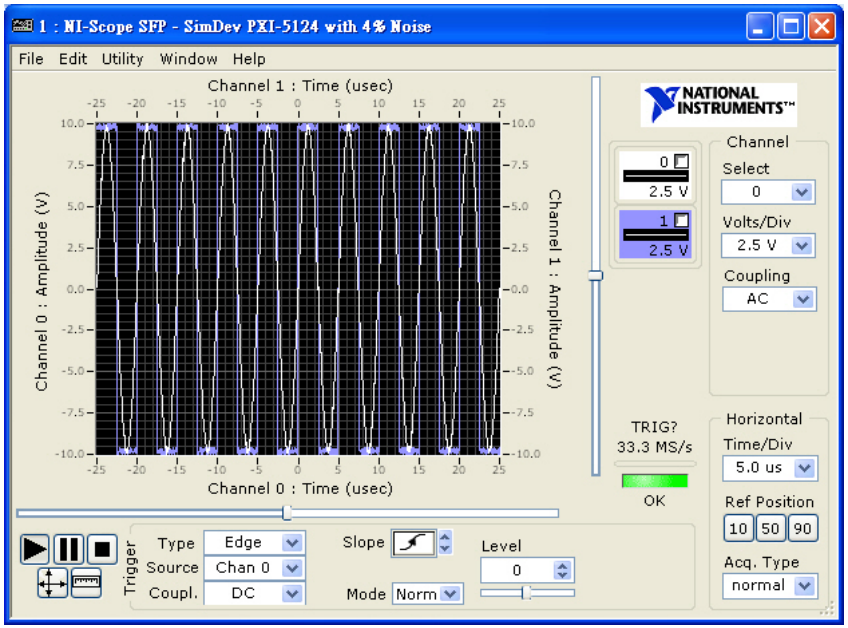

較不理想的介面設計



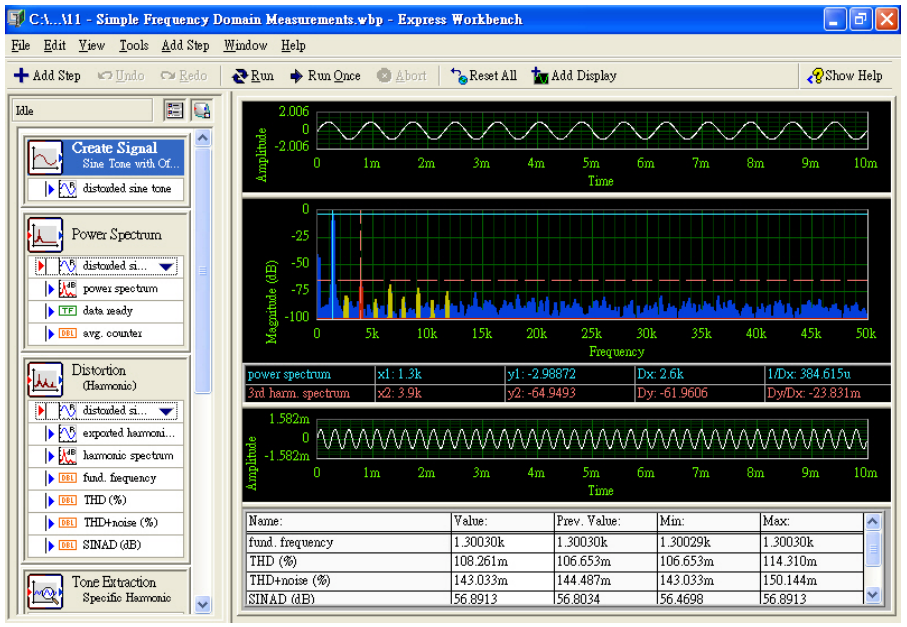

理想的介面設計

# 你也可以建立專業的人機介面

- Many powerful software are based on LabVIEW platform

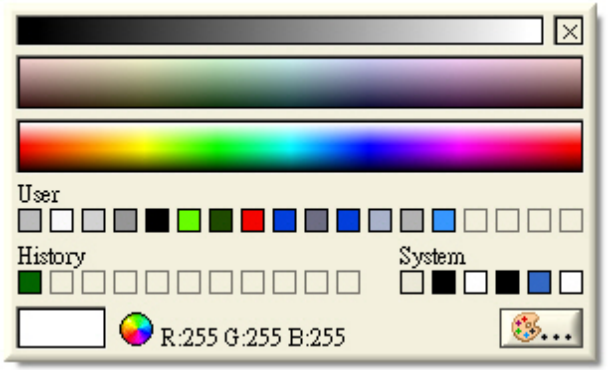



NI-Scope




Signal Express

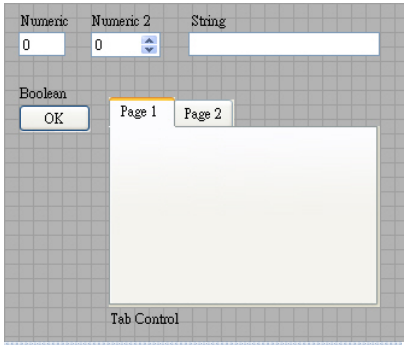
# 使用顏色



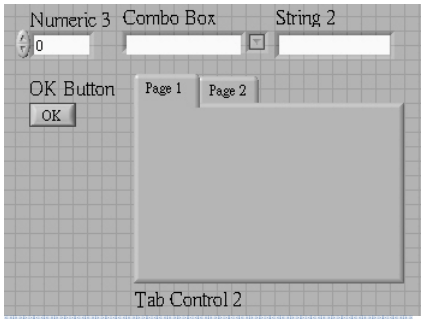

- 使用預設 LabVIEW 顏色。如果顏色無法顯示在此電腦中，LabVIEW 將以另一個最接近的顏色取代。
- 以灰階結構開始。選擇一個或兩個灰階色度並選擇能與背景成良好對比的重點顏色。
- 謹慎地增加重點顏色— 在圖形、停止按鈕或其他部分— 做重要的設定。較小的物件需要較亮的顏色及與較大的物件有明顯的對比。
- 使用空格或對齊群體物件來代替將相配顏色的物件將以群體化。
- 獨立儀器面板、地圖及雜誌等都是可以學習顏色的來源。
- 如果你要讓人機介面上的物件使用系統顏色，請在 Controls»All Controls»Dialog Controls面板來選擇物件。

# 使用系統控制項

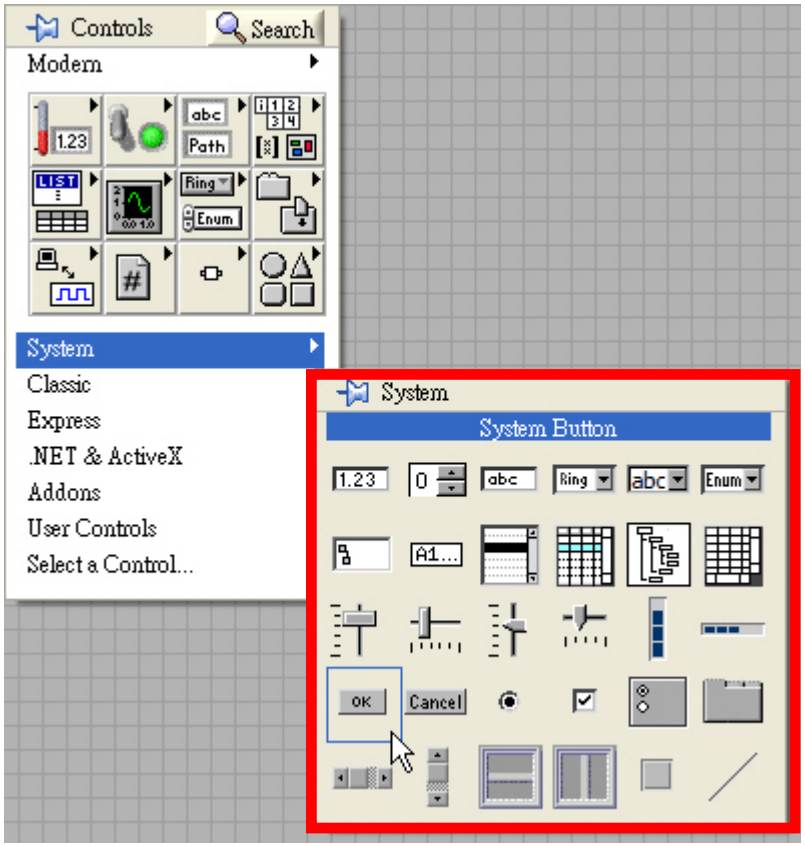
- 「系統」控制元與顯示元
- 可隨作業系統而改變呈現方式
- 建立不像LabVIEW，而像VB或C寫成的儀控程式



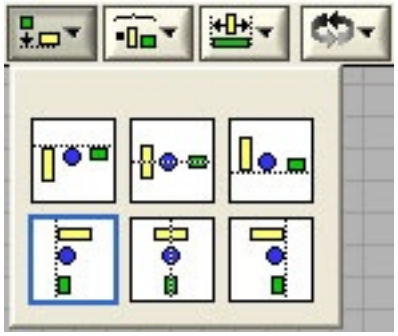

系統控制項



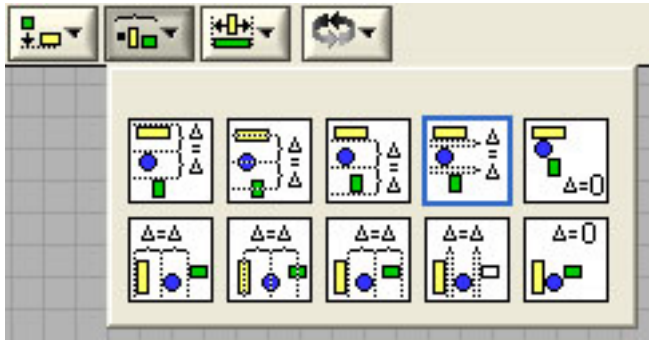

傳統控制項



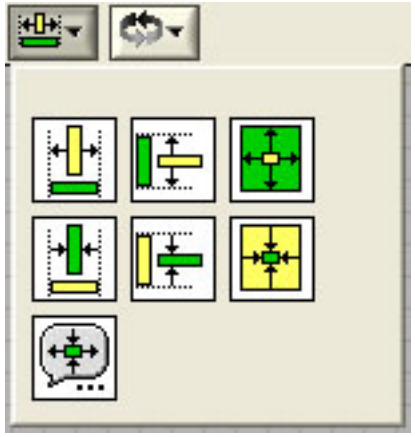
# 使用物件對其工具來讓人機介面更整齊



Align表單



Distribution表單

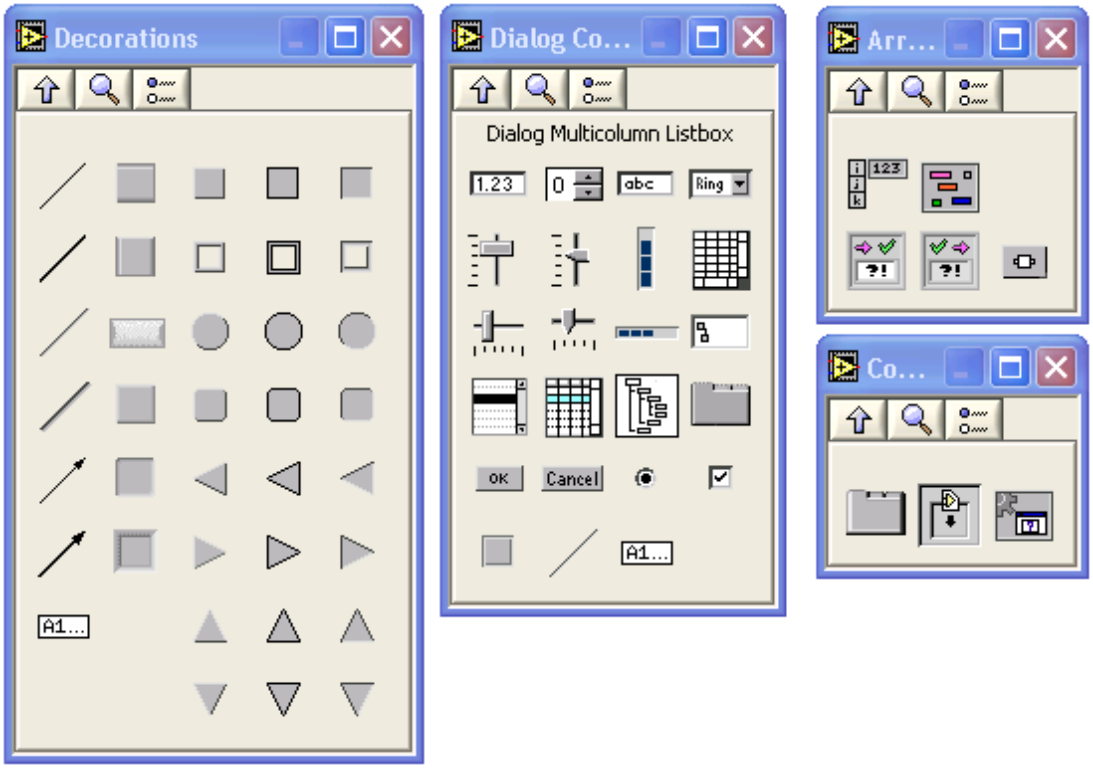


Resize表單


Group表單

# 使用者介面提示及工具

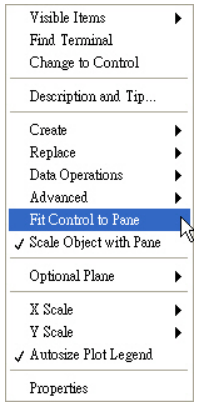
- 對話盒控制
- 標籤控制
- 自動重新調整人機界面上的物件
- 裝飾物件
- 目錄



# 配合視窗自動調整物件的尺寸

- 在人機介面按滑鼠右鍵，並選擇「**Scale Object With Pane**」，就可以人機介面上的物件自動隨視窗大小而調整尺寸


- 在人機介面按滑鼠右鍵，並選擇「**Fit Control to Pane**」，讓該物件佔滿整個視窗

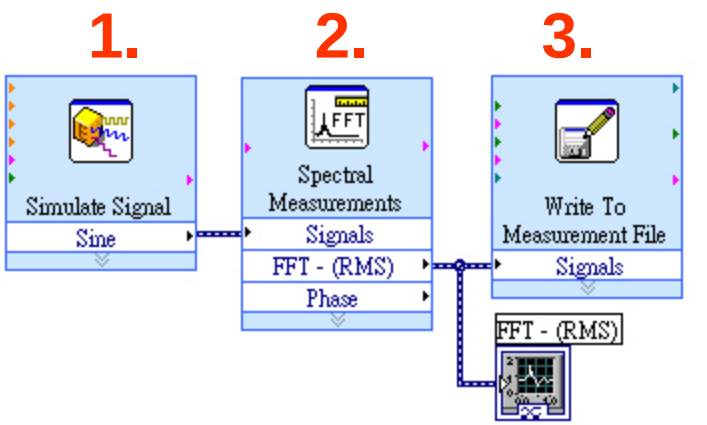


# 練習10.1 – 練習編排適當的人機介面

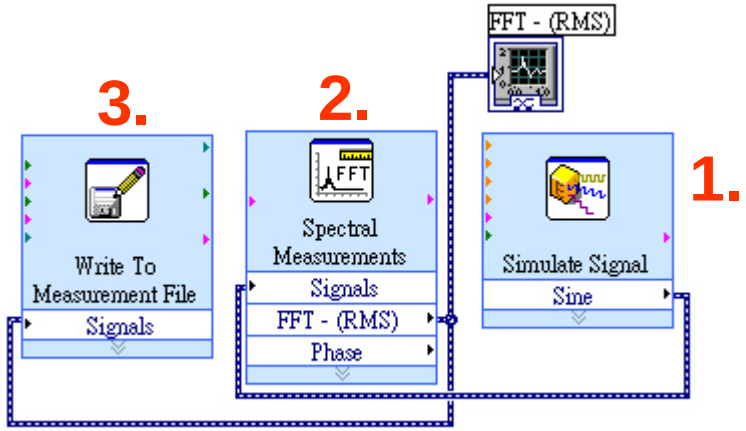
- 將人機介面上的物件與以妥善的排列或分隔，使得VI的人機介面變的更容易使用。
  - 重新改變尺寸、重新組織、及重新排列人機介面上的物件，使得VI的人機介面變的更容易使用。設定圖表使得其尺寸配合人機介面的大小。
  - 請開啟檔案：「<CD>\Ch10\Scope Panel.vi」
  - 開始進行工程吧……..!

# 程式區的佈局

- 資料流的流向並不限於由上至下、由左至右。但是位了閱讀習慣，多數人的程式寫法仍會遵循由上至下、由左至右。而這樣也是較為多數人所習慣的程式撰寫方式



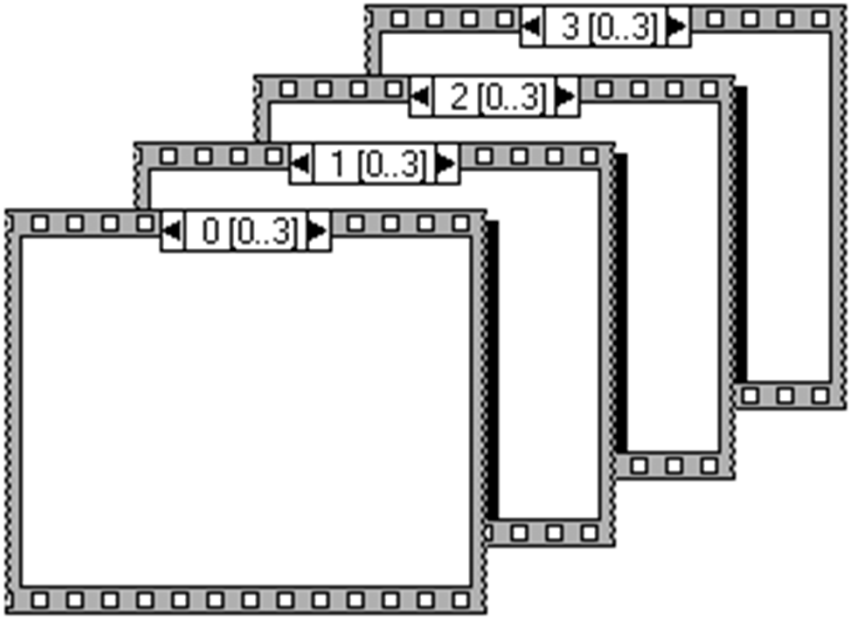

較容易閱讀的程式碼



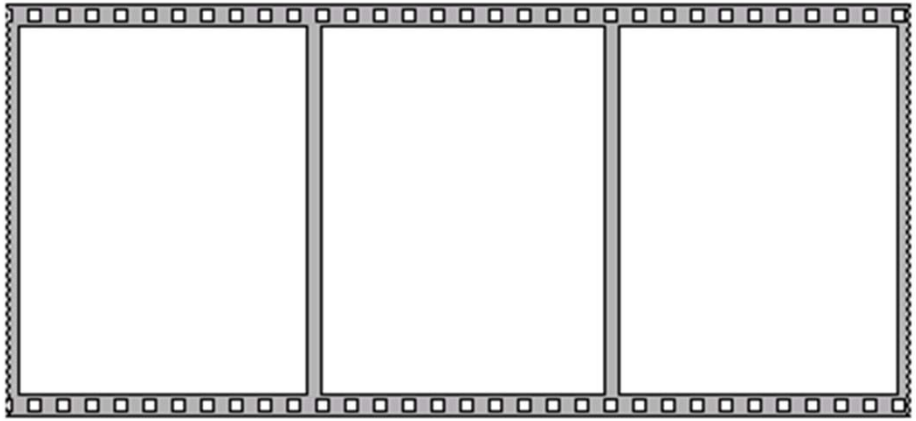

不易閱讀的程式碼

# 序列結構 (Sequence Structure)

- 在撰寫較多元件的程式時，可以使用序列結構來減少程式區的使用面積，讓程式更容易閱讀。此外，依序進行的結構也可以讓人更了解資料的處理順序。

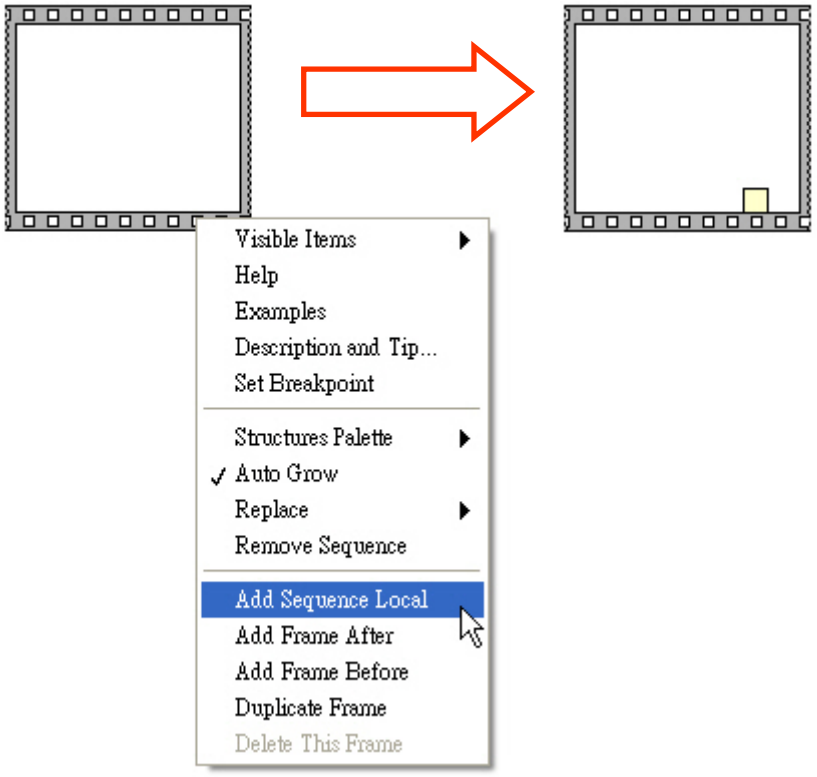


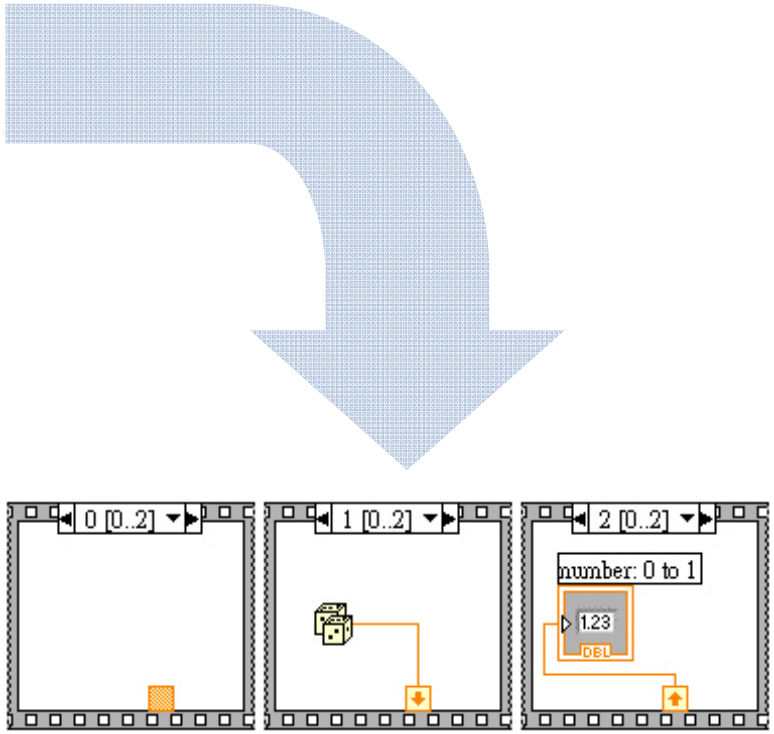

Sequence Structure



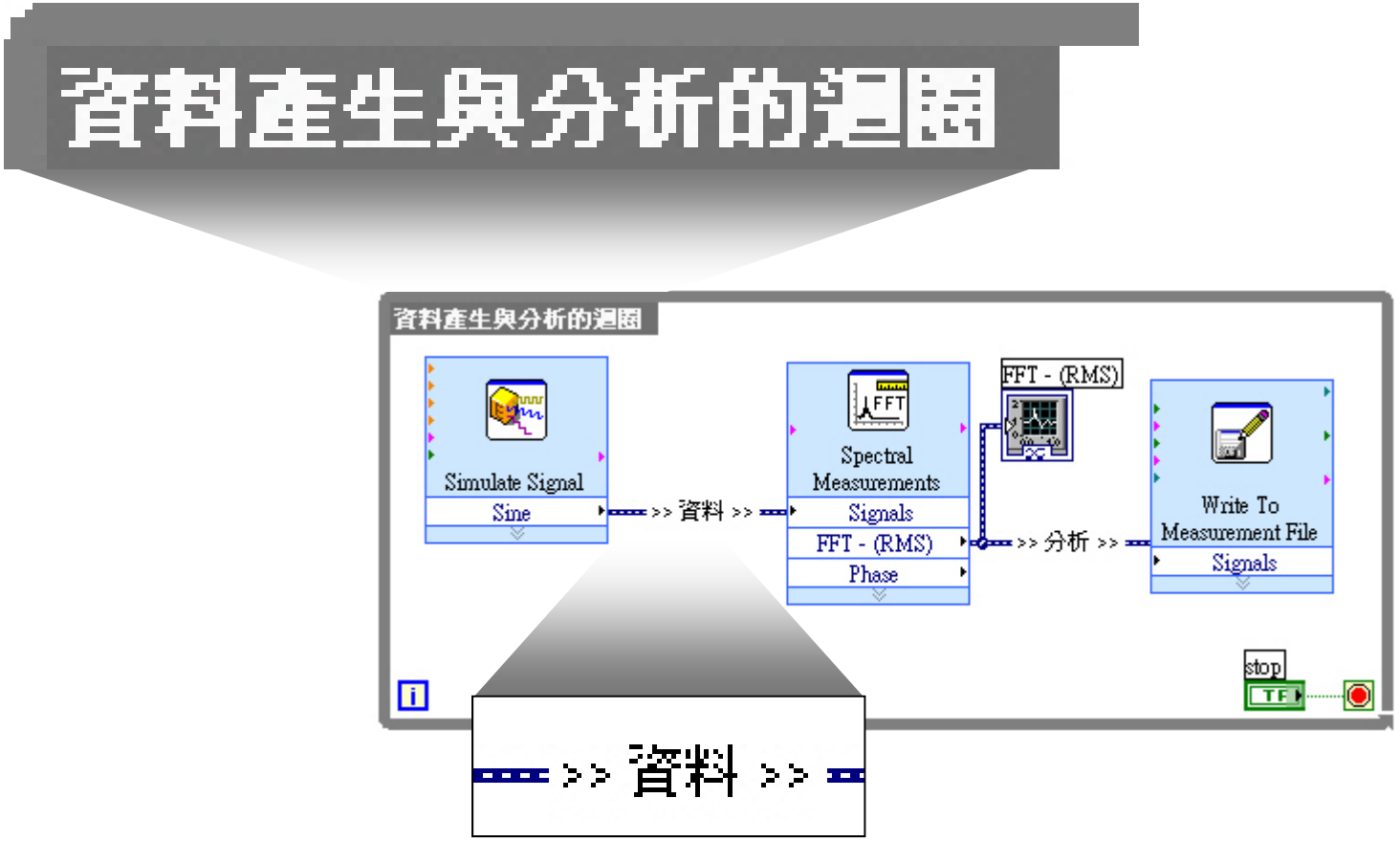
Flat Sequence Structure

# 序列區域變數


Visible Items
Help
Examples
Description and Tip...
Set Breakpoint
Structures Palette
Auto Grow
Replace
Remove Sequence
Add Sequence Local
Add Frame After
Add Frame Before
Duplicate Frame
Delete This Frame


0 [0..2]
1 [0..2]
2 [0..2]
number: 0 to 1
1.23
DBL

# 在程式區寫下註解以方便以後閱讀



# 加入註解的對策

- 在程式方塊圖中使用註解來解釋程式碼的功能。程式方塊圖中的標準註解為具有黃色背景的 Free Labels。
- 忽略較明顯易懂函式的註解。
- 對連線標記所傳遞資料的功能。這通常對於來自位移暫存器中的資料來說，相當有用。
- 對結構標記來指定說明結構的功能。
- 對常數標記來指定說明常數的特性。
- 使用註解來說明使用在程式方塊圖中的演算法。如果你從其他書上或參考文件上取得演算法，請提供參考資訊。

# LabVIEW 連線技術

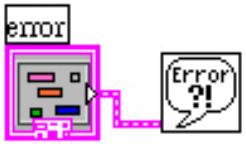
- 避免在程式方塊圖物件的下方置放任何連線，如subVIs或結構
- 盡量避免將連線轉彎，並保持連線在最短的長度。避免產生較長、複雜、容易使人混淆的連線
- 刪除任何多餘線段以保持程式方塊圖整潔
- 當可以直接用連線傳遞資料時，避免區域變數的使用。每一個讀取資料的區域變數將複製使用一個資料
- 當連線本身的資料並不在此結構中使用時，試著不要在不同結構中傳遞連線
- 平均地在不同線段之間保持同樣的距離
- 直接連線至通道及位移暫存器上，不要在結構邊界下方連線
- 直接連線到接頭上，即連線的接頭必須在程式方塊圖中明顯地顯示出來

# 練習10.2 – 重整程式區

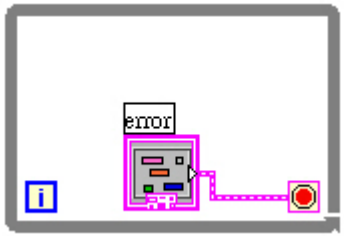
1. 開啟程式：「<CD>\Ch10\BadWiring.vi」，嘗試執行此程式
2. 這個VI將執行基本濾波以及顯示波形的相位
3. 停止此VI並開啟程式方塊圖
4. 整理程式區的程式，讓它變的更容易閱讀。
    1. 由左而右、由上至下
    2. 對齊函數
    3. 將連線的彎度最小化
    4. 點選兩次或三次較差連接的線段以發現相關的接頭
    5. 增加空間以避免混淆
    6. 使用「Clean Up Wire」來自動繞線
5. 儲存檔案

# 錯誤處理

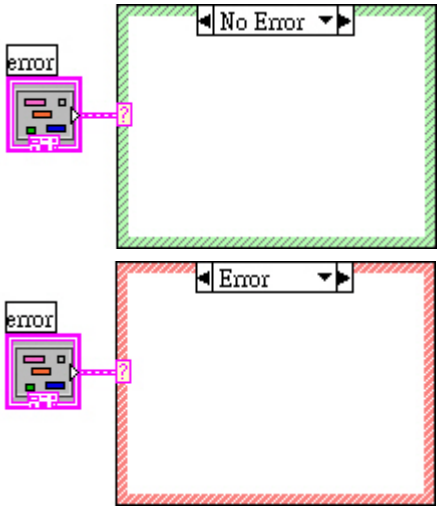
## 錯誤處理VI




## 使用 While Loops 來做錯誤處理

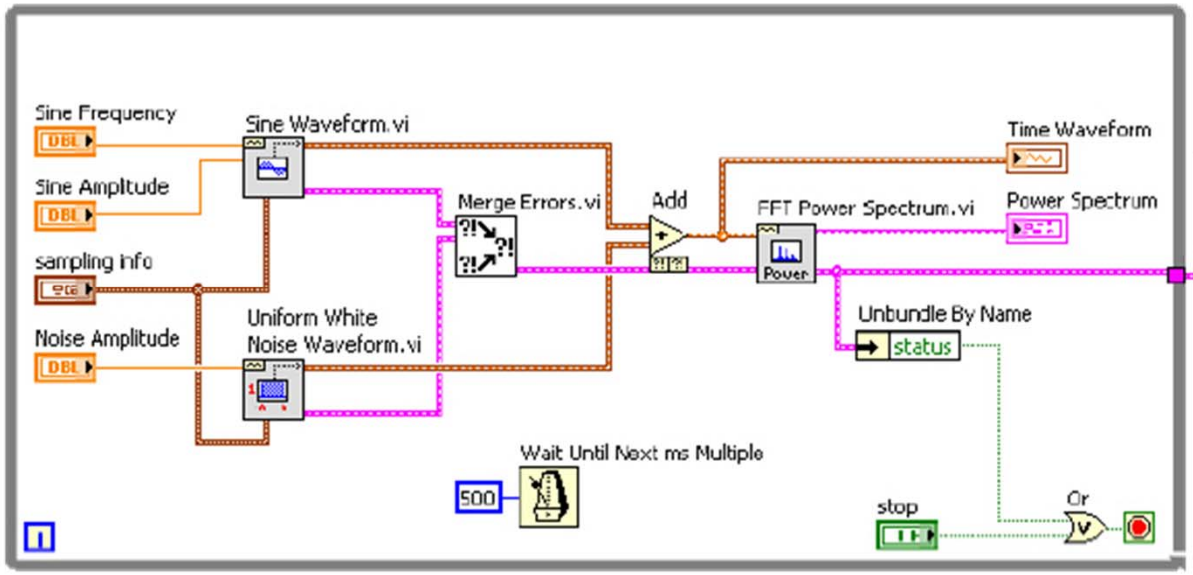


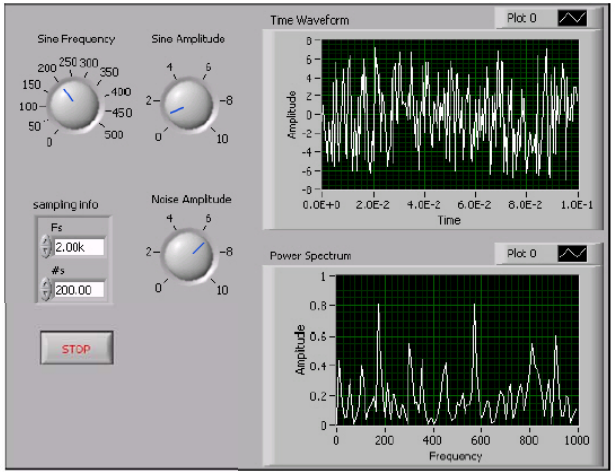

## 使用 Case 來做錯誤處理



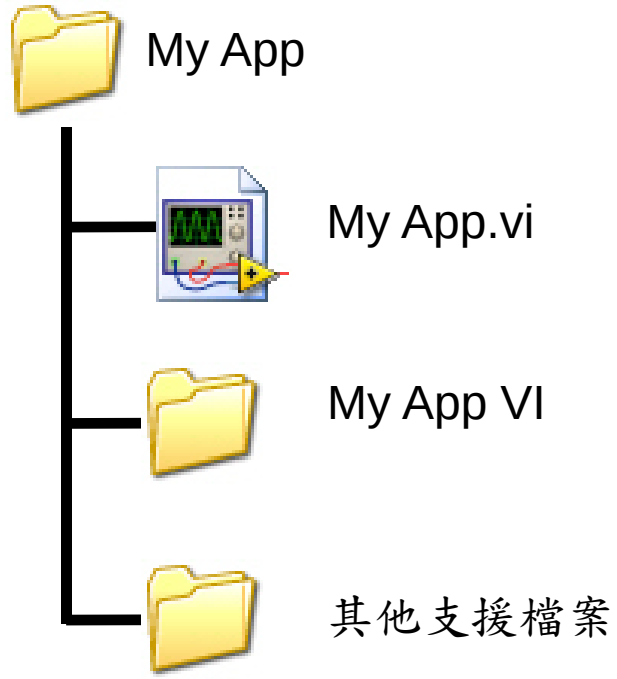
# 練習10.3 – 練習使用錯誤處理技術

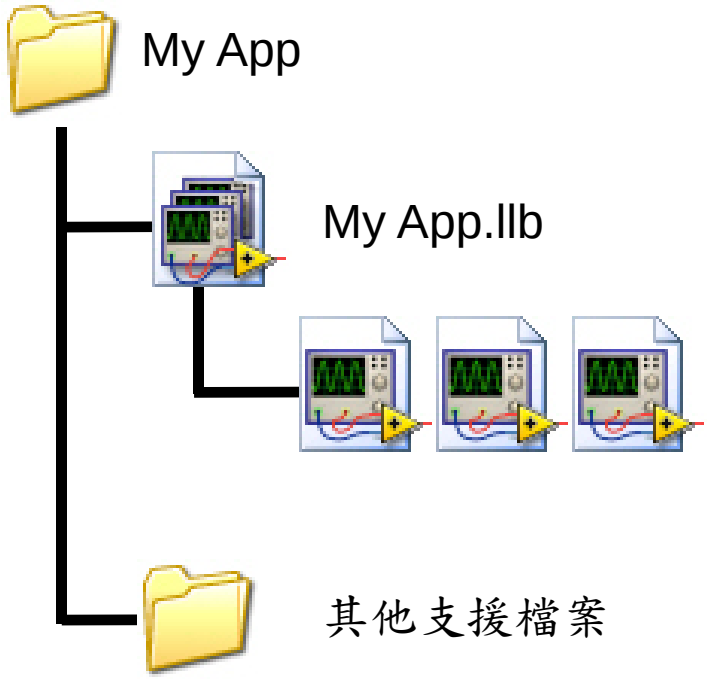
- 寫出下面的VI程式



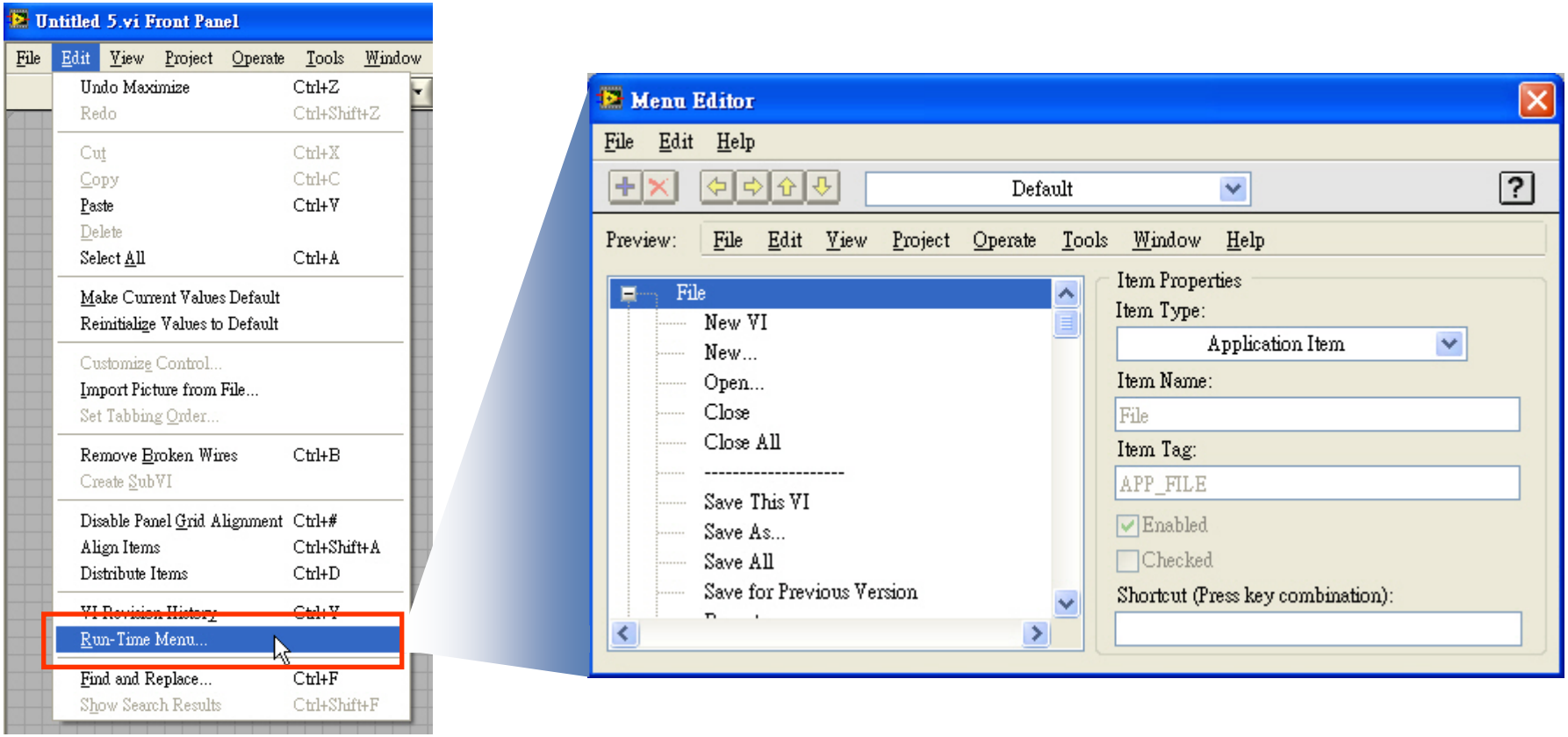



# 階層檔案組織



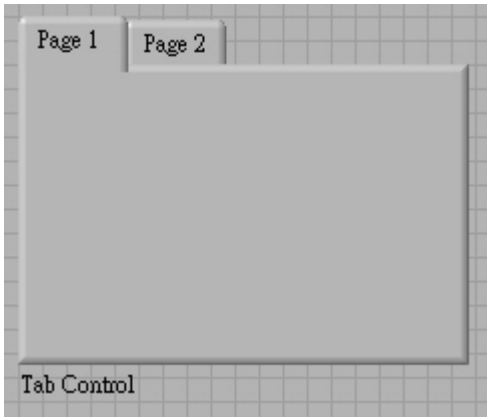
# LabVIEW Run-Time 目錄




注意：客製化的目錄只有當VI執行時才會出現。

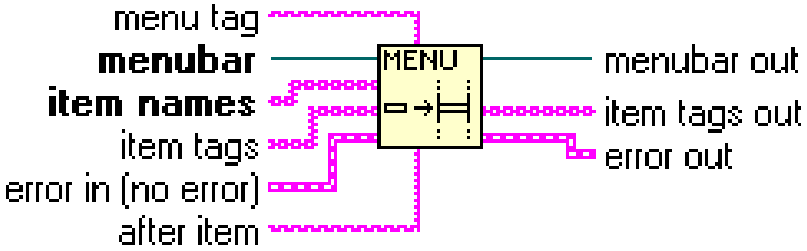
# 練習10.4 – 美化介面

- 開啟練習10.3檔案，加入以下功能：
  - LabVIEW Run-Time 目錄
  - Tab控制項

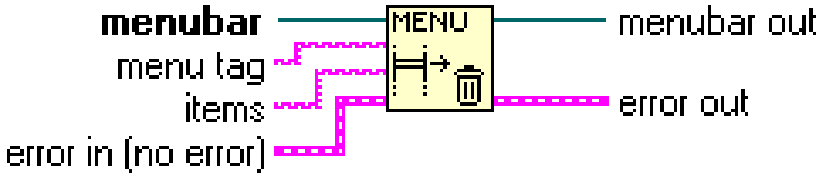


# 建立動態的Run-Time目錄

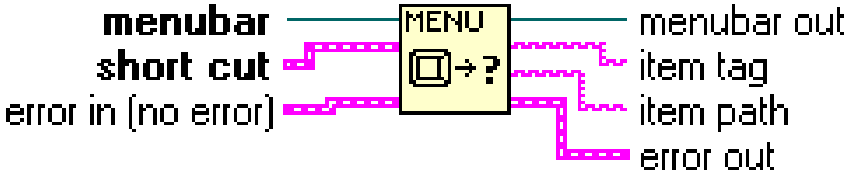
## Insert Menu Items



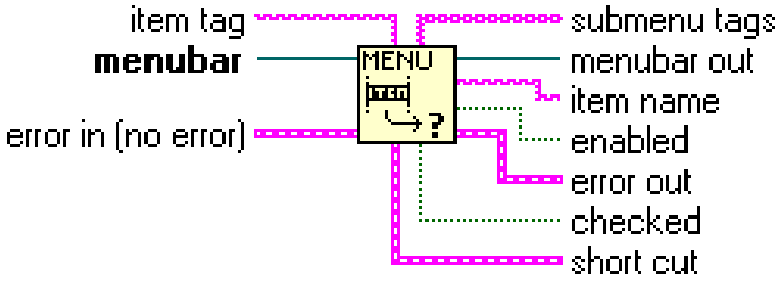

## Delete Menu Items



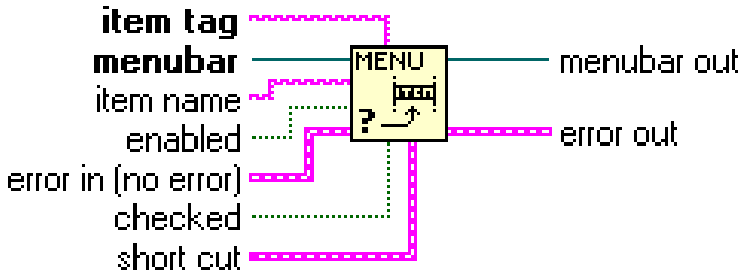

## Get Menu Shortcut Info




## Get Menu Item Info



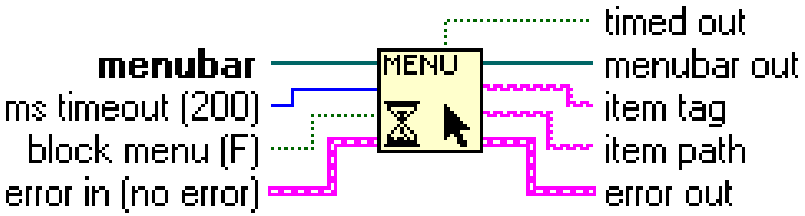

## Set Menu Item Info



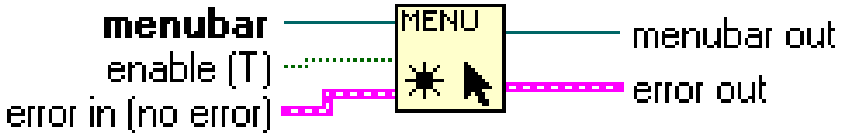
# 處理目錄的選擇

- 使用「**Current VI's Menubar**」來與當前的VI建立通道，並取得refnum
- 「Get Menu Selection」會回傳最後一次滑鼠在目錄所選擇的item tag
- 「Get Menu Selection」使run-time 目錄在它被 Get Menu Selection 函式所停止之後，開始啟動


Current VI's Menubar



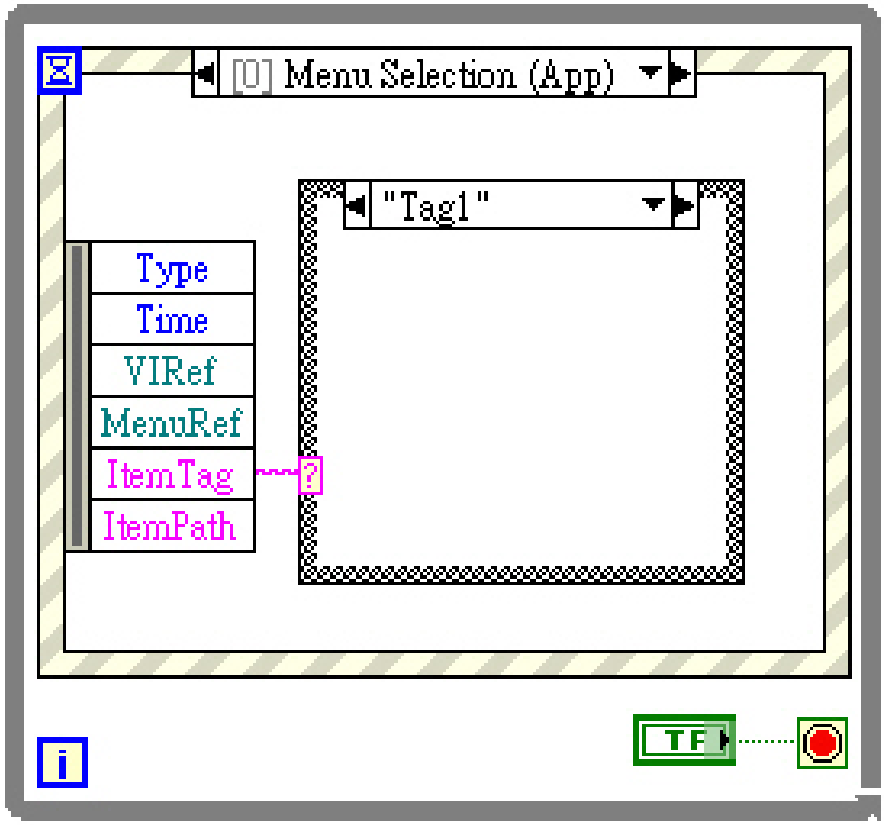

Get Menu Selection




Enable Menu Tracking

關於目錄的觸發，可以使用Event Structure來實現

# 使用Event Structure來讀取使用者的選擇




- 使用Event Structure是比較節省系統資源的作法，而且不會漏失使用者的人機介面動作事件

# 本章重點回顧

- 使用「系統控制項」而不要使用「傳統控制項」會讓你的程式看起來較不像是由LabVIEW所寫成的程式。
- 適當地在程式區加入註解
- 適當使用顏色來做區格或是介面一致性
- 注意字型、字體大小的一致性
- 人機介面的元件數量與呈現方式要特別注意：動線分明、輸入與輸出有區隔，不要混合放置
- 你可以產生一個自訂的run-time menu，並使用Event Structure來讀取使用者的選擇