

利用 Labview 撰寫光譜儀控制程式注意事項

2010/01/19 莊卓穎

目的：使用 PAT-001 控制光譜儀的光柵動作，量測特定波段的光譜。

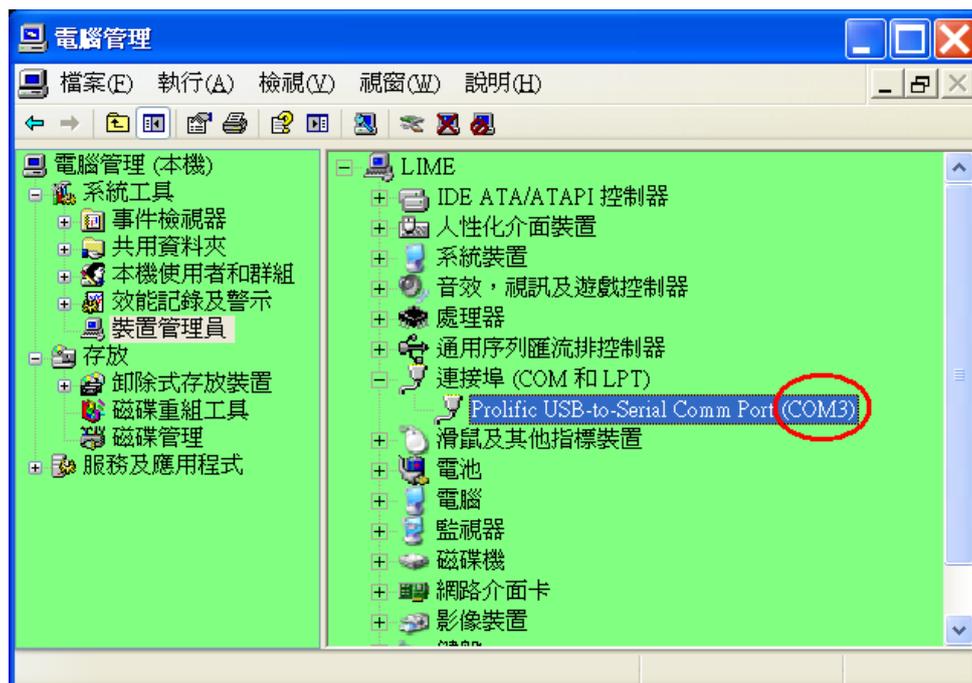
LabView NI-VISA 簡介： 不同於實驗室程式以往使用的 GPIB 物件，本程式採用 VISA (Virtual Instrument Software Architecture)物件，VISA 可由 GPIB 或 Serial 序列埠(例如 RS232)操控儀器，不受介面限制，使用上更廣。(詳細內容請參考實驗室「LabView NI-VISA 及 VISA 物件安裝與使用介紹」說明文件)

PAT-001 簡介：(詳情請見「安裝 PAT-001」說明文件)

1. PAT-001 要透過軟體更改參數 (Baud rate、Run current、Stop current 等等)。
2. 讀取回傳訊息：PAT-001 每下一次指令都要讀回一個 OK 或 NG。
3. 每次重新開啟 PAT-001 電源，下達 Move 指令前，要先執行 H: (回到 Mechanical Origin) 或 R: (設定 Electronic Zero)，讓 PAT 知道坐標位置，才會移動。
4. 機械原點問題：這不是 PAT 控制器的問題，是所有 Sigma-Koki Stage 的問題。不論 Stage 往正轉多少圈，反轉(逆時針)都無法過零點，會被機械原點卡住，這點在撰寫 LabView 程式時需特別注意。

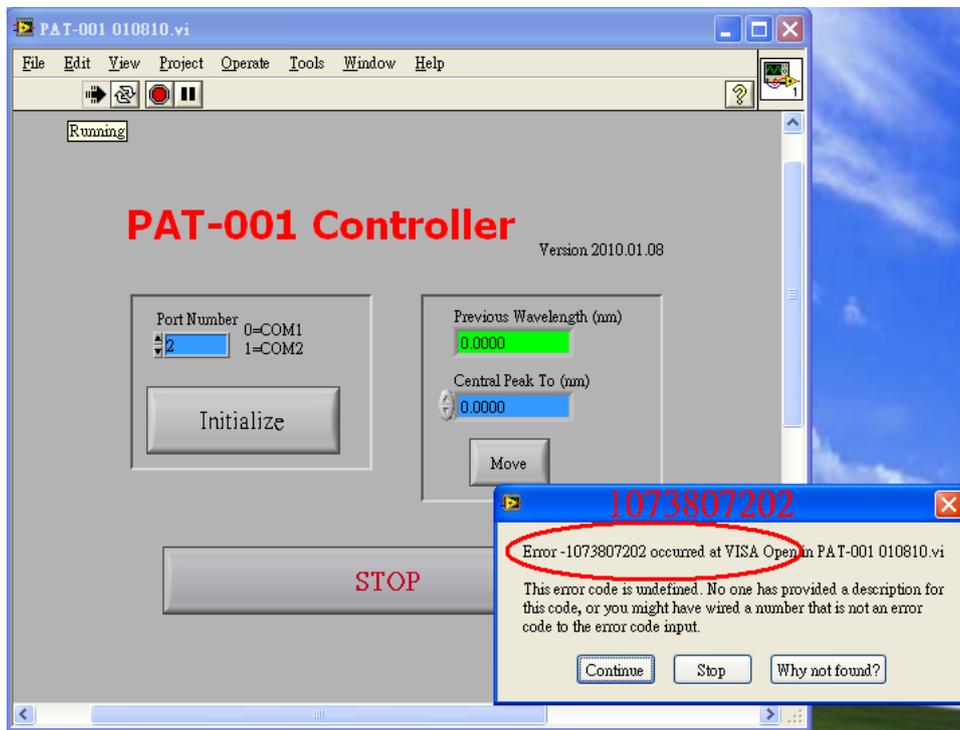
一、硬體設定：

1. 安裝 PL2303 USB to Serial 接線 (USB 轉 RS232) 的 driver，驅動程式位於 Z:\Lab Softwares \RS232\driver。(如果直接用 RS232 連接 PAT-001 與電腦，則可略過此步驟。)
2. 將 PAT-001 一端接到電腦，一端接到 Stage。
3. 在裝置管理員->連接埠，查詢 PAT-001 對應到的 COM port number，注意每個孔對應到的 number 都不同。



二、安裝 LabView NI-VISA :

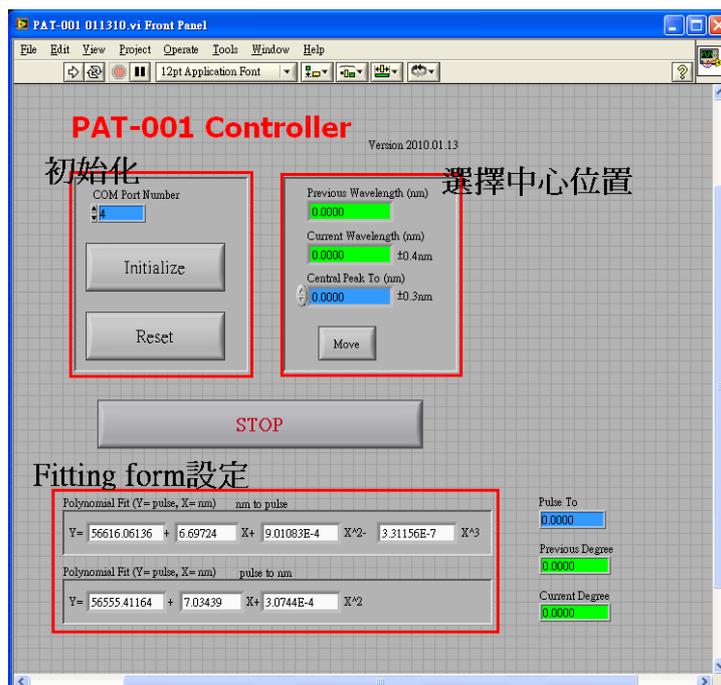
1. LabView 的 RS232 程式會使用到 NI-VISA，一般的 LabView 安裝程序並無安裝。如果在測試 PAT-001 之 LabView 程式時發現錯誤訊息：VISA Error- 1073807202 (0xBFFF009E)，代表沒有安裝 NI-VISA，請由 Data Server (Z:\Lab Softwares\LabView VISA 4.6\visa460full)下載安裝。



3. 自行撰寫的 PAT-001 controller 位於 Data Server (Z:\All Lab Programs Backup) 上，目前最新版本是 PAT-001 011310。

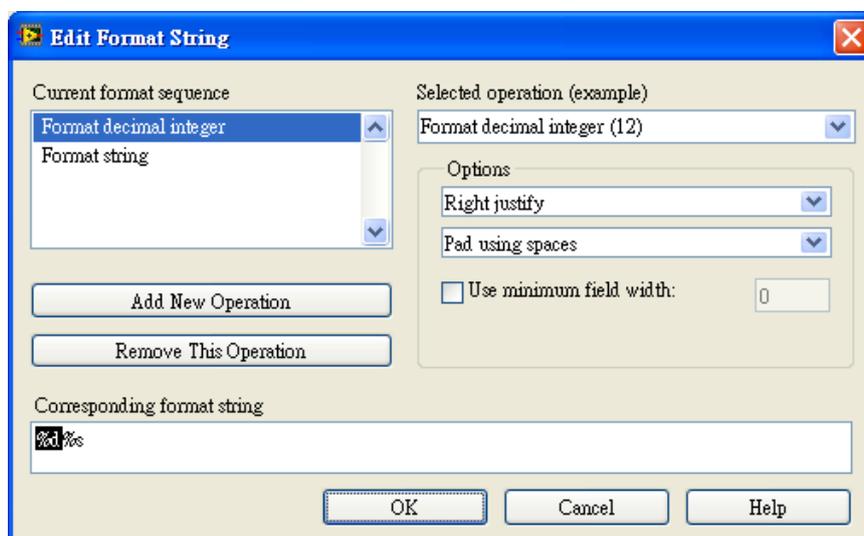
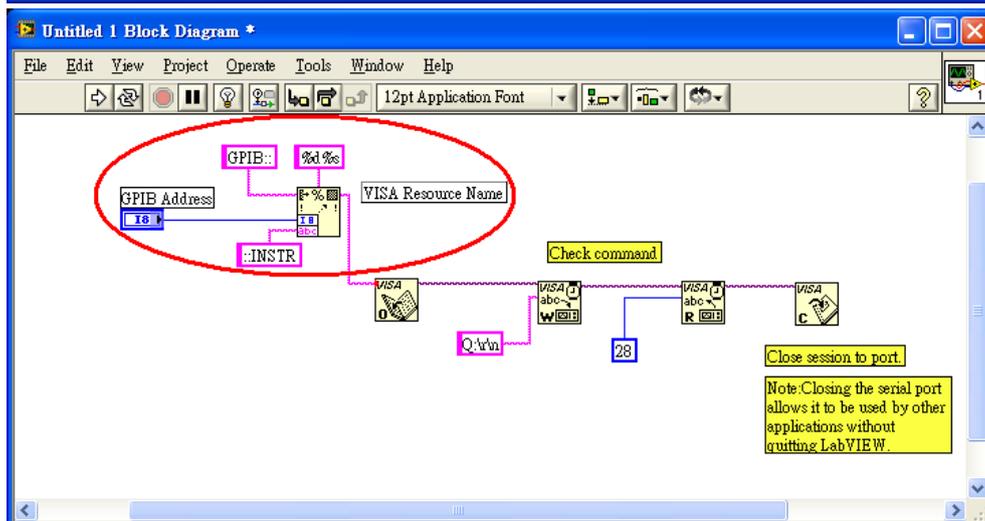
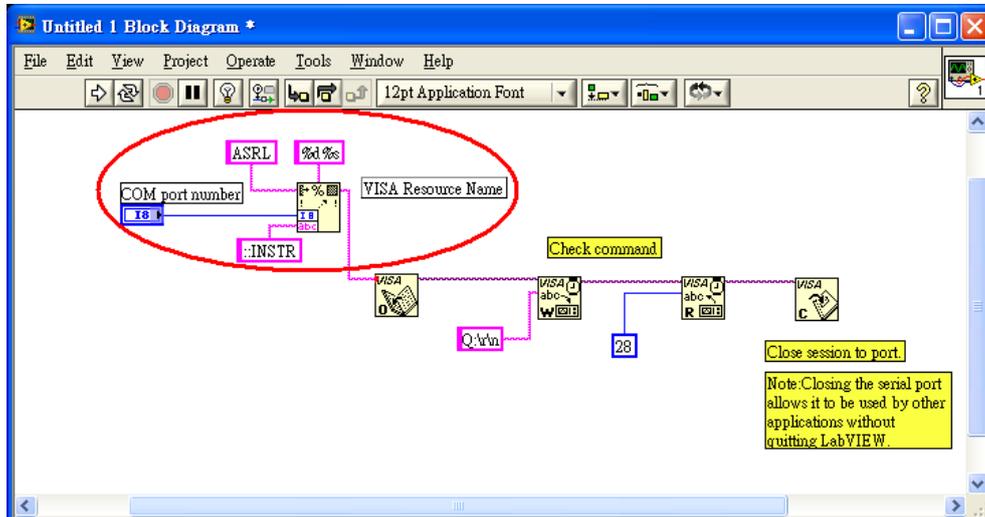
三、LabView 程式概觀：

1. 程式面板可分三部分：初始化、選擇中心位置、Fitting form 設定。



2. 程式基本組成為 VISA Open (開啟 VISA) 、VISA Write (寫入 VISA) 、VISA Read (讀取 VISA) 、VISA Close (關閉 VISA) 。

如下方例圖，每個 VISA 都要給 VISA Resource Name，若是 Serial 就輸入[ASRL][COM port number][:INSTR]，若是 GPIB 就輸入[GPIB::][GPIB Address][:INSTR]。可使用 Format Into String 物件將所有條件變成字串輸入，%d%s 代表字串格式 (%d 代表 Format decimal integer、%s 代表 Format string)，接上所有條件後，按右鍵選 Edit Format String，LabVIEW 會自動判斷格式。

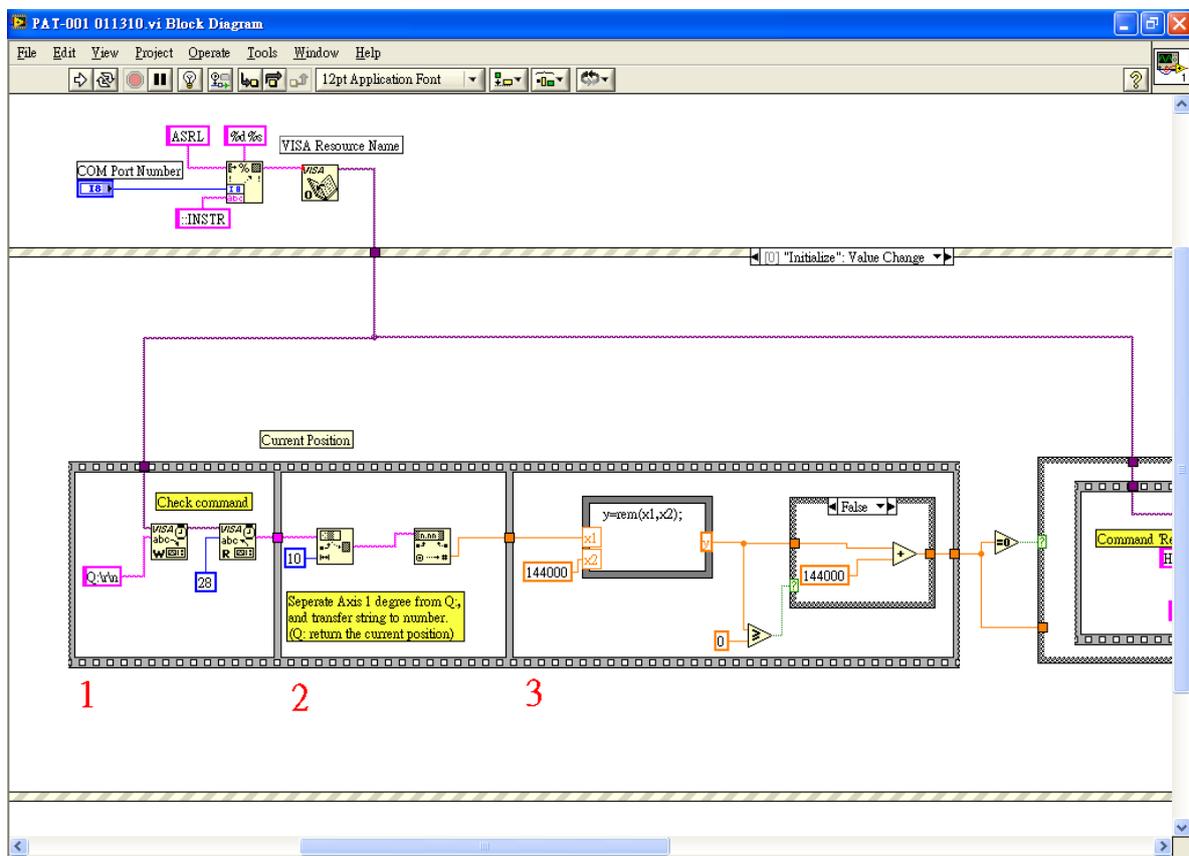


3. 程式最外層是一個 While Loop，而 Loop Condition 連接到面板前的 Stop 按鍵，所以按下 Stop 按鍵整個程式就會停止，此寫法可避免使用上方控制列的 Abort 鍵強制關閉造成的損害。

4. 第二層是一個 Event Structure，優點是可設定事件觸發的條件，例如按下 Move 鍵就只跑 Move Event 裡的程式，不用像 Sequence Structure 一樣從頭跑一遍，也不用像 Case Structure 一樣設定布林條件。

四、個別 Event 說明

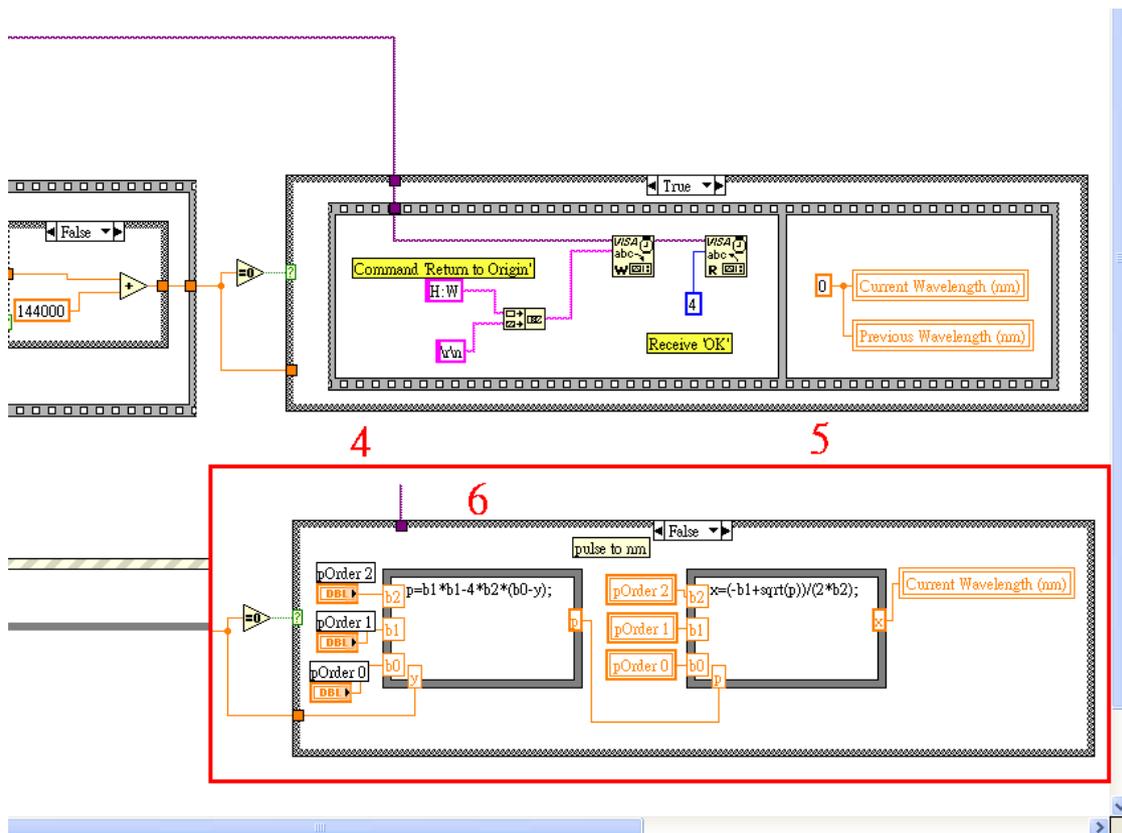
1. **Initialize** 可分為兩部份，一個是讀現在位置，一個是歸零。實驗室用的每台控制器，只要重啟電源就會以當下位置當做電子零點，但 PAT-001 還有一個特點是，重啟電源後要先下歸零或定零點的指令才可下移動指令。所以本事件的概念是，先讀現在位置，若為零則可能為重啟電源所致，因此下指令讓 Stage 回到機械零點；若非零，則表示 PAT-001 未重開，則讀回現在位置，即光柵對應到的波長。



(1) 用 VISA Write 寫入 Q:指令查詢現在位置，每個指令後方都要加上兩個 ASCII 設定：\r (回應輸入字元 Carrier Return)、\n (換行 Line Feed)。用 VISA Read 讀回現在位置。由 Hyperterminal 的測試得知，讀回的字串有 28 個字元，所以 byte count 給定 28。另外，若在 read buffer 的地方 create indicator，則可在面板顯示出讀回的值。

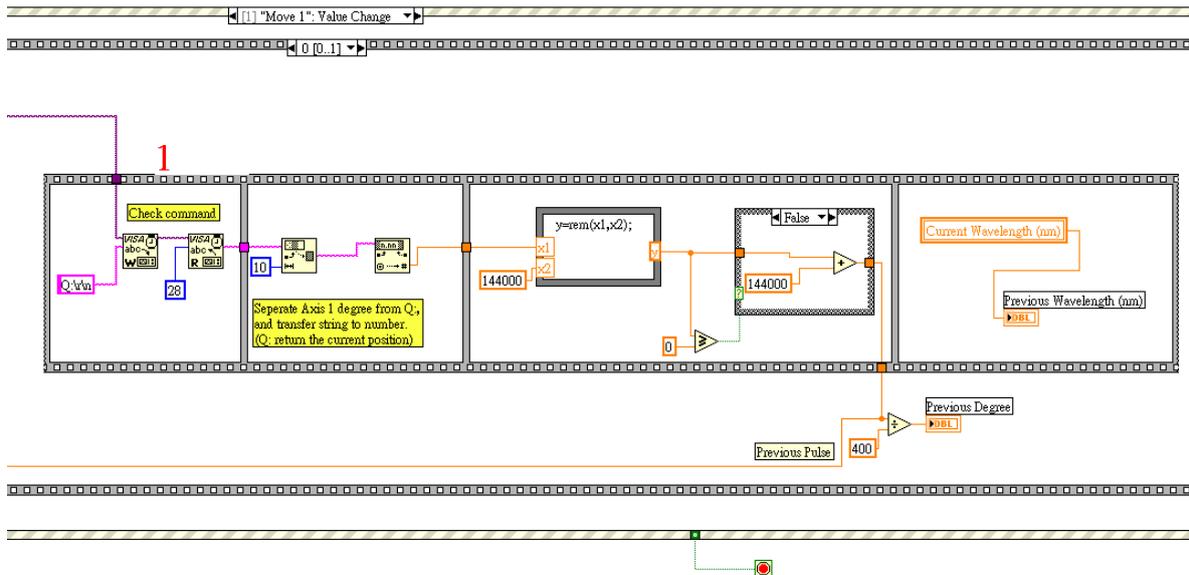
(2) 將讀回來的字串只取前 10 位，即為 Axis 1 的位置，後面英文字母為狀態顯示，目前不需要。將字串變成數值。

(3) 可移動的最小單位是 pulse，Stage 一圈是 144000 pulse，因此這裡取餘數，將數值全部轉換為 0-144000 間，方便後方計算。

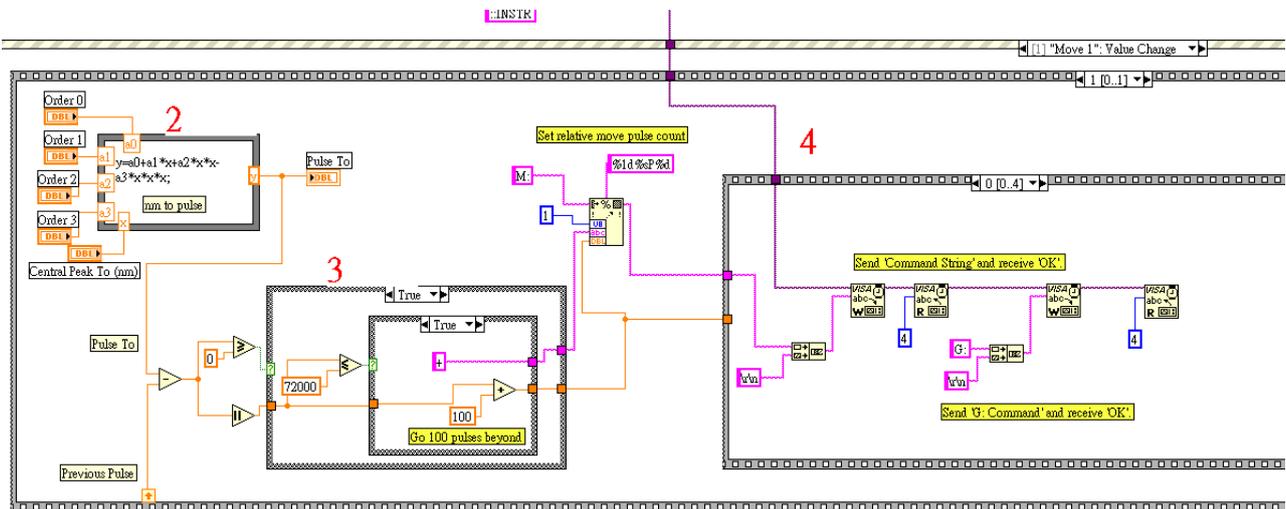


- (4) 若讀回值為零，表示可能為重新啟動，用 H:W 指令歸零，讀回 OK，再將目前值定為零。注意 PAT-001 一定要讀回一個 OK，不然 OK 會一直在 read buffer 內，等下次要讀回位置時，只能讀到 OK，每台控制器設定不同，有的要讀 OK 有的不用，詳見儀器手冊。
- (5) 將現在位置及前一個位置都定為零，此處使用 Local Variable，因為有很多地方都需要用到這個值，有時是 controller 有時是 indicator，用 local variable 方便。
- (6) 若讀回值非零，則使用二次方多項式公式解，換算出 pulse 對應到的波長。此例中三次方 Fitting 較準，只是二次方較好計算，讀值參考用。

2. **Move** event 可分兩大部分，讀回目前值、移動。



(1) 讀回目前值。



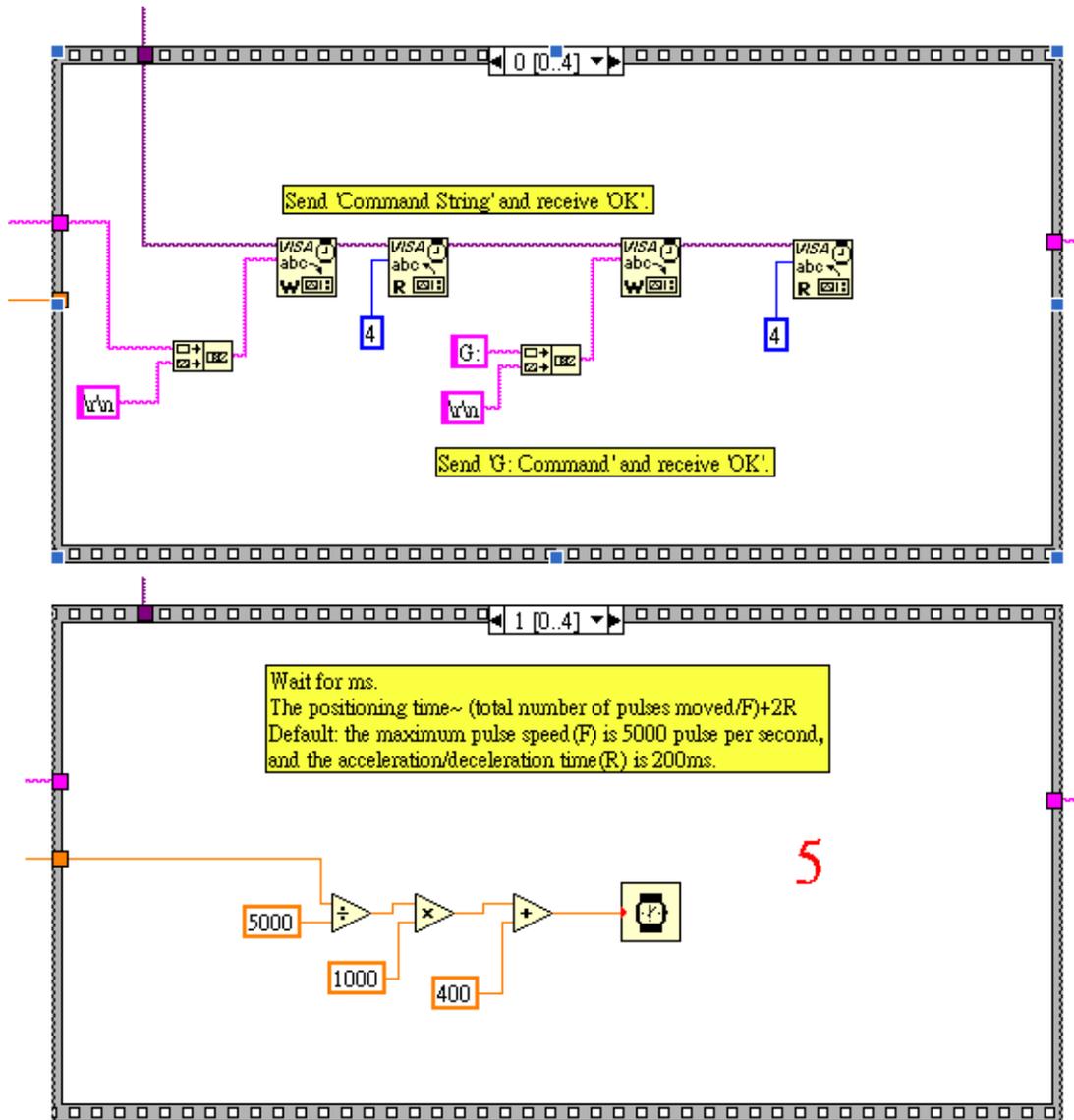
(2) 將面板上輸入的目標位置轉換成 pulse，與目前值相減，即為待移動步數。

(3) 判斷正反轉。

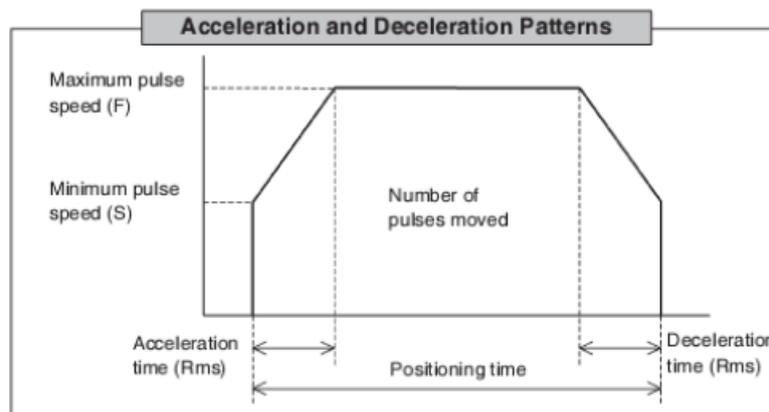
	待移動步數 $p \leq 72000$ (180度)	待移動步數 $p > 72000$ (180度)
待移動步數 $p \geq 0$	正轉 $ p $	反轉 $144000 - p $
待移動步數 $p < 0$	反轉 $ p $	正轉 $144000 - p $

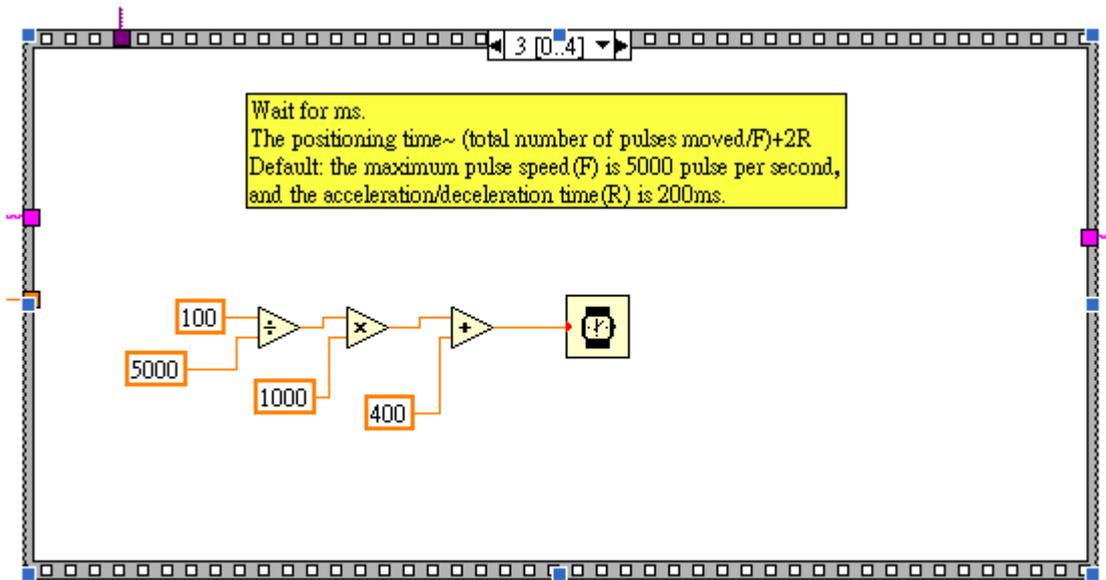
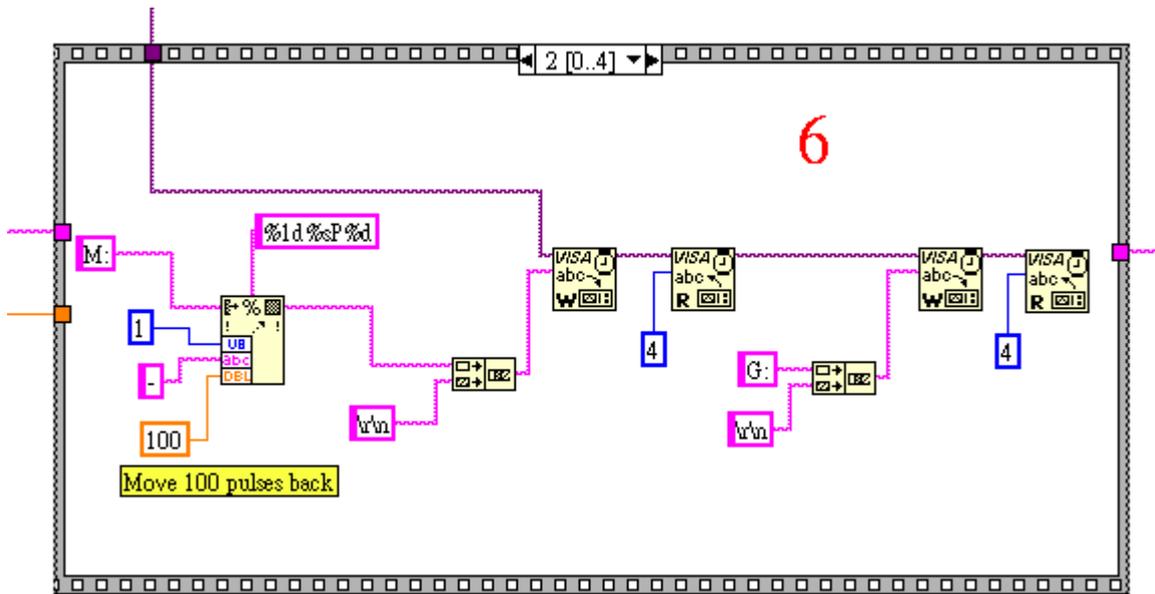
注意為了讓所有的移動都由同一方向驅近，此處全部往正方向先多跑 100 pulses，在接下來的程式會減回來。因為 Stage 是機械齒輪，左右不同方向趨近會有些微誤差，又因自製光譜儀相當精密，儘管左右趨近的誤差只有 1 pulse，但在 Mightex CCD 上呈現的是 2 pixels 的誤差，相當於 0.1 nm。

(4) 將所有移動指令用 Format into String 組成一個字串，例如 M:1+1000 即表示 Axis 1 往正方向移動 1000 pulses，加上\r（回應輸入字元 Carrier Return）及\n（換行 Line Feed）後，一起寫入 VISA。讀回 OK，寫入 G:指令，讀回 OK。

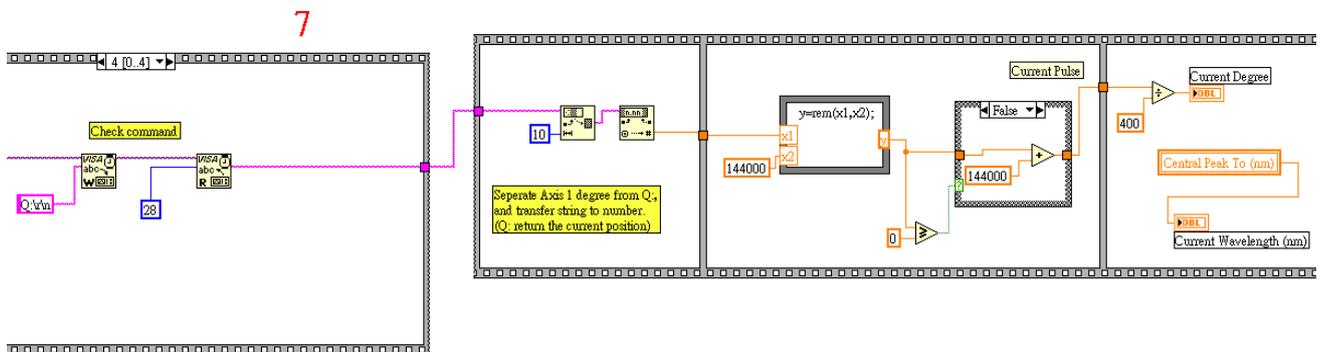


(5) 等待時間計算：PAT 預設每秒移動步數 $F=5000$ ， $S=500$ ，加速減速 R 各 200ms，所以把待移動步數除以 5000 換算成秒，再乘以 1000 換算成 ms，再加上加速減速共 400ms，即約等於等待時間。(參考下圖)



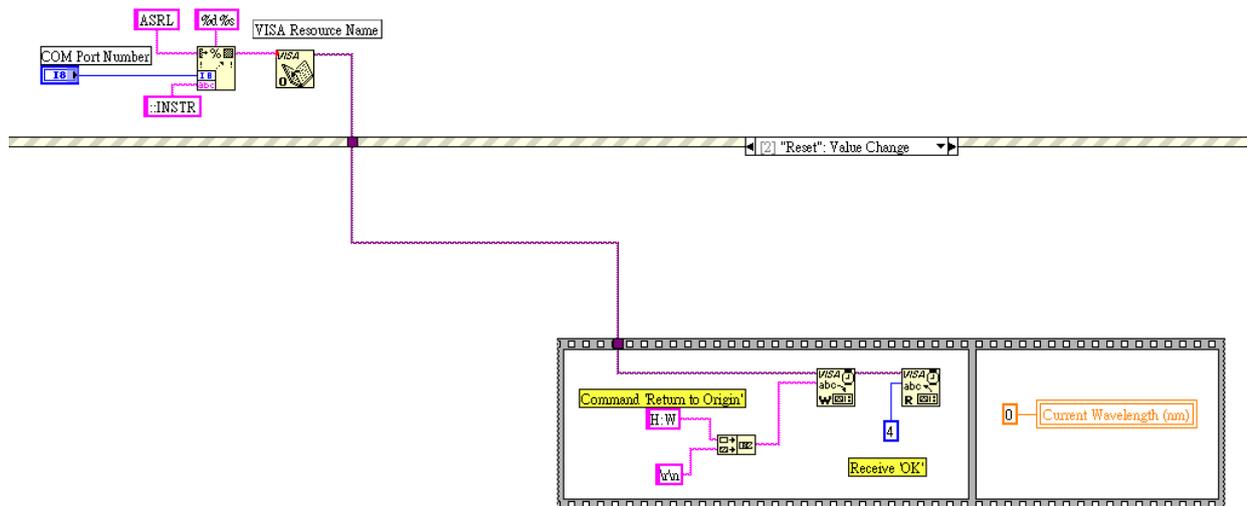


(6) 以同樣方式往回跑 100 pulses，並等待 100 pulses 的移動時間。



(7) 移動到目的地後，寫入 Q:指令，查詢現在位置，並換算成角度。因三次方多項式反函數換算不易，故直接將輸入值寫入 Current Wavelength。

3. **Reset** 即為回到機械原點，故使用 H:W 指令。



4. 按下 **Stop** 鍵即中止整個 While Loop，同時關閉 VISA 以便其他程式使用。沒有使用 VISA Close 的話，在 LabView 關閉之前，會一直佔線，其他程式都無法使用序列埠。

