

## 目錄

<b>第 1 章 產品介紹</b>	<b>1</b>
1. 什麼是 Acute-PG 可程式資料產生器？	2
2. 產品配備-PG1000 及 PG2000	3
3. 產品配備-Pocket PG	4
4. 產品規格及特性-PG1000 及 PG2000	5
5. 產品規格及特性-Pocket PG	7
6. 使用環境	9
<b>第 2 章 安裝</b>	<b>10</b>
1. 安裝步驟	11
2. Printer Port 的 BIOS 模式設定	18
3. 問題反應	19
<b>第 3 章 功能說明</b>	<b>20</b>
1. 系統內容及設定	21
2. 主機頻率設定詳述	22
3. 波形編輯器	23
3.1 畫面視窗調整	25
3.2 名詞定義	26
3.3 游標的使用方法	27
3.4 訊號移動和快速組合訊號	29
3.5 訊號選單及訊號選擇	30
3.6 新增訊號	30
3.7 新增訊號組	31
3.8 新增所有訊號	32
3.9 刪除訊號、刪除所有訊號	32

3.10 改變訊號名稱	32
3.11 內容及所有內容	33
3.12 組合訊號	34
3.13 組合訊號重整	34
3.14 分解訊號	34
3.15 波形檔	34
3.16 列印波形	35
3.17 波形放大	37
3.18 波形縮小	38
3.19 拖曳模式和指標模式	39
3.20 波形編輯方法	39
3.21 低電位、高電位和波形反相	39
3.22 編輯時脈	40
3.23 編輯訊號組資料	41
3.24 編輯指令	43
3.25 波形檢查	47
3.26 波形輸出與停止輸出	48
3.27 環境參數設定	49
4. 同步計數器	50
5. 非同步計數器	53
6. 並列訊號產生器	56
7. 串列訊號產生器	58
8. I <sup>2</sup> C 訊號產生器	61
9. 資料檔轉成波形	65
10.文字(向量)檔轉成波形	67
11.Acute LA 波形檔轉換	73

12. Altera 波形檔轉換-----	74
13. 批次輸出-----	76
14. SPI 訊號產生器-----	78
15. VHDL 波形檔轉換-----	81
16. LIN 訊號產生器-----	83
17. CAN 訊號產生器-----	90
18. HDQ 訊號產生器-----	96
19. 1-Wire 訊號產生器-----	103
20. I <sup>2</sup> S 訊號產生器-----	106
21. 編碼器-----	108
22. USB1.1 訊號產生器-----	110
23. SMBus 訊號產生器-----	113
24. PMBus 訊號產生器-----	118
25. PWM 訊號產生器-----	123
26. VCD 波形檔轉換-----	125
<b>第 4 章 注意事項-----</b>	<b>127</b>
1. 硬體注意事項-----	128
2. 軟體注意事項-----	130
<b>第 5 章 其他-----</b>	<b>131</b>
1. 故障排除-----	132
2. PG_Function 指令使用方法-----	134
3. 使用文字編輯器編輯文字向量檔-----	139

## 第1章 產品介紹

## 1. 什麼是 Acute-PG 可程式資料產生器？

皇晶科技股份有限公司研發之 PC-based 可程式資料產生器—Acute PG1000、PG2000 系列及 Acute Pocket PG 系列，是一台可以產生多種數位波形的可程式資料產生器，具有高速，多 Channels 及多用途的功能，可輕易的編輯波形，並輸出至測試板上，不需大費周章製作一個電路板。因此，擁有一台 Acute PG 可節省許多開發成本。

不管是電路模擬、IC 測試、ROM 模擬、Protocol 模擬、零件燒錄....等等波形，都可以用 Acute PG 來產生。常用的數位傳輸界面已做成制式功能，讓您輕易的產生您要的波形。提供多種波形資料編輯方式：直接畫波形、文字檔轉入、Altera 波型檔轉入、Acute LA 波形檔轉入...。完美的 Window 操作介面，可與 Notebook 連接，易於攜帶，完全滿足研發者挑剔的需求。

搭配 Acute 邏輯分析儀(以下簡稱 LA)時，可以用 LA 來擷取量測物的波形，再交由 Acute PG 將波形輸出給另一個電路。此外，更可組合 Acute LA 與 Acute PG 做成自動測試系統或自動驗證系統。皇晶科技將陸續提供各種常用的 PLD 與 FPGA Simulation 的波形轉換程式，讓已經畫好的波形或是寫好的 Test Vector 都能輕易的轉換成實際的波形，來驗證 PLD 和 FPGA。

**註 1：可程式資料產生器簡稱 PG。**

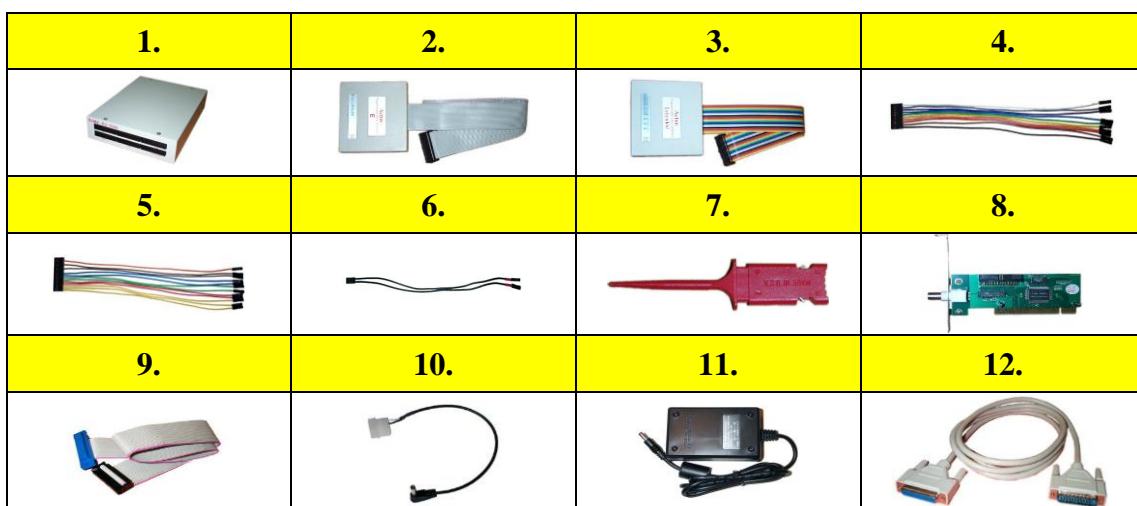
**註 2：Acute PG1000 系列包括 PG1020、PG1050。(簡稱 PG1000)**

**註 3：Acute PG2000 系列包括 PG2020、PG2050。(簡稱 PG2000)**

**註 4：Acute Pocket PG 系列包括 PKPG2016、PKPG2116、PKPG2116+。(簡稱 PKPG)**







## 2. 產品配備-PG1000 及 PG2000

	項 目	PGx020	PGx050
1.	PG1000 or PG2000 主機	1	1
2.	PG 訊號隔離放大器	2	5
3.	PG 擴充模組	1	1
4.	PG 訊號彩虹線/1x10 color line	2	5
5.	PG 擴充模組彩虹線/1x12 color line	1	1
6.	地線/black 1x2 line with red mark	3	6
7.	訊號探針(Red)	35	68
8.	PCI 傳輸介面卡(內接式配備)	1	1
9.	界面卡連接排線(25pins)(內接式配備)	1	1
10.	電源轉接電纜線(內接式配備)	1	1
11.	12VDC2A 整流穩壓器(外接式配備)	1	1
12.	Print Port 傳輸線(外接式配備)	1	1
13.	USB 轉換器(外接式選購配備)	1	1
14.	安裝光碟片	1	1
15.	說明書	1	1
16.	螺絲	1	1



### 3. 產品配備-Pocket PG

	項 目	PKPG 2016/2116/2116+
1.	Pocket PG 主機	1
2.	訊號彩虹線/1x20 color line	1
3.	地線/black 1x2 line with red mark	1
4.	訊號探針(Red)	22
5.	USB A-B 接線(1.8m)	1
6.	安裝光碟片	1

1.	2.	3.
		
4.	5.	6.
		



#### 4. 產品規格及特性-PG1000 及 PG2000

規格及特性		PGx020	PGx050
電源 (Power)	電源(Internal/External)	PC Power/Adapter(12V)	
	靜態消耗功率(Static Power Dissipation)	2.4W	
	瞬間最大消耗功率 (Max Power Dissipation)	<12W	
傳輸界面 (Interface)	內接式(Internal)	PCI Card	
	外接式(External)	Printer Port or USB	
通道(Channels)		20	50
內部時脈 (Internal Clock)	範圍(Range)	100MHz~1Hz	
	模式(Mode)	可微調	
	輸出通道(Output Channel)	1	
外部時脈 (External Clock)	範圍(Range)	<75MHz	
	模式(Mode)	Logic Not & Logic And	
	輸出通道(Output Channel)	2	
資料輸出控制方式 (Data Flow Control)		Loop Jump Wait for Event Branch (If command)	
輸出形式 (Output Type)		POD A-E: 1.5V-5.5V with 255Scakes. External POD: UART, I <sup>2</sup> C	
標準串列訊號輸出速度(UART Baud Rate)		110-256K	
輸出驅動能力(Fan Out)		8 個 TTL	
相位誤差(Data Skew)		<3ns	
記憶體(Memory)	波形儲存深度(Storage Depth /Channel)	標準規格 64K bits <sup>註</sup>	



內部事件(Internal Event)		Hot Key
外部事件 (External Event)	通道(Channel)	3
	模式(Mode)	Logic Not & Logic Or
	準位(Threshold)	TTL Level
工作溫度(Operating Temperature)		5°C~45°C (41°F~113°F)
保存溫度(Storage Temperature)		-40°C~75°C (-40°F~167°F)
使用語系(Languages)		中文、英文(可切換)
波形儲存、讀取(Save & Load Waveform)		可
波形列印(Print Waveform)		Windows 可接受的印表機皆可
線上輔助功能(Online Help)		有
體積 (Dimensions)	長 x 寬 x 高(mm) <sup>3</sup>	197 x 147 x 42(mm) <sup>3</sup>

註：加大容量選項：512K bits

## 5. 產品規格及特性-Pocket PG

規格及特性		PKPG2016/2116/2116+
電源(Power)	電源(Power Source)	USB bus-power (+5V)
	靜態消耗功率(Static Power Dissipation)	0.75W
	瞬間最大消耗功率 (Max Power Dissipation)	<2.5W
傳輸界面(Interface)		<b>USB</b>
內部時脈 (Internal Clock)	範圍(Range)	200MHz ~ 1Hz
	模式(Mode)	可微調
	輸出通道(Output Channel)	1
外部時脈 (External Clock)	範圍(Range)	<200MHz
	模式(Mode)	Logic Not & Logic And
	輸入通道(Input Channel)	1
資料輸出控制方式 (Data Flow Control)		Loop Jump Wait for Event Branch (If command)
輸出形式(Output Type)		5V, 3.3V, 3.0V, 2.8V, 2.5V, 2.1V, 1.8V, 1.5V, Hi-Z
輸出驅動能力(Fan out)		8 個 TTL
相位誤差(Data Skew)		<3ns
記憶體(Memory)	波形儲存深度(Storage Depth/Channel)	512K bits <sup>註</sup>
內部事件(Internal Event)		Hot Key
外部事件 (External Event)	通道(Channel)	2
	模式(Mode)	Logic Not & Logic Or

	準位(Threshold)	參考輸出形式
工作溫度(Operating Temperature)		5°C ~ 45°C
保存溫度(Storage Temperature)		-40°C ~ 75°C
使用語系(Languages)		中文、英文(可切換)
波形儲存、讀取(Save & Load Waveform)		可
波形列印(Print Waveform)		可
線上輔助功能(Online Help)		有
體積(Dimensions)	長 x 寬 x 高(mm) <sup>3</sup>	117 x 72 x 20(mm) <sup>3</sup>

註：PKPG2016：儲存深度為 64K bits/channel，PKPG2116，PKPG2116+：儲存深度為 512K bits/channel。PKPG2016，PKPG2116：內頻範圍 100MHz~1Hz，外頻範圍<75MHz，PKPG2116+：內頻範圍 200MHz~10Hz，外頻範圍 <= 200 MHz。

## 6. 使用環境

- PC INTEL 486 以上相容產品，建議使用 Pentium 100 以上等級。
- PC 記憶體容量 32M Bytes 以上。
- 硬碟可使用空間 10M Bytes 以上。
- 光碟機(安裝程式用)。
- 顯示規格 640x480 VGA 以上，建議使用 800x600 或 1024x768。
- 使用 101 鍵盤，建議使用 Windows 鍵盤。
- 二鍵或三鍵滑鼠。
- 一組 PCI Bus 插槽。<sup>註1</sup>
- Printer Port 傳輸界面。<sup>註1</sup>
- USB 傳輸界面。<sup>註1</sup>
- 印表機(非必需)。
- 中文版 Windows 95/98/ME/NT/2000/XP/7 作業系統。<sup>註2</sup>

註1：使用介面並非必需。Pocket PG 可使用的介面只有 USB；PG1000 及 PG2000 可使用的介面為 PCI、EPP 或 USB(依據選購配件而定)。

註2：使用 USB 介面，至少需要 Win 98se 以上。

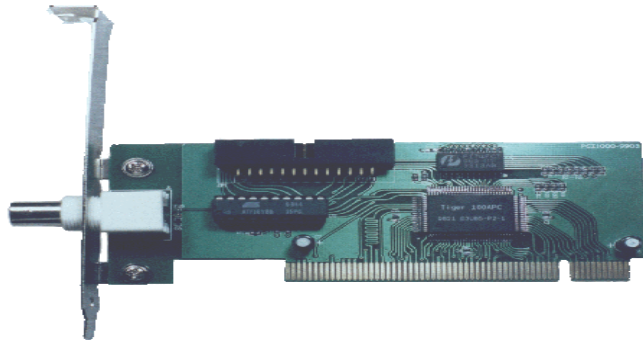
## 第2章 安裝

## 1. 安裝步驟

### 1.1 硬體安裝

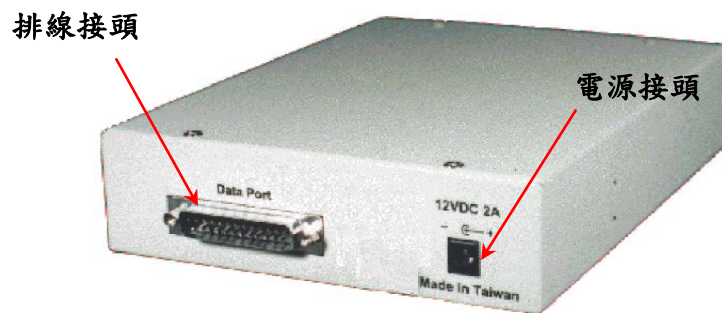
#### (1) PG1000、PG2000 內接式(使用 PCI 介面)

- a. 關閉 PC 的電源，並打開 PC 的外殼。
- b. 將傳輸界面卡(圖一)安裝在 PC PCI Bus 的插槽上，並將螺絲鎖緊。



圖一 傳輸界面卡

- c. 將 PG1000/PG2000 主機(圖二)安裝在 PC 的 CD-ROM 插槽。
- d. 將界面卡連接排線，從傳輸界面卡接到 PG1000/PG2000 主機背面排線接頭上。



圖二 PG1000/PG2000 主機

- e. 將 PC 的磁碟機的電源(5V,12V)透過電源轉接器，接到 PG1000/PG2000 主機的背面電源接頭。
- #### (2) PG1000、PG2000 外接式(使用 EPP 或 USB 介面)
- a. 將 Printer Port 傳輸線，連接在 Acute PG 與 PC 的 Printer Port 之間。(或是用 USB 傳輸線連接 PG1000/PG2000 與 USB Port)。
  - b. 將電源穩壓器(12V)接在 PG1000/PG2000 的電源接頭上。USB 界面請跳至步

驟三，Printer Port 界面請跳至步驟四。

- c. 如果使用 USB 傳輸界面，進入 Windows 環境時會出現安裝 USB 驅動程式的畫面，此時只要選擇磁片上的驅動程式即可。
- d. 打開 PC 電源，並進入 BIOS 設定模式，將您使用的 Printer Port 設定成 EPP 模式，假如沒有 EPP 模式，可以設成 EPP+ECP 模式、ECP 模式或是 Bi-direction 模式。但是如果您使用 Windows NT 的話，您只能選用 EPP 模式。注意!您的 PC BIOS 設定如果有 EPP1.7 和 EPP1.9 選項時，請選用 EPP1.9。(請參考故障排除一章說明)。

### (3) Pocket PG

- a. 將 USB Cable 正方形一端接在 PKPG 上。
- b. 將 USB Cable 長方形一端接在 PC USB Port 上。

## 1.2 量測套件安裝

- (1) 將訊號隔離放大器(請參考第五頁的配件圖)的排線接到 PG1000/PG2000 主機的面板上，記得要依照面板標示的 A、B、C、D、E 及 Extended Pod 一對一的與訊號隔離放大器的標示相符。(PKPG 無隔離放大器，只需將彩虹排線及地線接上即可)。
- (2) 將訊號連接線(10Pins)接在訊號隔離放大器上，另一端則接上探針。
- (3) 將訊號連接線地線(2Pins)接在訊號隔離放大器上，另一端也接上探針(只用 1Pin 即可)。
- (4) 將接地探針接(有紅色套環)在待測物的地線(Ground)上。
- (5) 將一般訊號探針接在標的物上。



## 1.3 驅動程式安裝

驅動程式安裝分為三種情況，會因使用不同界面及不同作業系統有所區別。

**注意! USB 無法與 Windows95/98SP1/NT 等作業系統搭配。**

### (1) 方式 1(不需要或不須安裝驅動程式)

以下環境組合請直間跳至軟體安裝。

- Printer Port + Windows 95
- Printer Port + Windows 98
- Printer Port + Windows ME
- Printer Port + Windows NT
- PCI + Windows NT

### (2) 方式 2(隨插即用 Plug & Play)

以下環境組合請跳至步驟 A1。

- PCI + 所有作業系統(不包含 Windows NT)
- USB + 所有作業系統

### (3) 方式 3(非隨插即用，但須驅動程式)

以下環境組合請跳至步驟 B1。

- Printer Port + Windows 2000
- Printer Port + Windows XP

**A1.** 硬體安裝完成後，打開 PC 電源並進入 Windows 作業系統。

**A2.** 將安裝磁片放入光碟片至光碟機中。

**A3.** Windows 在啟動時會找到硬體裝置的畫面，請依照 Windows 的指示操作。

**A4.** 如作業系統要求驅動程式時，請指示 Windows 至光碟機上尋找驅動程式。

**A5.** 如果 Windows 無法正確找到驅動程式時，請直接指示 Windows 到光碟機的根目錄，來尋找適合的驅動程式。

**A6.** 如果仍然無法正常安裝驅動程式，請參考故障排除章節或參考本公司網站

提供的 FAQ。仍無法解決時請與本公司聯絡。

**A7.** 如果正常安裝驅動程式時，您可從『裝置管理員』中找到一個『Acute PC-Based Instrument』的項目，該項目裡面會包括您安裝的驅動程式。接下來請安裝軟體。

**B1.** 硬體安裝完成後，打開 PC 電源並進入 Windows 作業系統。

**B2.** 將安裝磁片放入光碟片至光碟機中。

**B3.** 從『控制台』中選擇『新增/移除硬體』。

**B4.** 選擇新增裝置。

**B5.** 此時 Windows 會嘗試去尋找硬體裝置，但是 Windows 會找不到任何已知的硬體裝置。

**B6.** Windows 會要求選擇一種裝置，此時請選擇『新增一項裝置』。

**B7.** 選擇手動選擇硬體類型，不要讓 Windows 自行搜尋硬體。

**B8.** 在 Windows 2000 時選擇『其他裝置』，在 Windows XP 時選擇『顯示所有裝置』(Windows 可能會花很長的時間，請耐心等待)。

**B9.** 選擇『從磁片安裝』。

**B10.** 指定目錄至光碟機的根目錄。

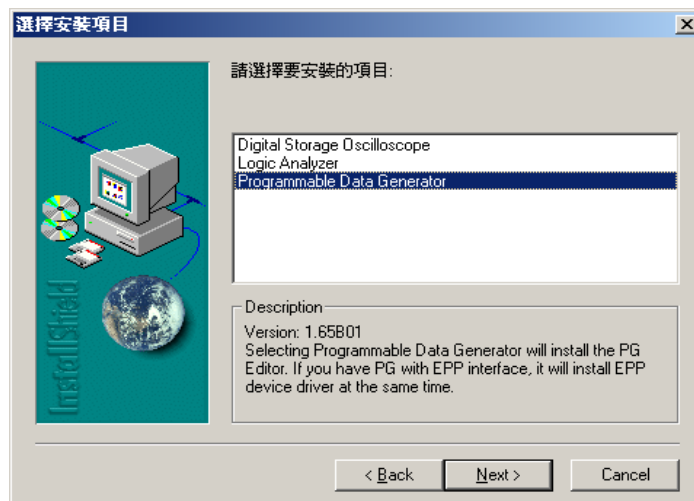
**B11.** 選擇『Acute EPP & ISA Interface Driver』。

**B12.** 如果仍然無法正常安裝驅動程式，請參考故障排除章節或參考本公司網站提供的 FAQ。仍無法解決時請與本公司聯絡。

**B13.** 如果正常安裝驅動程式時，您可從『裝置管理員』中找到一個『Acute PC-Based Instrument』的項目，該項目裡面會包括您安裝的驅動程式。接下來請安裝軟體。


## 1.4 軟體安裝(PG Editor 安裝)

- (1) 如果您已經安裝較舊 PG-Editor 版本時，請先移除舊版本，此移除動作不會將您舊環境設定消除請不用擔心。
- (2) 將安裝光碟片放入 CD-ROM 中。
- (3) 『我的電腦』中找到安置安裝光碟的光碟機，並點選該光碟機，此時應會自動進入安裝程式，如果沒有自動進入安裝程式，請選擇光碟中的 Setup.exe 程式並執行它。安裝程式會根據作業系統的語系自動安裝該語系的版本。(目前只支援中文及英文，非中文的語系皆會安裝成英文版)光碟片中可能含有其他 Acute 相關軟體，此時只要選擇可程式資料產生器的選項即可。如果安裝其他產品軟體，該軟體因沒有硬體配合，所以只會進入 Demo 模式。



- (4) 依照安裝軟體的提示輸入安裝目錄。



- (5) 開始安裝軟體。
- (6) 安裝結束後，Desktop 與程式集中都有可程式資料產生器的啟動圖示 ，可以任選一個來啟動 PG-Editor。
- (7) 如果執行 PG-Editor 時，出現 Demo Mode 就代表您的安裝上出了問題。請參考故障排除。

## 2. Printer Port 的 BIOS 模式設定

並列埠的規格有很多種，茲將一般 BIOS 有提供的模式與 Acute PG 支援的模式列表於下，請依下表的先後次序找尋您 BIOS 有提供的模式來設定，以達到外接式最佳的效率。

優先次序	BIOS 支援模式	備註
1	EPP 1.9	效率最好，建議使用
2	ECP + EPP 1.9	同上
3	EPP	有可能是 EPP 1.9，也有可能是 EPP 1.7
4	EPP 1.7	效率較 EPP 1.9 稍差
5	SPP + EPP 1.7	同上
6	ECP + EPP 1.7	同上
7	ECP + Bi-direction	效率比 EPP 差
8	Bi-direction	效率比 EPP 差
9	ECP	有些電腦可用，有些不行
10	ECP + SPP	有些電腦可用，有些不行
11	SPP	有些電腦可用，有些不行
12	Normal	不支援

### 3. 問題反應

您在安裝或使用上如果遇到任何問題，請參考注意事項及故障排除的章節說明，如果仍然無法解決問題，或是對本產品有任何的建議，請與我們聯絡，我們會儘快的處理您的問題。

我們的網頁是 <http://www.acute.com.tw>，網路上有許多新的訊息，包括新版軟體、FAQ、新產品介紹.....等等。如果網頁上的資訊無法滿足您的需求時，也可以用 E-mail 的方式，將您的問題告訴我們，我們的 E-mail 位址是 [service@acute.com.tw](mailto:service@acute.com.tw)。

## 第3章 功能說明

## 1. 系統內容及設定



系統內容設定主要是設定程式的使用語系，每一種語系可以隨時自由的切換，可以根據個人需求來做調整。另外還有兩個訊息欄位，一個是使用機種，另一個是使用界面。Acute PG 的機種分成 PG1020、PG1050、PG2020、PG2050、PKPG2016、PKPG2116 以及 PKPG2116+，PG1020 與 PG2020 為 20 個通道的機種，PG1050 與 PG2050 為 50 個通道的機種，PKPG2016、PKPG2116 與 PKPG2116+ 為 16 個通道的機種。而傳輸界面則分為內接式及外接式兩種，內接式為 PCI 界面卡，而外接式則有 Printer Port 及 USB 界面。使用 Printer Port 的界面時，要注意 BIOS 的設定，由於使用 Printer Port 界面會和主機板使用的 I/O Chip Set 以及 BIOS 的設定有關，所以如果設定不對有可能會變成無法與 Acute PG 連接或是使傳輸速度變慢。(請參考 Printer Port 的 BIOS 模式設定一節。)PG1000、PG2000 及 PKPG 系列提供了輸出電壓可調的功能。PG1000 及 PG2000 輸出準位 1 控制通道 0 到通道 19，輸出準位 2 控制通道 20 到通道 49；PKPG 只能調整輸出準位 1，輸出準位 1 控制通道 0 到通道 15，且準位只有 8 階。(PKPG 準位一但設定好後，16 個通道皆會變成該準位，因此切換準位前請先確認待測物的規格，否則待測物可能會因而損壞。)



## 2. 主機頻率設定詳述

Acute PG 的內部基頻有兩大系統，一是固定頻率，另一是微調頻率。所謂固定頻率是包含 100MHz、80MHz、50MHz、40MHz 和 100KHz 等 4 種固定頻率。而且 Acute-PG 有四組 10 倍數的除頻器，因此以 100MHz 的基頻來講，就可以產生 10MHz、1MHz、100KHz 和 10KHz 等頻率。依此類推，80MHz 就可產生 8MHz、800KHz、80KHz 及 8KHz 等頻率值。

另一種微調頻率系統有兩組基頻。第一組為 15MHz 到 75MHz，微調的基本單位為 100KHz。也就是可以產生 15MHz、15.1MHz、15.2MHz、...、74.9MHz 及 75MHz 等頻率值。第二組的頻率範圍為 1.25MHz 到 15MHz，但是它的微調基本單位為 50KHz，可以產生的頻率為 1.25MHz、1.3MHz、1.35MHz、...、14.95MHz 及 15MHz 等頻率值。當然微調的頻率系統下亦可以使用 Acute PG 的內部四組 10 倍數的除頻器。因此 15MHz 到 75MHz 且微調值為 100KHz 的環境，一旦除以 10 以後，就會變成 1.5MHz 到 7.5MHz 的範圍，而且微調值會變成 10KHz，換句話說可產生的頻率值為 1.51MHz、1.52MHz、1.53MHz、...、7.49MHz 及 7.5MHz。

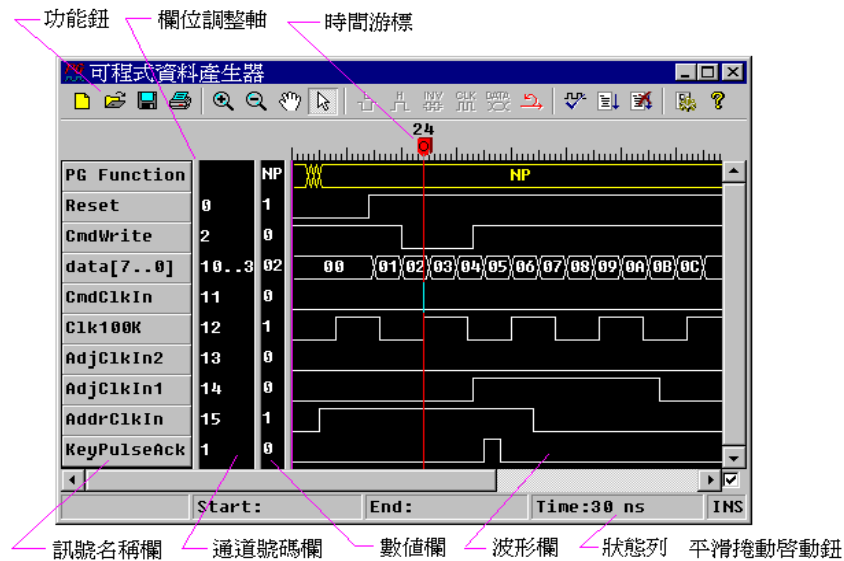
雖然頻率值的定義似乎有些複雜，但是在 PG-Editor 的程式環境中，您大可不必太擔心這個問題，因為一旦設定錯誤的頻率時，PG-Editor 都會給予最接近該設定的頻率值。

### 3. 波形編輯器



波形編輯器提供了一個畫波形的環境，透過波形編輯器可以畫出您想要輸出的波形，同時波形編輯器包含了許多快速畫波形的方式，例如 Clock 或 Counter 等的波形產生，並且包括了許多便於修改的使用環境。

#### • 波形編輯器的快速使用方法



- (1) 進入波形編輯器。
- (2) 用『環境參數設定』的功能來設定『工作頻率』及『可編輯時間長度』。
- (3) 啟動訊號選單(將滑鼠移至訊號名稱的欄位上，按下滑鼠的右鍵，畫面將出現選單)。
- (4) 請選擇訊號選單上的『新增訊號』，會出現一個對話盒。
- (5) 設定訊號名稱、訊號編號及訊號顏色。
- (6) 重複步驟三、四、五，直到所需要的訊號名稱都設定好為止。
- (7) 用滑鼠左鍵在波形區中，利用標記方式來選擇某一波形區塊，然後再用『低電位』、『高電位』、『波形反相』、『編輯時脈』或『編輯訊號組資料』等方式來改變波形。
- (8) 也可以用滑鼠的右鍵在波形區中，進入波形選單(將滑鼠移至波形區的欄位

上，按下滑鼠的右鍵，畫面將出現選單)，利用剪貼、複製及還原等功能來修改波形。

- (9) 將地線接到待測電路上的地線位置。
- (10) 將探針接到待測電路上，探針編號需與先前設定的訊號編號相同。
- (11) 按下『輸出波形』鈕。

### 3.1 畫面視窗調整

波形編輯畫面中的欄位(視窗)大小都是可以調整的。包含訊號名稱欄位、訊號編號欄位、數值欄位以及波形視窗。這四個欄位(視窗)的中間共有三條垂直的欄位調整軸，將滑鼠移到欄位調整軸時，滑鼠圖示會由箭頭變成一個橫移的圖示，此時只要按住滑鼠左鍵不放並拖動調整軸即可改變各欄位的大小。

注意！當靠左邊的調整軸被移動時，右邊的欄位會被等距離的被拖動，但是左邊的欄位將不會被改變。因此調整欄位大小時，可能將某些欄位完全移到視窗之外，此時只要將整個 PG Editor 的視窗變大就可以看到原來看不到的欄位或欄位調整軸，而順利的調整欄位大小。

每個欄位可以用適當調法將欄位調整到最適合的位置，只要用滑鼠左鍵在調整軸上按兩下，程式就會自動將欄位調整到需要顯示的字串都能完全被顯示的位置。

## 3.2 名詞定義

### (1) 訊號名稱的定義

訊號名稱可以是文數字或是如底線、中括號和句點等符號，也可以是中文字。

總長度不能超過 31 個字。(一個中文字相當於兩個英文字)。

### (2) PG1000、PG2000 訊號編號的定義 (面板標示由左而右)：

Pod A = 編號 0 ~ 編號 9

Pod B = 編號 10 ~ 編號 19

Pod C = 編號 20 ~ 編號 29(PGx020 沒有 Pod C)

Pod D = 編號 30 ~ 編號 39(PGx020 沒有 Pod D)

Pod E = 編號 40 ~ 編號 49(PGx020 沒有 Pod E)

Extended Pod = EVT1、EVT2、EVT3、RSI1、RSI2、RSO1、RSO2、I2CC、I2CD、

CKI1、CKI2、CKO1。

### (3) PKPG 訊號編號的定義 (面板標示由左而右)

一般通道 = 編號 0 ~ 編號 15

擴充通道 = EV1、EV2、CKI 及 CKO

EV1 = Event1


EV2 = Event2

CKI = Clock Input

CKO = Clock Output



### 3.3 游標的使用方法

波形編輯器的三個游標系統：

- (1)  『指示游標』代表在波形區時，可用來標記波形區塊或是設定波形的時間游標(紅色)。標記波形區塊的方法是將滑鼠移至波形區，然後按下左鍵不放並移動滑鼠，此時就會出現波形區塊，直到放開滑鼠的左鍵時，整個標記區才會被確定。當標記了某一個波形區塊後，即可利用『波形功能鈕』來加以改變標記區的波形，或是用滑鼠的右鍵來剪貼複製標記區的波形。而且在『指示游標』移動的時候畫面下方的狀態列也會出現目前『指示游標』所在的時間，一旦開始標記時狀態列也會出現標記的起始時間及結束時間，標記完成後狀態列上也會出現整個標記範圍的時間。

標記波形還有另一個方法，就是在波形區中按下滑鼠左鍵兩次(Double Click)。此時會出現一個標記區，這個標記區是在滑鼠游標位置的訊號，從滑鼠位置往前及往後延伸，直到該訊號值與滑鼠位置的訊號值不同為止。

至於如何在波形區設定『時間游標』呢？只要將滑鼠移至波形區然後按一下滑鼠左鍵，就會出現一條紅色的『時間游標』。當出現時間游標時，才可以設定『波形指令(PG Function)』。『時間游標』的另一個用途是當作貼上波形時的起始點，『時間游標』的中間會有一小段藍色的部份，這是代表如果要剪貼波形時的起始通道。

- (2)  『拖曳游標』是代表在波形區時，可用來移動波形的游標。使用時只要將拖曳游標移至波形區中，然後按住滑鼠左鍵並移動滑鼠，此時就可以看到波形跟著游標移動。
- (3)  『通道標記游標』只會在數值欄位出現，也就是當滑鼠移至數值欄位時，

游標形狀會變成通道標記游標。當通道標記游標出現時，就可以根據當時游標位置所指的通道進行標記。如果要複合標記時只要先按下『Ctrl 鍵』，再來選擇要標記的通道即可。


### 3.4 訊號移動和快速組合訊號


訊號名稱欄中的訊號名稱如果要重新排列，可以用滑鼠的左鍵拖動要移動的訊號。訊號被移動時有四種模式，這四種模式可以根據拖動訊號時的滑鼠游標來判斷。當滑鼠游標變成『訊號插入游標』時，就代表某一個訊號將要被移動到游標的位置。如果滑鼠游標變成『訊號快速組合游標』就代表被拖動的訊號將要與目前游標位置的訊號做組合的動作。前面兩種游標模式都是單一訊號被移動時所產生的游標。但是如果拖動的訊號不是單一訊號，而是被選擇的多個訊號時，就會出現『多訊號插入游標』及『多訊號快速組合游標』。所以非單一訊號被拖動時，就代表將所有被選擇的訊號一起移動或是被組合。


訊號快速組合功能可以將單一訊號組合成訊號組(Bus)。使用時只要用滑鼠左鍵去拖動某個訊號或是訊號組，將它拖到另一個訊號或是訊號組上面，就可將兩訊號或兩訊號組合併，組合後的訊號名稱為被加入的訊號名稱，訊號的順序安排為：新加入的訊號為 High Bits，被加入的舊訊號為 Low Bits。例如：


- (1) 將訊號 D1 拖入訊號 D0 中，組合後的新訊號名稱為 D0，而 D0 為 Bit0，D1 為 Bit1。
- (2) 將訊號組 A[2..0]拖入訊號組 D[2..0]中，新訊號名稱會變成 D[2..0]，訊號的順序由 MSB(Most Significant Bit)到 LSB(Least Significant Bit)依序為 A2、A1、A0、D2、D1、D0。

在多訊號快速組合時，所有被選擇的訊號都會被一起加入到新的訊號上，其加入的順序為訊號名稱欄由上而下排列。但是如果要被移入的新訊號也是屬於被選擇的訊號時，那就是代表將所有被選取的訊號做快速組合的動作。

 訊號插入游標

 訊號快速組合游標

 多訊號插入游標

 多訊號快速組合游標



### 3.5 訊號選單及訊號選擇

訊號選單的啟動方式，是將滑鼠移到訊號名稱欄，按下滑鼠的右鍵就可以啟動訊號選單。但是啟動時的滑鼠位置、訊號名稱增修情形或是通道的選擇狀況都會影響啟動的狀態。例如訊號名稱欄的資料被更改或是新增時，復原的選項就可以選擇。反之，如果沒有任何訊號被更動，復原的選項就會變成灰色而不能選擇。選單的動作主要是根據滑鼠所在位置的訊號來處理，如果所在的位置是被選擇的訊號(呈藍色)之一，那就代表選單的動作是對所有被選擇的訊號來處理。

訊號的選擇方式是 Windows 環境的標準用法，用滑鼠左鍵直接點選任何一個訊號就會變成藍色的訊號，也代表該訊號是被選擇的。一旦用滑鼠左鍵點選了某個訊號時，原來被選擇的訊號就會被還原。如果要複合選擇只要按住『Ctrl 鍵』不放，再用滑鼠左鍵來選擇，此時就可以重覆選擇。或是利用按住『Shift 鍵』的方式來做區域性的選擇。

### 3.6 新增訊號

新增訊號的方法是從訊號選單中，選擇新增訊號。此時會出現一個新增訊號對話盒，這個對話盒可設定三個項目，一是訊號名稱，二是訊號通道，三是訊號顏色。設定完成後，按下確定鈕，訊號名稱欄就會出現新增的訊號。

或是『新增一組訊號』來快速新增 POD 訊號組。(PKPG 每個 Pod 為 8 個通道)

### 3.7 新增訊號組



選用新增訊號組的功能後，畫面會出現一個如上圖的對話盒。這個對話盒包括了幾個部份：一是訊號名稱、二是訊號型態、三是通道組合區、四是通道區、五是訊號顏色區。請依照下列步驟來新增訊號組。在訊號名稱欄位中填入訊號組的名稱。

- (1) 選擇訊號型態，有 HEX、DEC、OCT、BIN 和 ASC 等選項。
- (2) 從通道區中選擇要組成訊號組的通道號碼，並從最低的位元開始選擇。例如要新增一個訊號組，使用的通道為 5、4、3、2 等 4 個通道，5 為最高位元，2 為最低位元。選擇時就必須依序選擇 2、3、4、5。如果選擇錯誤時，只要到通道組合區中，將錯誤的通道再選一次，錯誤的通道就會被移回通道區。而通道區中灰色的部份，是代表已經被使用的通道，所以不能被選擇。
- (3) 選擇訊號組的顏色。
- (4) 按下確定鈕，即完成設定。

畫面右方有兩個特殊的功能鈕，一為全開，另一為全關。全開代表將通道區的所有可用的通道全移至通道組合區。而全關剛好相反，是將通道組合區的所有通道全移至通道區中。

在點選通道區的通道時，有一個方法可以直接插入通道組合區的某兩個通道之間。方法是先將通道組合區上方的指標器先移至要插入的位置，然後再從通道區中來點選通道。

### 3.8 新增所有訊號

新增所有訊號會將所有的訊號依照通道編號來編訊號名稱，加到訊號名稱欄(如：CH-00,CH36 等)。PGx050 會加入 CH0-CH49 等 50 個通道，PGx020 會加入 CH0-CH19 等 20 個通道，而 PKPG 只會加入 CH0-CH15 等 16 個通道。

當新增所有訊號時，可能會出現一個警告對話盒，內容是『**要刪除其他訊號嗎？如不刪除，已經存在的通道將不會被新增**』，這是系統為避免通道重複，而出現的警語；如果要刪除原來訊號名稱欄內的訊號，就選擇『是』的按鈕；如果選擇『否』，代表原來已經佔用的通道將不會被新增。但是如果名稱相同時，該通道也不會被新增。

### 3.9 刪除訊號、刪除所有訊號

刪除訊號的作法有兩種。

- (1) 將選擇的訊號名稱刪除。將滑鼠移到要刪除的訊號上或是被選擇的訊號名稱上，按下滑鼠右鍵，此時會出現一個訊號選單，請選擇『刪除訊號』的項目，即可將該訊號刪除。
- (2) 刪除所有訊號，只要啟動訊號選單後，選擇『刪除所有訊號』就可以將所有的訊號全部刪除。

### 3.10 改變訊號名稱

要改變訊號名稱方法有二，一是啟動訊號選單，選擇『內容』來改變訊號名稱。另一種則是啟動訊號選單，然後選擇『改變訊號名稱』，此時要改變的訊號會變成一個可輸入的編輯盒，即可輕易的修改訊號名稱，但是修改名稱有幾個重點，就是訊號名稱不能是正在使用的名稱，也不能超過 31 個字，訊號名稱可以區分大小寫，所以相同名稱的大寫及小寫是代表不同的訊號。

### 3.11 內容及所有內容



設定訊號參數的啟動有兩種方法，一是將滑鼠移到需要改變訊號參數的訊號名稱上，並按下滑鼠右鍵，再從訊號選單中選擇『內容』的選項。另一個方法是將滑鼠移到需改變訊號參數的訊號名稱上，並按滑鼠左鍵兩下(Double Click)。兩種方法都會在畫面上出現設定訊號參數的對話盒。所以設定訊號參數的用途，是在已經存在的訊號需要改變設定時使用。訊號參數可以設定單一訊號或是訊號組，設定的部份有三項，一是訊號名稱，二是訊號模式，三是訊號顏色。但是如果設定的是單一訊號時，訊號模式無法設定。而訊號模式主要是設定訊號組的表示方式，有 HEX、DEC、OCT、BIN 和 ASC 等選項。

但是如果訊號名稱欄中有許多訊號是被選擇的時候，將滑鼠移至被選取的訊號名稱上並按下右鍵，此時出現的訊號選單的『內容』會變成『所有內容』，其意義是代表對所有被選取的訊號名稱做設定。所以在設定訊號參數對話盒的訊號名稱是無法設定的，因為每個訊號名稱是不能一樣的。只有訊號模式及訊號顏色是可以被一起改變的，如果原來被選取的訊號模式和訊號顏色並不是完全一樣時，訊號模式會變成灰色，但是仍然可以被改變，而顏色部份會出現被選取的第一個訊號的顏色。然而在還沒有改變設定以前所有的訊號都還會維持原來的設定，一旦改變了訊號模式或是訊號顏色時，所有的訊號設定就會被同時設定成一樣的。請注意！如果只改變其中一項，只會影響所有被選取訊號的該項的設定，其他的設定依然不會被改變。

### 3.12 組合訊號

組合訊號的方法分為快速組合及一般組合，快速組合的用法請參考『訊號移動和快速組合訊號』一節說明。而在一般組合的用法與『新增訊號組』的用法完全一樣，只是通道區裡可以被選擇的通道並不是全部，而是訊號名稱欄裡面所被選擇的訊號名稱。

### 3.13 組合訊號重整

組合訊號重整的用法與組合訊號及新增訊號組的用法一樣。組合訊號重整主要是對訊號組的每個位元重新安排順序的功能。

### 3.14 分解訊號

要將組合的訊號組(Bus)分解開來的作法是將滑鼠移到要分解的訊號組(Bus)上，用滑鼠右鍵來打開訊號選單，選擇分解訊號功能來做分解。被分解的訊號組會變成一些訊號，這些訊號會被重新命名，例如有一訊號組是由四個訊號所組合，名稱為XA，當分解之後，會變成四個訊號，名稱為XA-3、XA-2、XA-1、XA-0。

### 3.15 波形檔

透過波形編輯器所編輯出來的波形是可以被儲存成檔案形式。儲存的方式有兩種，一個是附屬檔名為(.PGW)的波形檔，另一個是附屬檔名為(.PGV)的向量檔。

PGW 波形檔，它的儲存內容包括了訊號名稱、波形、波形指令及相關參數。它的儲存格式是 PG Editor 的內定格式，所以無法用其他未支援 PGW 檔的軟體來讀取。

PGV 向量檔，是一個文字檔的格式。所以 PGV 檔可以用任何文字編輯程式來編輯它，如 Notepad、PE2 等等軟體。至於有關 PGV 的格式請參考『文字(向量)檔轉成波形』一節的說明。

由於 Acute PG 所用的波形檔和 Acute LA 所用的波形檔有一些差異，所以如果要使用 Acute LA 的波形檔必須經過轉換。轉換方法請參考『Acute LA 波形檔轉換』一節。

### 3.16 列印波形



#### (1) 印表機設定

PG Editor 會呼叫 Windows 的印表機設定，您可以選擇適合的印表機，並選擇橫印或縱印或其它印表機的特殊設定等。

#### (2) 列印範圍

波形列印可以根據個人的需求選擇任何的範圍來列印。列印範圍設定方面，可以設定成全部、選擇的通道或是標記區。選擇的通道是指在訊號名稱欄裡被選擇的部份，而這些被選擇的訊號會呈現成藍色。而標記區是波形區中被標記的部份。

#### (3) 列印比例

##### a. 『列印成一張報表紙』

橫向列印範圍會自動根據紙張寬度調整成適當的比例，而縱向列印範圍也會根據紙張高度自動調整比例。

**b. 『列印範圍與紙張同寬』**

橫向列印範圍會自動根據紙張寬度調整成適當的比例，但縱向列印範圍必須從垂直比例設定欄來設定，這種設定如果垂直列印部份會超過一張紙的話，就會被列印到下一張紙。

**c. 『列印訊號與紙張同高』**

縱向列印範圍會根據紙張高度自動調整比例，而橫向列印範圍必須從水平比例設定欄來設定。

**d. 『自定比例』**

橫向、縱向都必須自行從水平及垂直比例欄來設定。

**(4) 列印格線**

列印時也可以選擇是否要列印格線，要列印格線時，可以選擇格線間的距離。

**(5) 彩色列印或灰階列印**

黑白的列印會將波形的背景色當成白色，而其它的顏色都會設成黑色。但是彩色列印或灰階列印時，則是保持原來的設定顏色來列印，因此要注意波形的背景色最好先設定成白色，否則列印時整張紙都會變成波形背景色，而那些細細的波形會變得很難辨識。

**(6) 輸出至剪貼簿**

這個功能是將波形印到剪貼簿上，然後在其他的繪圖軟體或是有提供繪圖功能的軟體中，利用貼上的功能，就可以輕易的將波形剪貼到另一個軟體上。

**(7) 輸出至圖形檔(.BMP)**

這個功能是將波形變成一個圖形檔，當選擇這個選項時，畫面上會出現一個『Size』的按鈕，這個按鈕是用來設定要產生的圖形檔的長與寬。『Size』的設定就相當於印表時的紙張大小設定

### 3.17 波形放大



波形視窗的波形顯示是以螢幕的點為單位。在沒有任何的放大縮小倍率時，螢幕的一點相當基本頻率的倒數(時間)。

ex. 假設基本頻率為 1MHz，則螢幕的每一點為 1us。所以放大波形代表螢幕每一點為基本頻率的倒數除以放大倍數。

ex. 假設基本頻率為 1MHz，放大倍數為八倍時，則螢幕的每一點為  $1\mu\text{s}/8 = 125\text{ns}$ 。

所以波形放大時，我們會看到波形變寬。要注意的是當波形放大或縮小時，會以時間游標(紅色)為參考點來做放大，一旦在沒有時間游標的情況下會以波形區的最左邊為參考點。所謂的參考點就是放大縮小時的中心點，在放大或是縮小時會以參考點為中心往兩邊縮放。而放大的倍率有下列的倍數 2、4、8、16、32 或是 64。當按放大鈕按到 64 倍時，放大鈕將會變成灰色，而不能再放大，直到又按了縮小鈕為止。



### 3.18 波形縮小



波形視窗的波形顯示是以螢幕的點為單位。在沒有任何的放大縮小倍率時，螢幕的一點相當於基本頻率的倒數(時間)。

ex. 假設基本頻率為 1MHz，則螢幕的每一點為 1us。所以縮小波形代表螢幕每一點為基本頻率的倒數乘以縮小倍數。

ex. 假設基本頻率為 1MHz，縮小倍數為八倍時，則螢幕的每一點為  $1\text{us} * 8 = 8\text{us}$ 。

所以波形放大時，我們會看到波形變窄。要注意的是當波形放大或縮小時，會以時間游標(紅色)為參考點來做縮小。一旦在沒有時間游標的情況下會以波形區的最左邊為參考點。所謂的參考點就是放大縮小時的中心點，在放大或是縮小時會以參考點為中心往兩邊縮放。而縮小的倍率是沒有固定的，只要縮到波形區可完全被顯示出來時為止，此時縮小鈕也將會變成灰色，而不能再縮小，直到按下放大鈕為止。

### 3.19 拖曳模式和指標模式



拖曳模式和指標模式是互斥模式，所以兩個模式同時間只有一個模式可以被選擇。在波形編輯器中對游標系統的使用，可參考『游標的使用方法』的章節說明。

### 3.20 波形編輯方法

波形的編輯方法主要是用低電位、高電位、波形反相、編輯時脈及編輯訊號組的方式來修改波形，這些功能可以透過畫面上方的功能鈕達成。然而除了這些功能以外，也可以運用已經存在的波形來做剪下、複製及貼上等方法來修改波形。這些用法只要在波形區中按下滑鼠的右鍵，就會出現一個選單，選單就包括了以上的功能，而且還包括了一個還原的選項，這個選項是當編輯波形時，如果發現編輯錯誤，可以立即按下還原選項，讓波形恢復成原來的模樣。


### 3.21 低電位、高電位和波形反相



低電位、高電位和波形反相都是編輯波形的基本功能，當波形區有部份波形被標記的時候，就可以用這三個功能鈕來改變標記區的波形。如果按下『低電位』鈕那標記區的波形通通會變成低電位(Low)的訊號，如果是訊號組的部份就會變成 0。而按下『高電位』鈕標記區會變成高電位(High)的訊號，如果是訊號組會變成該訊號組的最大值，例如 8 個 Bits 組合的訊號組，就會變成 255(0FFh)。『波形反相』鈕就是將訊號反相，就是低電位的地方變成高電位，高電位的地方變成低電位。訊號組的地方就會變成 1 的補數。

### 3.22 編輯時脈



 首先必須先在波形區中標記一個區域出來，此時『編輯時脈』的按鈕才能使用。當按下『編輯時脈』鈕以後，會出現如上圖的對話盒。對話盒會有標記起始位置及標記結束位置，從這裡可以得知要產生時脈的區域在哪裡。對話盒中還有一個重要的訊息就是基本單位。所謂基本單位就是目前使用的頻率的倒數，例如目前所設定的頻率為 10MHz，那基本單位就是  $1/10\text{MHz}=100\text{ns}$ 。這也是波形區的游標每移一格的單位時間，所以稱之為基本單位。而在『編輯時脈』裡，這個基本單位也就代表所能產生的最快時脈的高電位週期或是低電位週期。同理，所能編輯出的最快時脈剛好是基本頻率的一半。當然要設定成最快的時脈，也必須將倍數值設成 1。倍數值增加代表時脈速度越慢，倍數值乘以基本單位就是時脈的半個週期。而為了讓使用者可以快速得知目前所設定的時脈頻率為何，所以在對話盒的最下方會顯示目前的設定會產生多少頻率的時脈。對話盒中還有一個參數值可以設定，那就是開始值，所謂起始值就是從標示區開始畫時脈波形時，要從高電位開始呢？還是要從低電位開始。如果要從低電位開始的話，那開始值就設定成 0，反之則設定成 1。

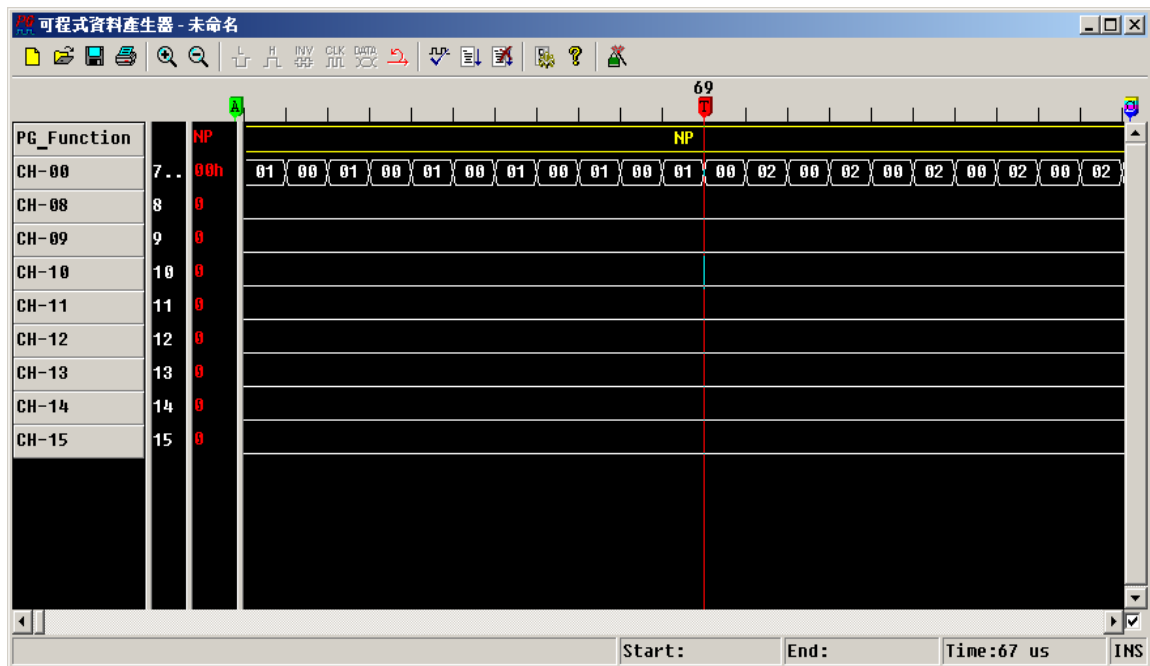
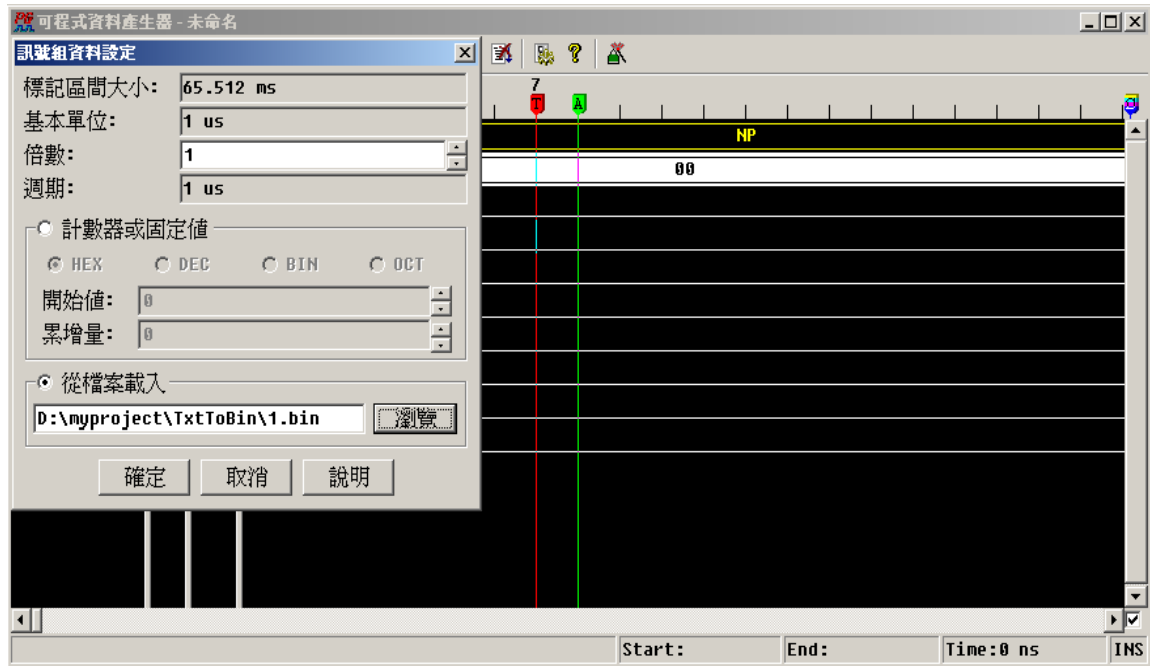
### 3.23 編輯訊號組資料



先在波形區中標記一個區域出來，此時『編輯訊號組資料』的按鈕才能使用。也就是說產生的訊號會出現在標記區域內，不會影響非標記區域的訊號內容。當按下『編輯訊號組資料』鈕以後，會出現如上圖的對話盒。對話盒中還有一個重要的訊息就是基本單位。所謂基本單位就是目前使用的頻率的倒數，例如目前所設定的頻率為 10MHz，那基本單位就是  $1/10\text{MHz}=100\text{ns}$ 。這也是波形區的游標每移一格的時間，所以稱之為基本單位。還有另一個欄位是『週期』，週期就是基本單位乘以倍數的結果，也就是編輯資料時每一筆資料的間隔。

訊號組的資料編輯方式有三種：第一是計數器，即是在標記區中填入一組累進的資料。而這個計數器可以設定開始值和累進值，如果開始值為 0，累進值為 3，那就會產生一組 0、3、6... 的數列。如果開始值和累進值要用其他的進位制來描述，只要選擇 HEX、DEC、BIN、OCT 等四個進位選項。第二種編輯資料組的方式是固定值，其實固定值與計數器是一樣的用法，只是累進值的設定為 0。因為累進值為 0，所以每一筆資料都會是開始值。

第三種編輯訊號組的方式是從檔案載入。如果要從檔案載入，必須先選擇『從檔案載入』的選項，並輸入一個檔案名稱。載入的方式是以每個 Byte 為基本單位，也就是從檔案的第一個 Byte 開始，每個週期填入一個 Byte 的資料。如果檔案的長度太短而不夠填入標記區時，程式會自動從檔案的開頭再讀取一次，直到標記區的資料被填滿為止。



### 3.24 編輯指令



在波形區中有一個訊號組叫做『PG Function』，它是一個不能被刪除的訊號組，這個訊號組就是波形指令訊號組。所謂波形指令顧名思義就是用來控制波形動作的指令，整個波形指令總共有 7 個，包括 NP (No Operation)、JP (Jump)、LP (Loop)、BE(Branch if Event)、LC (Loop Count)、SE (Set Event)和 WE (Wait Event)，如下圖的指令列表。

指令的介紹：

指令	指令全名	說明	週期
NP	No Operation	沒有動作	1
JP	Jump	跳躍到設定點繼續執行	3
LP	Loop	將迴圈數減一，如果迴圈數不為 0 就跳躍到設定點，如果為 0 就跳到下一個位置。	3
BE	Branch if Event	如果以設定的 Event 訊號出現就跳躍到設定點，否則就會跳躍到下一個位置。	3
LC	Loop Count	設定迴圈數，設定值為 2~65536。	2
SE	Set Event	選擇觸發的事件。	1
WE	Wait Event	等待事件被啟動。	1
OE	Output Enable	輸出允許(高阻抗模式) <sup>**1</sup>	3

註 1：OE 指令只在 PKPG 有用。

- (1) **NP (No Operation)**在波形執行時，並不會有任何控制流程的改變或是任何內部暫存器改變。用 CPU 的觀點來看，那 NP 指令的動作就是 New Address = Address + 1，也就是指標移到下一個位置，其他不做任何改變。
- (2) **JP (Jump)**是一個改變控制流程的指令。例如『JP 35』，這個指令的動作就是無條件跳至新位址 35(相當於 New Address=35)。
- (3) **LP (Loop)**和 JP 指令很類似，其最大差異就是 JP 是一個無條件跳躍指令，而 LP 是一個有條件跳躍，它必須與『LC』指令配合使用。Acute PG 有一

個內部暫存器稱為迴圈暫存器(Counter)，這個暫存器就是用來記錄迴圈數目的暫存器。這個暫存器是用 LC (Loop Count)指令來設定，例如『LC 32』就是將迴圈暫存器設定成 32。迴圈暫存器可以設定的值為 2~65536，所以不能設定成 0 與 1，這種設定方式與一般的 CPU 或 MCU 的用法有些許的差異，需注意。設定好迴圈以後就可以使用 LP 指令了，而每次執行 LP 指令，迴圈暫存器的值就會減 1，而跳躍的新指標就會參考 LP 指令的設定值來決定。

例如指令『LC 32』、『LP 16』，的整個動作步驟如下：

- a. 將迴圈暫存器 32 減 1。
- b. 檢查迴圈暫存器是否為 0。
- c. 如果迴圈暫存器為 0，New Address = Next Address。
- d. 如果迴圈不為 0，New Address = 16。有一點要注意的是如果當迴圈暫存器已

經被減成 0，又遇上 LP 指令時，這時候迴圈暫存器是不會再被減成-1，而是會產生特殊情況。所以迴圈暫存器被減成 0 以後，LP 指令的工作是不可預期的，所以要使用 LP 指令之前請注意是否已經設定 LC 指令了。

- (4) **BE (Branch If Event)**指令和 LP 指令又有點類似，因為他們都是一個有條件跳躍的指令。LP 的跳躍條件是迴圈暫存器，而 BE 的跳躍條件是 Event Bit。當波形執行到 BE 指令時，Acute PG 會立即判斷 Event Bit，如果 Event Bit 為 1 就會跳到 BE 所設定的新位址，Event Bit 為 0 的話，則繼續執行下一個位址。

- (5) **SE (Set Event)**是一個選擇事件的指令，Acute PG 有 3 個外部事件通道 (Event\_1、Event\_2、Event\_3)<sup>註1</sup>及 1 個內部鍵盤事件(Keyboard Event)，所以總共有 4 個事件來源。Acute PG 將這 4 個事件來源編輯成 16 種事件型態以茲利用，而這個事件型態會被存到一個 Acute PG 的內部暫存器，稱之為事件暫存器(Event Register)。這些事件型態包括：

- Keyboard Event

- Event\_1
- Event\_2
- Event\_3<sup>註2</sup>
- Event\_1 or Event\_2
- Event\_1 or Event\_3<sup>註2</sup>
- Event\_2 or Event\_3<sup>註2</sup>
- Event\_1 or Event\_2 or Event\_3<sup>註2</sup>

註 1：PKPG 只有兩個外部事件通道，所以共有 3 個事件來源，編輯成 8 種事件型態。

註 2：PKPG 不支援這種事件組合。

另外 8 種事件型態就是這 8 種型態的反相。例如事件暫存器的設定值為上述 8 種型態之一，此時不管從外部或是內部的事件通道都會被 Acute PG 所監控，Acute PG 會隨時比較目前的事件通道的狀態是否與事件暫存器的事件型態相同。如果相同就會將 Acute-PG 內部的旗標暫存器(Flag Register)的 Event Bit 設成 1(True)，如果不同就會將 Event Bit 設成 0(False)。然而反相型態就是當事件通道的狀態與事件暫存器的事件型態相同時，Acute PG 會將 Event Bit 設成 0，不同時則會將 Event Bit 設成 1。

而運用這個 Event Bit 的指令有兩個，一是 WE (Wait Event)另一是 BE (Branch If Event)。WE 指令是當波形執行到這個指令時，整個 Acute PG 會因此而暫停，而波形會停留在 WE 指令當時的波形，一直到 Event Bit 為 1 時，才會再繼續往下執行。

(6) **OE (Output Enable)**指令可以設定每個通道的輸出為高阻抗或是訊號輸出。

OE 指令的參數是一組 4 位數的 16 進位數字，這個值代表 16 個通道的狀態。

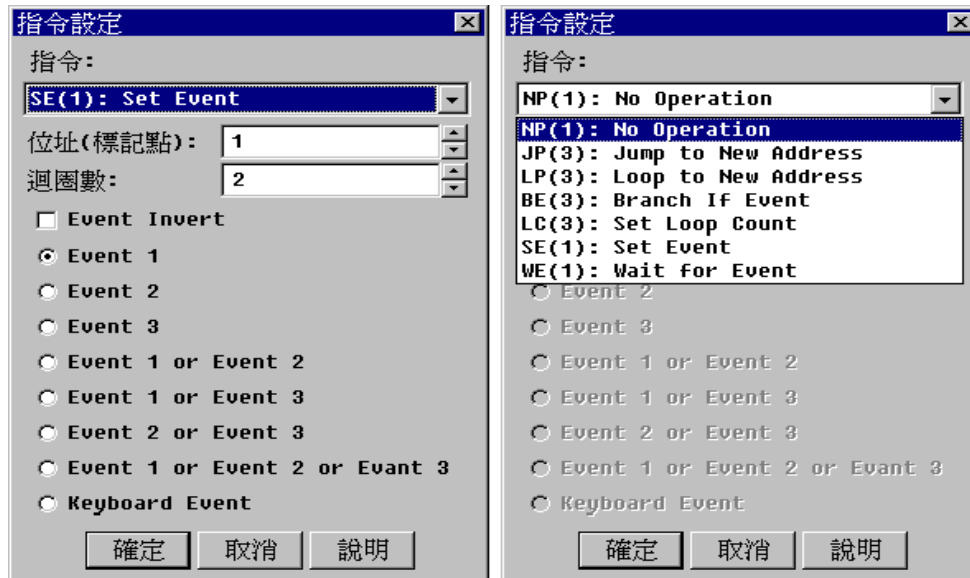
例如 FFF0 代表通道 0 到通道 3 為高阻抗，通道 4 到通道 15 則為正常訊號輸出。

#### 指令的編輯方法：

指令在編輯之前，必須先將游標模式切成指標模式，然後將游標移至波形區需要編



輯指令的位置上，然後按下滑鼠左鍵，這時候會出現一條紅色的時間游標。這時編輯指令的按鈕就會從灰色變成彩色，也代表編輯指令的功能是可以使用的。在按下編輯指令鈕時，就是對時間游標的位置做編輯指令的動作，此時也會出現如下圖的對話盒。



圖一

圖二

左圖是原始編輯指令對話盒的外觀，右圖則是顯示指令列表的外觀。使用時，首先選擇指令列表裡的指令，程式會根據選擇的指令，打開相關的控制項目。例如指令為 LC 時，迴圈數的項目才會被啟動，也才能改變迴圈值。設定好指令的項目以後，並按下確定鈕，指令就會出現在指令訊號組。

如果要將整區的指令清除成 NP(No Operation)時，用法比較特殊。將游標移至波形區的指令訊號組上，使用標記區塊的方式，將要清除的指令訊號組範圍標記起來，此時編輯指令鈕亦會從灰色變成彩色，但是按下編輯指令鈕時，會出現一個對話盒，內容是詢問是否要將指令區清除，只要按下確定的按鈕就會被清除了。

詳細介紹，請參閱

[PG Function 指令使用方法](#)

### 3.25 波形檢查



由於指令的運用有一些限制，所以在波形輸出之前最好先做波形檢查，否則輸出的波形可能會出現預期以外的情況。波形的檢查主要是檢查一個指令與另一個指令之間的距離，以及跳躍指令的跳躍點與其他指令的距離。以下是檢查方式之詳述(指令集請參考『[編輯指令](#)』一節)。


下列指令的前後 3 個指令週期(Clocks)之內不能有其他的指令，JP (Jump)、LP (Loop) 和 BE (Branch If Event)等 3 個跳躍指令。例如 JP 指令出現在 12 的位置，因為 JP 用了 3 個指令週期，所以 12、13 和 14 都屬於 JP 指令週期，而其前後 3 個指令週期就是 9、10、11 和 15、16、17。也就是說在前述的那些位置上不能有其他的指令，否則波形輸出時將會出現不可預期的結果。

上述的 3 個跳躍指令的跳躍點前後 3 個週期亦是不能有其他的指令。例如 LP 30 這個指令就是當波形執行到這個指令時，會跳躍到 30 這個週期繼續執行，而 30 這個點就是跳躍點。同理，27、28、29 和 31、32、33 就是它的前後 3 個週期，在這區間一樣不能有其他的指令。同時跳躍點也不能超出波形長度的設定範圍。

其他如 LC (Loop Count)、SE (Set Event)和 WE (Wait Event)等指令，是可以連在一起使用，沒有間隔的要求。

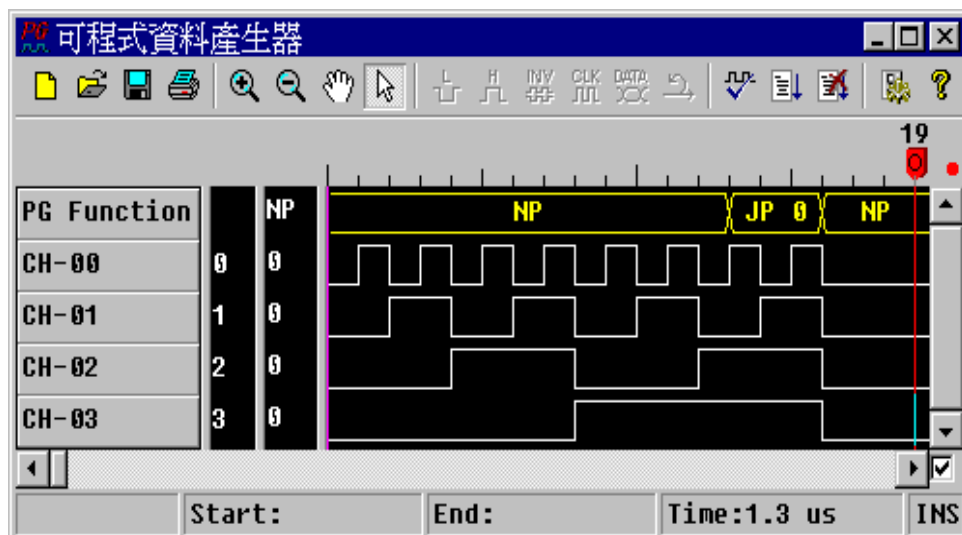
指令檢查時還有一項重要的檢查，那就是檢查是否有不明的指令，不明的指令在波形區的畫面上會出現『XX』的字串。

### 3.26 波形輸出與停止輸出

 當檢查波形如果一切正常時，就可以按下『波形輸出』鈕，此時波形就會從 Acute PG 的輸出端將波形輸出。由於系統必須將編輯好的波形先傳送到 Acute PG 的主機裡面，再通知 Acute PG 開始執行波形資料，所以當按下『波形輸出』鈕時並不是立即輸出，大概會延遲 0.5 秒的時間，所以使用時可以運用一些技巧。首先，先將標的物與 Acute- PG 連接，開始編輯波形，波形的最前端最好空出一小段時間，以便加入兩個指令，一個是 Set Keyboard Event，另一個是 Wait Event。加上這兩個指令後，當按下『波形輸出』時，Acute PG 的執行會停留再 Wait Event 上。此時可以操作您的標的物或是電路，以達到開始運作的狀態，這時再按下『事件熱鍵』就可以繼續輸出 Wait Event 之後的波形。


開始輸出波形時，狀態列會出現如同波形輸出的樣子，此時代表波形正在輸出。如果您想要停止波形繼續輸出，只要按一下『停止輸出』鈕，就可以讓波形輸出停下來。然而在按下『波形輸出』之前或是在按下『停止輸出』之後，Acute PG 的輸出端都是呈現高阻抗狀態。所以您的電路如果會因為高阻抗而造成工作不正常，此時就必須將上述連接標的物至 Acute PG 的時間，改成按下『輸出波形』和按下『事件熱鍵』之間來做連接即可。

下面是一個 4Bits Counter 的例子：



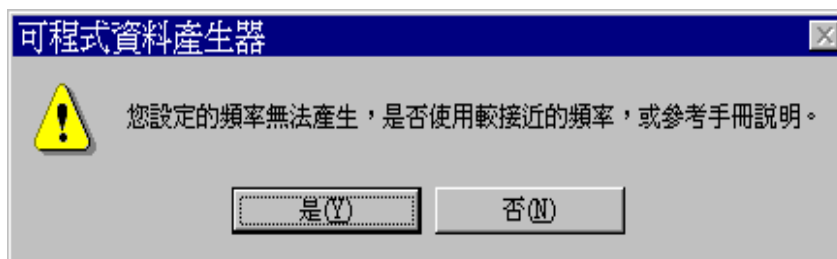
### 3.27 環境參數設定



 環境設定主要項目有 2 個，一是基本頻率的設定、二是事件熱鍵的設定。

調整基本頻率的方法是：先選擇『頻率形式』，可使用內部時脈、外部時脈或是混合時脈，然後調整頻率值即可。

當設定好頻率值後，如果按下確定鈕時會出現下方的對話盒時，就代表設定的頻率值是 Acute PG 所無法產生的頻率。此時對話盒會問是否要使用最接近設定值的頻率，如果回答『是』的話，Acute PG 就會自動選擇一個最接近的頻率當作基準時脈。

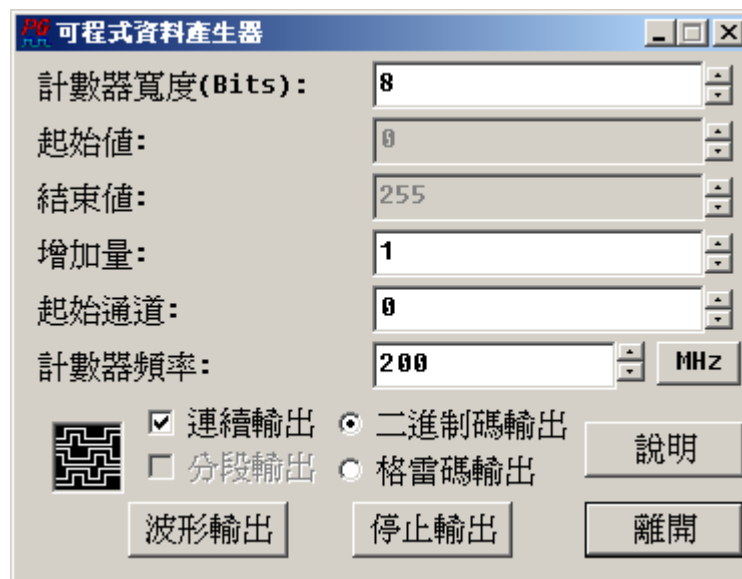


事件熱鍵設定：選擇哪一個按鍵當作『事件鍵』，有 Space、Return 可供選擇。而所謂的『事件鍵』就是當波形正在輸出的狀態下，按下這個熱鍵就會產生一個事件，並傳送到 Acute PG。

## 4. 同步計數器

**3456** 同步計數器可以很快製作出一個計數器的訊號，可以從任何一個值計數到另一個值，而且讓計數器重覆不斷的計數。由於 Acute PG 的記憶體深度有限，如果計數的範圍超過 Acute PG 的記憶體深度時，亦可以使用分段輸出的方式，來輸出計數器的訊號。計數器的位元數及頻率是可以依據使用的需要來做調整。

同步計數器使用說明：



### (1) 計數器寬度

計數器的寬度最大設可定範圍會依據使用的機型而有所不同。如 PGx020 最多寬度只能設成 20，PGx050 可設成 50，而 PKPG 只能設成 16。當進入同步計數器功能時，起始值和結束值會依計數器寬度自動改變起始值固定為 0，結束值為計數器寬度所能計數的最大值。例如計數器寬度設成 5 時，起始值為 0，結束值為  $31(2^5-1)$ 。直到改變了起始值或結束值後，它們就不會依計數器寬度來改變。

### (2) 起始值

起始值為計數器開始計數的值，如果要更改起始值只要將游標在起始值右邊的上下旋轉器輕按一下就可改變。當起始值大於計數器寬度所能產生的最大值時，起始值自動會被視為該值除以計數器寬度所能產生的最大值加 1 的餘

數。例如起始值為 45，而計數器的寬到為 5 時，起始值會被視為  $45-2^5=45-32=13$ 。

### (3) 結束值

結束值的設定方法與起始值相同。但是結束值是指當計數器計數到結束值時才算結束，所以當增加量不等於 1 時，計數器可能會超過結束值，此時計數器會根據起始值與增加值來計算下一個計數值。例如，當起始值為 0、結束值為 7、增加值為 3 時，整個計數器所產生的值為 0、3、6、1、4、7。

### (4) 增加量

增加量的設定方法與起始值和結束值的設定方法相同。增加量的意義是指每一計數的累進值。增加量亦可以設定成負數，一旦設成負數就代表是一個倒數計數器。

### (5) 起始通道

起始通道為輸出通道的最低位元，例如起始通道為 5、計數器的寬度為 6，則輸出通道為 10、9、8、7、6、5。通道 10 為最高位元，通道 5 為最低位元。

### (6) 計數器頻率

計數器的頻率是每計數一次的頻率。如果頻率為 10MHz 就代表每 100ns 計數一次。可以使用的頻率請參考頻率設定一節的說明。

### (7) 二進制碼輸出/格雷碼輸出

可以選擇訊號輸出方式為二進制碼(Binary code)或是格雷碼(Gray code)的輸出方式。


使用同步計數器可以將一串連續的數值輸出，使用『重覆輸出』的功能讓這些數值不斷的重覆。如果要輸出的數值超出 Acute PG 的記憶體深度時，也可以使用『分段輸出』的功能來達到輸出數值的目的。因為 Acute PG 有 64K 的記憶體深度，也就是說每個通道的深度為 64K。但是 64K 的記憶體深度中有 12 個是系統控制用，系統控制用的的長度是根據不同的功能會有所不同，在同步計數器中所需要的長

度為 12 個。所以整個同步計數器所能用的記憶深度為  $64K-12=65536-12=65524$ 。因此當同步計數器設定的值會超過 65524 時，程式會詢問要不要分段輸出，如果不分段輸出的話會有兩種情形。第一、如果沒有設定重覆輸出時，整個資料只能輸出到記憶體的底部就會被停止。第二、如果有設定重覆輸出時程式將會出現錯誤訊息。

由於 Acute PG 的記憶深度為 64K，所以超過 64K 的資料在使用『分段輸出』時，系統的處理方式分為下列幾個步驟：第一、將第一部分的資料寫入 Acute PG 的記憶體中。第二、輸出資料直到記憶體的底部。第三、將輸出訊號暫時停止。第四、再將下一組資料寫入 Acute PG 的記憶體中。重覆第二至第四步驟直到資料結束為止。由於輸出訊號會被暫停來輸入下一組的資料，所以整個輸出訊號將會有所延遲。也就是說當您要的是連續而且中途不能停頓的訊號，分段功能將是不能使用的。

當設定結束後，只要按下『波形輸出』就可以開始輸出波形，此時畫面左下角的輸出狀態圖示上會出現波形流動的圖示。如要停止波形輸出只要按下『停止輸出』即可。

## 5. 非同步計數器

 所謂非同步計數器就是一個連波計數器。Acute PG 能輕易的製作出一個連波計數器，可以從任何一個值計數到任何一個值，而且可以讓計數器重覆不斷的計數。由於 Acute-PG 的記憶體深度有限，如果計數的範圍超過 Acute PG 的記憶體深度時，亦可以使用分段輸出的方式，來輸出計數器的訊號。計數器的位元數及頻率是可以依據使用的需要來做調整。

### 非同步計數器使用說明：



#### (1) 計數器寬度

計數器的寬度最大可定範圍會依據使用的機型而有所不同。如 PGx020 最多寬度只能設成 20，PGx050 可設成 50，而 PKPG 只能設成 16。當進入同步計數器功能時，起始值和結束值會依計數器寬度自動改變起始值固定為 0，結束值為計數器寬度所能計數的最大值。例如計數器寬度設成 5 時，起始值為 0，結束值為  $31(2^5-1)$ 。直到改變了起始值或結束值後，它們就不會依計數器寬度來改變。

#### (2) 起始值

起始值為計數器開始計數的值，如果要更改起始值只要將游標在起始值右邊的上下旋轉器輕按一下就可改變。當起始值大於計數器寬度所能產生的最大



值時，起始值自動會被視為該值除以計數器寬度所能產生的最大值加 1 的餘數。例如起始值為 45，而計數器的寬到為 5 時，起始值會被視為  $45-2^5=45-32=13$ 。

### (3) 結束值

結束值的設定方法與起始值相同。但是結束值是指當計數器計數到結束值時才算結束，所以當增加量不等於 1 時，計數器可能會超過結束值，此時計數器會根據起始值與增加值來計算下一個計數值。例如，當起始值為 0、結束值為 7、增加值為 3 時，整個計數器所產生的值為 0、3、6、1、4、7。

### (4) 增加量

增加量的設定方法與起始值和結束值的設定方法相同。增加量的意義是指每一計數的累進值。增加量亦可以設定成負數，一旦設成負數就代表是一個倒數計數器。

### (5) 起始通道

起始通道為輸出通道的最低位元，例如起始通道為 5，計數器的寬度為 6，則輸出通道為 10、9、8、7、6、5。通道 10 為最高位元，通道 5 為最低位元。

### (6) 計數器頻率

計數器的頻率是每計數一次的頻率。如果頻率為 10MHz 就代表每 100ns 計數一次。但是在非同步計數器的頻率設定上會比同步計數器低，請參考以下的說明。

### (7) 每一級之延遲

每一級的延遲是每一步計數的時間乘以延遲設定。例如，計數器頻率為 10MHz，則每一步計數的時間為 100ns，而每一級延遲的設定為 1/10，就代表每一級的延遲為 10ns。

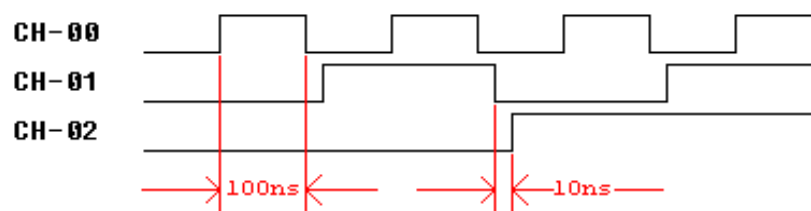
使用非同步計數器可以將一串連續的數值輸出，使用『重覆輸出』的功能讓這些數值不斷的重覆。在非同步計數器中是無法使用『分段輸出』的功能。因為

Acute PG 有 64K 的記憶深度，也就是說每個通道的深度為 64K。但是 64K 的記憶深度中有 30 個是系統控制用，系統控制用的的長度是根據不同的功能會有所不同，在非同步計數器中所需要的長度為 30 個，而且在使用重複輸出時記憶體深度只有一半可以使用。所以非同步計數器所能用的記憶深度分為兩種情況。一為在沒有設定『重覆輸出』時， $64K-30=65536-30=65506$ 。另一為在有設定『重覆輸出』時， $(64K-30)/2=(65536-30)/2=32753$ 。非同步計數器設定的值不能超過可用值，如果超過程式會出現錯誤訊息。

當設定結束後，只要按下『波形輸出』就可以開始輸出波形，此時畫面左下角的輸出狀態圖示上會出現波形流動的圖示。如要停止波形輸出只要按下『停止輸出』即可。

下面是非同步計數器的實例：

計數器寬度=3、起始值=0、結束值=7、增加值=1、頻率=10MHz、延遲=1/10、  
起始通道=0，結果為：

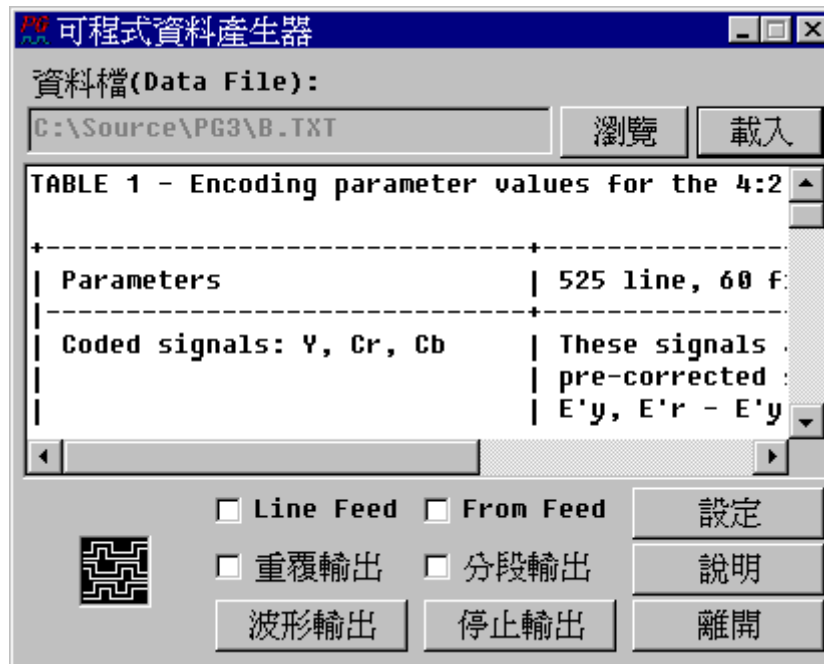


## 6. 並列訊號產生器



並列訊號產生器可以產生 SPP Mode 的印表機訊號，只要將 Acute PG 依設定中的腳位定義接上印表機，就可以將文字資料或是圖形資料從印表機列印出來。

並列訊號產生器使用方法：



並列訊號產生器的用法有二。一是，輸入一個印表機可以識別的檔案，這個檔案可以是純文字檔或是含有印表機控制碼的檔案。二是，直接在文字編輯區裡輸入純文字資料。如果輸入檔案的是一個純文字檔案，可以按下載入鈕，將檔案載入編輯區中做修改。資料的尾端也可以加入跳行(LINE FEED)或跳頁(FORM FEED)的指令。



資料輸出以前必需先設定通道，並且連接到印表機上，記得要接上地線，腳位的連接請參考附註說明。要設定的腳位有 Data[7..0]、nStrobe、nSelectIn、nAutoFeed 和 Busy 等接腳。除了 Busy 接腳是輸入訊號以外，其他皆為輸出訊號。所以 Busy 可以設定的通道為 Event\_1、Event\_2 或是 Event\_3，這三個 Event 通道是 Acute PG 的輸入通道，他可以通知 Acute PG 是否要繼續輸出資料。所以並列訊號產生器會將產生的訊號輸出至 Data[7..0]、nStrobe、nSelectIn 和 nAutoFeed 等腳位，但是當印表機送出 Busy 訊號時，PG 就會停止送出資料直到 Busy 訊號結束有後，才會繼續輸出資料。

當您資料因太大而超過 Acute PG 的記憶深度時，程式會主動通知是否要分段輸出。如果要列印多份時亦可選擇『重覆輸出』的選項。

當設定結束後，只要按下『波形輸出』就可以開始輸出波形，此時畫面左下角的輸出狀態圖示上會出現波形流動的圖示。如要停止波形輸出只要按下『停止輸出』即可。

**附註：(印表機腳位表)**

Data [7.0] = Pin 9.2

nStrobe = Pin 1

NSelectIn = Pin 17

NAutoFeed = Pin 14

Busy = Pin 11

Ground = Pin 25.18

## 7. 串列訊號產生器



串列訊號產生器可以產生 TTL 準位的 RS232 的訊號，而且這個 RS232 訊號是標準的訊號準位，所以 PG1000/PG2000 不需要額外增加驅動零件就可以直接接到 RS232 的輸入埠上。(若要將 PKPG 的 RS232 輸出訊號與標準的 RS232 連接的話，必須要加一個驅動器 “Drive IC”，將 TTL 準位變成正負 12 伏特的訊號。)

### 串列訊號產生器使用方法：



#### (1) 傳輸速率(Baud Rate)

可以從 110 一直到 256000 等所有標準的速率。

#### (2) 同位檢查(Parity)

None, Odd, Even, Mark, Space。

#### (3) 資料位元(Data Bits)

4~7

#### (4) 停止位元(Stop Bits)

1、1.5、2。

#### (5) 輸出埠(Output Port)

Acute-PG 的 Extended Pod 包含了兩個 RS232 的輸出埠，輸出埠的設定可以任選其中一個輸出埠或是兩個輸出埠同時輸出，也可以讓兩個輸出埠產生差動訊號。(只有 PG1000 和 PG2000 可用)

#### (6) 同步輸出通道

PG1000/PG2000 選擇任何一個通道來輸出未經驅動成 RS232 標準電位的 5V 或 3.3V 訊號(根據輸出 Pod 而定)。

#### (7) 資料格式

串列訊號產生器可以解釋 5 種資料格式，分別是 BIN、OCT、DEC、HEX、ASC，這些格式都是利用文字檔的模式來表示。BIN 是二進位的格式，OCT 是八進位的格式，DEC 是十進位的格式，HEX 是十六進位格式。以上 4 種格式必需是一行一筆資料，也就是每一筆資料都必需有 Carriage Return 和 New Line。例如，BIN 格式要輸出 1、2、3，資料內容如下：

**00000001**

**00000010**

**00000011**

如果格式設定為 ASC 模式，就不須加入 Carriage Return 和 New Line 等字元，所以任何字元都會被輸出。

串列訊號產生器的資料的輸入方式有二。一是，輸入一個資料檔案。二是，直接在文字編輯區裡輸入純文字資料。資料檔案如果要修改，可以按下載入鈕，將檔案載入編輯區中做修改。

由於資料位元、停止位元及同位位元的設定關係，所以使用 Acute PG 的記憶體情況都不一樣。當您資料因太大而超過 Acute PG 的記憶深度時，程式會主動通知是否要分段輸出。亦可選擇『重覆輸出』的選項讓資料不斷的重覆輸出。

當設定結束後，只要按下『波形輸出』就可以開始輸出波形，此時畫面左下角的

輸出狀態圖示上會出現波形流動的圖示。如要停止波形輸出只要按下『停止輸出』即可。

## 8. I<sup>2</sup>C 訊號產生器



**I<sup>2</sup>C** I<sup>2</sup>C 是一種串列通訊的標準，是由 Philips 公司所制定的一種通訊規格。他只用兩條訊號線，一條是時脈線(SCL)和一條資料線(SDA)所構成。其傳輸的方式是雙向的，資料格式分為 7Bits，8Bits 和 10Bits 三種。

I<sup>2</sup>C 訊號產生器的輸出模式分為[預設輸出]與[自訂輸出]，如果您選用的是[自訂輸出]，則您可以選擇 SDA 與 SCL 的輸出通道，但是所輸出的波形將不具有 OPEN COLLECTOR 的特性；如果您選用的模式為[預設輸出]，並且使用本公司 PG2020, PG2050 以及專門為 I<sup>2</sup>C 訊號產生器所設計的輸出腳(PG2020,PG2050 的 Extended Pod 的 I2CC, I2CD 腳)則將具有 OPEN COLLECTOR 的特性，但是如果您使用的是 PKPG 系列(PKPG2016, PKPG2116, PKPG2116+)，只能選擇[自訂輸出]，而且將不具有 OPEN COLLECTOR 的特性。

I<sup>2</sup>C 訊號產生器可以設定要輸出的資料，您可以自行編輯您要輸出的資料或是載入文字檔案。格式說明如下：



\*\*\*\*\*

7 bits Address:

Aw=12;

D=10,20,30;

Ar=3f;

D=0;

Aw=46;

D=21,3a;

\*\*\*\*\*

8 bits Address:

A=25;

D=0,0,0;

A=7f;

D=0;

A=8C;

D=21,3a;

\*\*\*\*\*

10 bits Address:

Aw=12c;

D=10,20,30;

Aw=23f;

D=4c;

Ar=18a;

D=0,0;

\*\*\*\*\*

A 表示 I<sup>2</sup>C 位址(address)，D 表示 I<sup>2</sup>C 資料(data)。在 I<sup>2</sup>C 位址數 7 bits 和 10 bits 的命令格式相同；Aw=12; 表示寫入位址(write address) 12h, Ar=3f; 表示讀取位址(read address) 3fh。而 I<sup>2</sup>C 位址數 8 bits 和上述 7 bits 的差異主要是將 r/w bits 納入 address, A=25; 即為 Ar=12,上述的數值皆為十六進制。

每行命令列以[;]作為結尾，若沒有使用[;]作為結尾，程式將會忽略該行指令，而在資料 D 中，每一筆資料以[,]間隔，如沒有使用[,]作為間隔則會忽略該筆資料。對於 I<sup>2</sup>C 編輯區的內容，程式會做簡易的格式檢查，並提示訊息告知編輯的格式是否有錯誤。

### (1) 自訂輸出

#### a. SCL 輸出通道

輸入要輸出 I2C 時脈通道。

#### b. SDA 輸出通道

輸入要輸出 I2C 資料通道。

#### c. 等候外部事件(EV1)

若勾選該項，EV1 腳若有脈波輸入，則資料產生器才會輸出 I2C 訊號。

### (2) 檔案設定

#### a. 存檔

將編輯區中的內容儲存成文字檔(.txt)

#### b. 讀檔

將儲存的檔案讀回，並將內容顯示於編輯區

### (3) I<sup>2</sup>C 位址數

選擇 I<sup>2</sup>C 位址(Address)模式

**(4) I<sup>2</sup>C 速度(bit/s)**

輸入 I<sup>2</sup>C 時脈

**(5) 編輯輸出設定(Hex)**

**a. 範本**

只要按下範本，會根據當 I<sup>2</sup>C 位址數，在編輯區中顯示每個 I<sup>2</sup>C 位址數的命令範例

**b. 清除資料**

清除編輯區的內容

**c. 轉換成波形**

會將所編輯 I<sup>2</sup>C 訊號命令，轉換成波形，顯示在波形編輯器上，可以讓使用者在波形的狀態下修改 I<sup>2</sup>C 訊號。

**d. 連續輸出**

可以連續輸出 I<sup>2</sup>C 訊號，不過有一點要注意，當要輸出的 I<sup>2</sup>C 訊號大於 PG 記憶深度，連續輸出的功能會 disable

**e. 波形輸出和停止輸出**

輸出 I<sup>2</sup>C 訊號和停止輸出 I<sup>2</sup>C 訊號

**(6) 說明**

呼叫說明文件

**(7) 離開**

離開 I<sup>2</sup>C 訊號產生器，回到 PG 主畫面

## 9. 資料檔轉成波形



任何資料檔可以透過轉換變成波形，再藉由 Acute PG 來輸出。設定方式如下：

資料檔轉成波形的使用方法：



### (1) 資料寬度(Bytes)

資料寬度是以 Byte 為計算單位，和檔案的儲存單位是一樣的。所謂資料寬度就是要將整個資料檔案以每次多少個 Bytes 來輸出。如果檔案長度為 1024Bytes，而資料寬度設定成 2Bytes，那整個資料檔就會被分成 512 組來輸出。

### (2) 輸出寬度(Bits)

輸出寬度的設定值可以從 1 設到最大輸出寬度，最大輸出寬度是指資料寬度換算成 Bits 後的值。也就是把資料寬度乘以 8。

### (3) 起始通道

起始通道為輸出通道的最低位元，例如起始通道為 5、輸出寬度為 6 時，則輸出通道為 10,9,8,7,6,5 等 6 個通道。通道 10 為最高位元，通道 5 為最低位元。

### (4) 輸出頻率

輸出頻率是每輸出一筆資料的頻率。如果頻率為 10MHz 就代表每 100ns 輸出一筆資料。可以使用的頻率請參考頻率設定一節的說明。

當您資料因太大而超過 Acute PG 的記憶深度時，程式會主動通知是否要分段輸出。如果要將資料不斷的重覆輸出時亦可選擇『重覆輸出』的選項。

當設定結束後，只要按下『波形輸出』就可以開始輸出波形，此時畫面右上角的輸出狀態圖示上會出現波形流動的圖示。如要停止波形輸出只要按下『停止輸出』即可。

## 10. 文字(向量)檔轉成波形



文字(向量)檔是 PG-Editor 專用的向量檔，這個檔案的附屬檔名為 PGV。所以您可以用任何的文字編輯程式先將向量檔寫好，然後在用這個轉換程式將向量檔轉成波形。

### PGV (PG Vector File) 資料格式說明：

PGV 有下列幾個指令(Keyword)，**INPUTS**、**UNIT**、**INTERVAL**、**FREQUENCY**、**PATTERN**、**ASSIGN**、**RADIX**。

#### (1) INPUTS

設定訊號名稱。每一個名稱用空白分隔開來，名稱可以為文字或是數字如果為 Bus(Group)時，可以用中括號來指定。例如某個 Bus 為 4 個通道所組成那就可以用 A[3..0]來代替，A[3..0]就代表 A3、A2、A1、A0 等 4 個訊號。**注意！『PG\_Function』**是一個保留字，如果在 **INPUTS** 指令的定義中有『**PG\_Function**』，就代表 **PATTERN** 中會包含 PG 的專用指令。而且『**PG\_Function**』也不能配合 **ASSIGN** 指令使用。

#### (2) ASSIGN

用來指定 **INPUTS** 指令所定義的訊號名稱之通道值。指定的方式如下：

**INPUTS A [3..0]; ASSIGN A [3..0]=8..11;**就表示 A3=CH8, A2=CH9,

A1=CH10, A0=CH11。**INPUTS Reset;ASSIGN Reset=32;**就表示 Reset=CH32

**注意！**未被指定的訊號將會根據通道順序(由小到大)來安排。**ASSIGN** 不能指定相同的通道或是超出 Acute PG 所能容許的最大通道。

#### (3) UNIT

設定 **PATTERN** 區域的單位。可以設定成 **ns**、**us** 和 **ms** 等三種單位。這個設定是配合有 Time Stamp 的 **PATTERN** 來使用。如果 **PATTERN** 沒有 Time Stamp 就不需要設定這個指令。

#### (4) INTERVAL

設定每個 PATTERN 之間的時間。如果設定了 INTERVAL 參數後就不能使用 Time Stamp 的用法。

#### (5) FREQUENCY

用途與 INTERVAL 一樣。只是 FREQUENCY 所指定的單位是用頻率，INTERVAL 所用的單位是時間。

#### (6) RADIX

設定 PATTERN 區域的進位值。如果 PATTERN 區域的數值有進位識別符號時，此時就要將 RADIX 的值設成 AUTO。例如 RADIX 設定成 AUTO 時，PATTERN 區域的某一數值為 35 與 35h 是不一樣的，但是當 Radix 設成 HEX(十六進位)時，這兩個值就是一樣的。但是當 RADIX 設成 DEC 時，而 PATTERN 區域的值為 35h 時，卻是會被當成 35。

RADIX 共有五種定義：

AUTO：由數值的進位識別符號決定。

HEX：16 進位。

DEC：10 進位。

OCT：8 進位。

BIN：2 進位。

而在 PATTERN 區域的進位識別符號定義是『h』代表 16 進位、『o』代表 8 進位和『b』代表 2 進位，而不加任何識別符號就代表 10 進位。

#### (7) PATTERN

是用來定義波形資料的區域。這個區域包含了兩個部份，一個是時間部份一個是波形部份。時間部份稱為『Time Stamp』，時間部份和波形部份用『>』(大於符號)來做分隔。時間部份也可以省略，此時波形部份每一行的時間就

會由 **INTERVAL** 或是 **FREQUENCY** 來指定，所以 **INTERVAL** 和 **FREQUENCY** 只能選用一種。如果使用 Time Stamp 方式，時間部份的單位就是 **UNIT** 所設定的值。波形部份就是根據 **INPUTS** 的定義順序來描述波形。(請參考下面例子)

#### (8) 注意事項

- a. 每個指令後面需用一個『;』(分號)來做結尾，**PATTERN** 後面的所有波形資料都屬於 **PATTERN** 指令，所以不用加『;』，一直到所有波形描述結束後才需要加『;』。
- b. 指令沒有區分大小寫，所以大小寫都視為同一個指令。每個指令與參數之間至少要有一個空白。參數與參數之間也是一樣需要有一個空白。但是**通道描述不能有空白，因為整個通道描述是當成一個參數**。例如：**3,5,6** 或是 **7..0**。
- c. 需要一些註解時，可以用『%』符號將要說明或是注釋的部份框起來，也就是註解的前面及後面各加入一個『%』符號即可。如果需要單行的註解時，可以直接加『//』兩個斜線符號，該符號一直到換行符號之前都會被認定為註解。這些註解都不會影響任何向量檔的結果。
- d. 使用及指定的通道數不能超過 Acute PG 的使用上限(PGx020P 為 19，PGx050P 為 49，PKPG 為 15)。
- e. PGV 向量檔的指令部份是不區分大小寫，但是訊號名稱是會區別大小寫的例如 Data [7..0]與 data [7..0]是不一樣的所以在 **INPUTS** 指令及 **ASSIGN** 指令的指定參數的要一致，否則會發現無法指定通道的情形。

對話盒中有『重覆輸出』及『按鍵後才輸出波形』的選項，是在轉換的過程中自動加入 Acute PG 控制指令。如果向量檔中已經包含控制指令的話，就不需要在加入這些選項。對話盒中還有一個選項是有關頻率的選項，這個選項是給沒有 Time Stamp 的 **PATTERN** 選用的。沒有 Time Stamp 的頻率是根據 **FREQUENCY** 或 **INTERVAL** 指令來設定，但也可以直接從對話盒來做



變更。

請參考下面例子(使用 Time Stamp 的例子)：

**INPUTS Reset KeyPulseAck CmdWrite data[7..0] CmdClkIn Clk100K**

**AdjClkIn2 AdjClkIn1 AddrClkIn;**

**Radix HEX;**

**UNIT ns;**

**PATTERN**

**0.0> 0 0 0 00 0 0 0 0 0**

**40.0> 1 0 0 00 0 0 0 0 0**

**50.0> 1 0 0 01 0 0 0 0 0**

**100.0> 1 0 0 02 0 0 0 0 0**

**150.0> 1 0 0 03 0 0 0 0 0**

**200.0> 1 0 0 04 0 0 0 0 0**

**250.0> 1 0 0 05 0 0 0 0 0**

**300.0> 1 0 0 06 0 0 0 0 0**

**350.0> 1 0 0 07 0 0 0 0 0**

**400.0> 1 0 0 08 0 0 0 0 0**

**450.0> 1 0 0 09 0 0 0 0 0**

**500.0> 1 0 0 0A 0 0 0 0 0**

**550.0> 1 0 0 0B 0 0 0 0 0**

**600.0> 1 0 0 0C 0 0 0 0 0**

**650.0> 1 0 0 0D 0 0 0 0 0**

**700.0> 1 0 0 0E 0 0 0 0 0**

**750.0> 1 0 0 0F 0 0 0 0 0**

**800.0> 1 0 0 10 0 0 0 0 0**

```
850.0> 1 0 0 11 0 0 0 0 0  
900.0> 1 0 0 12 0 0 0 0 0  
950.0> 1 0 0 13 0 0 0 0 0  
;
```

請參考下面例子(使用 INTERVAL 的例子)：

```
INPUTS Reset KeyPulseAck CmdWrite data[7..0] CmdClkIn Clk100K  
AdjClkIn2 AdjClkIn1 AddrClkIn ;  
Radix HEX;  
INTERVAL 12.5ns ;  
%FREQUENCY 8MHz ;%  
PATTERN  
0 0 0 00 0 0 0 0 0  
1 0 0 00 0 0 0 0 0  
1 0 0 01 0 0 0 0 0  
1 0 0 02 0 0 0 0 0  
1 0 0 03 0 0 0 0 0  
1 0 0 04 0 0 0 0 0  
1 0 0 05 0 0 0 0 0  
1 0 0 06 0 0 0 0 0  
1 0 0 07 0 0 0 0 0  
1 0 0 08 0 0 0 0 0  
1 0 0 09 0 0 0 0 0  
1 0 0 0A 0 0 0 0 0  
1 0 0 0B 0 0 0 0 0  
1 0 0 0C 0 0 0 0 0
```

**1 0 0 0 D 0 0 0 0 0**

**1 0 0 0 E 0 0 0 0 0**

**1 0 0 0 F 0 0 0 0 0**

;

關於 PGV 檔案更細節的部份，請參閱：

[使用文字編輯器編輯文字向量檔](#)

## 11. Acute LA 波形檔轉換



如果您已經使用了本公司的 Acute LA 時，您就可以用 Acute LA 來擷取電路上的波形。您只要取得該波形檔，就可以透過轉換程式將 Acute LA 的波形檔轉換成 Acute PG 的波形檔。PG-Editor 程式會在轉換結束的時候，將轉換結果放入波形編輯器中，此時您可以再經由波形編輯器加以編修，然後由 Acute PG 來輸出。也就是說當您沒有對方的電路所產生的波形時，也可以藉由 Acute LA 及 Acute PG 讓遠方的電路訊號在您這裡重現。

波形轉換時要注意目前只能轉換 Acute LA 64K 深度模式的波形檔。因為 Acute PG 可用的通道數為 16(PKPG)、20(PGx020)及 50(PGx050)，所以 Acute LA 的波形檔，不能設定超過 16、20 或是 50 個通道，否則將出現錯誤訊息。

**LA Viewer** (Acute LA 系列的操作軟體) 1.50 版之後，就支援 PGW 的波形格式，所以您有 Acute LA 系列邏輯分析儀時，請用 LA Viewer 1.5 版之後的版本。在儲存波形檔時，只要選擇存成 PGW 格式即可，詳細使用方法請參考 LA Viewer 使用手冊。

## 12. Altera 波形檔轉換



FPGA 與 PLD 可程式邏輯的運用已經越來越廣泛，許多的設計師都會用 FPGA 與 PLD 來當作控制電路，所以設計這些可程式邏輯時，零件商都會提供許多設計用的應用軟體，通常都會包括模擬程式，讓您在軟體上就可以輕鬆的模擬 FPGA 和 PLD 的功能。

但是一旦設計好了以後，您如何去驗證 FPGA 或是 PLD 的工作是否正常呢？此時就可以運用 Acute PG 來產生 FPGA 和 PLD 輸入腳的訊號，並將訊號餵入。再用 Acute LA 來擷取輸出腳的訊號，以此來驗證功能的正常與否。這種作法是非常好用的，因為有些模擬程式要將訊號完整的模擬必需花費很多時間，而且模擬的結果都是理想化的。運用真實的驗證可以真正得到零件的運作情形。同樣這種作法也可運用在 IC 的測試或是其他類似的測試及驗證環境。

一般模擬程式都必需建立一些波形檔案或是向量檔案，以模擬設計的電路。一旦建立好這些檔案後您就可以利用 PG-Editor 來將這些波形檔或是向量檔轉成 Acute PG 的波形檔，再由 Acute PG 來提供真正的波形給 FPGA 或是 PLD 的電路使用。由於模擬程式相當的繁多，所以本公司會陸續提供這些波形檔或是向量檔的轉換程式。

而 Altera 波形檔轉換就是將 Altera 的 Max Plus II 軟體所產生的波形檔轉換 Acute PG 的波形檔。Max Plus II 支援的波形檔案格式分成兩種，一種是 Binary File(.SCF)，另一種是 Text File(.VEC、.TBL)。而 Text File 是一種向量格式的波形檔，PG-Editor 只支援 Text File 格式。所以如果是 Binary File 格式的話，就必需用 Create Table File(請參考 Max Plus II 的用法)的功能來產生 Text File 的格式。然後再用 PG-Editor 的轉換程式轉成 PG-Editor 的波形。

## Altera 波形檔轉換的使用方法：



首先要先輸入一個 Max Plus II 的 Table 檔，然後按下載入鈕。此時在表列區中就會出現待轉換的訊號名稱和轉成 Acute PG 後的通道號碼。由於 Acute PG 是一個輸出訊號的儀器，所以轉換程式只會將 Table 檔案中的輸入腳的訊號取出並加以轉換。這時『單位設定』會出現經過換算的時間單位，由於 Acute PG 的最小時間單位為 10ns，所以如果 Table 檔所用的時間單位小於 10ns，轉換程式就只會設定成 10ns。但是這個時間單位亦可以自行修改，以配合實際的需求。在轉換的過程中也可以加入『重覆輸出』或是『按鍵後才輸出波形』的選項。『重覆輸出』就是讓 Table 檔的波形可以不斷的重覆輸出。『按鍵後才輸出波形』的選項是代表當您執行輸出波形前會暫時停止，然後再按下啟動熱鍵才會真正輸出波形。一旦兩個選項同時選擇時，可以達到重覆測試待測電路以外，而且每測試完一次都可[停下來](#)，等待下次按鍵再行測試。

### 13. 批次輸出



批次輸出可以讓您一次輸出大量的檔案，它所接受的格式為使用波形編輯器所編輯的波形檔(副檔名為.PGW 或是.PGV)。而且它提供一個很方便的功能，就是您能夠設定每個輸出檔案之間的時間間隔以及使用重覆輸出功能，讓您所要輸出的檔案就像迴圈一樣依序輸出，可以讓您的測試工作自動化，讓您的測試工作更簡單。

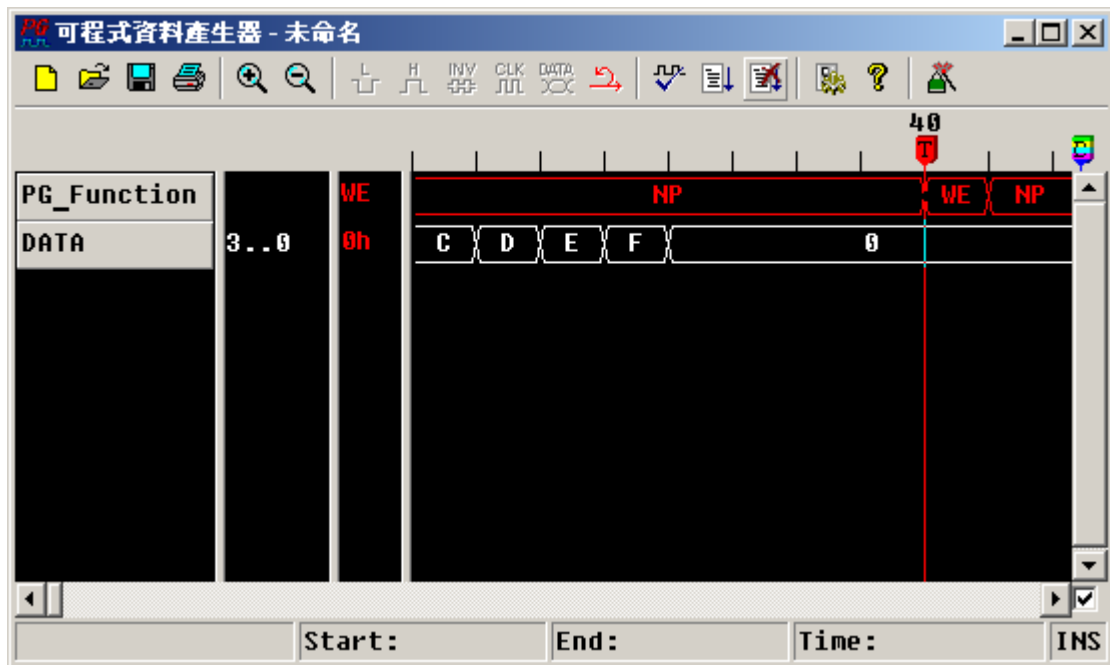
批次輸出的使用方法：



首先，按下『新增』選擇您要輸出的檔案，假如您要輸出多個檔案，您可以使用『上移』或是『下移』來決定您所要輸出檔案的順序。『刪除』是刪除單一檔案；而『全部刪除』顧名思義就是刪除你在表單中的所有檔案。您可以選擇勾選『間隔時間』並填入數字來決定你每個輸出檔案之間的時間間隔，或是不勾選而選擇『手動換檔』，而勾選『重覆輸出』可以讓您的檔案循環般的輸出，直到您按下『停止輸出』才會停止，而倘若您沒有勾選『重覆輸出』那只會輸出一遍就停止輸出了。


注意：多個檔案輸出時，您必須在每一個波形檔(.PGW 或是.PGV)的波形結尾部分

加上 Wait for Event(WE)的指令，來讓程式辨別檔案的結束處並輸出下一個檔案，否則將無法產生批次輸出的效果。您可以使用波形編輯器來下該指令，以下為使用範例。





## 14. SPI 訊號產生器

 SPI 訊號產生器可以產生 SPI 的訊號，它可以讀取副檔名為.bin 或.txt 的檔案來產生或是自行編輯 SPI 的資料，也可以使用程式來隨機產生資料。SPI 訊號產生器可以讓您的測試工作更有效率。

### SPI 訊號產生器的使用方法：

首先，是輸入您要產生 SPI 訊號的參數部份。在通道設定方面：您可以選擇 PG 要輸出 SPI 訊號的通道，預設為 CH0 為 SPI 訊號的 Chip Select，CH1 為 Clcck，CH2 為 Data。而在 Chip Select 和 Clock Latch Data 的參數設定上就根據您的需求來做設定，在此不多做敘述。

在 Data Bits Setup 的參數設定裡，輸出頻率為 SPI Clock 的輸出頻率；而 MSB 和 LSB 則是在 Data Bits 設定完成之後選擇高低位元的輸出順序，例如：1ah (選擇 8 Data Bits, MSB)那如果您選擇 8 Data Bits, LSB，將會變成 58h。

輸出資料個數則是會顯示您自行編輯的資料個數或是使用隨機的方式產生資料，使用隨機產生資料的方式其資料個數可以手動輸入資料個數或是隨機產生資料個數，以下為使用隨機輸入資料個數步驟：勾選『隨機輸出』→『波形輸出』。以下為手動輸入資料個數步驟：勾選『隨機輸出』→『隨機』→輸入要輸出的資料個數。勾選『連續輸出』可以連續輸出該資料，若不勾選，該資料只會輸出一次，隨機輸出只能產生十六進制資料。

使用檔案輸出的步驟：按下『瀏覽』→選擇 Bin 或是 Txt 檔案 →『波形輸出』，Bin 檔案只能以 Hex 的格式輸出，而 Txt 檔案只能以 ASCII 格式輸出。

Data Bits 項目必須詳細做說明，Data Bits 可以輸入的範圍為 8~24 Bits。倘若您是使用讀取 bin 檔案的方式來產生 SPI 的資料，如果此時您選擇的 Data Bits 的個數為 8Bits 則程式會以 8 Bits(1 byte)的方式讀取檔案資料並輸出；但是如果您選擇的 Data Bits 個數為 9~15，則程式會以 16 Bits(2 Byte)來讀取資料，不足 16 Bits 的部份將會被遮罩掉，也就是說，如果您是選擇 9 Bits 來讀取檔案資料，讀取的 16 Bits 中只保留低位元 9 Bits 的資料,而其他的部分將被遮罩掉，其他依此類推。而倘若你是選擇隨機產生 SPI 資料值的方式，而這種方式所能產生的資料值最大為 7fff (16

Bits)，如果您選擇的 Data Bits 為 16 bits 以上將無意義。

而如果您是選擇自行編輯 SPI 資料的方式，有一點必須要注意的是；如果您編輯的資料為 16 Bits，但是您卻選擇 Data Bits 為 8 Bits，則高位元組的部份將會被遮罩掉，例如：您輸入 162ah，這個數字必須使用 16 Bits 才足以表達該數字，但是您若是選擇 8 Bits，則 SPI 資料產生器只會輸出 2ah，其他依此類推。而自行編輯資料的格式如下(Hex)：

輸入：

2a;↵(Enter 鍵)

45;↵(Enter 鍵)

1c;↵(Enter 鍵)

67;↵(Enter 鍵)

52;↵(Enter 鍵)

如果您是要以 Hex 的方式輸出，您必須遵照上述格式，也就是輸入一個數字，然後輸入分號，再按下『Enter』鍵。但是倘若您是要以 ASCII 的方式輸出，就不需要遵照上述格式。清除資料則是可以清除資料編輯區的資料，而範例可以自動輸出一些範例的格式介紹。

## 15. VHDL 波形檔轉換

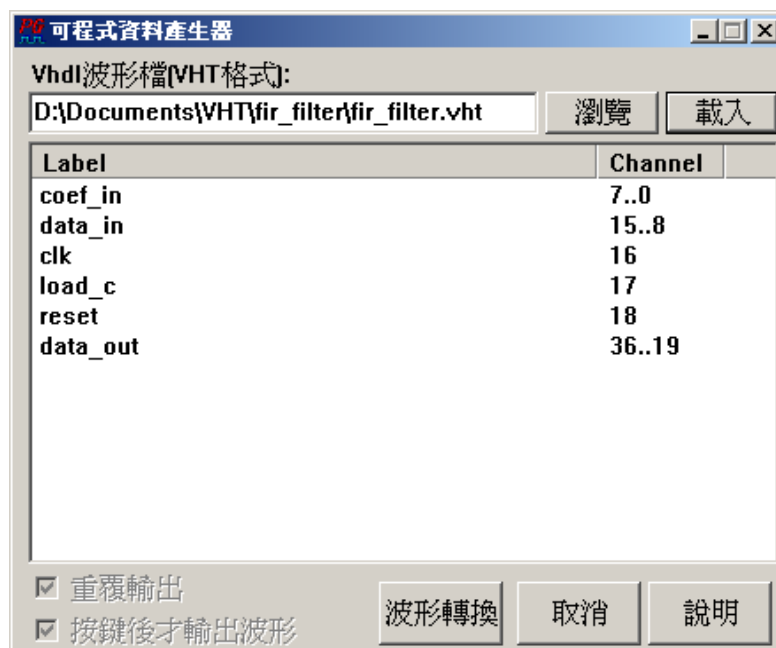


FPGA 與 PLD 可程式邏輯的運用已經越來越廣泛，許多的設計師都會用 FPGA 與 PLD 來當作控制電路，所以設計這些可程式邏輯時，零件商都會提供許多設計用的應用軟體，通常都會包括模擬程式，讓您在軟體上就可以輕鬆的模擬 FPGA 和 PLD 的功能。

但是一旦設計好了以後，您如何去驗證 FPGA 或是 PLD 的工作是否正常呢？此時就可以運用 PG 來產生 FPGA 和 PLD 輸入腳的訊號，並將訊號餵入。再用 LA 來擷取輸出腳的訊號，以此來驗證功能的正常與否。這種作法是非常好用的，因為有些模擬程式要將訊號完整的模擬必需花費很多時間，而且模擬的結果都是理想化的。運用真實的驗證可以真正得到零件的運作情形。同樣這種作法也可運用在 IC 的測試或是其他類似的測試及驗證環境。

而 VHDL 波形檔轉換就是將 Altera 的 Quartus II 或是 Xilinx 軟體所產生的波形檔轉換 Acute PG 的波形。然後再用 PG-Editor 的轉換程式轉成 PG-Editor 的波形。

### VHDL 波形檔的使用方法：



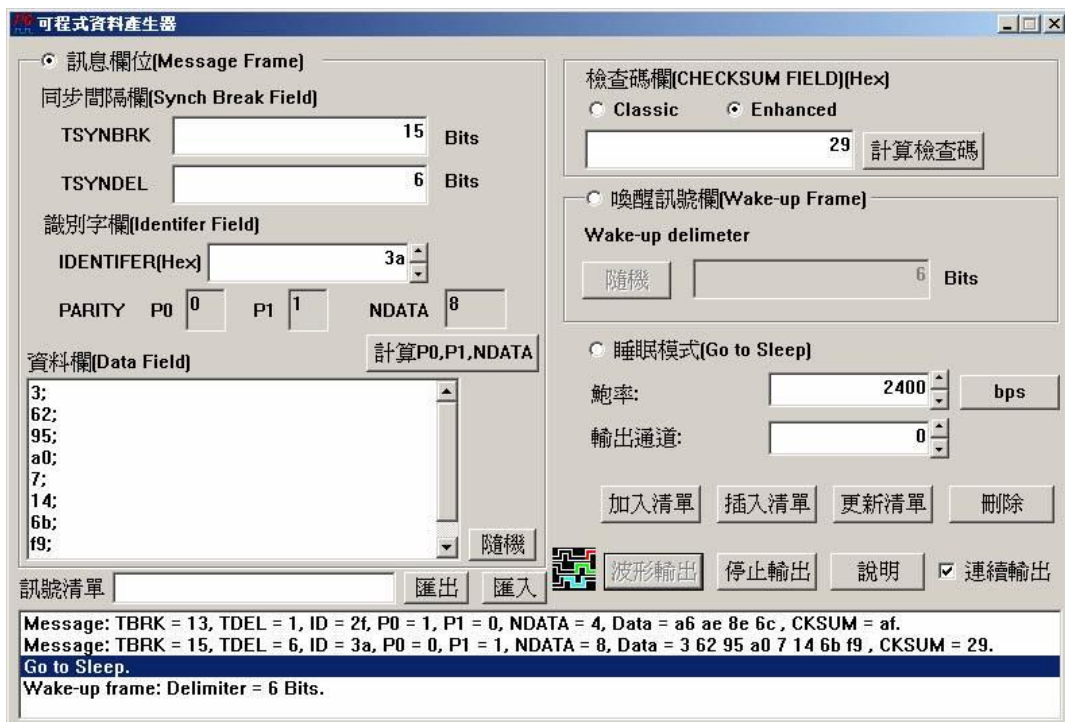
選擇要轉換的 VHT 檔案，然後根據您的需求，選擇是否要重覆輸出或是要按鍵後才輸出波形，當然也可以同時勾選或不勾選。再按下載入鍵，再按下波形轉換鍵。轉換程式會將 VHT 檔案中的訊號依輸入訊號組、輸入單一訊號、輸出訊號組以及輸出單一訊號的順序作轉換。

**註：訊號通道數目如果超過 50 個，將不會輸出。**

## 16. LIN 訊號產生器

隨著汽車市場的蓬勃發展，車用電子的傳輸控制也越來越重要；CAN 和 LIN 都是車用電子裡常見的傳控介面。而 LIN BUS 是車用電子中為因應低成本趨勢而產生的一種傳控介面，主要是使用在低速的周邊裝置，如車門控制、車窗控制等。

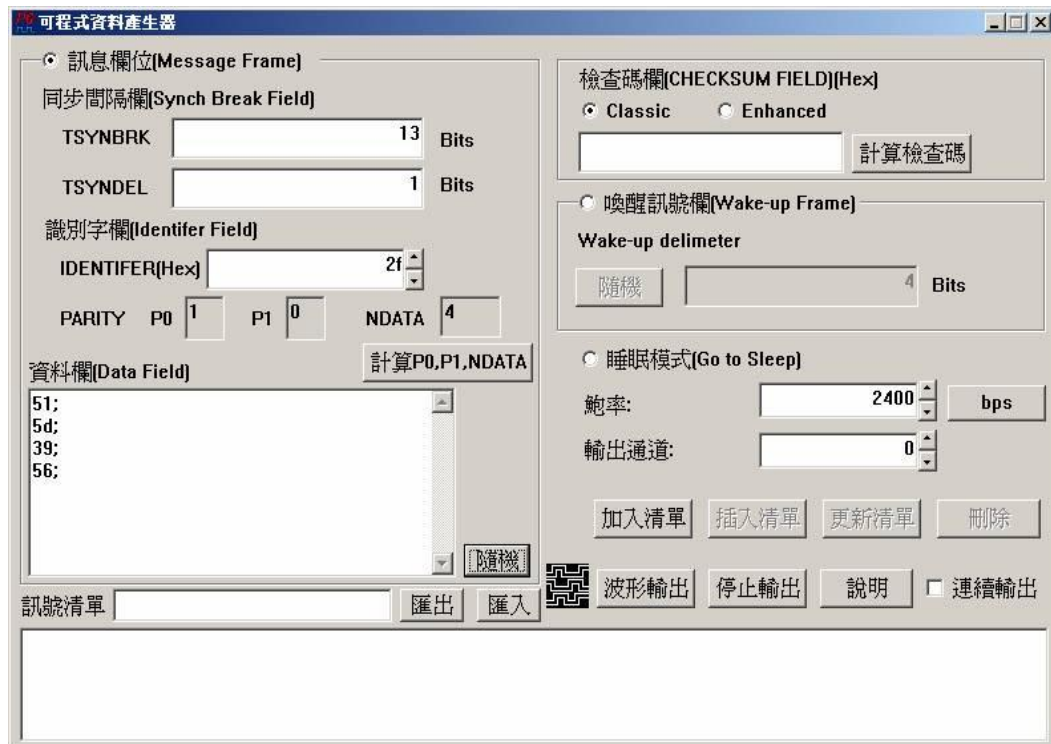
### LIN 訊號產生器的使用方法：



LIN 訊號中一個完整的 MESSAGE FRAME 是由 HEADER 和 RESPONSE 所組成。而 HEADER 可以依序區分成三個部份，分別為 SYNCH BREAK FIELD(同步間隔欄)、SYNCH FIELD(同步欄)和 IDENTIFER FIELD(識別欄)。而 RESPONSE 依序分為兩部分，分別為 DATA FIELD(資料欄)和 CHECKSUM FIELD(校驗和欄)。SYNCH BREAK FIELD(同步間隔欄)可以區分為兩部分，分別是 TSYNBRK 和 TSYNDEL(synchronization delimiter)。按照 LIN 通訊協定的規定，TSYNBRK 必須大於或等於 13 bits; TSYNDEL 必須大於或等於 1 個 bits。您可以自行輸入該值，如果您輸入的值超過限制，該程式會以訊息通知您，請您不用擔心。

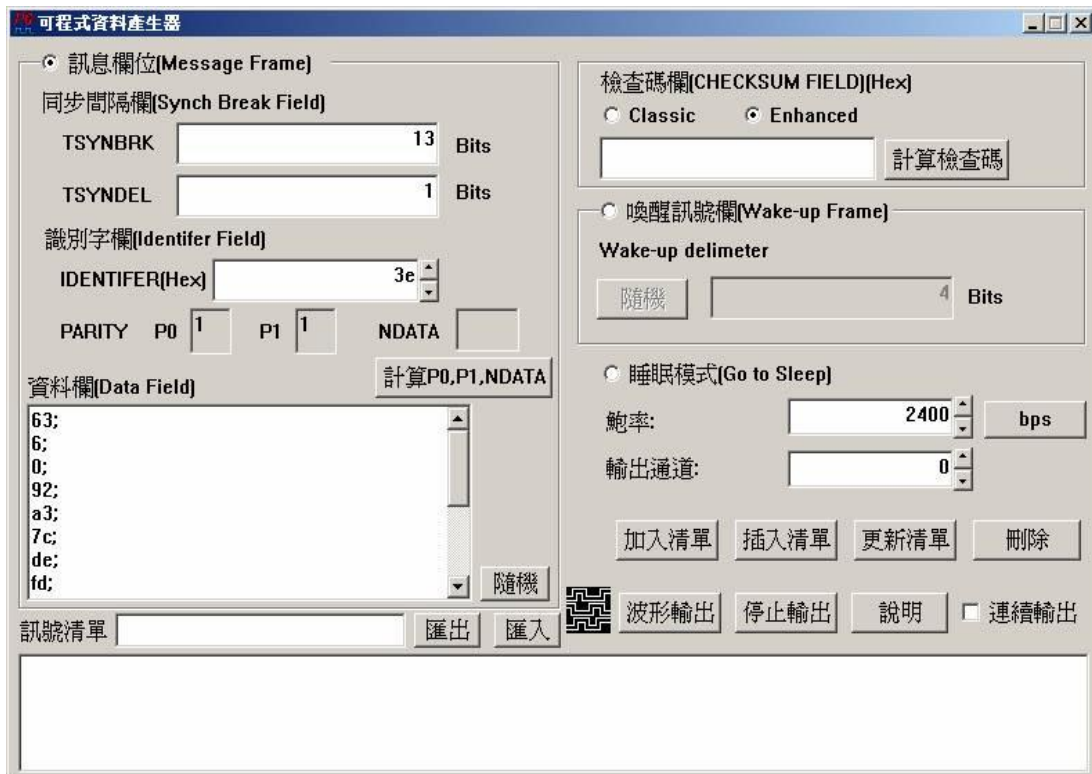
SYNCH FIELD(同步欄)固定為十六進制值 0x55，不需要做任何變動，所以程式中

並沒有顯示這方面的資訊。再來為 IDENTIFER FIELD(識別欄)，它依序區分為 6 個 bits 的識別字(IDENTIFER)和 2 個 bits 奇偶校驗位。您只要按下[計算 P0,P1,NDATA]按鈕，程式會自動幫您計算奇偶校驗位和 N-DATA，NDATA 是決定您所要傳送資料的個數。以程式來舉例做說明：



輸入 IDENTIFER 的值為 0x2F，接下來按下[計算 P0,P1,NDATA]按鈕，經過程式計算 P0(奇校驗)為 1，P1(偶校驗)為 0，NDATA 為 4。也就是說您必須自行輸入 4 個 8bits 的資料或是使用隨機來產生(該資料為十六進制值)。

這裡有一點必須要注意，如果您輸入的資料個數超過 4 個，那程式只會按照 NDATA 之值，也就是 4 去選取您所輸入的前 4 個的資料。如果您不想侷限於資料個數，您可以輸入 IDENTIFER 為 0x3E 或是 0x3F，此時就可以不用理會 NDATA 之值，程式會根據您所輸入的資料個數去選擇所有的資料。



這裡我們輸入 IDENTIFER 為 0x3E，並按下[隨機]，結果產生 14 個資料值，當然您也可以自行輸入想要傳送的資料值。

註：這裡隨機產生的資料個數為 0~20 個。

MESSAGE FRAME 的最後一個欄位為 CHECKSUM FIELD(校驗和欄)，您可以選擇勾選 Classic(Ver 2.0 之前的計算檢查碼的方式)或是 Enhanced(Ver 2.0 之後)。最後輸入速率以及決定使用 PG 的哪一個通道輸出，如果選擇 0 就是從 PG 的通道 0 輸出；1 就是從通道 1 輸出，依此類推。

然後按下[加入清單]，程式會幫您將該 MESSAGE FRAME 加入到 LIN 的訊號清單，PG 就是根據這份清單來輸出 LIN 的訊號；而[插入清單]、[更新清單]以及[刪除]可以幫助您來修改該訊號清單。接下來，就針對這 4 個修改清單的按鈕來做說明：

[加入清單]會依序將您想要的輸出欄位依序加入到輸出清單，舉例來說：我想讓 PG 送出一個 LIN 的 MESSAGE FRAME，然後讓 LIN 進入睡眠模式，然後再送一個喚



醒訊號，最後再緊接著一個 MESSAGE FRAME。做法如下：

- (1) 首先選取[訊息欄位]
- (2) 輸入 IDENTIFER，假設為 0x2F
- (3) 然後按下[計算 P0,P1,NDATA]，結果 NDATA = 4
- (4) 然後我按下[隨機]產生資料，也可以自行輸入，注意自行輸入資料有其格式，請遵照以下方式：

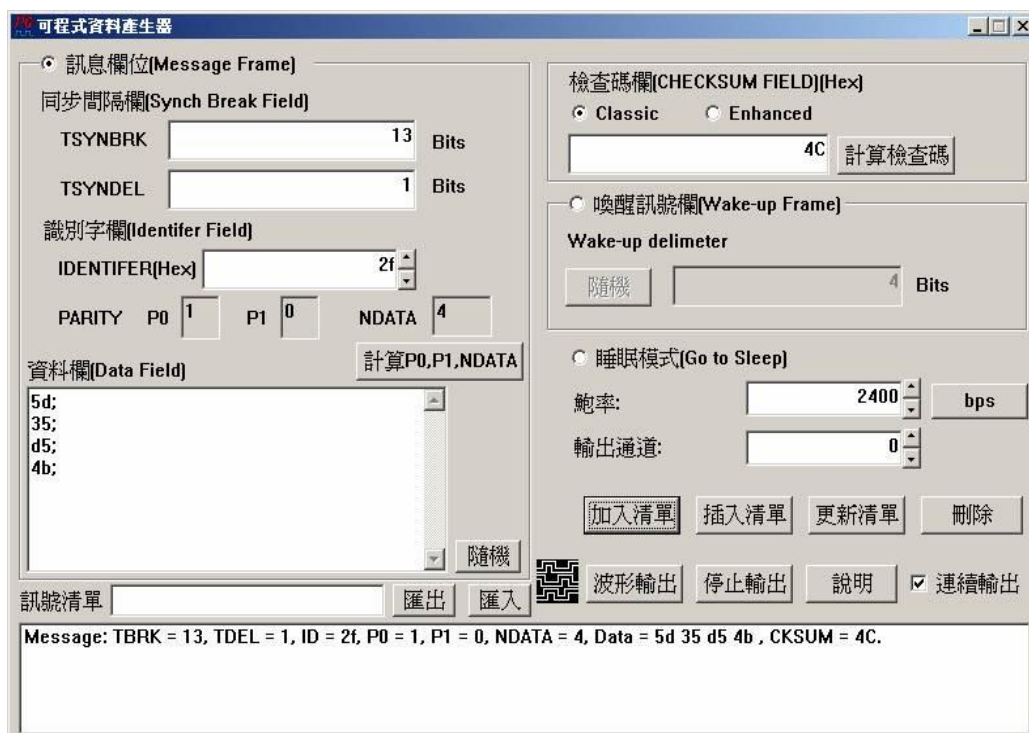
5d;↵

35;↵

d5;↵

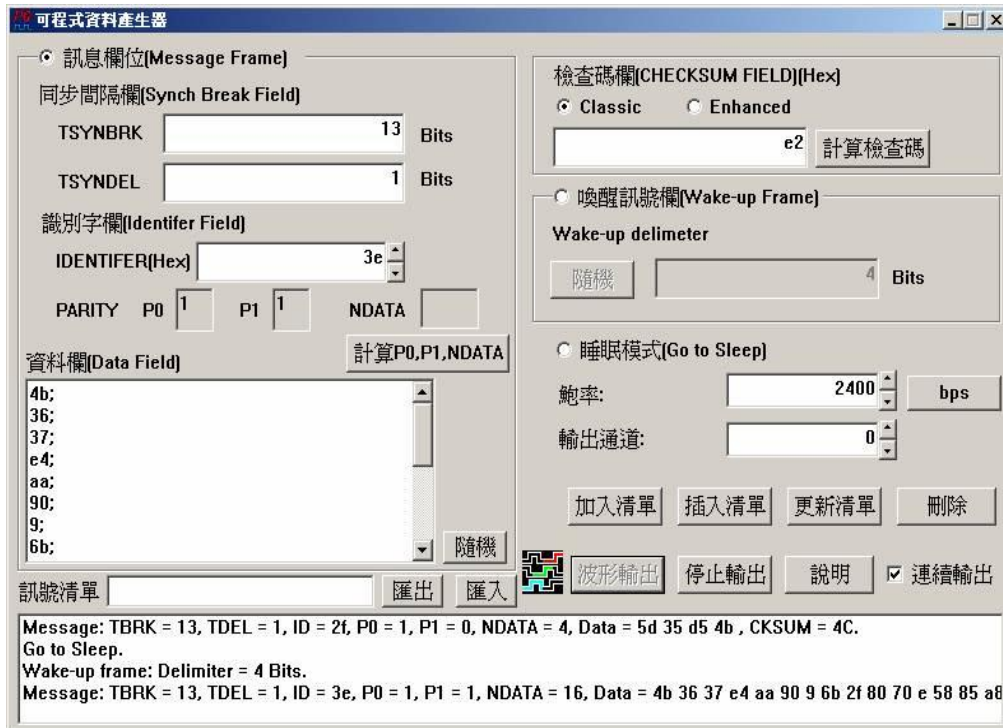
4b;↵

- (5) 按下計算檢查碼，我勾選 Classic 的計算方式
- (6) 按下[加入清單]，結果如下：

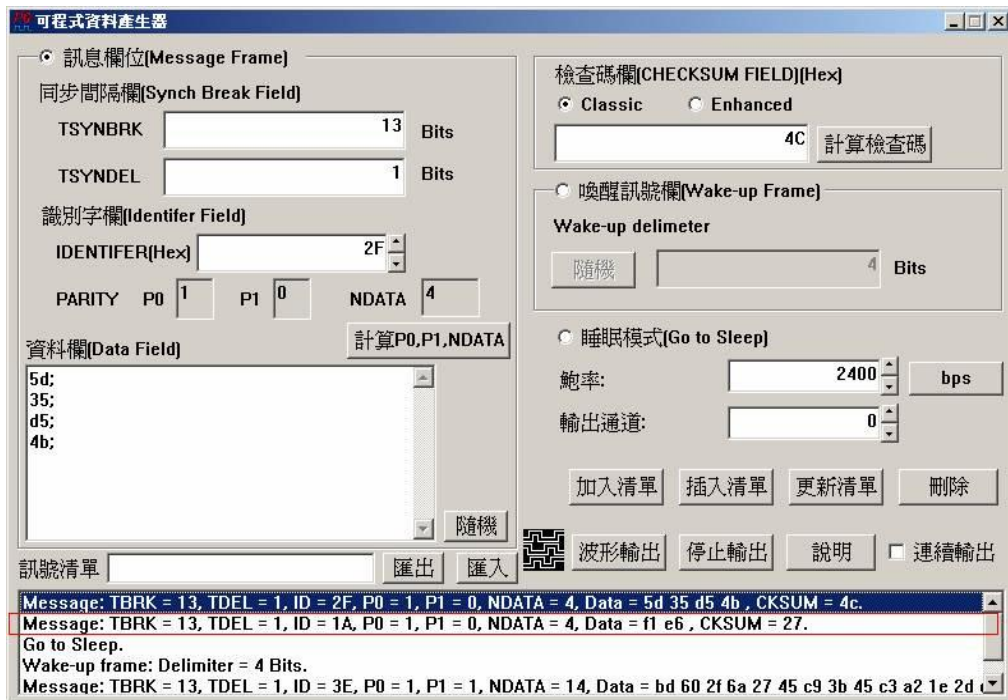


- (7) 然後選取[睡眠模式]，再按下[加入清單]
- (8) 選取[喚醒訊號欄]，再按下[加入清單]
- (9) 再選取[訊息欄位]，此時輸入的 IDENTIFER 為 0x3E，接下來的步驟同 1~4，

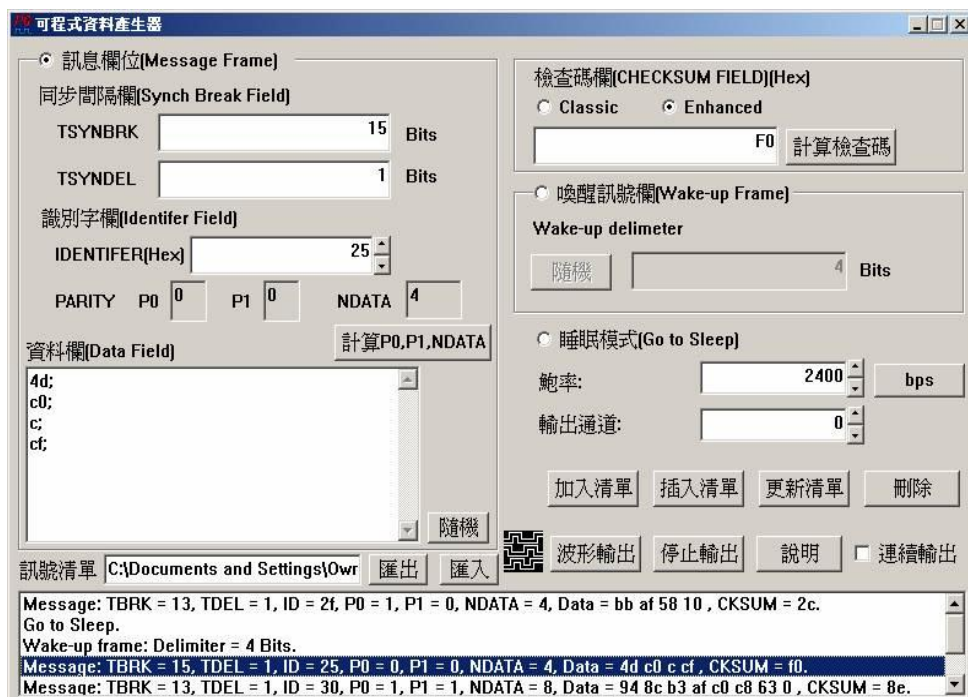
最後再按下[加入清單]，結果如下：



[插入清單]的使用方式：您可能會在上一例中，想在第一個 MESSAGE FRAME 之後再插入一個 MESSAGE FRAME，此時您就可以使用[插入清單]的功能，首先，使用滑鼠選擇您想要在哪一個項目之後插入新項目。我們選擇在第一項之後插入，也就是成為清單的第二項。所以使用滑鼠點選第一項，然後我們加入一個 IDENTIFER 為 0x1A 的 MESSAGE FRAME，然後遵照 1~4 的步驟，最後按下[插入清單]，您可以看到在點選的第一項之後多出了我們剛剛插入的項目。



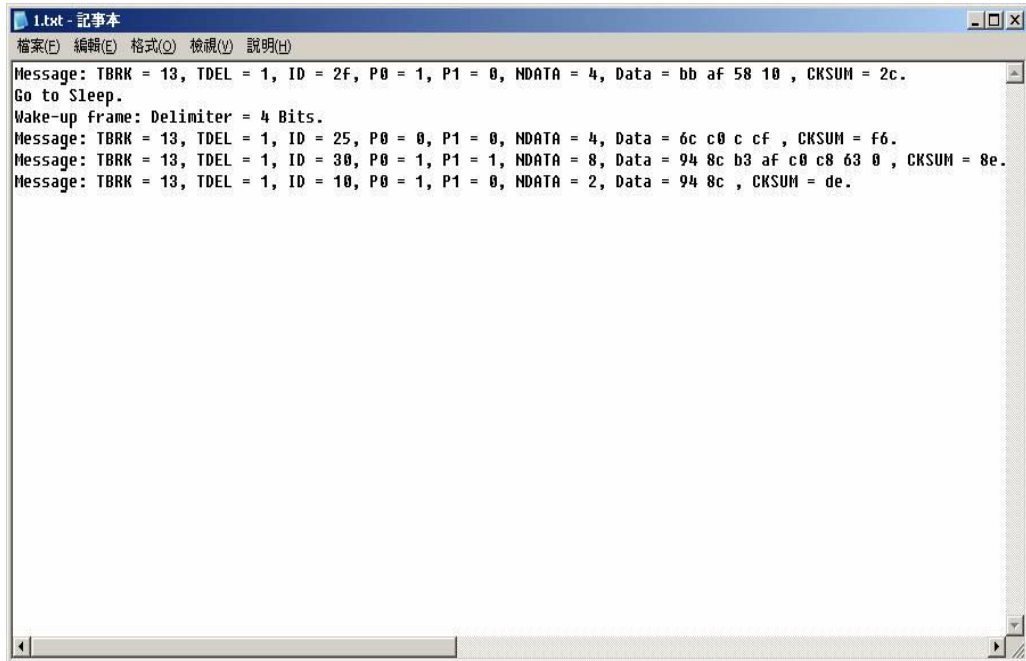
[更新清單]的使用方式：您可能想要在一長串的訊號中修改某項目裡的某一個欄位的內容，此時您就要使用[更新清單]的功能，假設您想要修改第四項的 TSYNBRK 以及將第一個資料從 0x6C 改為 0x4D，此時將滑鼠點選第四項，然後將 TSYNBRK 的 13 bits 改為 15 bits，將資料欄的第一個 0x6C 改為 0x4D，然後按下[更新清單]，就完成修改的動作。註：此時的 CHECKSUM 也必須做修改。



[刪除]的使用方式：您可以使用滑鼠點選項目，然後按下[刪除]鍵，則程式會

將該項目從訊號清單刪除。

[匯出]和[匯入]是將您的訊號清單存成文字檔或是將該文字檔載入，您也可以藉由修改該文字檔來更改您所要輸出的 LIN 訊號清單，不過，請依照該文字檔的格式，否則將會發生錯誤。



The screenshot shows a Notepad window titled "1.txt - 記事本". The menu bar includes "檔案(F)", "編輯(E)", "格式(O)", "檢視(V)", and "說明(H)". The text content is as follows:

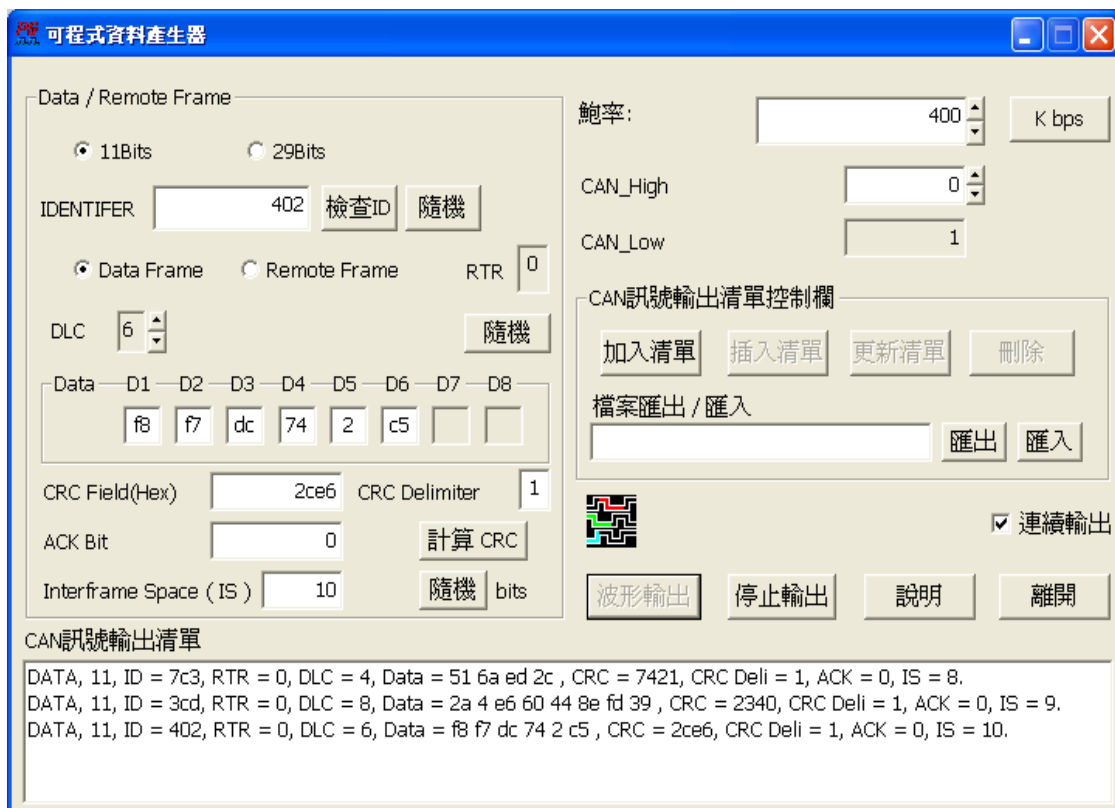
```
Message: TBRK = 13, TDEL = 1, ID = 2f, P0 = 1, P1 = 0, NDATA = 4, Data = bb af 58 10 , CKSUM = 2c.  
Go to Sleep.  
Wake-up frame: Delimiter = 4 Bits.  
Message: TBRK = 13, TDEL = 1, ID = 25, P0 = 0, P1 = 0, NDATA = 4, Data = 6c c0 c cf , CKSUM = f6.  
Message: TBRK = 13, TDEL = 1, ID = 30, P0 = 1, P1 = 1, NDATA = 8, Data = 94 8c b3 af c0 c8 63 0 , CKSUM = 8e.  
Message: TBRK = 13, TDEL = 1, ID = 10, P0 = 1, P1 = 0, NDATA = 2, Data = 94 8c , CKSUM = de.
```

## 17. CAN 訊號產生器

**CAN** CAN BUS(Controllor Area Network)協定技術的使用在汽車工業上已有相當一段時間，其目的為因應汽車日益進步，電子系統及配備也越來越多且複雜，需要有許多線路來連結，形成龐大而複雜的線束系統，直接影響造車成本提高、車體重量等問題。

CAN Bus 的原理是將電腦網路技術概念，運用在汽車上，每個汽車電子零件或作動器就像是網路用戶一樣，只需簡單一條線即可達成接收傳送溝通等功能，再由控制器整合所有訊號，傳遞至各個系統使用，而各個感知器、作動器、控制單元如有任何問題或故障，可以透過訊號傳遞，診斷問題所在。

### CAN 訊號產生器的使用方法：



CAN 訊號的 MESSAGE FRAME 可以分成 4 種，分別為 DATA FRAME、REMOTE FRAME、ERROR FRAME、OVERLOAD FRAME 等。DATA FRAME 是 CAN 中

最為普遍的類型，資料都是從一個節點傳送到所有節點；REMOTE FRAME 與 DATA FRAME 極為相似(只有 RTR bit 被設為 1)，主要的用途是用來向其他的 CAN 的節點要求資料；ERROR FRAME 是用來表示在 CAN BUS 上有錯誤狀況被偵測到，而 OVERLOAD FRAME 用來告知其他的 CAN 的節點，必須有更多的時間來處理收到的資料，主要功能是延遲下一個封包被傳送的時間。

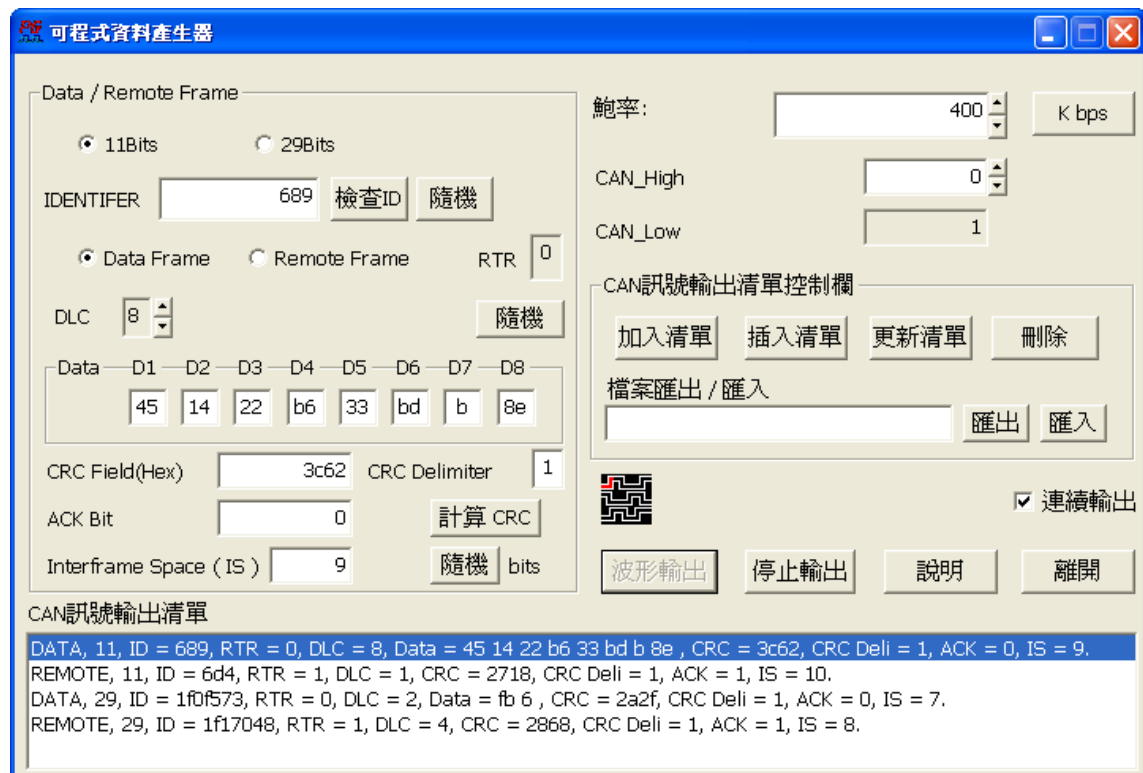
使用該程式時，首先，決定您要使用 STANDARD FRAME 還是 EXTENDED FRAME，如果您選擇 11 bits；那就表示您是使用 STANDARD FRAME 的模式，反之，您就是選擇 EXTENDED FRAME 模式。11 bits 和 29 bits 是表示您是使用哪一個 IDENTIFIER 的表示方式，以實例來說，如果您輸入的 IDENTIFIER 為 0x5A9，那就要請您選擇 11 bits，倘若您輸入的 IDENTIFIER 為 0x1F1518D，那就要選擇 29 bits。這裡有一點必須加以說明：當您選擇 29 bits 時，它是將您輸入的 IDENTIFIER 分為 11 個 bits MSB 和 18 個 bits LSB，所以是 29bits，然後，中間插入 SRR、IDE 2 個為 1 的 bit 以及最後加入 1 個 RTR bit 結合成 Arbitration Field。如果您擔心輸入的 IDENTIFIER 是否正確，您可以按下[隨機]由程式來幫您產生正確的 IDENTIFIER。

然後，RTR 會根據您是要使用 DATA FRAME 或是 REMOTE FRAME，如果是 DATA FRAME，RTR 為 0，REMOTE FRAME RTR 為 1。再者根據您的需求填入 DLC(Data Length Code)而 Data 欄位 D1~D8 會根據您輸入的 DLC 個數來運作。

在您填妥 IDENTIFIER、DLC、DATA 之後，接下來請按下[計算 CRC]，程式會自動幫您算出該段輸出資料之 CRC 值，再來根據您的需求輸入 CRC Delimiter、ACKnowledge 和 Interframe Space (間隔位元)，Interframe Space 可以隨機產生，而 ACKnowledge 和 CRC Delimiter 請輸入 0 或 1 和 1。

然後決定鮑率和 CAN\_HIGH、CAN\_LOW 的輸出通道，然後按下[加入清單]，將您

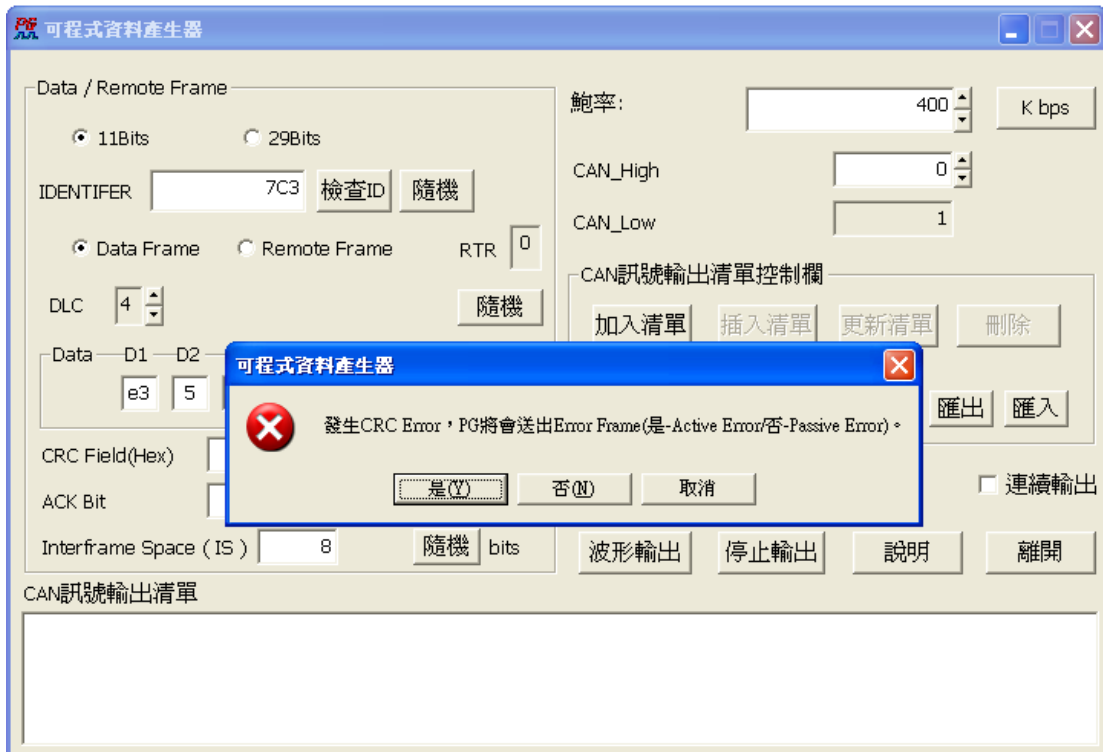
要輸出的 CAN 訊號填入輸出清單，以下以實例說明：



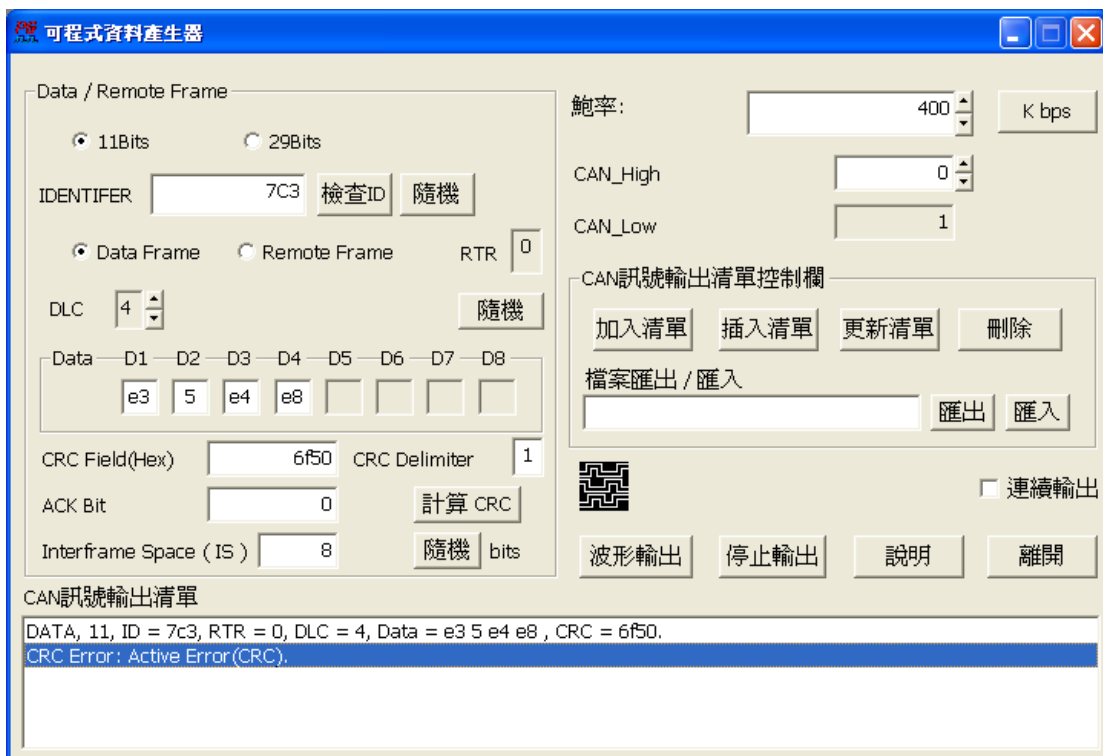
假設我要產生 4 個 FRAME，2 個 DATA FRAME 和 2 個 REMOTE FRAME，分別為 STANDARD FRAME 和 EXTENDED FRAME，請參考上圖的訊號輸出清單。

接下來為 4 個新增和修改 CAN 輸出清單的按鈕，分別為[加入清單]、[插入清單]、[更新清單]、和[刪除]。使用方式和 LIN 訊號產生器的方式一樣，請參考 LIN 訊號產生器的說明文件。

這裡有一點要說明，CAN 訊號產生器在您輸入錯誤的 CRC、CRC Delimiter 會產生相對應的 ERROR FRAME，以 CRC ERROR 做實例說明：假設我使用隨機方式產生 IDENTIFER 為 0x611，選擇 DATA FRAME，STANDARD FRAME，隨機產生 DLC 和 DATA 分別為 8，0x9F、0x1E、0xB8、0x6A、0x09、0x79、0xEE、0x30，然後計算過後，正確的 CRC 值為 0x4f80 它改成錯誤的 CRC 值為 0x4f81 受預設值，然後按下[加入清單]，您可以看到以下圖例的部分：



程式會出現發生 CRC ERROR，會請您選擇產生 Active Error 或是 Passive Error 或者是取消這個 FRAME，此例我是選擇產生 Active Error，可以參考底下圖例的部份：

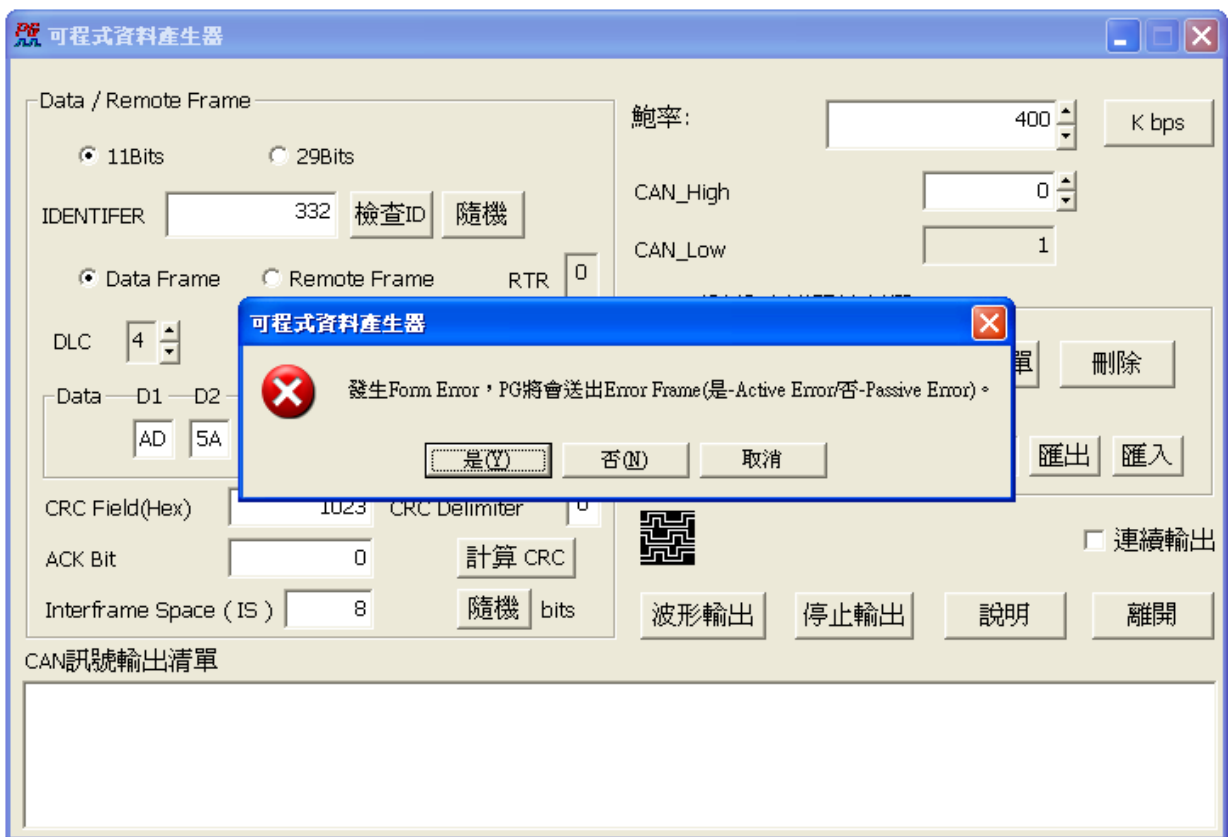


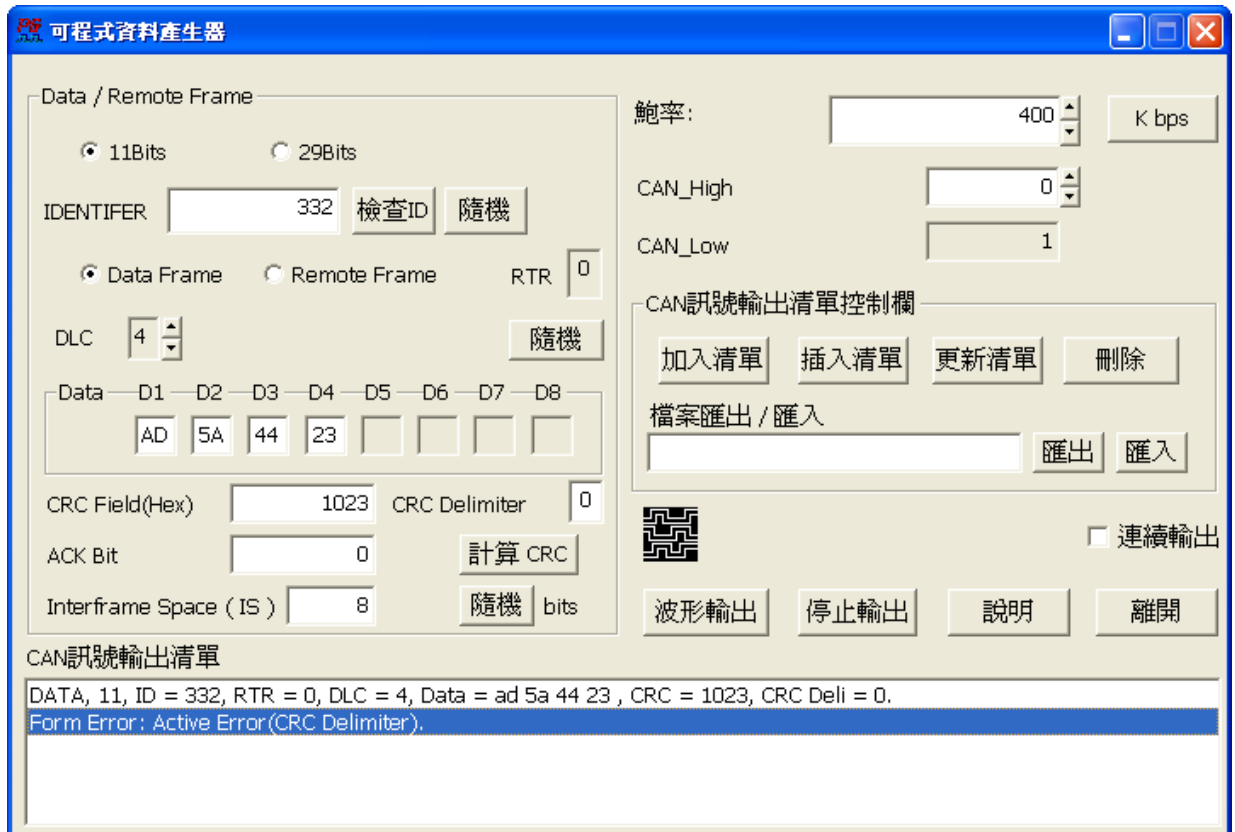
這個 CAN 訊號清單將會產生 DATA FRAME 11 bits IDENTIFIER，1 個 RTR bit，DLC = 8，Data = 0x9F, 0x1E, 0xB8, 0x6A, 0x09, 0x79, 0xEE, 0x30，CRC = 0x4f81。但是



因為這個 FRAME 是 CRC ERROR，所以這個 FRAME 後面的部份就不再傳送而緊接著由使用者所選擇的 Active Error Frame。

再者，說明產生 Form Error 的情形：將原本正確的 IDENTIFER 為 0x332，RTR = 0，DLC = 4，Data = 0xAD, 0x5A, 0x44, 0x23, CRC = 0x1023, CRC delimiter = 1, ACK = 0, Intermission = 4 的 11bits DATA FRAME 中將原本正確的 DATA FRAME 中的 CRC Delimiter 改為 0，即會產生 Form Error，如下圖：




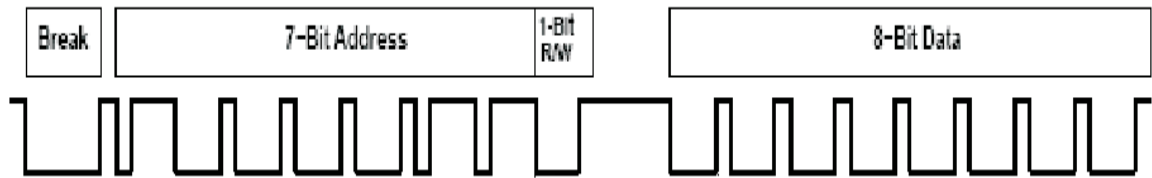


備註：每個發生錯誤的原因都有( )註明在 **ERROR FRAME** 之後。

而[匯出]和[匯入]的功能則和 LIN 訊號產生器的用法相同，請參考 LIN 訊號產生器的說明。

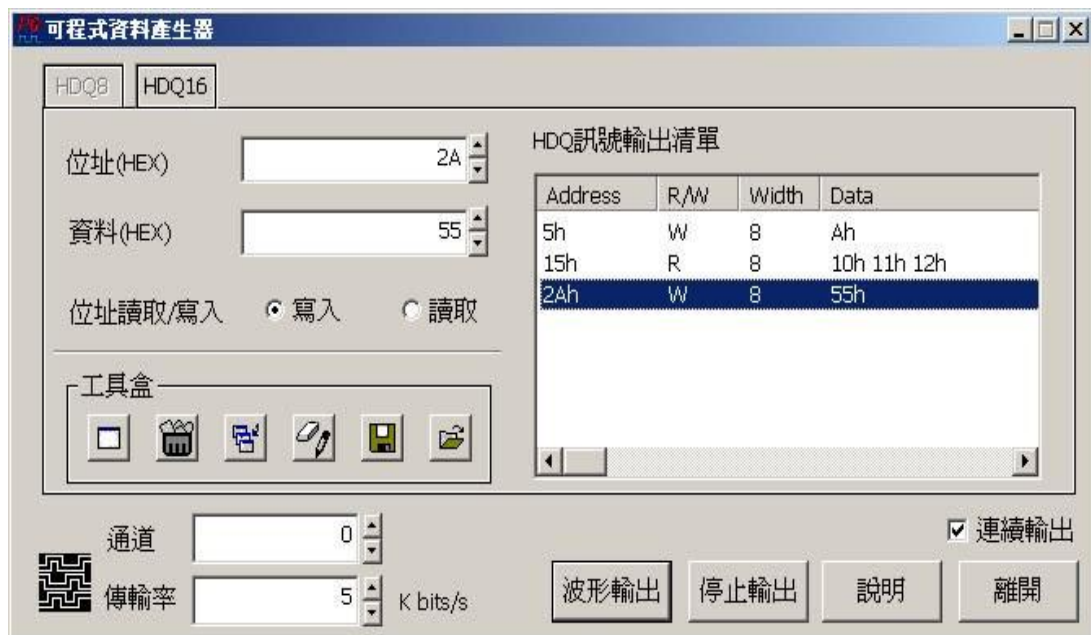
## 18. HDQ 訊號產生器

 HDQ 協定是由德州儀器(TEXAS INSTRUMENTS)所制定，使用於電池管理的顯示應用，主要是運用在消費性電子產品方面。



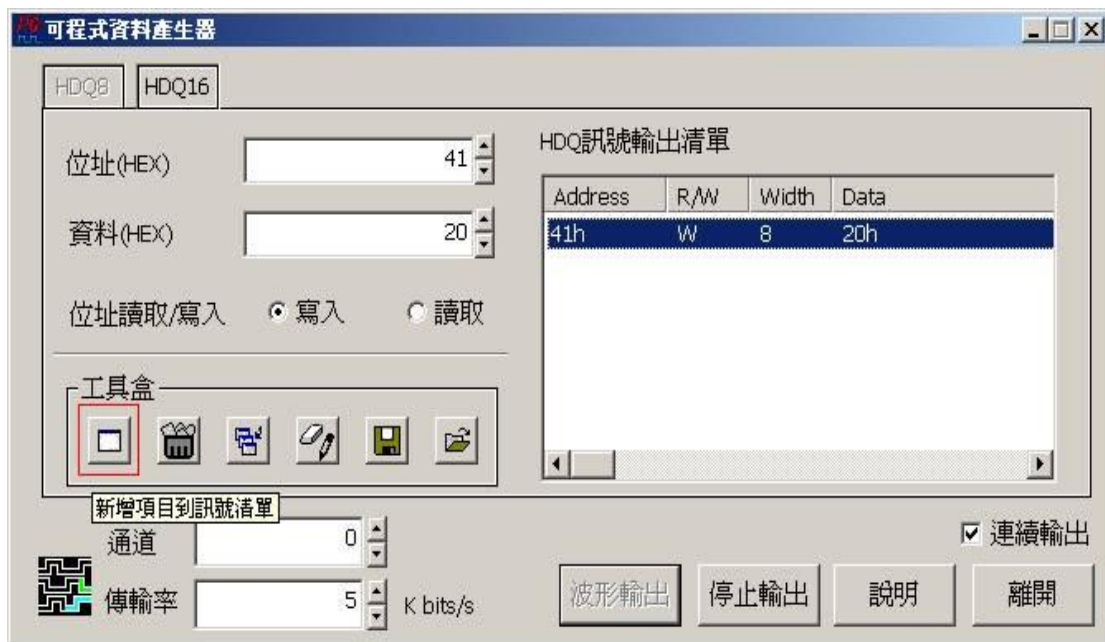
HDQ 分為 8 位元與 16 位元兩種資料寬度格式，位址固定為 7 位元。一個 HDQ 的封包主要由 **Break**、**7-Bit Address**、**1-Bit R/W** 和 **8-Bit Data** 或是 **16-Bit Data** 所組成。傳輸的方式為 LSB (Least-significant bit)到 MSB (Most-significant bit)，最大傳輸率為 5Kbits/s。

### HDQ 訊號產生器的使用方法



HDQ 訊號產生器的使用方法主要是根據您的訊號需求，輸出清單裡的每一列就代表一個封包。舉例來說：假設要產生一個位址為 41h、位址寫入、8 位元資料寬度然後資料為 20h 的 HDQ 封包。操作步驟如下：1. 按下 HDQ8 按鈕(選擇 8 位元資料寬度，此為程式預設項目)2. 分別在位址與資料填入 41h 和 20h，勾選寫入 3. 按下工

具盒中的『新增項目到訊號清單』請參考下圖。



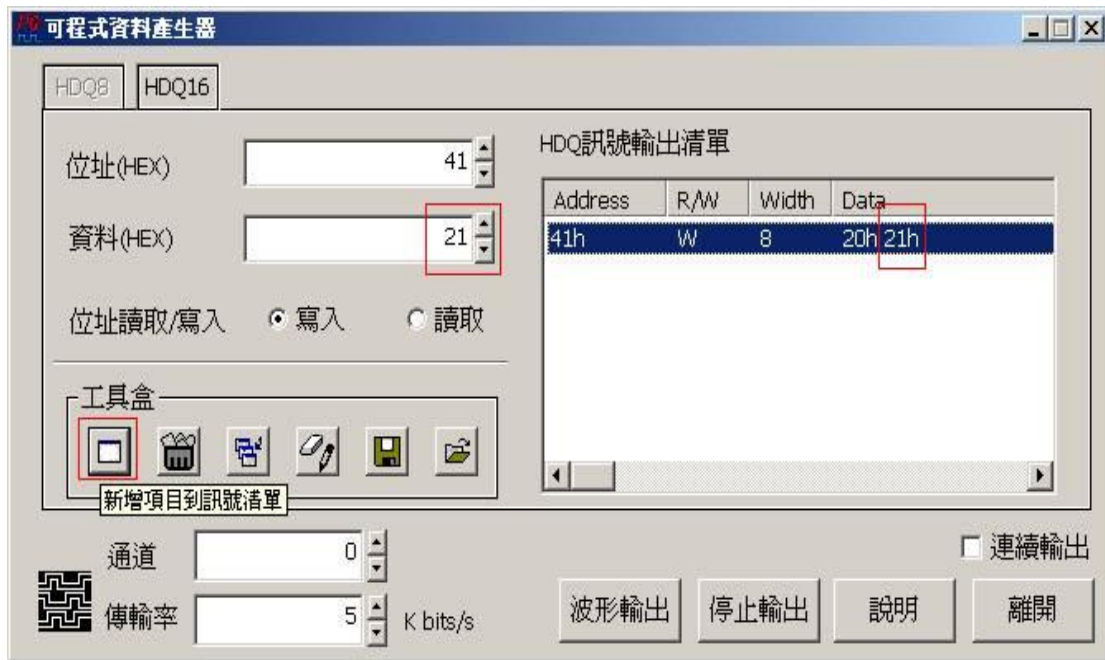
『新增項目到訊號清單』會將您的訊號封包增加在現有封包之後。但是如果想在相同位址之下，增加第二個訊號呢？

『新增項目到訊號清單』功能會根據上一個封包的位址、R/W、和資料寬度來判斷是否將該資料附加至上一個封包之後。只有在位址、R/W、和資料寬度和上一個封包相同時，程式會詢問使用者是否在該資料附加之上一個封包之後？

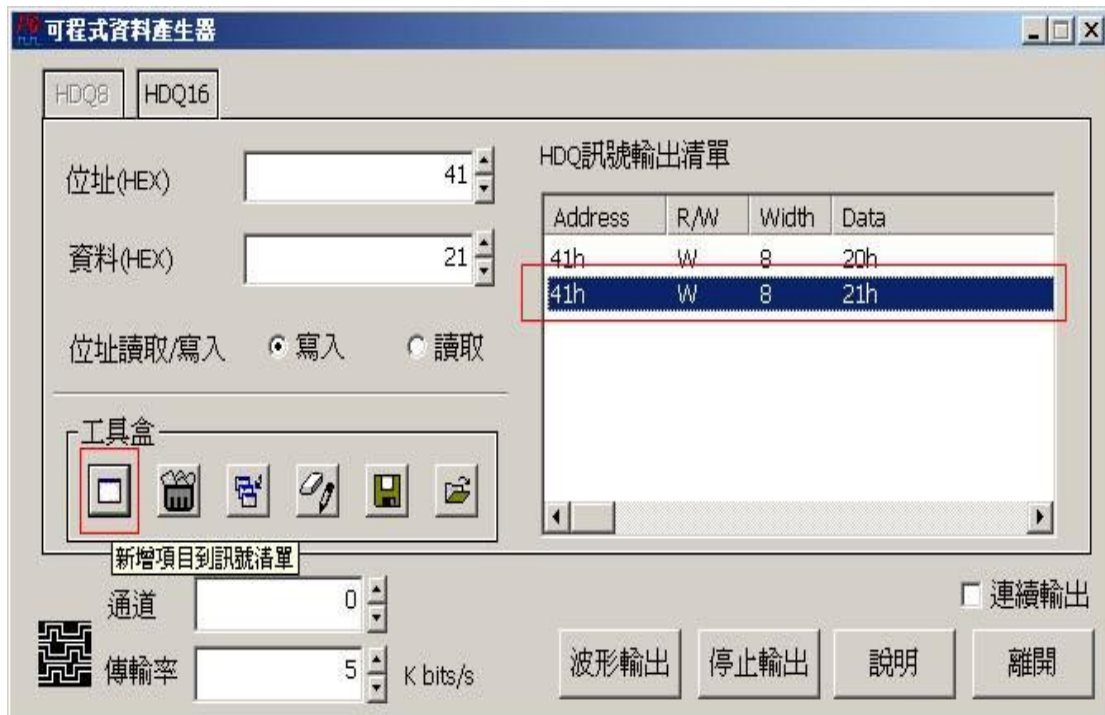
如果選擇“是”則會在資料附加至上一個封包之後，若選擇“否”則會增加一個新封包。請參考下圖：



選擇“是”的情形：



選擇“否”的情形：



『刪除所選擇的項目』功能可以將在『HDQ 訊號輸出清單』所選擇的項目刪除掉，只要按下該按鈕，就可以將目前選擇的該封包刪除。請參考下圖：



『插入新項目到所選擇的項目之後』功能以將在『HDQ 訊號輸出清單』所選擇的項目之後插入一個新封包。請參考下圖：



按下該按鈕之後，會將位址 41h、位址寫入、8 位元資料寬度然後資料為 22h 的 HDQ 封包插入至所選擇封包的下一個項目。請參考下圖：



請注意：『插入新項目到所選擇的項目之後』功能並無根據上一個封包的位址、R/W、和資料寬度來判斷是否將該資料附加至上一個封包之後的功能。『修改所選擇的項目』功能可以修改所選擇封包的位址、R/W、資料位元寬度和資料值，請參考下圖：



然後按下該鈕，程式會出現一個可以讓您修改的對話盒，該對話盒會顯示目前您所選封包之參數值，請參考下圖：

修改所選擇項目

位址(Hex) 41h

資料(Hex) 23h

位址讀取/寫入

寫入  讀取

資料寬度

HDQ8  HDQ16

確定 取消

假設要將所選擇的封包修改為位址 5Ah，資料為 24h、25h、26h，改為位址讀取然後將資料寬度改成勾選 HDQ16(16 位元資料寬度)，請參考下圖：

修改所選擇項目

位址(Hex) 5Ah

資料(Hex) 24h 25h 26h

位址讀取/寫入

寫入  讀取

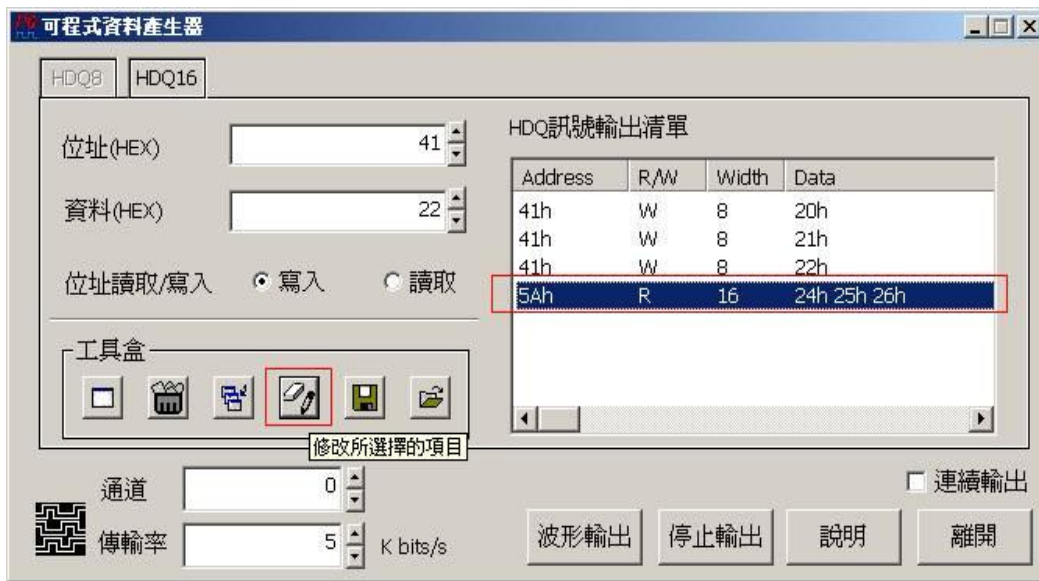
資料寬度

HDQ8  HDQ16

確定 取消



請注意：資料和資料之間請輸入一個空白鍵(space)。結果如下，請參考下圖：



『匯出訊號清單』功能可以將您編輯好的訊號清單存成檔案，之後可以使用『匯入訊號清單』功能將該檔案讀回，就不用再重新編輯訊號清單了。請參考下圖：



『通道』可以讓您選擇資料產生器的輸出通道，0 表示選擇通道 0 來輸出 HDQ 訊號，1 表示選擇通道 1 來輸出，依此類推。

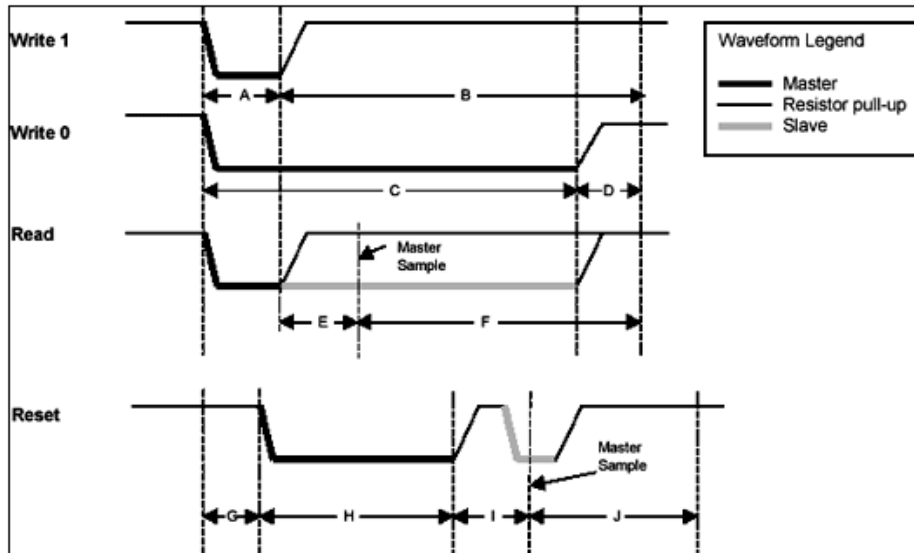
『傳輸率』根據 HDQ 協定，輸入範圍為 3.5K bits/s ~ 5.5K bits/s 之間，預設為 5K bits/s。

『連續輸出』則是選擇訊號是輸出一一次，還是重複輸出。

## 19. 1-Wire 訊號產生器

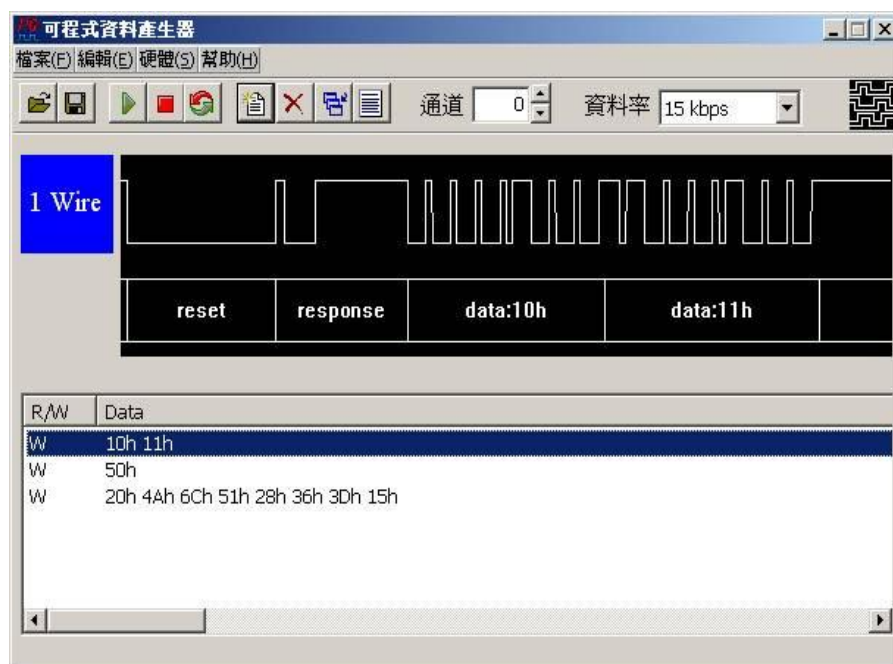


1-Wire 協定是由美國達拉斯公司(Dallas Semiconductor)所制定。1-Wire 協定定義了 Reset Pulse, Presence Pulse, Write 1, Write 0, Read 1, Read 0 等幾種訊號類型並由這些訊號類型組合成命令序列。通常應用於 EEPROM 的通信方面。



傳輸的方式為 LSB (Least-significant bit)到 MSB (Most-significant bit)，傳輸的速度分為高速(Overdrive speed)和低速(Standard speed)。

### 1-Wire 訊號產生器的使用方法



1-Wire 訊號產生器的使用方法主要是根據您的訊號需求，來編輯『1-Wire 輸出清單』讓資料產生器依照您的訊號清單來輸出該訊號，輸出清單裡的每一列就代表一個封包。

(1) 檔案

a. 清單匯出

將『1-Wire 輸出清單』儲存成文字檔。

b. 清單匯入

讀取所儲存的文字檔。

(2) 編輯

a. 新增封包

新增一個 1-Wire 的封包，它會附加至清單內現有封包之後。

b. 刪除所選取封包

刪除在清單內被選取的封包。

c. 插入新封包到所選擇的封包之後

插入一個新封包在清單內被選取的封包之後。

d. 修改所選擇封包

可以修改在清單內所選擇的封包。

e. 刪除所有封包

可以刪除在清單內所有的封包。



注意：在編輯封包時，請留意資料的編輯格式，每一個資料之間必須有一個空白(space)間隔，接受的資料為十六進制值，請參考上圖。

**(3) 硬體**

**a. 波形輸出**

將清單內的封包輸出一次。

**b. 停止輸出**

停止 **1-Wire** 封包輸出。

**c. 重覆輸出**

將清單內的封包重複輸出。

**(4) 幫助**

**a. 內容**

開啟關於 **1-Wire** 訊號產生器的線上使用說明。


**b. 通道**

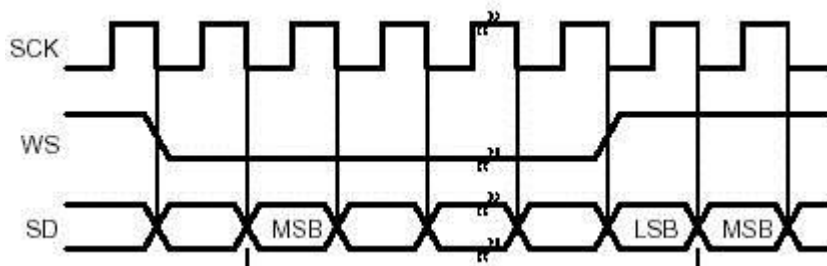
選擇 **1-Wire** 訊號輸出的通道。

**c. 資料率**

本程式支援低速 15kbps 和高速 111kbps。

## 20. I<sup>2</sup>S 訊號產生器

 I<sup>2</sup>S(Inter-IC Sound 或 Integrated Interchip Sound)是 IC 間傳輸數位音訊資料的一種介面標準，是飛利浦公司為數位音頻設備之間的音頻資料傳輸而制定的一種匯流排標準，常被使用在傳送 CD 的 PCM 音訊到 CD 播放器中的 DAC 上。在 I<sup>2</sup>S 的標準中，規定了硬體介面規範及數位音頻資料的格式，採用序列的方式傳輸 2 組(左右聲道)資料。由三條傳輸線組成，一條是時脈線(SCK)、另一條是字元選擇線(WS)以及資料線(SD)。資料格式最大到 20Bits。



### I<sup>2</sup>S 訊號產生器的使用方法



I<sup>2</sup>S 訊號產生器的使用方法主要是根據您的訊號需求，來編輯 I<sup>2</sup>S 資料設定讓資料產

生器依照您的訊號清單來輸出該訊號，輸出清單裡的每一列就代表一個封包。

**(1) I<sup>2</sup>S 通道設定**

設定資料產生器訊號輸出的通道，預設 SCK 為通道 0，WS 為通道 1，SD 為通道 2。

**(2) 資料寬度設定**

設定 I<sup>2</sup>S 資料寬度，有 16bits、20bits、24bits 可選擇。

**(3) I<sup>2</sup>S 時脈設定**

設定 I<sup>2</sup>S 時脈。

**(4) I<sup>2</sup>S 資料設定**

編輯 I<sup>2</sup>S 訊號資料內容 L = 75F5, R=5C75;表示第一筆的左通道資料為 0x75F5，右通道資料為 0x5C75。

L = DD35, R=65D1;表示第二筆的左通道資料為 0xDD35，右通道資料為 0x65D1。

編輯 I<sup>2</sup>S 訊號資料時請按照上述編輯格式以”,”分隔左右通道資料，然後以”;”作為結尾。

**a. 匯出**

將編輯的資料儲存成文字檔(.txt)。

**b. 匯入**

開啟儲存的文字檔。

**c. 範本**

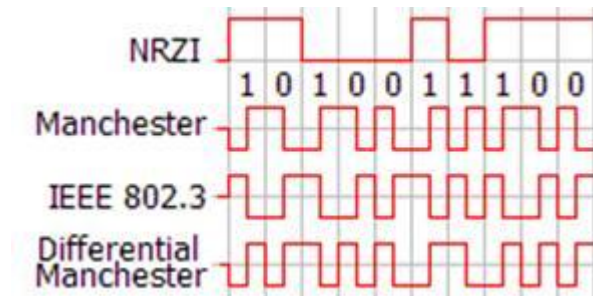
產生 I<sup>2</sup>S 訊號資料的範本。

**(5) 連續輸出**

連續輸出所編輯的 I<sup>2</sup>S 訊號。註：I<sup>2</sup>S 訊號產生器輸出的 I<sup>2</sup>S 訊號為 I<sup>2</sup>S 格式。

## 21. 編碼器

編碼器有四種編碼選擇，分別是 NRZI, Manchester(Thomas), Manchester(IEEE 802.3), Differential Manchester。



### 編碼器的使用方法



#### (1) Encoder 通道設定

選擇輸出的通道。

#### (2) 資料方向

選擇所要輸出的資料方向為 MSB 或是 LSB。例如 16 進制值 1a，MSB => 00011010, LSB => 01011000。

**(3) Encoder 編碼選擇**

可以選擇 NRZI、Manchester(Thomas)、Manchester(IEEE 802.3)、Differential Manchester。

**(4) Data Rate**


產生 1 bit 資料的時間。

**(5) 範本**

產生範例資料，此為輸出資料之正確格式(為 16 進制數值，一列一個並以分號隔開)。



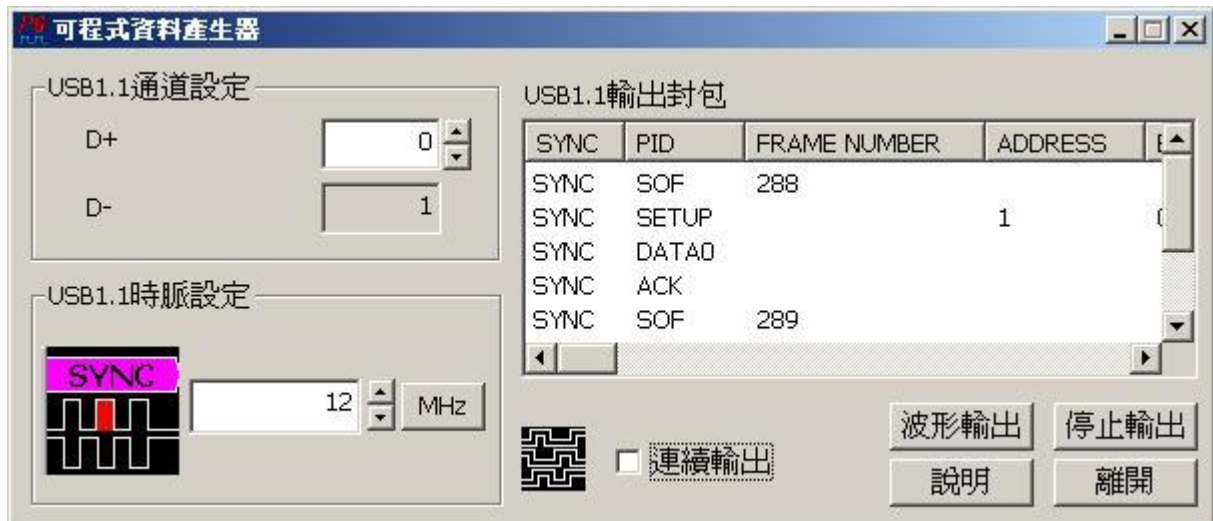
## 22. USB1.1 訊號產生器

 USB (Universal Serial Bus)稱為”萬用串列匯流排”，起初由 7 家公司所制訂的規格：Intel, Microsoft, National Semiconductor, Compaq, Nortel, NEC and AT&T。

### 標準 USB 介面

Pin	Name	Cable color	Description
1	VCC	Red	+5V
2	D-	White	Data -
3	D+	Green	Data +
4	GND	Black	Ground

### USB1.1 訊號產生器的使用方法



USB1.1 訊號產生器的使用方法主要是根據您的訊號需求，來編輯 USB1.1 輸出封包讓資料產生器依照您的訊號清單來輸出該訊號，輸出清單裡的每一列就代表一個封包。

**(1) USB1.1 通道設定**

主要是選定資料產生器的通道來代表 USB1.1 的 D+和 D-接腳。

**(2) USB1.1 時脈設定**

設定 USB1.1 時脈，預設為 12MHz。

**(3) USB1.1 輸出封包**

為 USB1.1 訊號封包清單，輸出的順序為由上而下。透過底下的表單(在清單區按下滑鼠右鍵)來編輯該輸出封包清單：



**a. 新增封包**

新增一個封包到 USB1.1 輸出封包清單。

**b. 刪除封包**

刪除所選擇的封包。

**c. 插入封包**

新增一個封包到所選擇的封包之後。

**d. 修改封包**

修改所選擇的封包內容。

**e. 範例**

顯示一個 USB1.1 輸出封包清單範例。

**f. 刪除所有封包**

刪除 USB1.1 清單中所有封包。

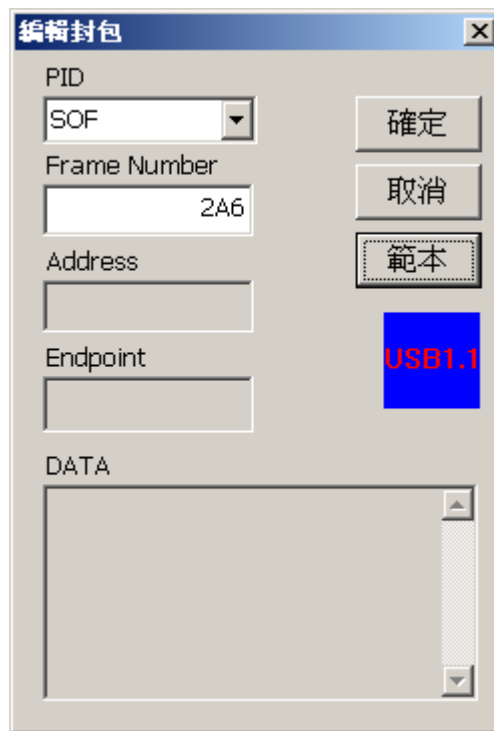
**g. 儲存檔案**

將清單內容儲存成一個文字檔。

**h. 開啟檔案**

開啟所儲存的清單內容的檔案。

- 新增封包、插入封包和修改封包都會出現底下的編輯封包的對話盒：



- 輸入封包的內容值，程式會自動計算出 CRC5 或 CRC16 的值並且根據所選擇的 PID 來組成 USB 的 Token, Data, Handshake 封包。

**(4) 連續輸出**

會重覆輸出 USB1.1 封包清單的內容。

註:所有輸入的數值都以十六進制表示。

## 23. SMBus 訊號產生器

**SMBus** SMBus 全名系統管理匯流排 (System Management Bus) 源自於 I<sup>2</sup>C 匯流排，是一種兩條訊號所組成的一種匯流排。SMBus 由 Intel 於 1995 年所定義。包含有 clock, data, 以及基於 Philips' I<sup>2</sup>C serial bus 協定的指令。其時鐘頻率範圍在 10 kHz 到 100 kHz。

### SMBus 訊號產生器的使用方法



SMBus 訊號產生器的使用方法主要是根據您的訊號需求，來編輯 SMBus 輸出封包讓資料產生器依照您的訊號清單來輸出該訊號，輸出清單裡的每一列就代表一個封包。

#### (1) SMBus 封包內容設定

SMBus 的封包種類共有 21 種，分別是 Quick command protocol, Send byte protocol, Send byte protocol with PEC, Receive byte protocol, Receive byte protocol with PEC, Write byte protocol, Write Word Protocol, Write byte

protocol with PEC, Write Word Protocol with PEC, Read Byte Protocol, Read byte protocol with PEC, Read word protocol, Read word protocol with PEC, Process Call, Process Call with PEC, Block Write, Block Write with PEC, Block Read, Block Read with PEC, Block Write - Block Read Process Call 和 Block Write - Block Read Process Call with PEC。封包種類的說明請自行參考 SMBus 規格(version 2.0)。首先選擇封包種類，程式會根據所選擇的封包種類對 Address, Command, Data 和 Sr (Repeated Start)做相對應的回應。假設選擇的封包種類是 Process Call with PEC，該種類的封包內容如下：

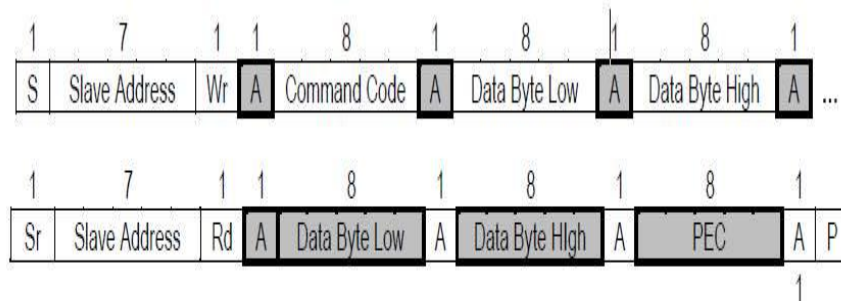


Figure 5-16: Process Call with PEC

- (2) 根據封包內容輸入 Address, Command Code, Data 和 Sr 然後按下確定。做法如下：



Address 和 Command 使用預設值 0x10 以及 0x20，Data 輸入 2F 和 3A(皆為十六進制值)。2F 為 Data Byte Low 的數值，3A 為 Data Byte High 的數值，數值之後必須接一個分號";"如上圖所示。然後根據 Process Call with PEC 格式，接著按下 Sr 按鍵產生一個 Repeated Start，然後輸入 slave 回應 master 的資料(例如：Data Byte Low: 3A, Data Byte High: 2F)，再按下確定即完成一個 Process Call with PEC 種類的封包，而封包中的 ACK 或是 NACK，READ/WRITE 或是 PEC，程式會自動加入。結果如下：



上圖列表中的 S 表示 Start, ADDR 表示 Address, CMD 表示 Command, W 表示 WRITE, R 表示 READ, L 表示 Data Byte Low, H 表示 Data Byte High。該封包後面看不見的部份，您可以按下滑鼠右鍵，選擇修改封包即可。



**a. 刪除封包**

將選擇的封包刪除。

**b. 插入封包**

插入一個新的封包到所選擇封包的下一個(順序由上到下)。

**c. 刪除所有封包**

刪除列表中所有封包。

**d. 儲存檔案**

將列表中的內容存成文字檔。

**e. 開啟檔案**

開啟所儲存的封包內容文字檔。

**(3) 波形輸出**

根據列表中的內容輸出波形。

**(4) 停止輸出**

停止輸出波形。

**(5) 說明**

顯示該工具程式的說明文件。

**(6) 離開**

離開該程式。

**(7) 連續輸出**

連續輸出列表中的封包內容。



## 24. PMBus 訊號產生器

**PMBus** **PMBus** **PMBus** PMBus 全名電源管理匯流排 (Power Management Bus) 是一種兩條訊號所組成的一種匯流排。包含有 clock, data。和 SMBus 相同皆源自於 I<sup>2</sup>C 匯流排，而 PMBus 最大傳輸速度為 400KHz 高於 SMBus 的 100KHz。

### PMBus 訊號產生器的使用方法

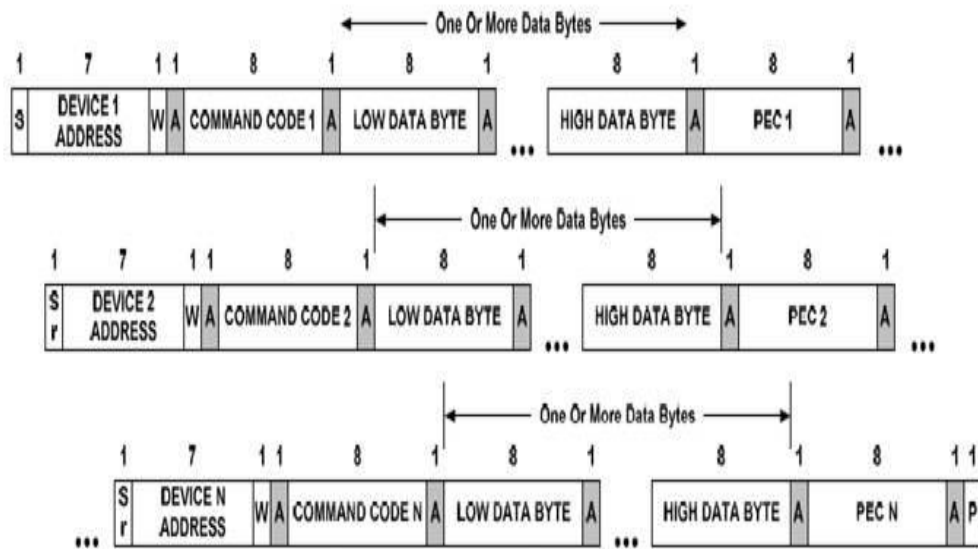


PMBus 訊號產生器的使用方法主要是根據您的訊號需求，來編輯 PMBus 輸出封包讓資料產生器依照您的訊號清單來輸出該訊號，輸出清單裡的每一列就代表一個封包。

#### (1) PMBus 封包內容設定

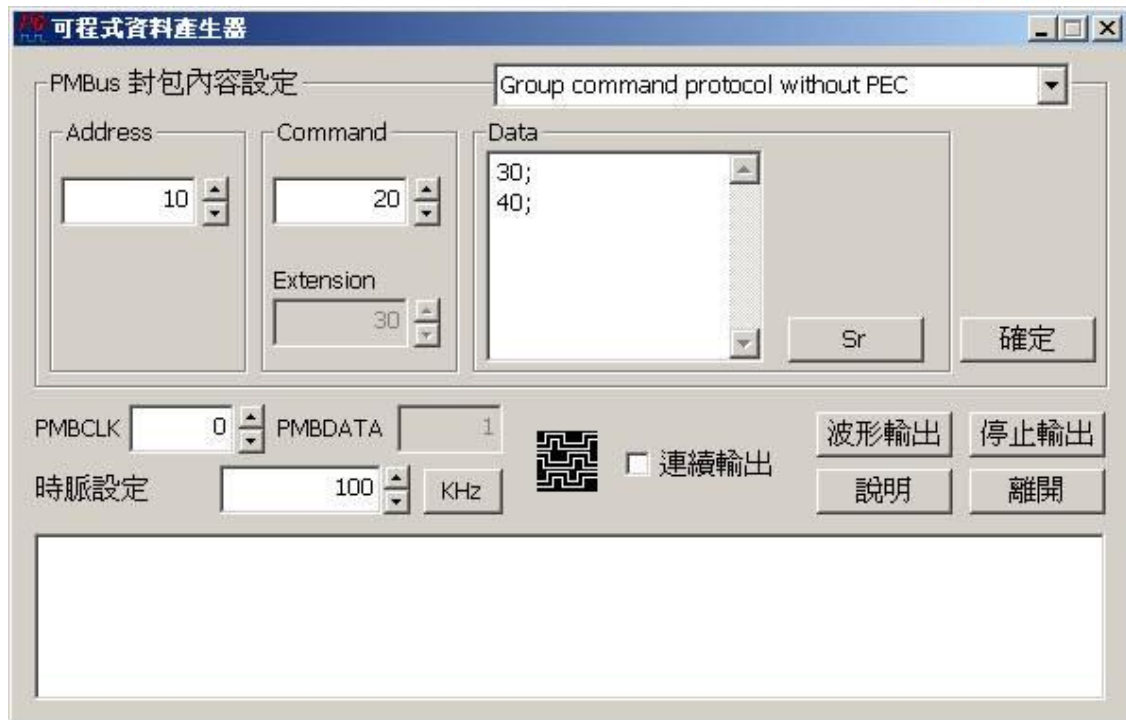
PMBus 的封包種類共有 10 種，分別是 Group command protocol without PEC, Group command protocol with PEC, Extended command read byte protocol, Extended command read byte protocol with PEC, Extended command write byte protocol, Extended command write byte protocol with PEC, Extended command

read word protocol, Extended command read word protocol with PEC, Extended command write word protocol 和 Extended command write word protocol with PEC。封包種類的說明請自行參考 PMBus 規格(Revision 1.1)。首先選擇封包種類，程式會根據所選擇的封包種類對 Address, Command, Data 和 Sr (Repeated Start)做相對應的回應。假設選擇的封包種類是 Group command protocol with PEC，該種類的封包內容如下：



**Figure 3. Group Command Protocol With PEC**

- (2) 根據封包內容輸入 Address, Command, Data 和 Sr 然後按下確定。做法如下：



Address 和 Command 使用預設值 0x10 以及 0x20，Data 輸入 30 和 40(皆為十六進制值)。30 為 Data Byte Low 的數值，40 為 Data Byte High 的數值，數值之後必須接一個分號”；”如上圖所示。然後根據 Group command protocol with PEC 格式，接著按下 Sr 按鍵產生一個 Repeated Start，然後輸入 slave 回應 master 的資料(例如：Data Byte Low: 50, Data Byte High: 60)，再按下確定即完成一個 Group command protocol with PEC 種類的封包，而封包中的 ACK 或是 NACK，READ/WRITE 或是 PEC，程式會自動加入。結果如下：



上圖列表中的 S 表示 Start, ADDR 表示 Address, CMD 表示 Command, W 表示 WRITE, R 表示 READ, L 表示 Data Byte Low, H 表示 Data Byte High。

該封包後面看不見的部份，您可以按下滑鼠右鍵，選擇修改封包即可。





**a. 刪除封包**

將選擇的封包刪除。

**b. 插入封包**

插入一個新的封包到所選擇封包的下一個(順序由上到下)。

**c. 刪除所有封包**

刪除列表中所有封包。

**d. 儲存檔案**

將列表中的內容存成文字檔。

**e. 開啟檔案**

開啟所儲存的封包內容文字檔。

**(3) 波形輸出**

根據列表中的內容輸出波形。

**(4) 停止輸出**

停止輸出波形。

**(5) 說明**

顯示該工具程式的說明文件。


**(6) 離開**

離開該程式。

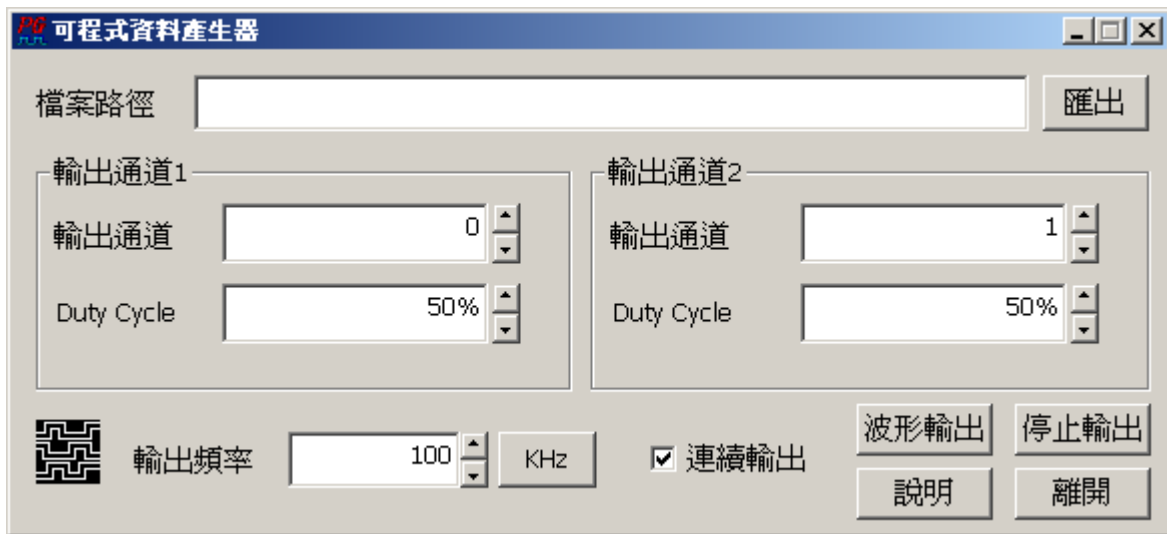
**(7) 連續輸出**

連續輸出列表中的封包內容。

## 25. PWM 訊號產生器

 PWM (Pulse Width Modulation)，稱為脈寬調變，它不是一種匯流排分析協定。主要是利用脈衝寬度之週期對類比電路進行控制的一種非常有效的技術，廣泛應用在一些轉速控制、亮度控制和溫度控制等。

### PWM 訊號產生器的使用方法



PWM 訊號產生器可以同時輸出 2 個不同 Duty Cycle 的 PWM 訊號，還可以將設定好的 PWM 訊號儲存成文字向量檔 (\*.PGV)。

#### (1) PWM 通道設定

主要是設定 PWM 的輸出通道 1/2 以及該 PWM 訊號的 Duty Cycle。

#### (2) PWM 輸出頻率

設定 PWM 頻率，預設為 100KHz。

#### (3) 匯出檔案

將設定的 PWM 波形匯出為文字向量檔(\*.PGV)。

#### (4) 波形輸出

根據列表中的內容輸出波形。

#### (5) 停止輸出

停止輸出波形。

**(6) 說明**

顯示該工具程式的說明文件。

**(7) 離開**

離開該程式。

**(8) 連續輸出**

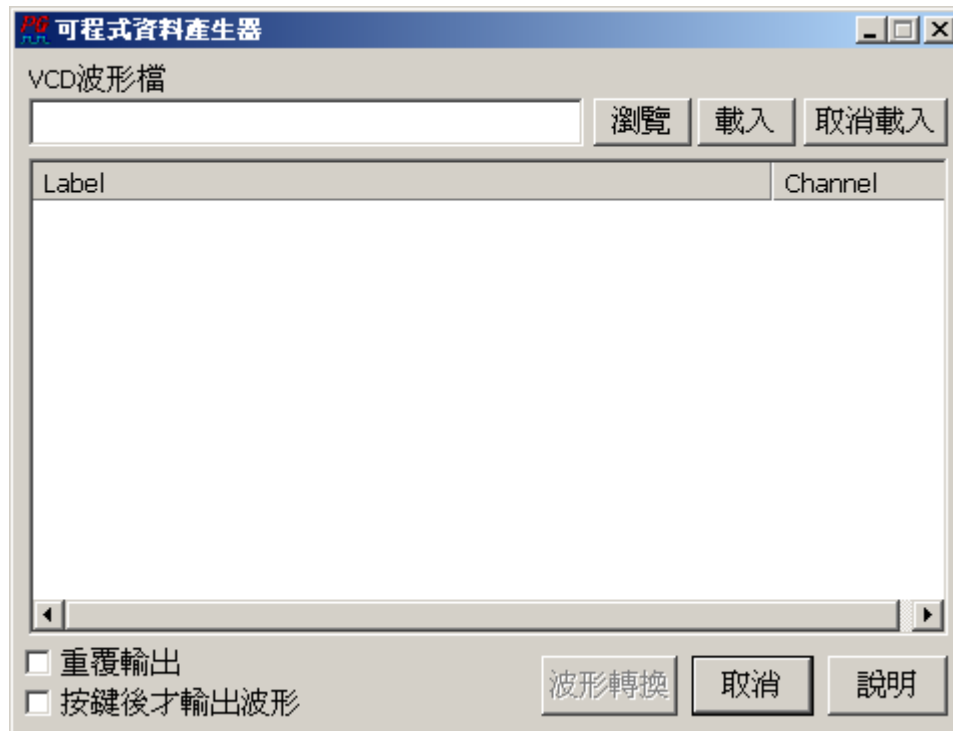
連續輸出列表中的封包內容。

## 26. VCD 波形檔轉換



Value change dump，簡稱 VCD。用於記錄由 EDA 模擬工具所產生的訊號訊息。

### VCD 波形檔轉換的使用方法



VCD 波形檔轉換可以將 VCD 檔案的內容轉換成 PG 的波形。

#### (1) 瀏覽

選擇 VCD 檔案。

#### (2) 載入

載入 VCD 檔案。

#### (3) 取消載入

取消載入 VCD 檔案。

#### (4) 重覆輸出

連續輸出列表中的封包內容。



**(5) 按鍵後才輸出波形**

在轉換成 PG 波形的前端加入 Keyboard event (space)。

**(6) 波形轉換**

轉換成 PG 波形。

**(7) 取消**

離開該程式。

**(8) 說明**

顯示該工具程式的說明文件。

## 第4章 注意事項

## 1. 硬體注意事項

- (1) 將 Acute PG 與標的物連接時，請注意不要把輸出端接到 Ground 或 VCC 上。因為 Acute PG 是一種輸出裝置，如果將輸出端與標的物的 Ground 或 VCC 接在一起，會造成 Acute PG 的前端隔離放大器的損壞。
- (2) 請不要使用非本公司提供的整流穩壓器(Power Adapter)，來連接 PG1000/PG2000 的電源接頭。本公司提供的整流穩壓器是一個符合 PG1000/PG2000 電源需求的電源供應器，他的穩壓與功率都是經過計算的。如果使用自行購買的整流穩壓器，可能造成 PG1000/PG2000 工作不正常或是零件破壞。
- (3) Pocket PG 是可以調整輸出電壓值所以調整電壓時，請注意標的物電壓容許值，否則將影響或破壞該標的物。
- (4) 每個隔離放大器及 Pocket PG 的地線至少要接一支到標的物上，如量測出的波形訊號干擾太大時，就多接一支地線到標的物上以改善量測的品質。
- (5) 隔離放大器雖然有標示 A、B、C、D、E，但是每個隔離放大器都是一樣的，所以任何一個隔離放大器都可以對調。可是建議不要亂接，避免通道編號與 PC 端的對應，產生不必要的錯誤。但是 Extended Pod 是特殊的隔離放大器，請不要將它接到 A、B、C、D、E 等 Pod 的接頭上，同樣的 A、B、C、D、E 的 Pod 也不能接到 Extended Pod 的接頭上，否則會對 PG1000/PG2000 主機造成傷害。
- (6) PG1000/PG2000 的主機與界面卡的接頭和 Printer 所使用的 Cable 相同，但是不要將 PG1000/PG2000 與 Printer 同時並聯接在一起，否則將有不可預期的後果。而且請注意，不要使用自行購買的 Printer Cable 來連接 PG1000/PG2000 和 PC，請使用本公司提供的連接 Cable，如此我們才能保證 PG1000/PG2000 的工作正常。
- (7) Pocket PG 使用的是 USB 傳輸界面，所以必需用 USB Cable 連接 Pocket PG 與 PC 主機，使用 USB Cable 時請不要讓接頭部份搖晃，否則連接器一旦鬆動，將造成 Pocket PG 無法正常工作。

- (8) 將 PG1000/PG2000 主機安裝在 PC 的磁碟機插槽時，儘量放在下方的位置，因為 PG1000/PG2000 的面板會接上排線，所以可能會擋住放在下方的磁碟機或光碟機，造成使用上的不便。
- (9) 標的物的測試腳，如果用 Acute PG 所提供的探針，仍很難正常的接在標的物上時，可以向本公司選購較細的探針，或是測試夾等配備，或是自行購買合適的探針或測試夾。

## 2. 軟體注意事項

- (1) 軟體程式或是驅動程式安裝的順序不會影響正常工作，但是要執行軟體之前，必須已經裝好硬體驅動程式才能正常工作。
- (2) 安裝新版軟體時，請先將舊版軟體移除。
- (3) 請至 <http://www.acute.com.tw> 網站查詢新軟體版本，自行更新。

## 第5章 其他

## 1. 故障排除

(1) 如果執行 PG Editor 時出現『Demo Mode』，就代表您的安裝有問題，請按下列步驟處理：

### ● PCI 與 EPP 介面

- a. 檢查電源接頭是否接好。
- b. 檢查傳輸界面的接頭是否接好。
- c. 如果您使用內接式，請至控制台-系統-裝置管理員-系統裝置中尋找是否有一個 LP1000 的裝置。如果沒有就代表界面卡沒有接好，請關閉 PC 電源並將界面卡重新插入 PCI 插槽或換另一個插槽。
- d. 如果您使用外接式，請檢查 PC BIOS 的 Printer 設定是否已經設成 EPP 模式(請參考『Printer Port 的 BIOS 模式設定』)。
- e. 做完上述檢查後，請重新新啟動 PG-Editor 程式。
- f. 如果仍然無法正常使用本產品，請與我們聯絡，讓我們為您服務。

### ● USB 介面

- a. 至裝置管理員中，檢查驅動程式是否存在。(參考硬體安裝一節)
- b. 如果驅動程式存在的話，請確認是否已安裝最新版本的 PG-Editor。
- c. 如果已安裝最新版本的 PG-Editor，請重新啟動 Windows 作業系統。
- d. 如果未安裝最新版本的 PG-Editor，請至 Acute 網站 <http://www.acute.com.tw> 上 Download 並重新安裝。
- e. 如果驅動程式不存在時，請重新插拔 USB Cable，並檢查驅動程式是否已出現。
- f. 驅動程式如果仍然不存在時，請重新開機，並再次檢查驅動程式是否存在。
- g. 經過第六步驟後，驅動程式還是不存在，此時請注意 USB Cable 插拔時，是否新增一個其他的裝置，而並非是 Acute PG 的驅動程式，如果是這樣，請與本公司聯絡。

- (2) 輸出波形時，如果訊號未正常輸出，請按下列步驟處理：
- a. 檢查探針是否正常接在訊號連接線上。
  - b. 檢查訊號連接線是否正常接在隔離放大器或 Pocket PG 上。
  - c. 檢查探針是否與標的物正常連接。
  - d. 檢查標的物是否有訊號產生(可將有訊號產生的通道與有問題的通道對調來確定問題所在)。
  - e. 再重新輸出波形一次。
- (3) 當使用波形編輯器時，如果波形無法如預期的結果時，請按波形檢查鈕，看看編輯波形有沒有錯誤。或是按環境設定鈕，檢查基本頻率的設定有沒有設錯。

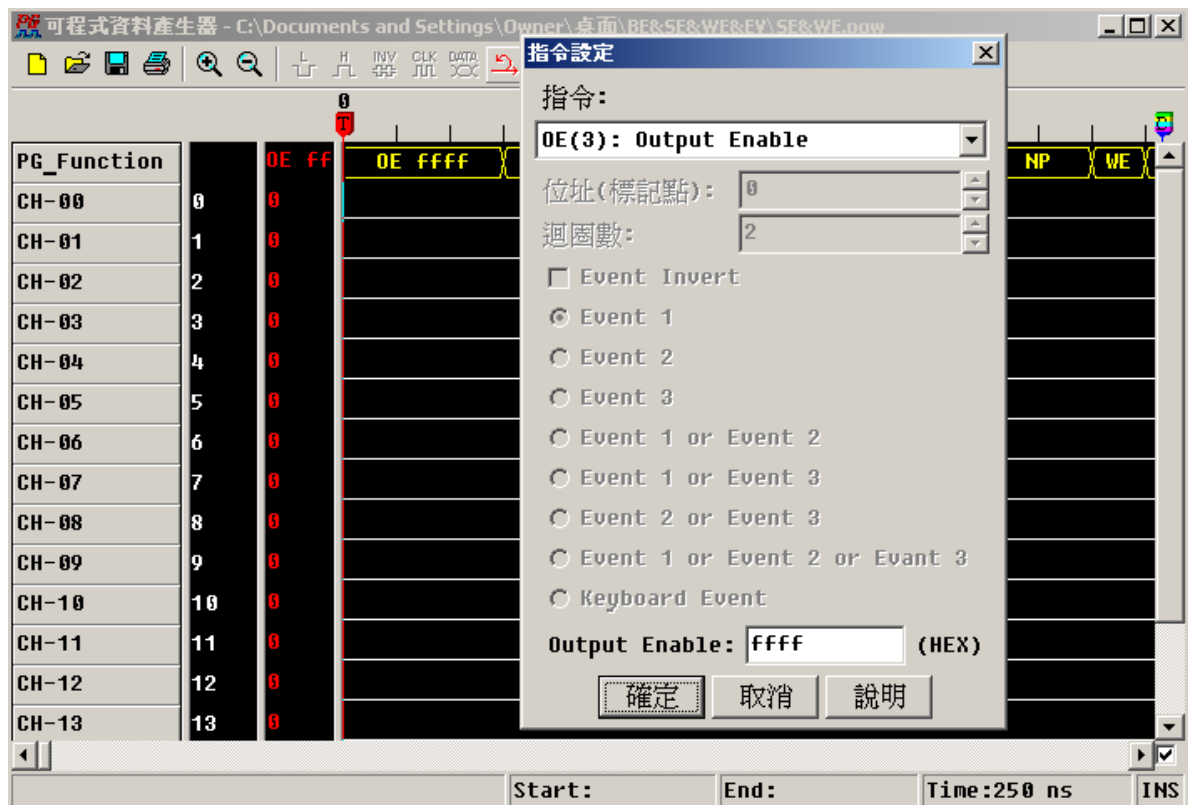


## 2. PG\_Function 指令使用方法

### (1) Output Enable OE(3):

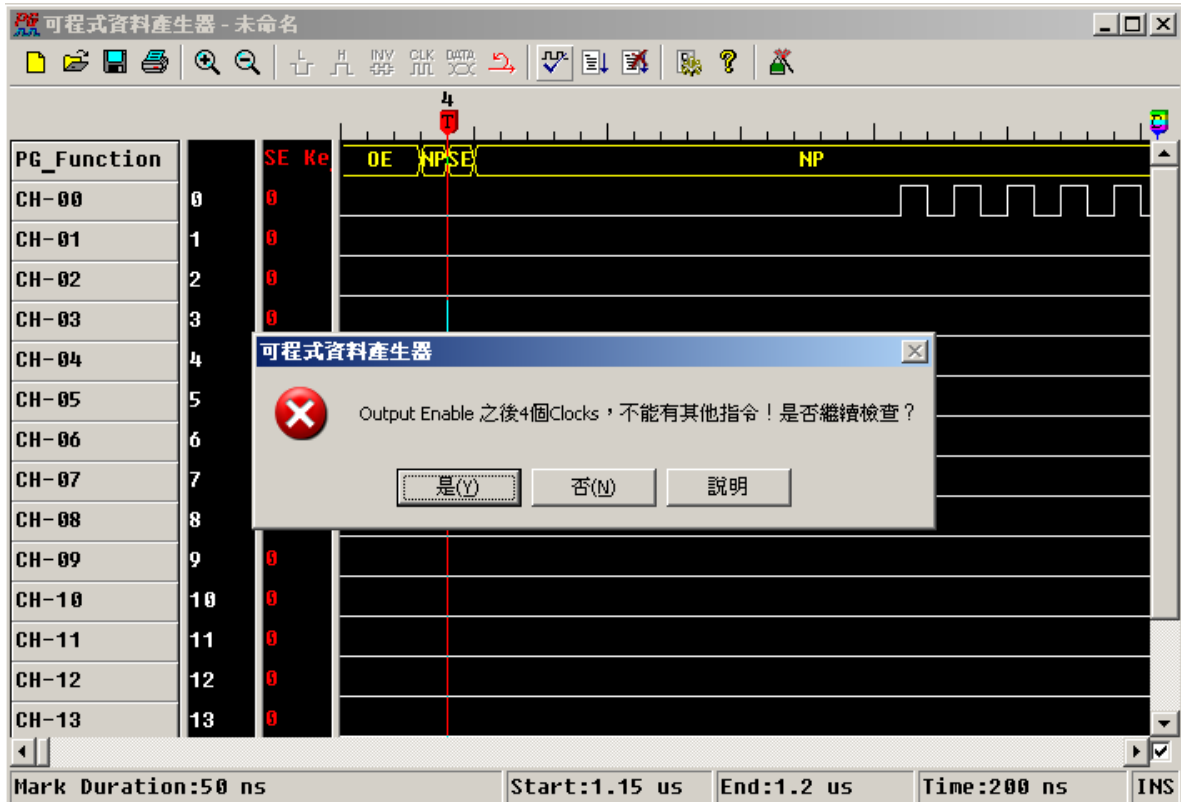
OE 該指令是 PKPG<sup>註</sup>系列才支援的指令而 PG<sup>註</sup>系列並不支援。這裡輸入 0xFFFF 是表示 PKPG 的 16 個資料輸出腳輸出致能，如果是輸入 0x000F 則表示 PKPG 的 CH-00~CH-03 4 個資料輸出腳為輸出致能，其他資料輸出腳為高阻抗；如果您輸入為 0xF000，則表示 PKPG 的 CH-12~CH-15 4 個資料輸出腳為輸出致能，其他為高阻抗。

OE(3) 括號裡的數字 3 表示該指令使用 3 個 Clock。而使用該指令之後的 4 個 Clock 不能在使用其他指令。



<sup>註</sup> PKPG 系列包含 PKPG2016/2116/2016+/2116+

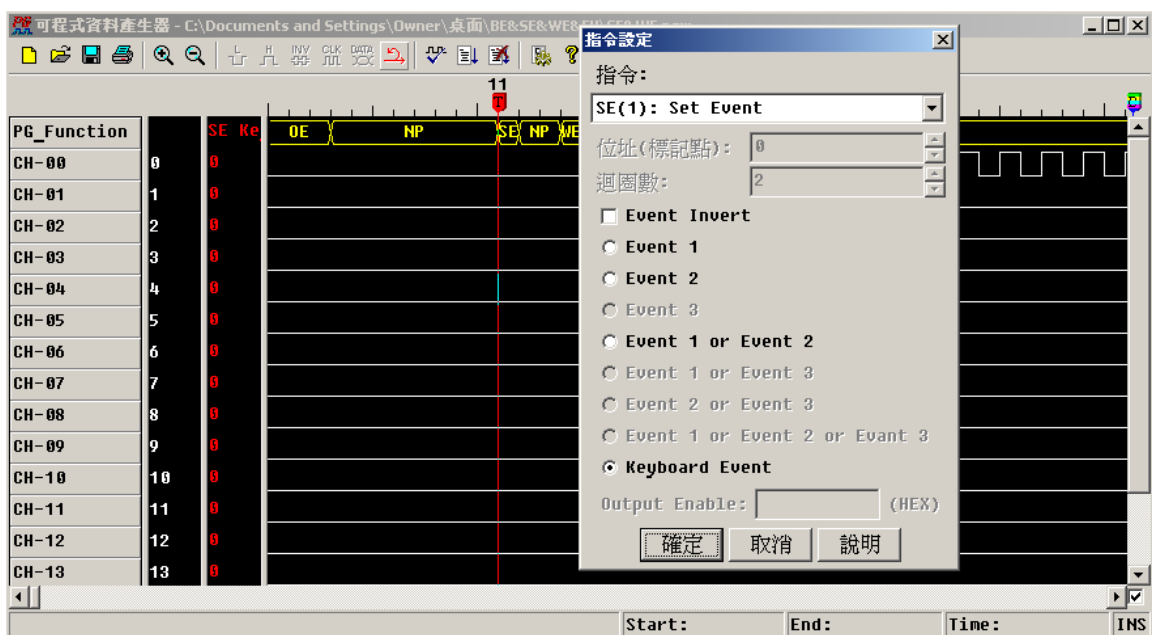
<sup>註</sup> PG 系列包含 PG1020/2020/1050/2050

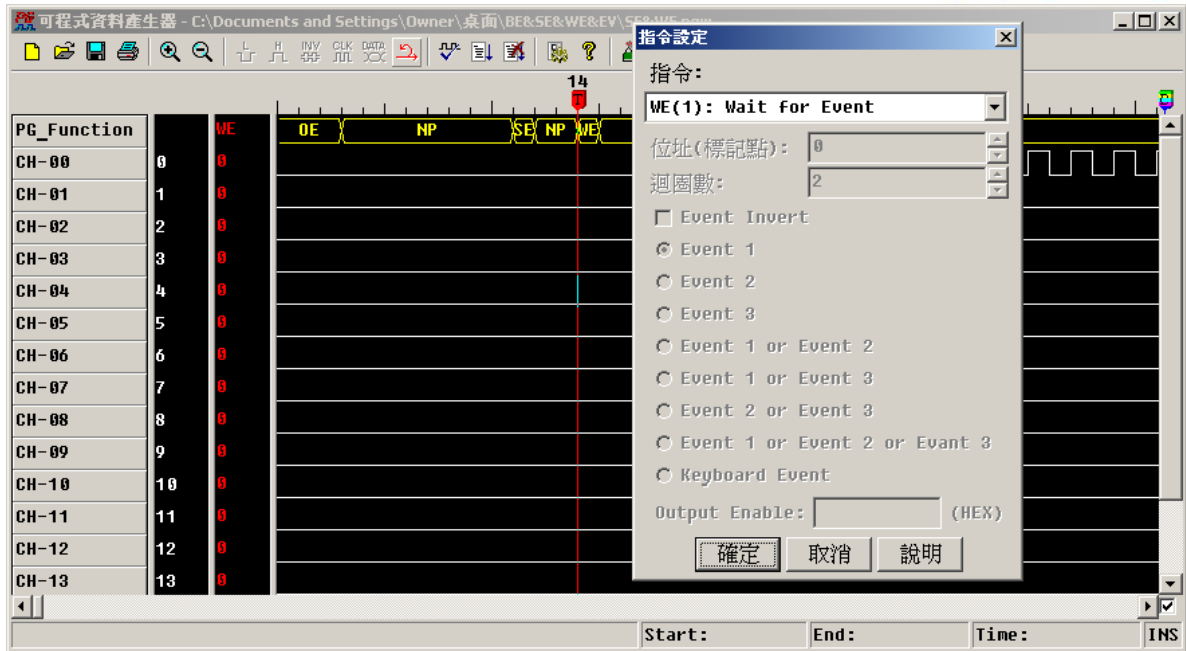


上圖即是在使用 OE 該指令後未間隔 4 個 Clock 而接著使用 SE 該指令導致錯誤發生。再每次輸入指令時，如果您不確定每個指令的間隔是否正確，都可以按下[波形檢查]按鈕來檢查每個指令的間隔時間是否正確。

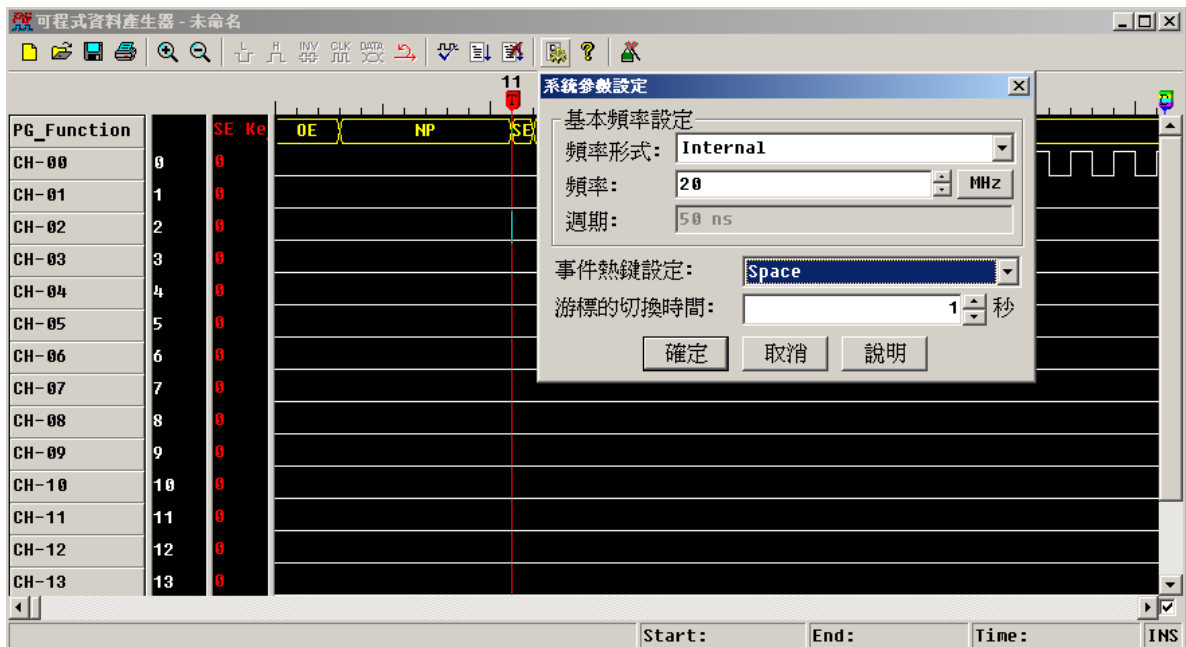
(2) Set Event SE(1) 和 Wait Event WE(1):

SE(1) 和 WE(1) 是設定 PG 在碰到某事件發生時，接下來要執行的動作。



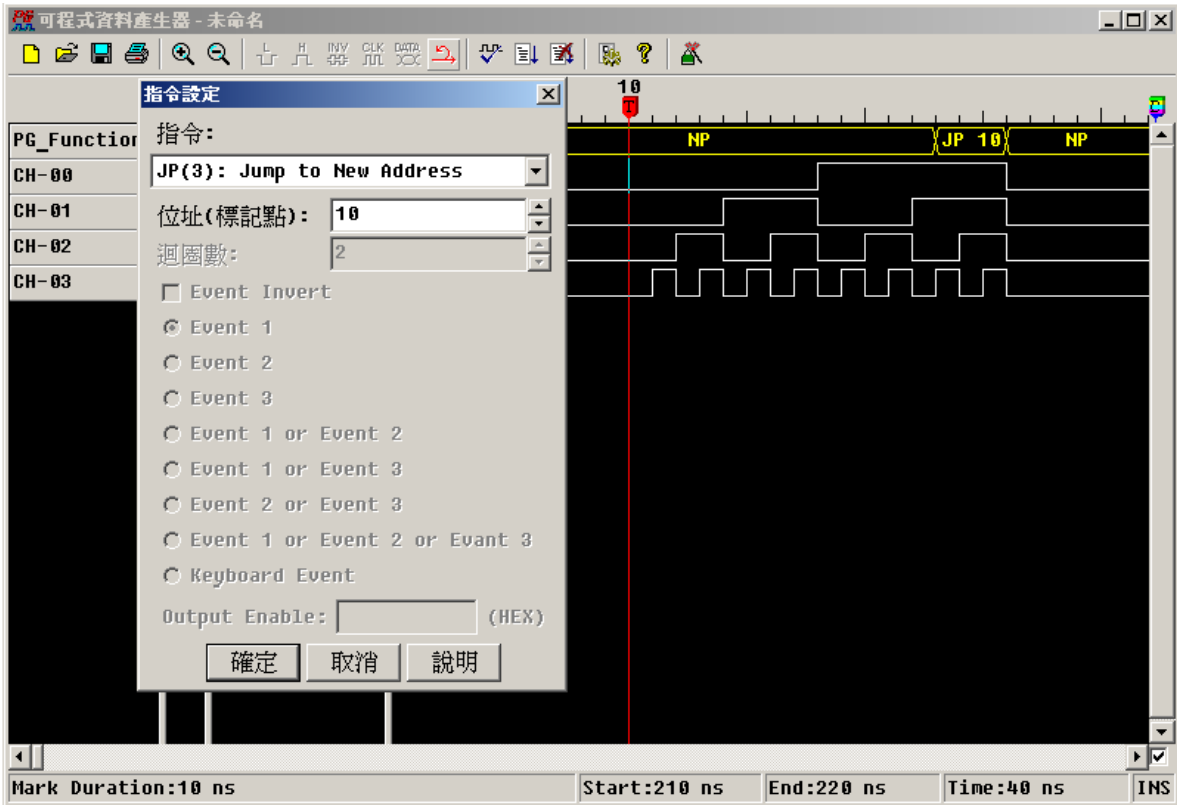


這一組 SE 和 WE 指令的輸出結果：在按下[波形輸出]按鈕時，PG 並未輸出波形，直到按下 PC 鍵盤的 <space> 鍵，才會輸出連續的波形。這是因為設定 PG 要等到 Keyboard 事件的發生，PG 才會動作，否則 PG 只會一直等待事件的發生。而 Keyboard 可以選擇 <space> 或 <enter>。



### (3) Jump to New Address JP(3):

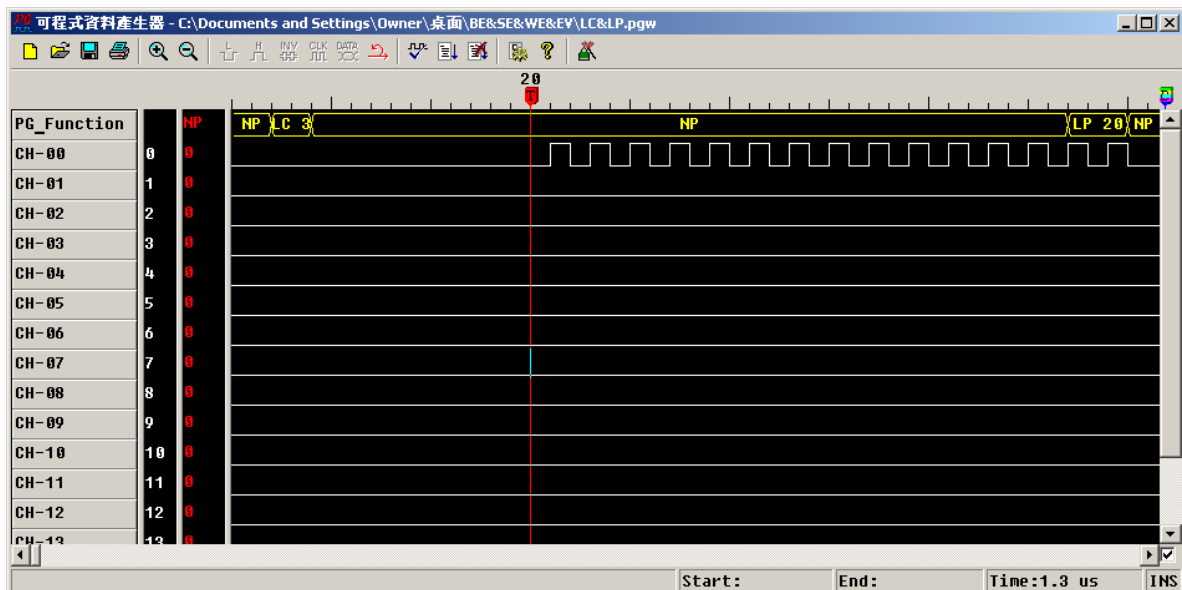
表示 PG 無條件跳躍至新位址。



上圖表示 PG 會無限次的輸出位址 10~位址 26 的波形，直到按下[停止輸出波形]。

(4) Set Loop Count LC(3) 和 Loop to New Address LP(3):

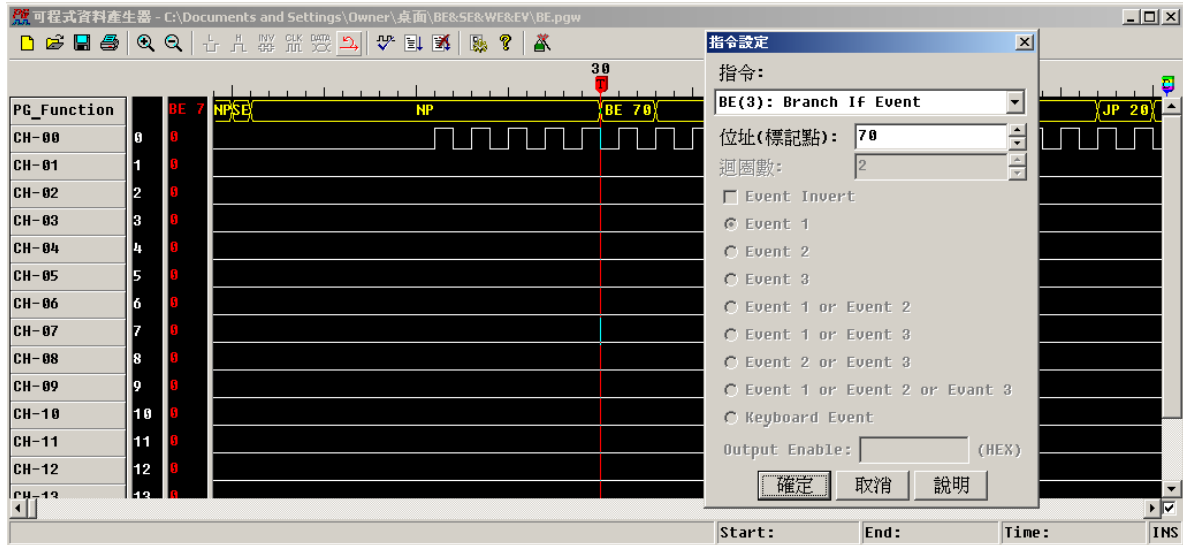
表示設定 PG 有次數限制的輸出波形。LC 是設定輸出次數；LP 是設定輸出波形的範圍，請參考擷圖說明的部份。



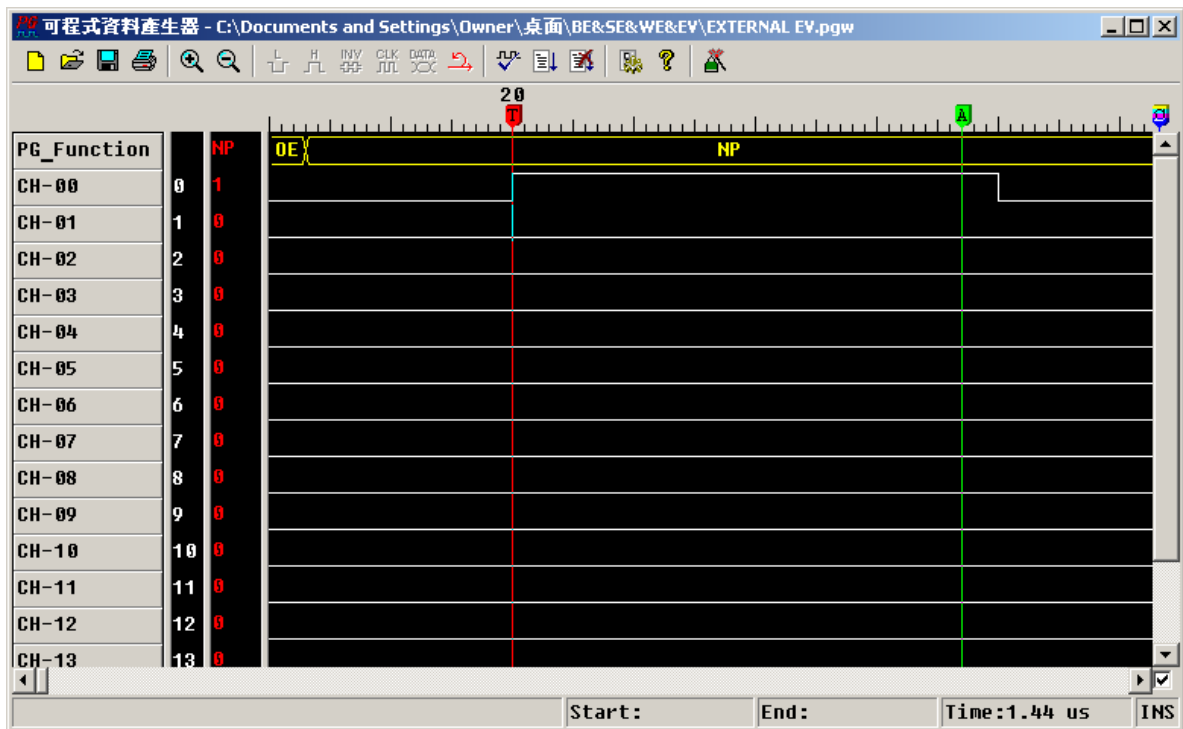
上圖表示 PG 會輸出圖上的波形 3 次。

(5) Branch If Event BE(3):

表示當事件發生時，PG 跳躍至指定的新位址。請參考擷圖說明：



圖中的 SE 設為外部事件 Event1。這個例子是使用另一部 PG 當作外部事件的訊號源，將它簡稱為 PG2。而原先那一個則簡稱為 PG1。接法如下:接 PG2 的 CH-00 連接到 PG1 的 EV1 接腳，然後 PG2 送出如下圖之訊號：



由上圖可知，若 Event1 一直沒發生，則 PG1 會重複送出位址 20~位址 60 的波形。若 Event1 事件發生，則 PG1 跳躍至位址 70，輸出位址 70 後的波形。

### 3. 使用文字編輯器編輯文字向量檔

可程式化資料產生器除了可讀取波形檔(\*.PGW)之外,還可以讀取以文字編輯器所編輯的文字檔,而該文字檔.\*PGV (**PG Vector File**)的內容主要是您想要觀察的一些資料以及給 PG 的指令等。

而該文字檔有一定的格式,以下的內容就該文字檔的格式及使用方法做說明,首先以下為擷取某文字向量檔的部分來做為範例並透過它來說明 **PGV** 檔的格式部分。

文字向量檔還有根據您所下的指令(Keyword)的不同而有不一樣的檔案內容,在定義波形資料的區域,可以區分為使用『Time Stamp』與否。除了介紹基本指令,還有許多在應用上會使用到的 PG\_Function 訊號組的波形指令,整個波形指令總共有 7 個,包括 **NP(No Operation)**、**JP(Jump)**、**LP(Loop)**、**BE(Branch if Event)**、**LC(Loop Count)**、**SE(Set Event)**和 **WE(Wait Event)**等。以下會針對這些指令來做說明:

INPUTS PG\_Function DATA;

ASSIGN DATA 3..0;

RADIX AUTO;

FREQUENCY 1000 Hz;

%INTERVAL 1ms;%

『%..%』:註解符號

PATTERN

8FFh 0h // 0 ( MOV RL, 255 )

2FFh 0h // 1 ( MOV RH, 255 )

900h 0h // 2 OE 65535

000h 0h // 3

000h 0h // 4

000h 0h // 5 『//』:註解符號

000h 0h // 6

000h 0h // 7

000h 0h // 8

000h 0h // 9

000h 0h // 10 START PATTERN

000h 1h // 11

000h 2h // 12

000h 3h // 13

000h 4h // 14

000h 5h // 15

000h 6h // 16

000h 7h // 17

```

000h    8h  //  18
000h    9h  //  19
000h   Ah  //  20
000h   Bh  //  21
000h   Ch  //  22
816h   Dh  //  23          ( MOV RL, 22 )
200h   Eh  //  24          ( MOV RH, 0 )
100h   Fh  //  25          JP 10
000h   0h  //  26
;

```

上述的範例為一個 4Bits、1KHz 計數器的應用。以上述範例先來說明 PGV 的指令：

#### **INPUT PG\_Function DATA**

設定訊號名稱。每一個名稱用空白分隔開來，名稱可以為文字或是數字如果為 Bus(Group)時，可以用中括號來指定。例如：某個 Bus 為 4 個通道所組成那就可以用 A[3..0]來代替，A[3..0]就代表 A3、A2、A1、A0 等 4 個訊號。

注意！『PG\_Function』是一個保留字，如果在 INPUTS 指令的定義中有『PG\_Function』，就代表 PATTERN 中會包含 PG 的專用指令。而且『PG\_Function』也不能配合 ASSIGN 指令使用。

#### **ASSIGN DATA 3..0**

用來指定 INPUTS 指令所定義的訊號名稱之通道值。結果如下

DATA0 = CH0 , DATA1 = CH1.....DATA3= CH3 等。



## **RADIX AUTO**

RADIX 為區域的進位值。設定 PATTERN 區域的進位值。如果 PATTERN 區域的數值有進位識別符號時，此時就要將 RADIX 的值設成 AUTO。例如 RADIX 設定成 AUTO 時，PATTERN 區域的某一數值為 35 與 35h 是不一樣的，但是當 RADIX 設成 HEX(十六進位)時，這兩個值就是一樣的。但是當 RADIX 設成 DEC 時，而 PATTERN 區域的值為 35h 時，卻是會被當成 35。

RADIX 共有五種定義：

AUTO：由數值的進位識別符號決定。

HEX：16 進位。

DEC：10 進位。

OCT：8 進位。

BIN：2 進位。

而在 PATTERN 區域的進位識別符號定義是『h』代表 16 進位、『o』代表 8 進位和『b』代表 2 進位，而不加任何識別符號就代表 10 進位。

## **FREQUENCY 1000 Hz**


這裡是指定使用 PG 主頻為 1000Hz, 主要為了設定計數器的頻率 1KHz。

## **PATTERN**

是用來定義波形資料的區域。這個區域包含了兩個部份，一個是時間部份一個是波形部份。時間部份稱為『Time Stamp』，時間部份和波形部份用『>』（大

於符號)來做分隔。時間部份也可以省略，此時波形部份每一行的時間就會由 INTERVAL 或是 FREQUENCY 來指定，所以 INTERVAL 和 FREQUENCY 只能選用一種。如果使用 Time Stamp 方式，時間部份的單位就是 UNIT 所設定的值。波形部份就是根據 INPUTS 的定義順序來描述波形。上述 PGV 的檔案中實際的資料為 0h,1h,2h,3h.....4Fh, 上述為十六進制值, 如果以十進制表示, 則 0,1,2,3.....15。

PS:在第 25 筆的註解中的 JP 10 是表示在輸出第 25 筆的資料值之後, 在跳回第十筆來重複的輸出資料。

我們使用 PG EDIT  來觀察上述計數器的文字向量檔所產生的結果, 使用步

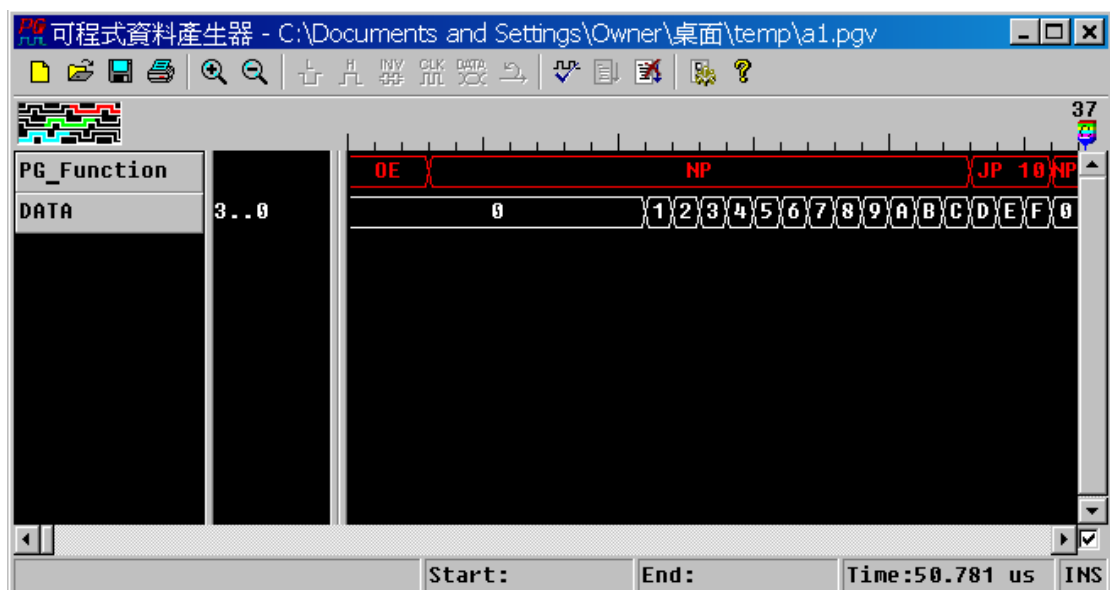
驟如下:1. 執行 PG 程式 



2. 執行文字(向量)檔轉成波形該程式  介面如下, 並且選取編輯好的文字向量檔並載入該檔案



然後再按下『波形轉換』會出現以下畫面, 該畫面為 PG EDIT 的程式介面:



上述為文字向量檔 PGV 格式中的其中一種, 主要的分別為以上的 PATTERN 的波形資料部分使採用 FREQUENCY 的例子, 您可以看到 PGV 檔檔頭的部分:

```
FREQUENCY 1000 Hz;    %INTERVAL 1ms;%
```

而另一種格式為上述 PATTERN 說明中關於 Time Stamp 的部分, 底下為使用 Time Stamp 的例子:

```
INPUTS PG_Function DATA;  
  
ASSIGN DATA 3..0;  
  
RADIX AUTO;  
  
UNIT    ms;  
  
PATTERN  
  
0.0> 8FFh    0h  
1.0> 2FFh    0h  
2.0> 900h    0h  
10.0>000h    0h  
11.0>000h    1h  
12.0>000h    2h  
13.0>000h    3h  
14.0>000h    4h  
15.0>000h    5h  
16.0>000h    6h  
17.0>000h    7h  
18.0>000h    8h
```

以上為 Time stamp 格式的簡單範例, 您可以看到與 FREQUENCY 的不同之處在於檔頭與 PATTERN 內容開頭的部分:

UNIT ms

1.0> 1h

如上所說明的, UNIT ms 是指定資料的時間單位 ms(毫秒), 而 1.0> 1h 是表示在 1.0ms 這個時間點有 PG\_Function 指令的改變或是資料的改變, 在 Time Stamp 的例子中都是記錄狀態有改變的時間點。『>』是區隔時間部分以及波形資料部分。而關於每筆資料採用的時間單位為 1.0> 主要是要配合所設定的計數器頻率 1KHz, 您可以注意到只要是 PG\_Function 的指令或是 DATA 改變, Time Stamp 才會加 1 ms, 例如說時間 3.0>~9.0> ms 之間並沒有任何的 PG\_Function 或是 DATA 改變, 所以沒有記錄在檔案內, 您可以跟使用 FREQUENCY 的 PGV 檔案內容做比較。

註 1: 8FFh、2FFh、900h、816h、200h、100h 為 PG\_Function 指令, 該指令的功能都有在該行末的註解說明, 關於 PG\_Function 指令的完整說明請參考底下註 2

註 2: 關於 PG\_Function 指令的說明:

表一

指令	指令全名	說明	週期
NP	No Operation	沒有動作	1
JP	Jump	跳躍到設定點繼續執行	3
LP	Loop	將迴圈數減一, 如果迴圈數不為 0 就跳躍到設定點, 如果為 0 就跳到下一個位置。	3

BE	Branch if Event	如果以設定的 Event 訊號出現就跳躍到設定點，否則就會跳躍到下一個位置。	3
LC	Loop Count	設定迴圈數，設定值為 2~65536。	2
SE	Set Event	選擇觸發的事件。	1
WE	Wait Event	等待事件被啟動。	1

PG 內部有下列幾個暫存器：RT, REX, RC, ROE。使用時都是透過

PG\_FUNCTION 的指令控制。而 PG\_FUNCTION 是一個 12Bits 的指令集。

PG_Function (12Bits)			
4Bits (MSB)	8Bits (LSB)		
8	XX	MOV RL,XX	將 PG_Function 的 LSB 8Bits 值移入 RL 暫存器中。
2	XX	MOV RH,XX	將 PG_Function 的 LSB 8Bits 值移入 RH 暫存器中。
1	XX	JP RT	跳至 RT-12 的新位址。

RT 有 16 個 Bits，可拆成兩個 8 Bits 的暫存器來用分別是 RL 及 RH。RT 的用途主要是用來傳遞資料用的。填入 RT 值的方法是藉由對 RL 以及 RH 這兩個 8Bits 的暫存器填值，例如：

```
MOV      RL      16h          //對暫存器 RL 填入 16h
MOV      RH      00h          //對暫存器 RH 填入 0h
會等同於: MOV      RT      016h
```

PG\_Function 的指令主要是告知 PG 內部的暫存器指標做什麼對應的動作，然後由記憶體取出資料傳送出去。以下就 PG\_Function 內部的指令說明如下：

**NP(No Operation):**在波形執行時，並不會有任何控制流程的改變或是任何內部暫存器改變。用 CPU 的觀點來看，那 NP 指令的動作就是 New Address = Address + 1，也就是指標移到下一個位置，其他不做任何改變。

**JP(Jump):**是一個改變控制流程的指令。例如『JP 35』，這個指令的動作就是無條件跳至新位址 35(相當於 New Address=35)。而在文字向量檔裡(\*.PGV)該如何使用這個指令，以下皆使用前述 1KHz 的計數器的 PGV 檔案內容做說明：

```
000h Bh // 00021:
000h Ch // 00022:
816h Dh // 00023:      (MOV RL, 22)
200h Eh // 00024:      (MOV RH, 0)
100h Fh // 00025:      JP 10
000h 0h // 00026:
;
```

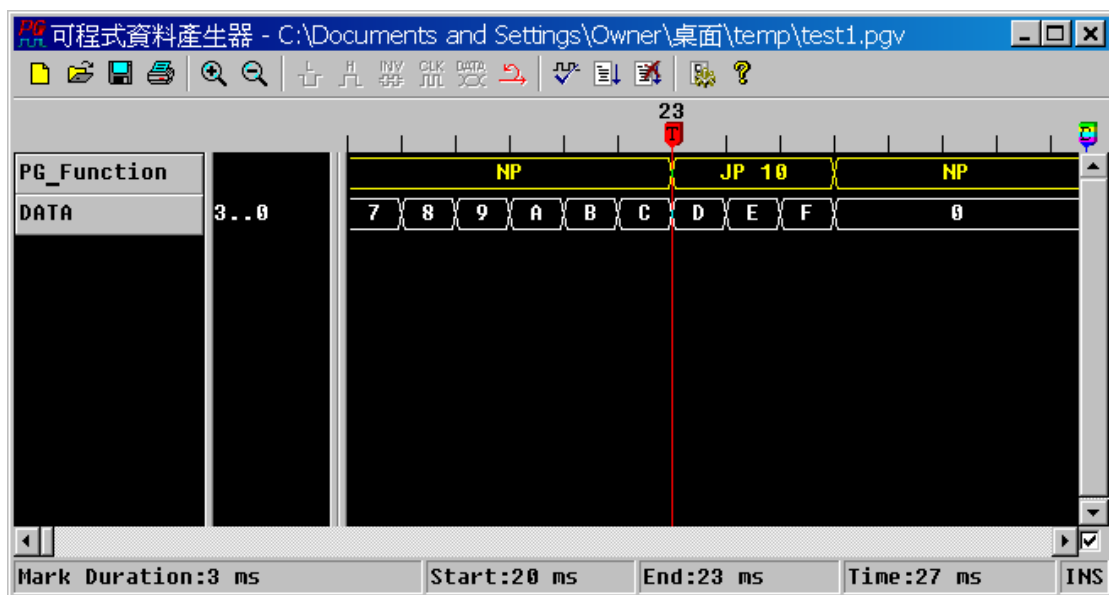
以上三行紅色字體的十六進制的 PG\_Function 的指令，816h:對

暫存器 RL 填入數值 22(MOV RL, 22), 200h: 對暫存器 RH 填入數值 0

(MOV RH, 0), 100h: 跳至 RT-12 的新位址(22 - 12 = 10), 也就是說送出第 25 點資

料後, 跳躍到第十點後再繼續傳送資料。以下是透過 PG EDIT  來觀察文字向

量檔(\*.PGV)內的資料與 PG\_Function 的指令:



**LP(Loop):** LP 和 JP 指令很類似，其最大差異就是 JP 是一個無條件跳躍指令，而 LP 是一個有條件跳躍，它必須與『LC』指令配合使用。PG 有一個內部暫存器稱為迴圈暫存器(Counter)，這個暫存器就是用來記錄迴圈數目的暫存器。這個暫存器是用 LC(Loop Count)指令來設定，例如『LC 32』就是將迴圈暫存器設定成 32。迴圈暫存器可以設定的值為 2~65536，所以不能設定成 0 與 1，這種設定方式與一般的 CPU 或 MCU 的用法有些許的差異，需注意。設定好迴圈以後就可以使用 LP 指令了，而每次執行 LP 指令，迴圈暫存器的值就會減 1，而跳躍的新指標就會參考 LP 指令的設定值來決定。

例如指令『LC 32』、『LP 16』，的整個動作步驟如下：



- 一、將迴圈暫存器 32 減 1。
- 二、檢查迴圈暫存器是否為 0。
- 三、如果迴圈暫存器為 0，New Address = Next Address。
- 四、如果迴圈不為 0，New Address = 16。

```

200h 0h // 00007: (MOV RH, 0)
401h 0h // 00008: LC 3
000h 0h // 00009:
000h 0h // 00010:

=====

820h Dh // 00033: (MOV RL, 32)
200h Eh // 00034: (MOV RH, 0)
300h Fh // 00035: LP 20
    
```


PG_Function (12Bits)			
4Bits(MSB)	8Bits(LSB)		
3	XX	LP RT	跳至 RT-12 的新位址。

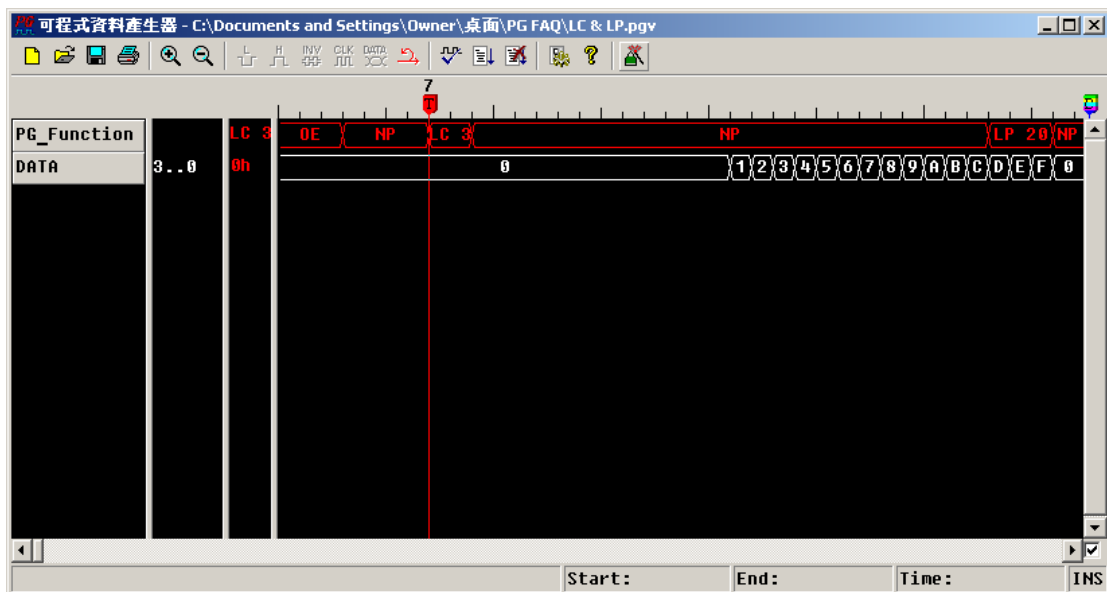
先看到前三行紅色字體的十六進制的 PG\_Function 的指令, 820h: 對暫存器 RL 填入數值 32(MOV RL, 32), 200h: 對暫存器 RH 填入數值 0 (MOV RH, 0), 300h: 跳至 RT-12 的新位址( 32 - 12 = 20 )做迴圈的動作。

PG_Function (12Bits)			
4Bits(MSB)	8Bits(LSB)		
4	XX	LC RC	設定 RC 暫存器的迴圈數

接下來看到後兩行的部分: 200h: 對暫存器 RH 填入數值 0(MOV RH, 0), 401h:對迴圈暫存器 RC 填入數值 3(1 + 2 = 3,因為迴圈數從 2 開始),也就是說暫存器 RC 裡的值為 1,因為 RC 也是 16Bits 的暫存器,所以這兩行所代表的意義為 :

RC	00000000 (00h)	00000001(01h)
----	----------------	---------------

以下為透過 PG EDIT  來觀察文字向量檔(\*.PGV)內的資料與 PG\_Function 的指令:



有一點要注意的是如果當迴圈暫存器已經被減成 0，又遇上 LP 指令時，這時候迴圈暫存器是不會再被減成-1，而是會產生特殊情況。所以迴圈暫存器被減成 0 以後，LP 指令的工作是不可預期的，所以要使用 LP 指令之前請注意是否已經設定 LC 指令了。

**SE(Set Event)**:是一個選擇事件的指令,PG 有 3 個外部事件通道(Event\_1、Event\_2、Event\_3)及 1 個內部鍵盤事件(Keyboard Event),所以總共有 4 個事件來源。PG 將這 4 個事件來源編輯成 16 種事件型態以茲利用,而這個事件型態會被存到一個 PG 的內部暫存器,稱之為事件暫存器(Event Register)。這些事件型態包括:

1. Keyboard Event
2. Event\_1
3. Event\_2
4. Event\_3
5. Event\_1 or Event\_2
6. Event\_1 or Event\_3
7. Event\_2 or Event\_3
8. Event\_1 or Event\_2 or Event\_3

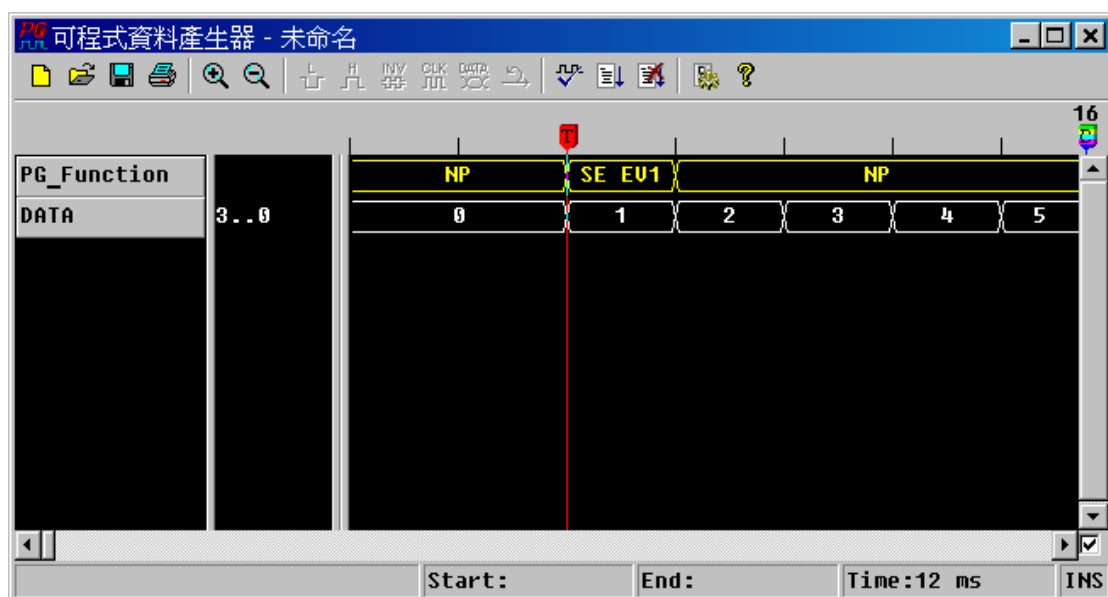
另外 8 種事件型態就是這 8 種型態的反相。例如事件暫存器的設定值為上述 8 種型態之一,此時不管從外部或是內部的事件通道都會被 PG 所監控,PG 會隨時比較目前的事件通道的狀態是否與事件暫存器的事件型態相同。如果相同就會將 PG 內部的旗標暫存器(Flag Register)的 Event Bit 設成 1(True),如果不同就會將 Event Bit 設成 0(False)。然而反相型態就是當事件通道的狀態與事件暫存器的事件型態相同時,PG 會將 Event Bit 設成 0,不同時則會將 Event Bit 設成 1。

```
000h 0h // 00010:  
609h 1h // 00011: SE EV1  
000h 2h // 00012:
```

PG_Function (12Bits)			
4Bits(MSB)	8Bits(LSB)		
6	XX	SE EV	設定 REX 暫存器事件值

8Bits 的 LSB 中:00h 代表內部鍵盤事件(Keyboard Event)、01h:代表外部事件通道 1 的反相型態(!EV1)、02h: 代表外部事件通道 2 的反相(!EV2)型態、03h: 代表外部事件通道 1 的反相型態與外部事件通道 2 反相型態的交集(!EV1 & !EV2)、04h: 代表外部事件通道 3 的反相型態(!EV3)、05h:代表外部事件通道 1 的反相與外部事件通道 3 的反相的交集(!EV1 & !EV3)、06h:代表外部事件通道 2 的反相與外部事件通道 3 的反相交集(!EV2 & !EV3)、07h:代表設定外部事件通道 1 的反相與外部事件通道 2 的反相與外部事件通道 3 的反相的交集(!EV1 & !EV2 & !EV3)、08h: 代表內部鍵盤事件的反相(!KEY)、09h:代表外部事件 1(EV1)、0Ah:代表外部事件 2(EV2)、0Bh:表示 Event\_1 or Event\_2,也就是(!EV1 & !EV2).....其他的依此類推,總共會有 16 種不同的事件。

只要在 8Bits 的 LSB 中輸入想要的情況即可。以下為上述例子之圖示:



而運用 Event Bit 的指令有兩個，一是 WE(Wait Event)另一是 BE(Branch If Event)。

**WE(Wait Event):**指令是當波形執行到這個指令時，整個 PG 會因此而暫停，而波形會停留在 WE 指令當時的波形，一直到 Event Bit 為 1 時，才會再繼續往下執行。

```
000h 0h // 00010:
700h 1h // 00011: WE
000h 2h // 00012:
```

PG_Function (12Bits)			
4Bits(MSB)	8Bits(LSB)		
7	XX	WE	PG 暫停,等待事件發生



**BE(Branch If Event):**指令和 LP 指令又有點類似，因為他們都是一個有條件跳躍的指令。LP 的跳躍條件是迴圈暫存器，而 BE 的跳躍條件是 Event Bit。當波形執行

到 BE 指令時，PG 會立即判斷 Event Bit，如果 Event Bit 為 1 就會跳到 BE 所設定的新位址，Event Bit 為 0 的話，則繼續執行下一個位址。

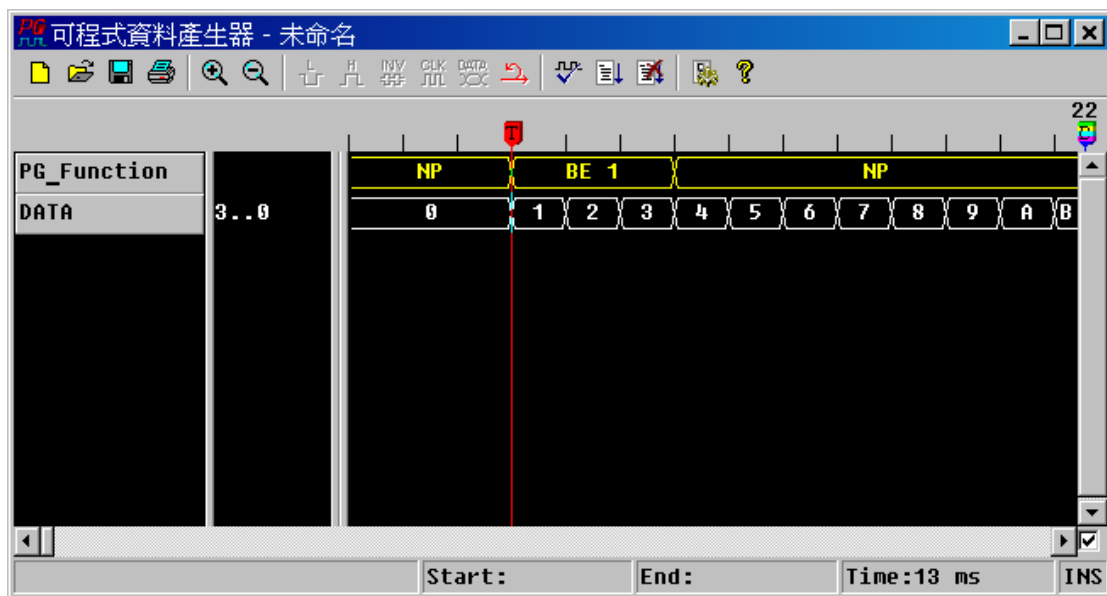
```

000h 0h // 00010:
80Dh 1h // 00011: (MOV RL, 13)
200h 2h // 00012: (MOV RH, 0 )
500h 3h // 00013: BE 13
000h 4h // 00014:
    
```

PG_Function (12Bits)			
4Bits(MSB)	8Bits(LSB)		
5	XX	BE	跳至 REX 的新位址。

而 80Dh: 對暫存器 RL 填入數值 13(MOV RL, 13), 200h: 對暫存器 RH 填入數值 0(MOV RH, 0),500h: 跳至 REX 的新位址。

以下為上述例子之圖示:



在 PKPG 系列中,PG\_Function 的指令還有一個 Output Enable 『OE』

,說明如下:

```

8FFh 0h // 00000: (MOV RL, 255)
2FFh 0h // 00001: (MOV RH, 255)
900h 0h // 00002: OE 65535
000h 0h // 00003:
000h 0h // 00004:
    
```

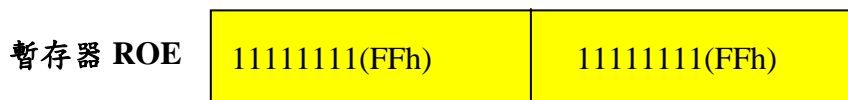
PG_Function (12Bits)			
4Bits(MSB)	8Bits(LSB)		
9	XX	OE	將通道輸出致能

8FFh: 對暫存器 RL 填入數值 255(MOV RL, 255),2FFh:對暫存器 RH

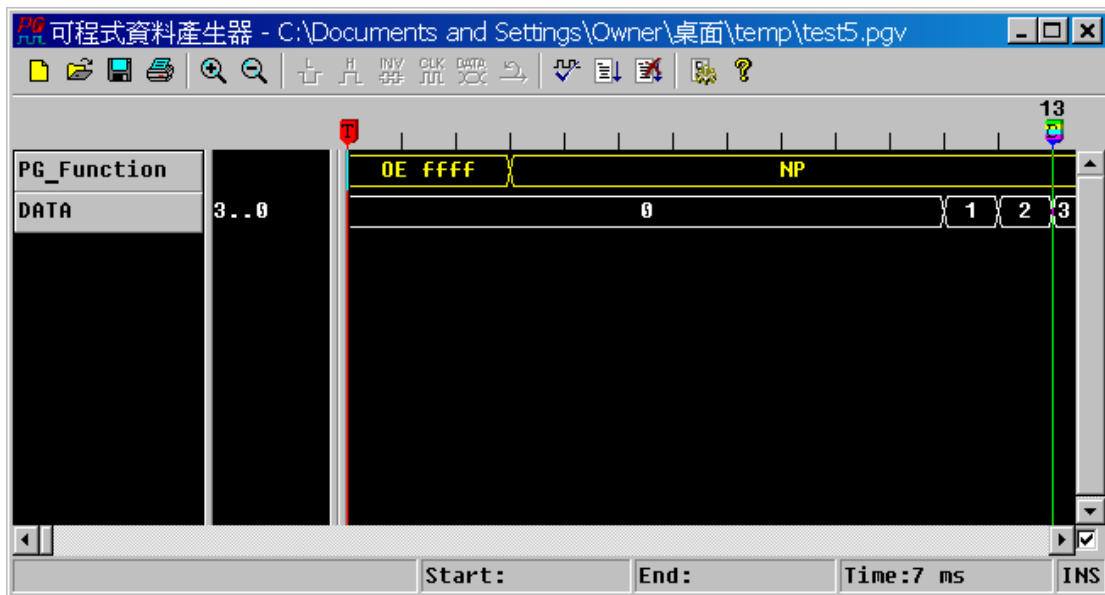
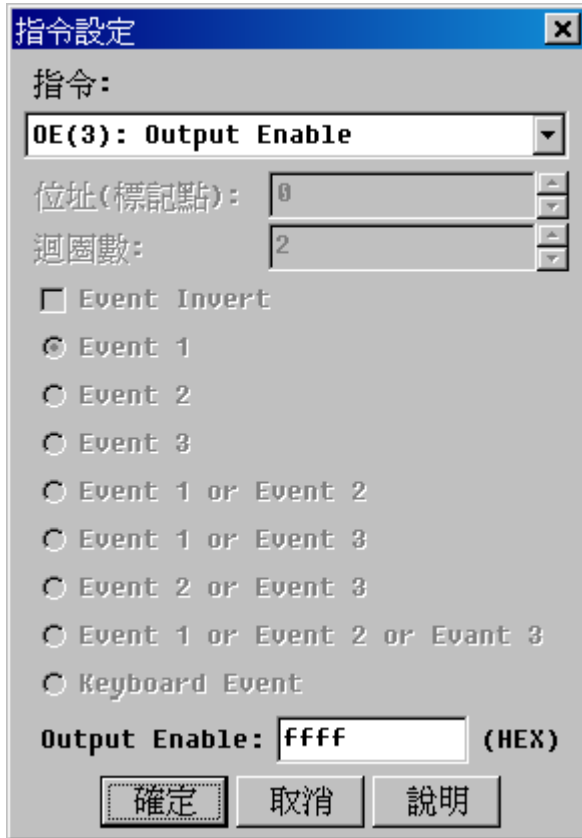
填入數值 255(MOV RH, 255), 900h: 輸出致能。

暫存器 ROE 就如同暫存器 RT 一樣,都是由 8Bits 的 RL 與 RH 組成,所以整個 ROE

暫存器也可以看成如下所示:



底下為圖示:





**PC-Based Pattern Generator Manual**

**Copyright © 2012 Acute Technology Inc. All Rights Reserved.**

資料產生器使用手冊 ©2012 皇晶科技・版權所有



皇晶科技股份有限公司

[www.acute.com.tw](http://www.acute.com.tw)

地址：新北市三重區 24159 重新路五段 609 巷 12 號 6 樓之 7 (湯城園區)

連絡電話：(02)2999-3275

傳真：(02)2999-3276

E-mail: [service@acute.com.tw](mailto:service@acute.com.tw)