PGRUN.DLL 使用說明

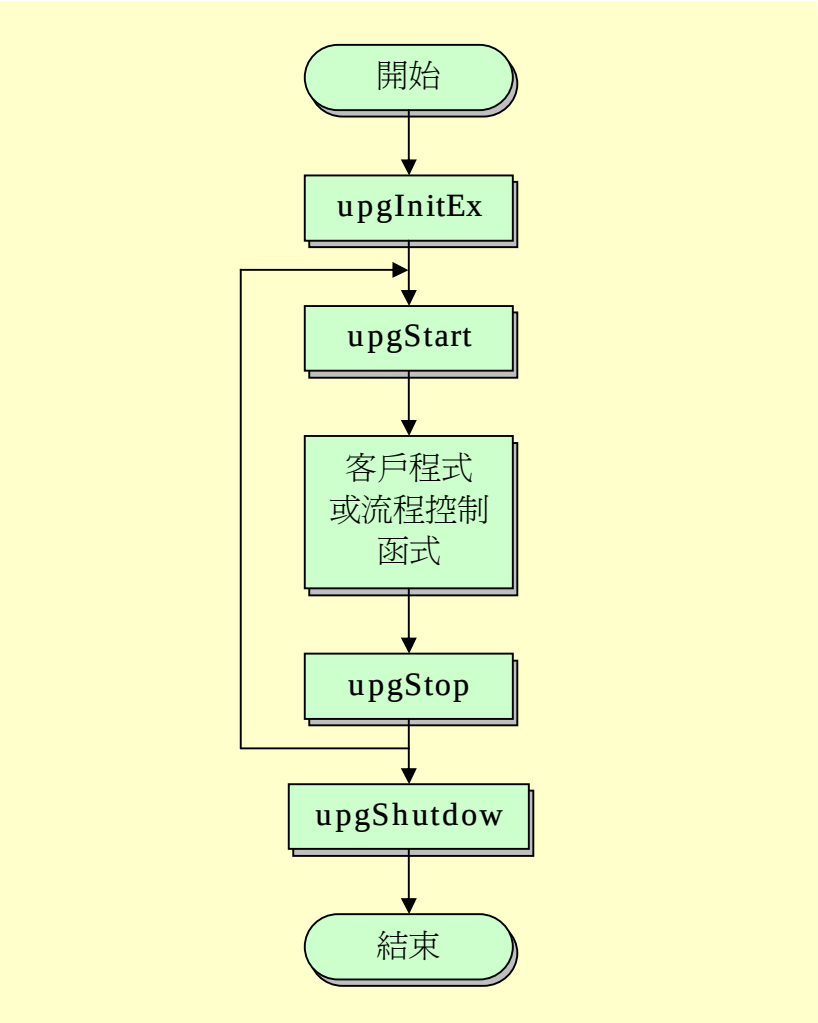
： 什麼是 PGRUN.DLL？

PGRUN.DLL 是一個 Windows 界面的動態連結程式庫，這個程式庫提供了 Pattern Generator(PG1000 及 PG2000 系列)的驅動副程式。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，並與 PGRUN.DLL 連結來控制 Pattern Generator(簡稱 PG)，達到待測物與 PG 的結合。如果您不是使用 Visual C 或是 Visual Basic，而是其他支援 DLL 界面的高階語言亦可，但是我們只提供 Visual C 及 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。

PGRUN.DLL 總共包括了下列函式:

```
BOOL upgInitEx( int iModel, int iSlot );
int upgStart( LPCSTR szFileName, int iFileType,
              LPPGSI lppgsi );
BOOL upgStop();
BOOL upgKeyEvent();
unsigned int upgGetStatus();
BOOL upgShutdown();
```

： 使用流程



： 函式介紹

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

upgInitEx

啓動 Pattern Generator。

```
BOOL upgInitEx(int iModel, int iSlot);
```

參數說明：

iModel

PG 裝置 ID, 請填入零做初始化的動作。

iSlot

USB slot 或是 PCI slot 或是 LPT slot,端看您使用何種介面與 PC 連結，也填入零做出初始化。

回應說明：

如果 PG 正常啓動，這個功能將回應一個非零的值(TRUE)。如果無法正常啓動 PG、找不到 PG 或是參數設定錯誤，這個功能將回應一個零的值(FALSE)。

附加說明：

使用 PG 之前一定要先呼叫 upgInitEx 這個功能來啓動。在還沒呼叫 upgInitEx 此函式之前，所有 PGRUN.DLL 的其他

函式都無法使用。這個函式不只有啓動的作用，他還兼具有檢查 PG 是否存在的意義。一旦 PG 啓動以後，就可以使用 upgStart 來送出波形。注意！這個函式只要呼叫一次。

upgStart

將波形檔案載入 PG 中，並輸出波形。

```
int upgStart(
  LPCSTR szFileName,   // file name
  int iFileType,       // file type
  LPPGSI lppgsi        // system information
);
```

參數說明:

szFileName

波形檔名稱。載入波形是以檔案格式的方式，將該波形檔經由本函式解譯，並傳輸到 PG 的記憶體。

iFileType

波形檔案的格式。波形檔的格式有兩種，一種叫 PGW、另一種叫 PGV。PGW 是屬於 Binary 的檔案格式，而 PGV 則是一種文字向量的格式。當 iFileType 爲 0 時，代表檔案格式選用 PGW，iFileType 爲 1 時，代表選擇 PGV 的格式。注意！不管是 PGW 檔案或是 PGV 檔案都必須由 PGEditor 這個應用程式所產生，且

需經由 PGEditor 的波形檢查功能來確認該波形檔是可用的。

lppgsi

保留(需填 NULL)。

回應說明:

如果回應值爲 0，代表波形載入成功且正常輸出。如果回應非 0 的值則代表執行此函式產生錯誤。錯誤碼如下:

1 ： 找不到檔案。

2 ： 檔案讀取錯誤。

3 ： 檔案型態錯誤。也就是說 iFileType 的值錯誤。

4 ： 檔案格式錯誤。波形檔案的內容與設定的檔案型態不同。

5 ： 檔案格式版本太舊。PGW 的檔案格式有版本的區別。

6 ： 記憶體不足。記憶體大小是根據檔案大小及波形長度而定。

upgStop

停止波形輸出。

```
BOOL upgStop();
```

回應說明:

如果 PG 正常停止則此函式會回應非零的值(TRUE)，否則回應零(FALSE)。

附加說明：

在 upgStart 和 upgInitEx 還沒執行前，這個函式是沒有作用的。

upgKeyEvent

送一個 Key Event 到 PG。

```
BOOL upgKeEvent();
```

回應說明：

如果 PG 正常送出 Key Event 則此函式回應非 0 的值(TRUE)，否則回應 0。

附加說明：

當您的波形資料有 Wait Key Event 指令時，此時 PG 的輸出將會停下來，直到執行這個函式後，波形才會繼續送出。但是執行 upgKeyEvent 的時候 PG 並不在 Wait Key Event 狀態下時，upgKeyEvent 將不會對 PG 有任何作用，而且 upgKeyEvent 也不會保留。因爲 upgKeyEvent 函式會送出的訊號是一個 System Clock 的週期寬度。例如 PG 的 System Clock 設定成 1MHz(Mega-Herts)，則 upgKeyEvent 會送出一個寬度爲 1us(micro-second)的訊號到 PG。但要如

何得知 PG 目前是在 Wait Key Event 狀態呢?您可以使用 upgGetStatus，來取得目前的 PG 狀態。

upgGetStatus

取得 PG 目前的狀態。

```
unsigned int upgGetStatus();
```

回應說明：

回應的值是一個 32 位元的資訊。定義如下:

Bit7~0 : PG Device ID number

1 : PG1020
2 : PG1050
3 : PG2016
4 : PG2116

Bit10~8 : Active Event

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

Bit11: Event Invert

0 ： Invert

1 ： Non-Invert

Bit12: Wait Event

0 ： PG 目前在等待狀態。

1 ： PG 目前不在等待狀態。

Bit31~13: Reserve, will be zero

附加說明：

如果要知道現在是否在等待 Event1 的狀態下時，只要呼叫 upgGetStatus 函式，當回傳的值爲 0x20，就代表已經進入等待 Event1 的狀態了。注意！PG1000 系列無法取得 Event Invert 的資訊。

upgShutdown

關閉 Pattern Generator。

```
BOOL upgShutdown();
```

回應說明：

如果 PG 正常關閉則此函式會回應非零的值(TRUE)，否則回應零(FALSE)。

附加說明：

當要結束您的應用程式之前，記得要先呼叫此函式，來關掉 PG

與電腦的連結，如此才不會影響下次使用。如果您的應用程式結束之前，未呼叫此函式來關閉 PG 的話，將造成下次進入 PG 時的不穩定。此時必須要重新開機，才能解決這個問題。