

AxPgRun.ocx 使用說明

： 什麼是 *AxPgRun.ocx* ？

AxPgRun.ocx 是一個 Windows 的 **ActiveX** 物件。這個物件用來控制 **Pattern Generator(PG1000 及 PG2000 系列)**的運作。使用者可以運用 **Visual C** 或是 **Visual Basic** 等高階電腦語言來撰寫應用程式，並呼叫 **AxPgRun.ocx** 的 **Method** 來控制 **Pattern Generator(簡稱 PG)**，達到待測物與 **PG** 的結合。如果您不是使用 **Visual Basic**，而是其他支援 **ActiveX** 界面的高階語言亦可，但是我們只提供 **Visual Basic** 的 **Sample Code**(可從我們的網站獲得)，其他高階語言請參考該語言的使用說明。注意！**AxPgRun.ocx** 必須與 **PgRun.dll** 搭配使用沒有 **PgRun.dll** 的話，**AxPgRun.ocx** 是不能使用的。以下介紹的使用方法是以 **Visual Basic** 為主。

: 使用流程

1. 請先將 **AxPgRun.ocx** 與 **PgRun.dll** 兩支程式拷貝至 **Windows** 的 **System32** 目錄中。
2. 啓動 **Visual Basic** 並新增物件將 **AxPgRun.ocx** 加入。(由於我們並未提供 **AxPgRun.ocx** 的安裝程式，所以必需手動加入此物件)
3. 將 **AxPgRun** 的物件拖入您的專案中即可使用。

: Property 介紹

AxPgRun.ocx 有以下幾個 **Property**。

FileName Property

設定輸出的波形檔。此波形檔可以用 **PgEditor** 應用程式的 **Wave Editor** 功能來編輯。

Syntax

```
Object.FileName [= File Name]
```

Remarks

波形檔必須先在 **PgEditor** 檢查過，確認沒有問題。

FileType Property

設定輸出波形檔的檔案型態。此波形檔型態目前只支援 **PGW** 的格式。

Syntax

```
Object.FileType [= File Type]
```

```
PGW : 0
```

```
PGV : 1
```

Remarks

因爲目前這個 **AxPgRun** 的版本支援 **PGW** 的格式，所 **FileType** 只能填入 **0**。

VoltPodA Property

VoltPodB Property

VoltPodC Property

VoltPodD Property

VoltPodE Property

設定 **PG** 每個 **Pod** 的輸出電壓。

Syntax

Object.**VoltPodA** [= Number]

Object.**VoltPodB** [= Number]

Object.**VoltPodC** [= Number]

Object.**VoltPodD** [= Number]

Object.**VoltPodE** [= Number]

Pod 電壓值的單位是 milli-Volt (mV)。

Remarks

不同 **PG** 裝置有不同數量的 **Pod**，所以有些 **Pod** 設定值是對某些 **PG** 裝置是無效的。例如，PKPG2016 有兩個 Pod 所以只能用到

VoltPodA 及 VoltPodB。又例如 PG2050 有五個 Pod，所以每個設定值都可以使用。

由於不同機種可設定的電壓值就不同所以，當您設定的電壓值是該裝置無法產生的電壓時，本物件會自動將電壓值調至最接近的值。電壓設定請參考 PG 使用手冊說明。

Frequency Property

設定 **PG** 的主時基頻率。

Syntax

Object.**Frequency** [= Number]

PG 頻率值的單位是 Hz。

Remarks

您可以設定 **PG** 的主時基頻率，如果您想用波形檔所設定的頻率值，只要將此 Property 填入 0 值即可。

FrequencyMode Property

設定 **PG** 的主時基頻率的模式。

Syntax

Object.**FrequencyMode** [= Number]

目前這個物件版本只支援波形檔的內部使用頻率模式。所以必須填入 0。

Remarks

您可以設定 **PG** 的主時基頻率模式，如果您想用波形檔所設定的頻率值，只要將此 Property 填入 0 值即可。新版的物件將可以改變此波形檔的主時基頻率模式。

ID Property

取得 **PG** 裝置的型號。

Syntax

Object.**ID**

PG1020 : &h0A020
PG1050 : &h0A050
PG2016 : &h0B016
PG2116 : &h0B316
PG2020 : &h0C020
PG2050 : &h0C050
PG2120 : &h0C320
PG2150 : &h0C350

Remarks

如果您的 **PC** 同時安裝了多台的 **PG** 裝置就必須用讀取此值來確認目前您取得的 **PG** 裝置是否正確。

ModelName Property

取得 **PG** 裝置名稱。

Syntax

Object.**ModelName**

: Method 介紹

PGRUN.DLL 函式介紹是採用 C 語言的風格。函式版本為 1.0.0.1 以上。

GetHardware Method

尋找並啟動 PG 裝置。

Syntax

```
fRet Object.GetHardware lModel, lSlot
```

GetHardware 有下列幾個參數：

lModel	設定欲取得的裝置 ID(請參考 ID Property 說明)。
lSlot	設定欲取得的裝置順序編號。
fRet	呼叫此 Method 的回應值，此值為 True 代表已經順利找到並啟動 PG 裝置。

Remarks

lSlot 的設定是以裝置的界面來區分(如 USB、EPP 或 PCI 等)。假設您安裝了五台 PG 裝置，一台 PKPG2016、一台 PKPG2116、一台 PG2050 USB 界面及兩台 PG2050 PCI 界面。您欲取得該裝

置得方法如下。

PG2016	AxPgRun.GetHardware &h0B016, 0
PG2116	AxPgRun.GetHardware &h0B316, 1
PG2050USB	AxPgRun.GetHardware &h0C050, 2
PG2050PCI	AxPgRun.GetHardware &h0C050, 0
PG2050PCI	AxPgRun.GetHardware &h0C050, 1

由於有三台 USB 界面的 PG 裝置所以 lSlot 的設定必須從 0 到 2。至於 USB Slot 的順序是依照裝置與 PC 連接的順序。而 PCI 的 lSlot 值則是依照插在 PCI SLOT 的編號。而 EPP 的順序為 Port 378 = Slot 0, Port 278 = Slot 1。

Start Method

開始輸出波形。

Syntax

```
iRet Object.Start
```

Start Method 會回應一個值這個值的定義如下：

0	波形已正常輸出。
1	找不到設定的波形檔。
2	讀取波形檔時發生錯誤。

- 3 檔案型態錯誤或是不支援該檔案型態。
- 4 檔案格式錯誤。
- 5 檔案的版本太舊。
- 6 記憶體不足。
- 其他 嚴重錯誤或不明錯誤。

Stop Method

停止輸出波形。

Syntax

Object.**Stop**

KeyEvent Method

送出一個 **Key Event** 到 PG 裝置。

Syntax

Object.**KeyEvent**

Remarks

當 PG 進入 **Wait Key Event** 時，可以用 **KeyEvent Method** 產生

一個 **Key Event**，來觸發 PG 裝置。

WaitStatus Method

檢查 PG 裝置是否進入 **Wait** 狀態。

Syntax

iRet Object.**WaitStatus**

WaitStatus 的回應值定義如下：

Bit2~0: Active Event

- 0 Key Event
- 1 Event0
- 2 Event1
- 3 Event0 # Event1
- 4 Event2
- 5 Event0 # Event2
- 6 Event1 # Event2
- 7 Event0 # Event1 # Event2

Bit3 Event Invert

- 0 Invert
- 1 Non -Invert

Bit4 Wait Event

- 0 In wait state
- 1 Not in wait state

Remarks

以下是使用 WaitStatus 的 sample。

```
If AxPgRun1.WaitStatus = 0 then      ' Wait Key Event
    AxPgRun1.KeyEvent                ' Send a Key Event
to PG
End if
```

Shutdown Method

關閉 PG 裝置。

Syntax

Object.Shutdown

Remarks

當您不再使用 PG 裝置時，您可以用 Shutdown Method 來關閉 PG 裝置。但是一旦用了 Shutdown Method 以後，如果還要使用該 PG 裝置，就必須再呼叫一次 GetHardware Method 來啟動 PG 裝置。