一、概說

VP-894 爲一 8bit 語音通信介面卡，可使用於 PC 386/486/586 以上機種，具有一卡四線 (或一卡兩線，稱 VP-894J) 之多工電話通信能力，並可多卡共用。如裝入工業用 PC 內，最多可同時操作 64 線。VP-894 語音介面卡具有多項一般語音卡無法比擬之特點。首先，它具有將 PCM 數位語音資料以 1:4 或 1:8 壓縮後再進行傳輸的能力，使資料儲存時能節省電腦記憶裝置 (如硬碟) 的大量空間，在網路傳輸上也更能發揮效率。其次，VP-894 採用 8088CPU 爲主處理器配備 256K 的 Voice Buffer，不但加速了語音卡與 PC 間數位資料的存取時間，也強化了系統的擴充性。由於實際進行擴充時所有卡片均使用同一 IRQ，因此無須擔憂因 IRQ 過多而產生與其它介面卡衝突的問題。

VP-894 在語音錄製時，先以內部之 PCM CODEC 進行類比 / 數位資料轉換，將語音轉成 64Kbps 之數位資料，再經特殊設計之數位訊號處理器(DSP)將 64Kbps 之數位語音資料以 1:4 或 1:8 比例進行壓縮，因而最終速率成爲 16Kbps 或 8Kbps (如加上靜音壓縮可降低爲 9.8Kbps 或 4.9Kbps)。進行語音播放時，則將上述過程反轉。由於它包含了電話週邊線路，故除了語音錄放外，也具備振鈴偵測、DTMF 偵測及自動撥號 (含 CPM 偵測，註) 以及較爲特殊的內部轉接等功能。在軟体支援上，本公司提供了用戶程式介面-DEVICE DRIVER 及 API(詳見本手冊中支援各種高階語言的相關章節)，使用戶在程式寫作時，可利用一部 PC 同時操作多片 VP-894 卡，並輕鬆完成 2～64 線的各種語音查詢、語音信箱、電話拜訪、民意調查、電話轉接等用途。如欲增加傳真功能，亦可搭配使用本公司之傳真卡以達成。

1-1、 軟體簡介

VP-894 的 API function 可以概分爲兩類：一類爲不產生事件(event)的 function 組合；另一類則爲事件驅(event-driven)，亦可稱爲 multi-tasking 的 function 組合。在傳統的 programming 習慣觀念上，當執行控制權由被呼叫的 function return 至呼叫者時，即代表該被呼叫的 function 已結束並完成工作。但是在 VP-894 內各個 API function 完成工作所需使用的時間長短不一，例如播放語音的 Play() function 甚至可以指定播放長達數小時的語音檔案，而且早期的 PC 作業系統(DOS)沒有提供類似即時多工作業系統 multi-thread 的功能，因此我們無法完全採取傳統的 programming 習慣來 implement 所有 VP-894 API function，相反的我們須將比較耗時的函數區隔出來，使用另一種方式來處理。這些函數即爲前面所說的第二類事件驅動的 Function (event- driven function) 。application 寫作者在引用 event-driven function 時，必須特別注意執行該功能的 VP-894 channel 是否處於 idle 狀態；一旦 event-driven function 被成功呼叫，執行該功能的 VP-894 channel 即由 idle 進入 busy 狀態，一直要到該功能執行完成，或因爲其他狀況，造成該工作終止，此 VP-894 channel 才會再由 busy 恢復成 idle 狀態。而在 VP-894 處於 busy 狀態的期間，除 StopCh()外，VP-894 不再接受其它 event-driven function 的呼叫 (ERR894_CHANNEL_BUSY 的結果碼會被 device driver 傳回)，但不產生事件的函數則可在任何時刻呼叫。
event-driven function 在執行結束時，VP-894 會送出代表執行結果的事件通知 application。

VP-894 的 device driver 會在其內部 maintain 一個 event queue，將 VP-894 各個 channel 送出的 event，按先後順序排列，因 application 的 flow-control 中必須有一個 LOOP 不斷呼叫 GetEvent()函數，向 device driver 查詢是否有新的事件正等待被處理。
Application programmer 所必須切記的一點就是執行控制權由 event-driven function 返回時，傳回成功碼僅是表示 VP-894 接受該函數的呼叫，且該功能正在進行中，至於函數的執行結果，則可由收到的 event 類別中獲知。

VP-894 API 目前支援 DOS 及 Windows 兩種作業系統。在支援 DOS 的版本提供了 C 及 Clipper 兩種程式語言介面；在 C 語言方面，您必須使用 Microsoft C 7.00 或以上版本，或者 Borland C++2.00 或以上版本；Clipper 的 Programmer 則必須使用 5.01 或以上版本。在 Windows 方面我們提供了兩種 solution－VP894.DLL 及 VP894CC.VBX。前者是一個 Windows 的 dynamic link library，因

此基本上任何可以呼叫 DLL export funtion 的程式語言應該皆可使用 VP894.DLL，不過因爲 VP-894 所產生的 event 是透過 Windows 的 messaging 系統來傳送，所以您還必須確定您所使用的程式語言是否能讓你的 application 收到自 Windows 的 message。
VP894CC.VBX 則是一個標準的 custom control，雖然許多程式語言皆宣稱支援 VBX，不過多半僅符合 Visual Basic Custom Control 1.00 specification，而 VP894CC.VBX 採用 2.00 specification，所以在使用 VP894CC.VBX 之前，請確定您所使用的程式語言能否支援 2.00 的 specification。雖然 VP894.DLL 及 VP894CC.VBX 目前還是 16bits 的軟體，但仍能在 Windows 95 上執行，本公司預定在未來會將此兩項軟體更新爲 32bits 的 DLL 及 OCX。

註：CPM 爲 CALL PROGRESS MONITOR 簡稱。VP-894 可利用所測得能量(energy)數據，來分析撥號結果爲忙線(Busy)或無人接聽(No answer)或回答(Answered)等不同狀態。

1-2、 硬體簡介

請見附圖一之 VP-894 硬體方塊結構。本卡除採用 8088 做爲主信號處理器外，另外配有 32K SRAM，主機(PC)上的 CPU 與 VP-894 上的 8088 即是透過這一塊兩邊可以 Access 的 memory 來交換訊息及令。當任何一塊 VP-894 需要與 PC 交換資料時，會送出 IRQ，並等待 PC 經由 300H 送來的掃瞄信號。 換言之，I/O port 300H 是啓動任一片 894 卡 32K SRAM，及隨後 I/O 301H，302H gate 的先決控制信號。因此送卡號至 300H 的位置，而 PC 能得到相對卡號所送來之 IRQ 信號時，相對卡號 894 adapter 的 gate 即被打開。例如送卡號 4 至 300H，接下來的 memory R/W 或 I/O 301H，302H，皆是針對卡號 4。

由於 32K shared buffer 不能同時被 PC 和 VP-894 所選擇，
I/O 301H、302H，被利用來控制 894 方塊圖內的
I/O control circuit 及 bus buffer，以決定 PC 或 894 local address，data 和 R/W 信號那一方可連接至 share buffer 上。每一片 894 卡可接 4 line，每一 line 有各自獨立的 A/D，line interface circuit 及 answering machine processor 處理 voice，DTMF decode，和複頻 dialing，另外每一 line 還有 48K DRAM 作爲錄放音的 buffer，實際工作時這 48k 的 Buffer 將分割爲兩部分，以 PING-PONG 方式進行資料的存取動作。因此如果某一線是在 16K bps 的錄音模式工作中，則每 12 秒 894 adapter 上的 CPU 即會透過 32K shared buffer 要求 PC 上的 device driver 將這 24K 的語音資料寫入檔案中。另外手冊最後的附錄 C 則是 VP-894 對 Busy，Ringback 和 Answered 狀態之能量偵測所得數據，每一數值代表 8ms 的 energy 值，因此一列是 200ms。 測試環境是 record gain 等於 0dB，off threshold 等於 -50 dBmO。(指在 PCM codec 輸入點之測量值。 1dBmO=+4dBm)

圖一：VP-894 系統方塊圖

1-3 、新版 Device Driver 4.00

VP-894 device driver 在 4.00 版增加支援 VP-894 的兩線版本，對於原有已利用 VP-894 API 所發展的 application，如果確定將來不會在您的系統內安裝 VP-894 的兩線卡版本，則可以不做任何修改，直接昇級至 4.00 版。如果您要共用 VP-894 的兩線版本，則有可能要對部份程式作必要的修改。 基本上所有的 API function 除 Init894() 外 **(**VP-894.DLL 尚包含 Query894()，Get894Owner()**)**，均與 4.00 以前的版本相同。在 4.00 以前的版本中 device driver 會在 application 呼叫 Init894()時所提供的一個 16 個元素的 array 內分別填入 0 或非 0 值表示系統內 VP-894 的安裝組態，然而在 4.00 及未來的版本中，此 array 內的填入值除了代表各卡號 VP-894 是否安裝外，各元素的值的 bitmap 亦代表該卡號 VP-894 上各個 channel 的組態，0 表示 absent，1 代表 exist，由 LSB bit 0 →bit 3 分別代表卡上的 channel 0～ channel 3。若由軟體最大的相容性上來考量，不論您在您的系統內是否僅使用四線或兩線的 VP-894，最好以 Init894()所傳回 array 內值的 bitmap 做爲判斷 channel 是否 exist 的依據，請參考 API 磁片中各個 demo 程式的實作部份 **(**因爲 Clipper 語言缺乏位元運算子，在 UTILITY.PRG 內已預先定義一個 function IsChExist()用來判斷任一 VP-894 channel 是 Valid**)**。

1-4、 電氣特性規格

電 氣 特 性	規　　格
On-board CPU	Intel 8088
On-board Buffer	256Kbyte DRAM
Shared Buffer	32Kbyte SRAM
Voice Digitization	μ-Law PCM with compression
Voice Data Rate	4.9K-16Kbits/sec
Voice DSP	DALLAS 2132A
Bus Structure	≧8Mhz,8 bits
Interrupt Level	one for all ports
I/O Address	fixed
Max.Boards/System	16
Phone Interface	analog
Start Method	loop start
Connector	4×RJ11
Impedance	600Ω/900Ω
Frequency Response	300-3.4KHZ @16Kbps
Ring Detection	40-130Vms,15-68Hz
Loop Current	20-120ma
Local Phone Leads	Equipped
Outbound Dialing	DTMF＆Pulse
Inbound Reception	DTMF＆Pulse
Crosstalk	＜50dB
CPM Detection	Yes
Internal Bridging	Adjacent Channels
Power Requirement	±5Vdc
Operating Temperature	0℃to +50℃
Storage Temperature	-10℃to +60℃
Humidty	10％to 80％non-condensing
Certifications	CE,FCC＆BZT

1-5、VP-894 在 Computer Telephony 上的應用

1-5-1、 Computer Telephony 系統之分類

Computer Telephony(簡稱 CT)是近幾年來逐漸演變而成的一個專有名稱，涵蓋了幾乎所有與 電腦語音與電話結合的應用，也已含了傳真功能的整合，未來並有逐漸往 ISDN 與 INTERNET 發展的趨勢。現在僅就目前在市場上應用較爲普及的系統加以介紹：

1.互動式語音查詢系統

電腦語音系統在美國已有 20 餘年之歷史，其應用五花八門、不勝枚舉，但大多數均不脫互動式語音查詢(INTER-ACTIVE VOICE RESPONSE 簡稱 IVR)之特性。簡而言之，語音系統須事先錄製並儲存各種語音檔，並藉電 話線與外界連通。使用者在進行資料查詢時，須透過電話按鍵輸入所要知道的訊息代碼。電腦除根據系統流程適時播放語音提示 (VOICE PROMPT)外，並應能立即辨認所收到的 DTMF 碼(即按鍵信號)並以語音實際回應使用者所欲知道的信息。一般訊息查詢系統均屬此類，此種系統必要時亦可設計成爲電話答錄用途，因而成爲具備“語音信箱”功能之系統。

2.主動外撥系統

除了被動式的電話撥入式查詢應答外，電腦語音系統亦可以主動進行對外撥號，所撥之電話號碼可以事先設定或經由資料庫軟體控制撥號號碼及順序。此種語音系統須有精確偵測撥號成功與否之能力，並在電話接通後，播放一段通知訊息，或依題庫進行選項式樹狀問答，一般自動催費、民意調查系統，均屬此類。

3.電腦轉接系統

語音系統的另一應用是進行電話轉接服務，使不同電話線路能透過語音卡彼此通話。此種系統最重要的是電話連接時，不影響原有通話音量。同時在通話完畢，任何一方掛斷電話後，系統能加以判斷並進行拆線動作。市面的國際電話回撥與電腦交友系統均屬此類。

4.電腦數位多軌錄音系統

由於電腦科技的日新月異，不斷帶動其周邊產業的發展,而進步最明顯的莫過於電腦記憶能力提高與價格下降。不論是內接式電腦硬碟機、或外接式數位磁帶機(DAT)及可寫式光碟機(MO DISK)，容量均不斷增加，性能愈趨完善，因而語音卡與上述裝置結合成爲多軌數位錄音系統自是水到渠成。此類產品在不久將來勢將取代傳統盤帶式錄音系統，成爲市場主流。

1-5-2、 VP-894 在 Computer Telephony 上的應用

除了無法與 ISDN 或數位式電話系統連接外，幾乎所有之 Computer Telephony 應用均能採用 VP-894 爲其核心硬件。比較市面上其他廠牌之語音卡，或是限於擴充性不足，或是線路偵測能力欠佳，或是無法執行內部轉接，或缺乏語音壓縮能力，均相對突顯出 VP-894 的系統應用能力。茲分別說明如下：

1. VP-894 使用於互動式語音查詢系統

此爲最爲普遍之語音系統應用。VP-894 以其完善之硬體設計及軟體支援 **(**包含 Inbound Pulse Dialing 之偵側**)**能力，無論在 DOS 或 Windows 環境上進行程式開發，均可順利完成可擴充至 64 線之互動式語音查詢或語音信箱系統。市面所有廠牌的語音卡均爲了此類系統應用而設計。但比較重點在 1.線路擴充性 2.遠方掛斷之偵測能力 3.API 功能是否齊全。VP-894 均符合此一方面之要求。

2. VP-894 使用於主動外撥系統

此種系統於最近數年逐漸流行而成爲主要應用之一。VP-894 以其對撥號過程 **(**CALL ROGRESS MONITOR 簡稱 CPM**)**之精確判斷能力而成爲軟體開發人員在設計此類系統之優先考慮對象。雖然大多數的語音卡均號稱具有 CPM 偵測能力，但由於世界各國線路信號標準不一，因而產品對不同地區的線路適應能力便成爲此類系統的考慮重點之一。VP-894 對此提供了線路忙音、回鈴音的多種參數選擇，因而大幅提高了對不同地區之適應能力。

3. VP-894 使用於電腦轉接系統

VP-894 內部具有數位迴路**(**Digital LoopBack **)**設計，使經過 A/D 轉換的 PCM 輸入信號能轉 接至隔壁波道的 Codec 輸入點，因而不但可成功進行電話轉接，且在通話時能維持原有音量，達到信號不衰減的目的。在電腦轉接系統應用愈來愈普遍之今日，此種轉接功能已成爲市場新趨勢。

4. VP-894 應用於多軌電腦錄音系統

VP-894 的下列數項功能使其在應用於多軌電腦錄音時，明顯優於其他語音卡: 1.較高的語音壓縮能力 2.本地電話的連接能力及 3.錄音時的能量數據提報。本公司利用 VP-894 卡之上述優點設計之 VLR-100 多軌電腦數位錄音系統，已於 1995 年 12 月完成，1996 年 2 月起正式開始行銷。

二、硬體安裝說明

VP-894 上有 32k bytes 連續的記憶體，用來與 PC 上之驅動程式交換訊息資料，此 32K 記憶體的位置可設定 A000，A800，C000，C800，D000，D800，E000 或 E800。以下爲 VP-894 卡上 dip switch (S1) 及 Jumper (W1～W12)之設定及跳線方式說明。(請參考圖二)

2-1、S1 設定方式

S1-1：用來連接共同 IRQ，不論一片或多片僅有一片必須設爲 on。

S1-2～S1-5：用來選擇卡號，編號爲 0～15，以 Binary 方式排列。舉例如下:

(bit3) S1-2	(bit2) S1-3	(bit1) S1-4	(bit0) S1-5	卡號	
0	1	0	0	**4**	1:on
0	0	1	1	**3**	0:off
1	0	0	1	**9**	

S1-6～S1-8：用來選擇 32K SRAM 的起始位置

S1-6	S1-7	S1-8	記憶體位置
0	0	0	**C000～C7FF**
0	0	1	**C800～CFFF**
0	1	0	**D000～D7FF**
0	1	1	**D800～DFFF**
1	0	0	**E000～E7FF**
1	0	1	**E800～EFFF**
1	1	0	**A000～A7FF**
1	1	1	**A800～AFFF**

圖二:VP-894 硬體設定及跳線位置表

2-2、 W1～W12 跳線說明

1.W5—W8 (供選擇 IRQ):

W5 ON 時，使用 IRQ3
W6 ON 時，使用 IRQ5
W7 ON 時，使用 IRQ7
W8 ON 時，使用 IRQ9

註:多片使用時必須共用一個 IRQ

2.W1－W4 及 W9－W20:

爲選擇 VP-894 與電話外線及本地電話 (Local Phone) 之連接方式。若 VP-894 僅連接電話外線，可適用於一般語音查詢用途。若 VP-894 分別與 Local Phone 及電話外線相連則通常用於電話錄音 / 監聽用途。(請參考圖三)

跳線方式如下：

① 用於一般語音查詢用途時 (EVT_DETECT_RING 可正常使用)

W3，W13 對應 LINE 1
W1，W15 對應 LINE 2
W4，W17 對應 LINE 3
W2，W19 對應 LINE 4

② 連接 LocalPhone 時 (EVT_DETECT_RING 無法使用)：

W11 ，W14 對應 LINE 1
W 9 ，W16 對應 LINE 2
W12 ，W18 對應 LINE 3
W10 ，W20 對應 LINE 4

圖三:本地話機(LOCAL PHONE)連接圖示

2-3、 I/O Port 設定

VP-894 的 I/O port 固定於 300H～302H 不可調整或更動。如有特殊需求，請與本公司連絡。

2-4 、J1 說明

用於提供 L1，L2，L3，L4 的通路訊號，可經由扁平排線連接 EX24 或 EX-2424 語音交換卡來完成某些特殊應用。

三、軟體發展介面程式(API)大要

3-1、軟體事件(EVENT)定義

API functions 依其使用時機及特性，可分爲下列兩組：

1.不產生 Event 的 functions(Non Event-Driven Type)

Init894()，Close894()
GetCtrlParam()，GetEvent()
FlushEvent()，GetEnergy()
InsertEvent()，GetCPMParam()
SetCPMParam()，GetHungUpParam()
SetHungUpParam()

2.產生 Event 的 functions(Event-Driven Type)

SetCtrlParam()，PickUp()
HangUp()，Flash()
Play()，Record()
GetDTMF()，FlushDTMF()
Dial()，StopCh()
CallLocal()，CallRemote()
CallBeeper()

各個 function 的使用方式及傳遞參數的詳細說明請參閱各 API 章節的內容。以下對在 API894.H 檔案內定義的所有 AP 在執行過程中可能 receive 的 event type 常數名稱分別說明其意義:

1.EVT_EOP_NORMAL
指示先前啓動的工作已正常執行完成。

2.EVT_DTMF_INTERCEPT
指示先前啓動的工作被 remote 輸入之 DTMF 中斷。

3.EVT_TIME_OUT
先前啓動的 GetDTMF()超過設定最大電話按鍵間歇等待時間。工作終止。

4.EVT_ENDOF_STOP

先前啓動的 StopCh()工作已完成。

5.EVT_INTRN_QUEUE_OVERFLOW

指示 894 系統內部 **(**firmware or driver**)** queue overflow。可能造成的原因有: 1.未正確即時處理所有的 event 2.PC 執行速度不足以支援目前 install 的 channel 數量。當有本狀況發生時 894 系統自動重新初始化所有 channel 終止目前工作，回到 idle 狀態。

6.EVT_DETECT_RING (註)

指示偵測到 ring。

7.EVT_LOCAL_PHONE_PICKED_UP (註)

指示偵測到並接的電話被拿起。

8.EVT_LOCAL_PHONE_HUNG_UP (註)

指示偵測到並接的電話被掛上。

註： 使用 Local Phone 功能時，VP-894 無法偵測 EVT_DETECT_RING 但可偵測 EVT_LOCAL_PHONE_PICKED_UP 或 EVT_LOCAL_PHONE_HUNG_UP。前者功能與後二者功能無法並存，請參考第二章硬體跳線設定說明。

9.EVT_DETECT_DTMF

指示接收到一個有效的 DTMF 輸入。

10.EVT_REPORT_ENERGY

指示接收到一組信號能量紀錄。

11.EVT_PCMIO_ERROR

指示先前啓動的錄、放音工作於檔案輸出入發生錯誤。**(**如產生這樣的 event，該 channel 的工作並不終止如要中斷工作可呼叫 StopCh()**)**

12.EVT_NO_DIAL_TONE

指示先前的 call out 工作，因偵測不到 dial tone 而終止。

13.EVT_CPM_COMPLETE

指示先前的 call out 工作完成。如果此 call out 工作有指定要進行 CPM，則 CPM 的結果是 GetEvent()傳回的 struct typeEvent 中之資料成員 Data 內的值，可能爲下列任一結果。

① CPMR_NO_ANSWER

回鈴的持續時間超過 CtrlParam WaitAnswer-Duration 所設定的時間，表示受話方未接答。

② CPMR_BUSY

受話方正忙線中。

③ CPMR_INVALID_NUM

受話方號碼爲一無效的電話號碼。

④ CPMR_USER_DEFINED1，CPMR_USER_DEFINED2

偵測到使用者定義之信號 1 或信號 2，請參考第四章 API 程式說明中有關 GetCPMParam()或 SetCPMParam()部分。

⑤ CPMR_NO_SIGNAL

CPM 偵測超過 CtrlParam 中 NoSignalTimeOut 的設定時間且沒有偵測到任何訊號。

⑥ CPMR_ANSWER

受話方已接答。

⑦ CPMR_NO_RINGBACK

CPM 偵測超過 CtrlParam 中 WaitAnswer- Duration 所設定的時間，沒有偵測到回鈴音。

⑧ CPMR_CALL_BEEPER_SUCCESS (適用台灣地區)

執行 CallBeeper()成功。

以上各個 CPM 結果常數皆定義在 API894.H 中。

14.EVT_LINE_ROARING_REMOTE_HANG_UP

指示 VP-894 偵測到 Remote 掛斷後之 roaring tone 出現。

15.EVT_LINE_BUSY_REMOTE_HANG_UP

指示 VP-894 偵測到 remote 掛斷後之 busytone 出現。

16.EVT_LINE_VOLT_REVERSE_TOGGLE

指示 VP-894 在佔線之後,偵測到線路上有電壓極性反轉的現象發生。請注意並非所有的交換機系統都會提供這種訊號，application 如要偵測電壓極性反轉，必須確認 VP-894 所連接的通信主機確實有送出這種訊號；另外 VP-894 firmware 內定是將此種訊號視爲遠端掛斷的通知訊息，因此如果 application 設定控制參數啓動遠端掛斷偵測，而且控制參數中的 StopOperationRemote- HangUp 被設爲 TRUE，VP-894 在偵測到電壓極性反轉時，會自動中斷連線。另一種狀況是少數通信主機會將這種訊號當作遠端 answer 的通知訊號，送給執行 outbound call 的 calling party，此時 calling application 如要避免錯誤的偵測，必須在 call function **(**CallLocal()， CallRemote()**)** 完成後，才啓動掛斷偵測的功能或事前先 disable StopOperationRemote-HangUp，以防止 firmware 誤將通信連線切斷。*17.EVT_LINE_SILENT* 指示線路上持續的 silent 時間超過設定值。*18.EVT_DISPOSE_PCM_DATA* 當 VP-894 在多線同時錄音狀態下,因執行效率與 storage I/O 的速度有密切的關係，如果 storage I/O 的速度無法即時把每一個 channel 送出的頁框語音資料輸出到 storage 設備，就極易發生 event queue full 的情形，一旦發生這種狀況，隨後產生的 event 即有可能 lose，而影響系統正常運作。爲支援某些可犧牲頁框資料而維持系統持續運作的應用，當有 internal I/O queue overflow 的情形發生時，該頁框資料即自動丟棄，並產生一個 EVT_DISPOSE_PCM_ DATA 的事件通知 Application，application 可自行決定是否結束執行。

3-2、錯誤碼常數定義

894 API functions 中除了 Close894()及 Flush- Event()不傳回值外，其它的 function 均會傳回一整數值指示呼叫後之結果。在這些有傳回值 function 中除了 GetEvent()外，其餘 functions 在執行成功時均會傳回 0 值，否則就傳回以下所述的錯誤碼。在 API894.H 中所定義由 API functions 所傳回錯誤碼常數所代表的意義如下:

1.ERR894_INVALID_FUNC
本項功能未提供。

2.ERR894_INVALID_PARAM
傳遞參數不正確。

3.ERR894_INVALID_CHANNEL
指定的 channel 號碼無效。

4.ERR894_CHANNEL_BUSY
指定的 channel 正在 busy 的狀態中。

5.ERR894_NO_MORE_VOICE_DATA
呼叫 Play()時檔案指標已指到檔尾。

6.ERR894_EVENT_QUEUE_OVERFLOW
894 系統 event queue overflow。

7.ERR894_PCMIO_QUEUE_OVERFLOW
894 系統內部錄放音檔案輸出入 queue overflow。

8.ERR894_PCM_FILE_IO
錄放音檔案輸出入錯誤。

9.ERR894_DRIVER_NOT_INSTALLED
DRV894.EXE 尚未 install。

3-3、控制參數定義

在 API894.H 中定義了一個結構類別 typeCPB，這　個結構類別用的所有成員參數可以依照不同的組態　設定，控制各個 VP-894 channel API 功能的執行方　式，client application 可以透過 SetCtrlParam()函數來更改其預設值，進而達到 customize VP-894，使其依照 client application 的期望，完成 telecommunication 的工作。以下針對各項結構成員用途加以說明:

1.DialMode

設定撥號模式，0 爲 Tone dialing，1 爲 pulse daling。預設爲 Tone dialing。

2.LineToPBX

指定本 channel 電話線是接在分機上或局線。0 爲局線，1 爲分機。預設爲局線。

3.TriggerMode

指定本 channel 偵測振鈴信號或並接電話 Hook 狀態。0 爲偵測振鈴，1 爲偵測 Local Phone Hook 的狀態。預設爲偵測振鈴。**(**請參考 894 硬體設定中 jumper W1 －W4，W9－W20 之說明**)**

4.MonitorDTMF

指定本 channel 接收到一有效的 DTMF 資料，是否產生一 EVT_DETECT_DTMF event，1 爲是，0 爲否，預設爲否。

5.MonitorEnergy

指定本 channel 是否定時產生一 EVT_REPORT_ENERGY event，回報線上 Energy 變化。1 爲是， 0 爲否，預設爲否。**(**請一併參考 GetEnergy()說明**)**

6.OffHookDelay

設定 channel 執行 Pick Up 後，延遲多少時間才發出 EVT_EOP_NORMAL event。

7.OnHookDelay

設定 channel 執行 Hang Up 後，延遲多少時間才送出 EVT_EOP_NORMAL 的事件。

8.FlashTime

設定 flash time 時間長度。

9.PulseMake，PulseBreak，PulsePostDigitPause

這三個參數用來設定 Pulse dialing **(**脈衝方式撥號**)**的時間間隔。圖四表示撥出"213"波形及每一個參數所代表的意義。

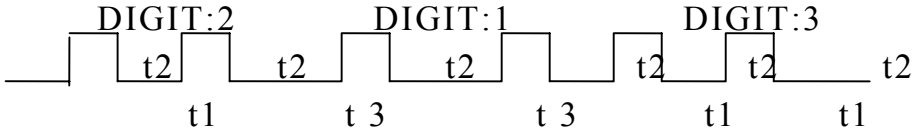


圖 四

t1 PulseMake
t2 PulseBreak
t3 PulsePostDigitPause

10.ToneDuration，InterTonePause

這兩個參數用來設定 DTMF dialing **(**複頻方式撥號**)**的時間間隔。

11.OutsideLineAccess

設定佔用外線所須撥的碼。 **(**若本 channel 接在 PBX 分機線上，亦即 LineToPBX 設定爲 1 時，本參數方爲有效**)**

12.RingsToAnswer

設定接收到幾響振鈴送出 EVT_DETECT_RING 的事件，預設值爲 1。**(**若 TriggerMode 設定爲偵測 Local Phone，本欄無作用**)**

13.WaitAnswerDuration

設定執行 call out 工作，最大等待 remote answer 的時間。

14.InterDigitPause

設定最大等待 remote 輸入按鍵的間歇時間，若在執行 GetDTMF() 時，間歇時間超過，系統會送出 EVT_TIME_OUT 的事件。

15.NoSignalTimeOut

設定 call out 後，執行 CPM 工作偵測不到任何訊號最大等待時間。

16.MaxRoarDuration

判定偵測到聒噪音，產生 EVT_DETECT_ROARING 事件的聒噪音持續時間。

17.PlayGain，RecordGain

PlayGain 用來設定輸出的音量大小 **(**放音或撥號**)**，RecordGain 用來調整輸入能量的放大比率。兩者的調整範圍皆是由 10～-10，預設皆爲 0，亦即爲 0dB。每一階度相差 3dB.所以其實際調整範圍爲-30dB～30dB。注意 RecordGain 的大小會影響 DTMF 信號的解碼及 energy 的值，如果有 DTMF 接收不到的情況發生，通常在降低 RecordGain 後即可改善。系統在執行聲音的 Record 及偵測 DTMF 信號時，可分別設定不同的 RecordGain。

18.PlayMode

894 可使用以下四種不同的放音模式：

① PM_HIGHER_SAMPLING
(9.8K or 16K bps**)**

② PM_HIGHER_SAMPLING_ECHO_CANCEL
(9.8K or 16K bps with echo cancellation**)**

③ PM_LOWER_SAMPLING
(8K or 4.9k bps**)**

④ PM_LOWER_SAMPLING_ECHO_CANCEL
(8K or 4.9Kbps with echo cancellation**)**

此四個模式常數均定義於 API894.H 中。其中具有 echo cancellation 功能者，是 894 上之 DSP 當偵測到有類似可疑之 DTMF 信號時，會暫時 Mute 輸出，以確定是否確實收到 DTMF 信號，以避免不正確的誤判，不過因此也會使輸出的聲音有斷續現象。故如考慮輸出之聲音品質建議勿使用 PM_HIGHER_SAMPLING_ECHO_CANCEL 及 PM_LOWER_SAMPLING_ ECHO_CANCEL 兩種模式。

19.RecordMode

894 可設定爲下列四種不同的錄音模式:

① RM_STANDARD_RATE **(**9.8Kbps**)**

② RM_PREMIUM_RATE **(**16Kbps**)**

③ RM_INTERMEDIATE_RATE **(**8Kbps**)**

④ RM_EXTENDED_RATE **(**4.9Kbps**)**

其中 RM_STANDARD_RATE 或 RM_EXTENDED_RATE 取樣率同於 RM_PREMIUM_RATE 或 RM_INTERMEDIATE_RATE 加上靜音壓縮之功能，因此 9.8Kbps 及 4.9Kbps 僅爲估計值，實際的錄音資料量，得視錄音內容及 Off Threshold 而定。

20.OffThreshold

設定錄音時雜訊能量的程度。在具有靜音壓縮的錄音模式下，輸入的信號能量低於 Off Threshold 的設定位準時，將被視爲雜訊壓縮，亦即不被列入波形取樣數據，可設定的範圍由 0 ～ 15。

[0] -50 dBm　　[8]　-32 dBm
[1] -49 dBm　　[9]　-29 dBm
[2] -47 dBm　　[10]　-26 dBm
[3] -44 dBm　　[11]　-23 dBm
[4] -42 dBm　　[12]　-20 dBm
[5] -40 dBm　　[13]　-17 dBm
[6] -38 dBm　　[14]　-14 dBm
[7] -35 dBm　　[15]　-11 dBm

21.DetectRemoteHangUpWhenRecording

22.DetectRemoteHangUpAlways

這兩個參數是用來設定偵測 remote Hang up 的系統狀態，設計程式時僅能二者選一；當 Detect-RemoteHangUpWhenRecording 設爲 TRUE 時，VP-894 只在系統錄音時，進行偵測 remote 端是否掛斷；若 DetectRemoteHangUpAlways 設定爲 TRUE 時，則 VP-894 自 pick up 開始即不斷地檢查是否有 remote 的掛斷訊號產生，直到 hang up 爲止**(**但如果目前進行的是 out-going 的 function，例如 CallRemote()，CallLocal()等，則是在 CPM 判斷爲 answer 後，才開始 remote hang up 的偵測**)**。至於偵測到何種線路訊號變化被視爲遠端掛斷，則可以藉由參數 23，24，25 進行設定。VP-894 播放語音時，如在 DetectRemoteHangUp-Always 條件下預先設定由 busy cadence 及 Threshold 參數偵測 remote 掛斷訊號，須嚴格限制系統播放出來的聲音屬純粹語音，如此便不致造成誤判。反之，如播放的內容爲一持續的音樂或 remote 端有固定持續性的噪音，且 energy 大於 threshold 的設定時則造成誤判之可能性增高 **(**請注意此處的 threshold 不同於壓縮模式錄音的 threshold**)**。

在系統錄製 Remote 留言時，由於線路狀態較爲單純，可以 DetectRemoteHangUpWhenRecording 做爲判斷 Remote 是否掛斷的條件。Application programmer 可依個別的需要在二者擇一使用；如果決定採用 DetectRemoteHangUpAlways 的方式，須在不同的噪音環境下測試，以確保 VP-894 不致產生判斷錯誤。

23.DetectBusyRemoteHangUp

24.DetectRoaringRemoteHangUp

25.DetectVoltReverseRemoteHangUp

這三個參數在上述 21、22 任一設定爲 TRUE 時， 可用來指明何者被視爲遠端掛斷的訊號。**(**即 Busy、Roaring 或 Voltage reverse**)** 三者可同時爲 TRUE，但不可同時爲 FALSE。

26.DetectSilentWhenRecording

用來設定在錄音時是否要偵測 Silence。**(**如果線路上持續 Silent 超過設定時間，則 VP-894 會依照參數 StopOperationRemoteHangUp 的設定回應本狀況**)**

27.StopOperationRemoteHangUp

決定系統偵測到 Remote Hang Up 後的處置方式。如果設定爲 TRUE，則在偵測 remote 掛斷後， VP-894 自動停止目前正在進行的工作，回復至 idle 的狀態，並且執行 HangUp()。如設定爲 FALSE 則只送出 event 通知 application 而不做任何處置。

3-4 、電話掛斷參數定義

在 API894.H included file 中定義了一個資料類別爲 typeHungUpParam 的結構，這個結構內的成員描述了被視爲遠端掛斷的各種訊號的特徵及指明如可控制 VP-894 辨識這些訊號。爲了調適不同電話系統間的差異，使 VP-894 達到最佳的辨識效果，故提供這些參數讓 client application 作調整，client application 可以利用 API function 中 SetHungUp-Param()來更改系統的預設值。下面是這個結構的原型定義:

```
struct  _tagHungUpParam{
 typeHungUpBusy Busy；
unsigned       MinBusyDuration；
unsigned       MinRoarDuration；
unsigned char  BusyThreshold；
unsigned char  RoarThreshold；
unsigned       MinSilentDuration
unsigned char  SilentThreshold
 }typeHungUpParam；
```

1.參數成員 — Busy

Busy 定義出掛斷 busy 訊息的 cadence 特徵，它是一個資料型別爲 typeHungUpBusy 的結構，此結構的原型如下:

```
struct _tagHungUpBusy{
        int          Varieties；
        typeDutyDuration Cadence[5]；
      } typeHungUpBusy；
```

其中 Varieties 指示有幾種 Cadence type 可判定爲掛斷 busy 訊號，最大爲 5 **(**因爲 Cadence[ ]的 size 爲 5**)**；Cadence[ ]定義各種 busy 訊號的 ON -OFF duration 特徵。**(**typeDutyDuration 請參考附錄 B: VP-894 之 CPM 原理解析中之說明**)**

2.參數成員 — MinBusyDuration

本參數設定至少必須持續偵測到與前述 Busy adence 參數相吻合的訊號多少時間，才判定爲掛 斷 busy 訊號。

3.參數成員 — MinRoarDuration

本參數設定至少必須持續偵測到 energy value 大於 RoarThreshold 設定多少時間，才判定爲掛斷 Roaring**(**howling**)**tone。

4.參數成員 — BusyThreshold

此參數設定判定 busy 訊號的 ON-OFF 的 energy 臨界點，例如:若本參數爲 5，則偵測到之 energy 值大於 5 者視爲 ON-DUTY，否則爲 OFF-DUTY。

5.參數成員 — RoarThreshold

本參數設定判斷爲 Roaring present 的 energy 臨界值。

6.參數成員 — MinSilentDuration

本參數設定在偵測到持續多久的線路 silent，才發出 EVT_LINE_SILENT。

7.參數成員 — SilentThreshold

本參數設定線路上 energy 值在多少以下視爲 silence。

以上 4. 5. 7.三項設定時，須考量目前 RecordGain 的設定值，因爲對同一強度的訊號，不同的 Record- Gain，會獲得不同的 energy 值，因此在設定 Busy- Threshold，RoarThreshold 及 SilentThreshold 的值時，應以實際使用的環境 **(**各地區不同**)** 中，在一標準的 RecordGain 值測試下所得的結果，做爲調整的依據。

四、C 語言介面程式(DOS 版)說明

使用 894 API 前必須先 install 一個常駐的驅動程式 DRV894.EXE，如果要 Release DRV894.EXE，只須執行 DRV894 時加上/R。大多數的 API function 均需指定一個 channel number 的參數。在 894 系統內 channel number 的定義是依照卡號依序排列，例如卡號 0 有 channel 0～3，卡號 1 有 channel 4～7，依此類推。以下詳述各個 function 的使用方式：

4-1、功能函數說明

*1.Init894(int * Install)*

功能：client application 在呼叫其它 894 API 函數之前，必須先執行本函數，令驅動程式對系統內安裝的所有 894 設備進行初始化工作。本函數在返回時，會將 894 的安裝組態填入 client application 呼叫時所提供的 array 中。

輸入參數：Install 是由 client application 在呼叫本函數時，所傳來的一個指到含 16 個整數元素的陣列啓始位置，本函數會在返回 client application 時，於 Install 所參考到的陣列的每一個元素內填入代表該卡號 VP-894 上 channel 組態的 bitmap 值，各個位元以 0 表示 channel absent，1 表示 channel exist，而由 LSB bit 0→bit 3 分別代表卡上的 channel 0→channel 3。例如，如果 install[2]nstall[7] 內分別被填入值 12(00001100b)及 15(00001111b)，則表示系統內安裝有卡號 2 及 7 的 VP-894，且有效的 channel 爲 10. 11.28.29.30.31。

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：無

2.Close894()

功能：本功能呼叫使所有 channel 終止工作，回復至 idle 狀態。
Application 應在程式結束前呼叫本功能

輸入參數：無

傳回值：無

送出 event type：無

*3.GetCtrlParam(int ChNum，typeCPB * CtrlParam)*

功能：本功能用來讀取目前指定 channel 之控制參數

輸入參數：ChNum channel Number

CtrlParam:這個輸入參數是一個指到資料類別爲 typeCPB 的結構變數指標，用來儲存所讀取的 channel control parameter，typeCPB 的原型定義，及此結構內成員的說明，請參考第三章、3-3 節。

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：無

4.SetCtrlParam(int ChNum，typeCPB * CtrlParam)

功能：本 function 用來設定 ChNum 所指定 channel 的控制參數

輸入參數：請見 GetCtrlParam()說明

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：EVT_EOP_NORMAL

5.PickUp(int ChNum)

功能：使由 ChNum 所指定之 channel 佔線

輸入參數：ChNum channel number

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：EVT_EOP_NORMAL

6.HangUp(int ChNum)

功能：使由 ChNum 所指定之 channel 掛上電話

輸入參數：ChNum channel number

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：EVT_EOP_NORMAL

7.Flash(int ChNum)

功能：使由 ChNum 所指定之 channel 暫時斷線，並於 Flash Time 所指定的時間終了重新 Pick Up **(**此動作在交換機內是作爲轉接之用**)**

輸入參數：ChNum channel number

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：EVT_EOP_NORMAL

8.Play(int ChNum，int Handle，long Length， unsigned AcceptDTMF)

功能：使由 ChNum 所指定之 channel 由與 Handle 相關的檔案之目前檔案指標位置開始播音，同時 DTMF queue 被清除輸入參數：ChNum channel number

Handle：要放音的檔案 Handle 號碼，在放音的過程中此檔案必須保持開啓的狀態

Length：指定要放音的長度，單位爲 byte，] 如果本參數設定爲 0，則檔案由目前指標位置放音到檔案尾端。如果指定長度超過檔尾，聲音僅放至檔尾即結束

AcceptDTMF：在放音中可接受的 DTMF 輸入。如果有接受的 DTMF 輸入，放音工作會提前終止。本項參數可由下列一個或多個常數以 bitwise OR(¦)合併形成

E_DTMF0,E_DTMF1,E_DTMF2,E_DTMF3
E_DTMF4,E_DTMF5,E_DTMF6,E_DTMF7
E_DTMF8,E_DTMF9,E_DTMFa,E_DTMFb
E_DTMFc,E_DTMFd,E_DTMF_ASTERISK
E_DTMF_POUND

以上常數定義在 API894.H 中

傳回值：0 代表成功，否則傳回錯誤碼 **(**如果錯誤碼小於 255，代表的是 DOS 的 function call error，請參考 DOS 技術手冊以獲得完整的錯誤訊息資料**)**

送出 event type：EVT_EOP_NORMAL
EVT_DTMF_INTERCEPT
EVT_PCMIO_ERROR

9.Record(int ChNum，int Handle，unsigned Length，unsigned AcceptDTMF)

功能：使由 ChNum 所指定之 channel 由與 Handle 相關連的檔案之目前檔案指標位置開始錄音，同時清除 DTMF queue

輸入參數：ChNum channel number

Handle：要錄音的檔案 Handle 號碼，在錄音的過程中此檔案必須保持開啓狀態，且須爲准許寫入模式

Length：指定最大的錄音長度，單位爲秒，如果本項參數設定爲 0，則錄音長度無限制

AcceptDTMF：在錄音過程中，可接受的 DTMF 輸入。如果偵測到有接受的 DTMF 輸入，則終止錄音工作。
(請參考 Play()參數 Accept- DTMF 說明**)**

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type： EVT_EOP_NORMAL
EVT_DTMF_INTERCEPT
EVT_PCMIO_ERROR

*10.GetDTMF(int ChNum，char * Buffer，unsigned Count，unsigned AcceptDTMF， unsigned ExitDTMF)*

功能：由 ChNum 所指定之 channel 讀取一串 DTMF 輸入

輸入參數：ChNum channel number

Buffer：爲一指到要儲存 DTMF 輸入的緩衝區起始位置

Count：指明要讀取 DTMF 輸入的個數 **(**此值必須小於或等於緩衝區之大小減 1，驅動程式會在 DTMF 字串尾端自動補入 NULL 字元**)**

AcceptDTMF：指明可接受的 DTMF 輸入

ExitDTMF：當偵測有與本項參數 enable 的 DTMF 信號輸入時 channel 會停止 DTMF 的讀取工作，並且將此 DTMF 之前的 DTMF 資料放入緩衝區

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type： EVT_EOP_NORMAL
EVT_DTMF_INTERCEPT
EVT_TIME_OUT

11.FlushDTMF(int ChNum)

功能：清除由 ChNum 所指定 channel 的 DTMF queue

輸入參數：ChNum channel number

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：EVT_EOP_NORMAL

*12.Dial(int ChNum，char * String)*

功能：由 ChNum 所指定的 channel 撥出一串號碼

輸入參數：ChNum channel number String－指到欲撥出號碼起始位置之指標

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type:EVT_EOP_NORMAL

*13.GetEnergy(int ChNum，unsigned char * const Buffer)*

功能：讀取由 ChNum 所指定 channel 的信號能量紀錄。如果 control parameters 中 MonitorEnergy 設定爲 1 時系統會每隔 240 ms 送出一組 30 個線上信號能量變化的紀錄 **(**每個信號能量間隔 8 ms**)**並且塞入一 EVT_REPORT_ENERGY 到 event queue 中。程式寫作者可利用這些資料做 call progress monitor**(**CPM**)**的設定參數。附錄 C 是 Ring Back 及 Busy 信號的例子。此表列出之 energy values 並非固定，依照測試線路信號之強弱、線路本身雜音及測試時之 RecordGain 的大小而有所不同

輸入參數：ChNum channel number

Buffer：指到大小 30 bytes 儲存 energy 資料的緩衝區起始位置。下表列出 energy value 所代表之信號大小

value	energy level
0	小於 -48dBmo
2	-45dBmo
4	-42dBmo
5	-39dBmo
6	-36dBmo
7	-33dBmo
8	-30dBmo
9	-27dBmo
10	-24dBmo
11	-21dBmo
12	-18dBmo
13	-15dBmo
14	-12dBmo
15	-9dBmo
16	-6dBmo
17	-3dBmo
18	0dBmo
19	3dBmo
20	6dBmo
21	9dBmo

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：無

14.StopCh(int ChNum)

功能：終止目前由 ChNum 所指定 channel 所進行的工作

輸入參數：ChNum channel number

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：EVT_ENDOF_STOP

*15.GetEvent(typeEvent * EvtBuf)*

功能：由系統 event queue 中 fetch 一個 new event

輸入參數：

EvtBuf：爲一指到資料類別爲 typeEvent 的緩衝區起始位置 **(**此資料類別爲一資料結構.定義在 API894.H**)** 以下分別說明各項資料成員代表之意義

Issuer：送出此 event 的　　channel number

Type：此 event 的類別。請參考第三章 3-1 節的說明

Data：當 type 爲 EVT_DTMF_INTERCEPT 或 EVT_DETECT_DTMF 時，此內容爲收到的 DTMF 碼；當 type 爲 EVT_PCMIO_ERROR 時，本項內容爲 DOS 系統錯誤碼；當 type 爲 EVT_INTRN_QUEUE_OVERFLOW 時此項資料爲 ERR894_EVENT_QUEUE_OVERFLOW 或 ERR894_PCMIO_QUEUE_OVERFLOW。若爲其他 event types，本項資料無定義

傳回值：0 代表目前 event queue 是空的，none ZERO 代表傳回新的 event

送出 event type：無

*16.InsertEvent(typeEvent * EvtBuf)*

功能：894 系統保留 OxAO～OxFF 讓程式寫作者自行定義不同的 event type。本項功能即是讓程式寫作者可放置一 event 至系統的 event queue 中

輸入參數：

EvtBuf：請參考 GetEvent()中之輸入參數說明

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：無

17.FlushEvent(void)

功能：清除 894 系統 event queue

輸入參數：無

傳回值：無

送出 event type：無

*18.GetCPMParam(typeCPMParam * CPMParam)*

功能：讀取目前系統內 CPM 參數的設定值。

輸入參數：

CPMParam：此指標是一要儲存讀取 CPM 參數的緩衝區起始位置，它所指示的資料類別爲被定義於 API894.H 的 typeCPMParam 的結構，此結構的原形如下：

```
struct _tagCPMParam {
        typeCadenceType  CentralOffice;
        typeCadenceType  PrivateSystem;
        typeCadence      Beeper;
      } typeCPMParam;
```

其中結構成員 CentralOffice 及 PrivateSystem 的資料類別是一預先定義的結構 typeCadenceType；CentralOffice 中紀錄的是局線 **(**外線**)** 信號的特徵；PrivateSystem 中則紀錄 PABX 內線的信號特徵。typeCadenceType 的結構原型定義如下：

```
struct _tagCadenceType {
        typeCadence  Ring;          〃回鈴音
        typeCadence  Busy;          〃忙音
        typeCadence  InvalidNum;    〃空號
        typeCadence  DialTone;      〃撥號音
        typeCadence  UserDefined1;  〃使用者定義 1
        typeCadence  UserDefined2;  〃使用者定義 2
      } typeCadenceType;
```

以上各個信號的資料類別與 typeCPMParam 中的成員 Beeper (呼叫器)同屬結構類別 typeCadence，下面是此種結構類別的原型定義：

```
struct _tagCadence {
        unsigned char   WaveCount  3;
        unsigned char   Type       :1;
        unsigned char              :4;
        unsigned char   RecognizeCycle;
        typeDutyDuration Wave[6];
      } typeCadence;
```

這種結構類別 (typeCadence)定義出各不同信號是如何被量化的紀錄，詳細關於 CPM algorithm 的說明，請見附錄 B (CPM 原理解析)

傳回值：無

送出 event type：無

*19.SetCPMParam(const typeCPMParam * CPMParam)*

功能：設定 CPM 的參數

輸入參數：

CPMParam：此指標是一要設定 CPM 參數的緩衝區中的起始位置，詳細說明請參考 GetCPMParam()中輸入參數之說明。

傳回值：0 代表成功，否則傳回錯誤碼。

送出 event type：無

*20.CallLocal(int ChNum，const char * TargetNum，unsigned Mode)*

功能：啓動由 ChNum 所指定 channel 執行內線分機的撥號 **(**如果所指定 channel 的控制參數中 LineToPBX 設定爲 False，則此功能與 CallRemote()相同**)**

輸入參數：ChNum channel Number

TargetNum：受話方的分機號碼

Mode:：指定撥叫電話的工作模式，可由以下預先定義於 API894.H 中的一個或多個常數以 bitwise OR (¦) 合併形成

CM_NO_DO_CPM	不要執行 CPM
CM_NO_WAIT_DIAL_TONE	不要等待及偵測 Dial tone
CM_DETECT_INVALID_NUM	要偵測空號
CM_DETECT_USER_DEFINED1	要偵測使用者定義的信號 1
CM_DETECT_USER_DEFINED2	要偵測使用者定義的信號 2

如果要使用 default 的工作模式，只要將 Mode 指定爲 0 即可**(**換句話說，default的工作模式是要執行CPM及等待偵測 dial tone，不偵測空號、使用者定義信號 1 及使用者定義信號 2**)**

傳回值：0 代表成功，否則傳回錯誤碼。

送出 event type： EVT_NO_DIAL_TONE
EVT_CPM_COMPLETE

*21.CallRemote(int ChNum，const char * TargetNum，unsigned Mode)*

功能：啓動由 ChNum 所指定的 channel 執行外線電話撥號，如果所指定 channel 的控制參數中 LineToPBX 設定爲 TRUE。本功能會先撥出控制參數中 OutsideLine- Access 所指定的碼佔外線後，再撥出受話方之電話號碼(本功能與 CallLocal()功能一樣，啓動時如在 Hang up 的狀態，會自動啓動 PickUp())

輸入參數：請參考 CallLocal()中輸入參數的說明。

傳回值：0 代表成功，否則傳回錯誤碼。

送出 event type： EVT_NO_DIAL_TONE
EVT_CPM_COMPLETE

*22.CallBeeper(int ChNum，const char * TargetNum ，const char * Message*

功能：啓動由 ChNum 所指定的 channel 執行呼叫器的撥號傳呼。如果所指定的 channel 是連接在內線上 **(**LineToPBX=TRUE**)**。本功能會先撥出控制參數 OutsideLine-Access 所指定的碼佔外線。再撥出呼叫器的號碼 **(**本功能啓動時，如在 hang up 的狀態。會自動啓動 PickUp()**)**

輸入參數：ChNum channel number

TargetNum：呼叫器的號碼

Message：要傳送給呼叫器的訊息 **(**要包含 prolog 與 epilog digit，例如'#'**)**

傳回值：0 代表成功，否則傳回錯誤碼。

送出 event type： EVT_NO_DIAL_TONE
EVT_CPM_COMPLETE

*23.GetHungUpParam(typeHungUpParam * Param)*

功能：讀取目前系統內掛斷訊號的參數設定值

輸入參數：

Param：這個輸入參數是一個指到資料類別爲 typeHungUpParam 的結構變數指標，用來儲存所讀取的掛斷參數， typeHungUpParam 的原型定義，及此結構的成員說明，請參考第三章 3-4 節。

傳回值：無

送出 event type：無

*24.SetHungUpParam(typeHungUpParam * Param)*

功能：設定掛斷訊號的參數

輸入參數：

Param：請參考 GetHungUpParam()中之 Param 說明

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：無

25.SetInterLink(int ChNum，int Connect)

功能：VP-894 硬體上連續奇偶成對(例如 0 與 1 或 2 與 3...等依此類推)的兩個 channel，可藉由此函數設定是否形成內部迴路 (即轉接)的狀態。由於當成對的 channel 形成迴路後僅偶數(0，2，4...) 的 channel 可作錄音及監視 energy 的變化，故建議實際使用此轉接功能時，由偶數 channel 做爲撥入方，再由奇數 channel 撥出，如此對偵測電話掛斷之判斷會較爲精準。

輸入參數:ChNum channel number

connect：此參數爲 0 時，所指定成對的 channel 設爲斷接(各自獨立)的狀態，若此參數爲非 0 時，則爲迴路狀態

傳回值：0 代表成功，否則傳回錯誤碼

送出 event type：EVT_EOP_NORMAL

註:在正常狀況下 VP-894 執行轉接功能時，應可維持良好之通話音量。如遇到線路狀況不佳(如衰減過大或匹配不良)時，可試下列方式改進①修改硬體 Gain 值②加裝本公司 HM-101A 線路增益器，詳情請與本公司連絡。

26.CheckWhetherVP894StillAlive *(int AdptrNum，unsigned EchoTimeOut)*

功能：檢查所指定的 VP894 設備是否仍然在正常的工作。

輸入參數：

AdptrNum：指明所要檢查 VP894 設備的卡號

EchoTimeOut：指明最大等待 VP894 設備回應的時間，以電腦的系統時脈每秒 18.2 次 ticks 爲計時單位。例如若本參數設定爲 182， 即代表 10 秒。

傳回值：0 表示成功，否則傳回錯誤碼

可能傳回的 event types： EVT_VP894_ALIVE
EVT_VP894_DEAD
(結構 typeEvent 的成員 Issuer 代表的是送出事件的卡號**)**

4-2 、脈衝(PULSE)撥號解碼功能說明

VP-894 之 V2.6X 版以上 API 增加了 VP-894 pulse 撥號解碼的能力，以適應不具複頻撥號功能的線路環境 (如東歐及中國大陸)。應用本項功能時，須考慮各種複雜外在因素，例如 VP-894 必須接受來自各個不同交換機房的 in-coming call，亦可能發生複頻及脈衝撥號混合使用的狀況。況且各個話機的脈衝撥號的訊號並不完全一致，在經過不同機房線路搭接路徑而到達 VP-894 後，其所造成訊號的衰減程度也各有大小不同，所以很難用一組固定的模式達成 pulse 撥號的正確解碼。爲了解決以上的問題，VP-894 的 firmware 設計採用每通電話學習的方式，以靈活地變動脈衝撥號訊號的特徵參數，提高系統的辨識率。因此應用 pulse 撥號解碼功能時，就必須遵循上述流程，在進入 application 的程序之前先要求 remote 輸入 9 或 0 中任一碼，讓 VP-894 可針對收到的碼進行學習，調整隨後接收到的輸入碼 (如果是 in-coming call，VP-894 在 off hook 後，即開始偵測是否收到相關的 pulse 訊號；out-going call 則是在判斷對方接聽後再進行偵測)。圖五是進入 application 程序前完成 pulse learning 的基本流程圖，在 API 磁片上有一個依照此流程圖設計的 demo 程式 (depulse.c)，application programmer 可直接修改或截取本程式的片斷來發展 application；如果 application programmer 要自行撰寫此段程式，也必須符合本流程圖的程序，以使 VP-894 的 pulse 撥號解碼功能正常運作。(提示應答語音內容可依不同應用而更改)

圖五：

4-2-1、 typeCPB 中相關的參數

① *LearnPulse*

本項參數用來指明是否要學習並偵測 pulse 撥號，預設爲 FALSE。 若本項參數設定爲 TRUE 時，VP-894 上的 firmware 即依先前所述之程序進行 pulse 撥號解碼及學習，所以 application 必須在進入本身程序前，先依照前面的 pulse learning 流程圖完成 pulse 撥號的訊號學習。

② *MonitorPulse*

本項參數是用來指明當 VP-894 firmware 解出一個有效的 pulse 撥號時，是否要送出一個 EVT_DETECT_PULSE 的事件給 application，預設值爲 FALSE。VP-894 在收到一個有效的 pulse 撥號後，即將解出的相對 ASCII 碼放入 DTMF queue 中，所以對於 application 而言，取得遠端輸入資料的程序，不論複頻或脈衝撥號都是一樣的**(**都是呼叫 GetDTMF()**)**；如果 application 確實有必要分辨遠端使用的是何種撥號方式，即可啓動本項功能。

4-2-2、 相關的 EVENT TYPE

① *EVT_LEARN_PULSE_SUCCESS*

本事件是在啓動 pulse 撥號解碼功能後， VP-894 能在每一通電話撥號學習模式偵測階段中，成功地解出 pulse 撥號'9'或'0'並送出通知 application 的事件。

② *EVT_DETECT_PULSE*

本事件是指示 VP-894 接收到一有效的 pulse 撥號。**(**僅有在 channel 的控制參數中之 MonitorPulse 設定爲 TRUE 時，VP-894 才會送出這個事件，在這個事件結構成員 Data 中包含著所收到 pulse 的 ASCII 碼**)**

4-2-3、 使用 PULSE 撥號解碼功能須注意事項

① application 在進入電話應答的流程迴路後，切勿再更改 record gain，因爲 VP-894 上的 firmware 是利用每一通電話輸入的第一個撥號資料爲樣本來辨別同一通電話隨後的輸入資料，如果 application 在 VP-894 完成 pulse learning 後改變 record gain，則隨後輸入資料的 energy 變化即會與原來的樣本有誤差，而使 VP-894 辨認發生錯誤。

② VP-894 在播放語音時，無法正確辨別 pulse 撥號，因此爲避免發生誤判，VP-894 在執行語音播放工作時，不偵測 pulse 撥號。
(firmware 內部自動暫時 disable 這項功能**)**

4-3、 錄/放音程式(UTY894.EXE)說明

UTY894.EXE 是提供給客戶自行錄/放音應用的程式。使用前請先將介面轉換器依圖六的說明配線完成，再執行 UTY894.EXE。請注意在操作錄/放音功能時，應將介面轉換器上的錄/放音選擇開關切換在適當的位置，否則無法正確使用。如使用 9V 電池操作， 請注意在測試完畢後立即關掉電源，以免消耗電池電力。UTY894 可分爲兩個 menu，分別是 play menu 及 record menu。 程式初始啓動時先在目前目錄尋找 temp.voc，如果找不到就自動 create temp.voc 的檔案。如果 temp.voc 爲新建的檔案或其檔案長度爲 0，系統即自動進入 record menu，否則會進入 play menu。在 record menu 中可設定的項目有 channel number，file name，record gain，record mode 及 off threshold，(本項爲在有靜音壓縮時才使用)在 play menu 中可設定的項目有 channel number， file name，play gain，play mode，increment， start position 及 end position。

圖六:VP-894 與測試盒(Tester)連接圖示

4-3-1、系統 HOT KEYS

① ***Ctrl-P*** 切換至 play menu。如果目前的檔案長度爲 0 系統會顯示錯誤訊息並要求先作錄音。

② ***Ctrl-R*** 切換至 record menu，且錄音模式設定爲 append。**(**螢幕右方會顯示"Append"字樣**)**

③ ***Ctrl-S*** 指明一檔案路徑，並將目前的聲音檔拷貝至此檔案中。如果目前系統是在 record menu 中，所有的資料均會複製；如果系統是在 play menu 中，複製的檔案內容僅由目前之由 start position 至 end position。

④ ***Ctrl-O*** 開啓新的聲音檔案。如果檔案長度不爲 0，系統進入 play menu，否則進入 record menu。

⑤ ***Alt-X*** 結束程式，回到 DOS。

⑥ ***F1*** 本鍵僅在 record menu 可使用，用來在 Append 及 Overwrite 兩種錄音模式間切換。在 Append 模式中，後續的錄音均接續在原聲音檔後面；而在 overwrite 模式中，新的錄音內容會覆蓋掉原來的內容。

4-3-2、 錄音說明

在 record menu 中可執行錄音功能及設定 channel number，file name，record gain， record mode 及 off threshold 五項資料，使用者可用↑↓鍵移動 select bar 選擇要設定的項目或←→鍵更改各項目之內容，以下分別說明這五項設定:

① channel number：

選擇錄音的 channel，使用← →鍵更改。

② file name：

設定目的與 Ctrl-O 同，使用者可使用→鍵或 End 鍵或直接輸入檔案路徑名稱。

③ record gain ：

設定錄音的輸入訊號放大比率，有效範圍由-10 至 10 使用← →鍵更改。本項設定與 Off Threshold 皆可影響錄音之品質，尤其是在有 靜音壓縮的錄音模式時 **(**9.8K 或 4.9Kbps**)**；如果有發生錄音音質不良的狀況可增加本項參數及降低 Off Threshold 以改善。

④ record mode ：

選擇錄音模式可使用← →鍵更改。模式種類有 4.9Kbps、9.8Kbps、16Kbps、8Kbps 四種，請注意 4.9K 及 9.8K 是一般平均人類語言經過靜音壓縮後的結果，並非絕對速率。

⑤ off Threshold：

設定輸入訊號雜訊能量比率之認定程度，有效範圍由 0 至 15。使用← →鍵更改。在有靜音壓縮的錄音模式中，當輸入的訊號能量低於 off Threshold 之設定時，將被視爲背景雜音而不予接受，僅有訊號大於 off Threshold 設定時，才被接受進行數位錄音。

4-3-3、 放音說明

在 play menu 中可執行放音功能及設定 channel number、file name、play gain、play mode、increment、start position 及 end position 等七項資料，使用者可用↑↓鍵移動 select bar 選擇要設定的項目或← →鍵更改各項目之內容，以下分別說明這七項設定:

① channel number：

選擇放音的 channel，使用← →鍵更改。

② file name：

請參考 record menu **(2)** file name 說明。

③ play gain：

設定放音時，音量的放大比率有效範圍由-10 至 10 使用← →鍵更改。

④ play mode：

選擇放音的模式，使用← →鍵更改。

模式種類如下:

9.8 or 16kbps

9.8 or 16kbps with echo cancellation

4.9 or 8kbps

4.9 or 8kbps with echo cancellation

(echo cancellation 說明請參考第三章 3-3 節中之 18.PlayMode**)**

並建議勿使用 with echo cancellation

⑤ increment：

設定調整放音 start position 及 end position 時，每一 step 的增減量，由 50 至 10000**(**bytes**)**，使用← →鍵更改。

⑥ start position：

設定放音的啓始位置，使用者可用下列各鍵調整。

← → 鍵每次增減 increment 設定之長度
＋ － 鍵每次增減 1 byte
Home 鍵設定爲 0，亦即檔案開始之位置

使用者亦可直接輸入欲設定的位置**(**digits**)**

⑦ end position：

設定放音的結束位置，使用者可用下列各鍵調整。

← → 鍵每次增減 increment 設定之長度
＋ － 鍵每次增減 1 byte
End 鍵設定至檔尾位置

使用者亦可直接輸入欲設定的位置**(**digits**)**

4-4 、示範應用程式說明

在 894 API 磁片中附有七個 DEMO 應用程式來示範每個 API function 的使用方法，它們分別是 DTMF.EXE，TEST.EXE，ENERGY.EXE，DEMO.EXE， OHDETECT.EXE，CALLOUT.EXE，INTRLINK.EXE。這七個程式的原始檔案也附在磁片中，下頁起即分別說明這七個 DEMO 應用程式:

1.DTMF.EXE (須照圖七所示連接後再執行)

這個程式除了示範撥號及讀取 DTMF 資料的功能外,且可做爲上述兩項功能的測試程式。它先各別由雙號 channel 撥號至單號 channel,單號 channel 接收到後檢查接收是否正確,若正確無誤則撥回一'*',通知對方接收正確,然後單雙號 channel 工作對調,繼續前述工作程序,中間若有錯誤發生,則程式終止,並 Display diagnostic message。因爲單雙 channel 要互相 communication,所以在執行本程式前,必須先用電話線將單雙 channel 連接起來。

圖七:示範程式 DTMF.EXE 測試前之連線方式

2.TEST.EXE

本程式是示範 multi-line 錄音，放音及 StopCh()的呼叫方式。 執行本程式前請先依圖六所示將 VP-894 Tester 盒接好(多路測試時，需搭配多個 Tester 盒)；錄音音源可選擇由 MIC 或 TAPE 輸入；放音前須確定每一個 channel 已錄有音源檔才可執行。

3.ENERGY.EXE

本程式是示範如何使用 GetEnergy()monitor 線路上信號能量的變化，Application 可利用這些能量的紀錄來判斷忙音或 Answer，附錄 C 中的能量資料即是由本程式所產生，以下是在 DOS command line 執行之命令格式：

＞ENERGY 電話號碼 (＞ 檔案名稱)

4.DEMO.EXE

本程式是用來測試 VP-894 的 DTMF 接收及電話的錄放音功能。執行本程式後，系統便處於 Wait ring 的狀態，使用者可用一般話機撥入系統，依照系統的提示應答選擇任意鍵入一串數字 **(**最多 10 個，用來測試 DTMF 解碼功能**)** 或輸入'*'號，選擇錄音。 **(**執行本程式前請先確定提示語音檔 Greet.voc 是否在現目錄中**)**。 若鍵入數字不到 10 個，系統會 Timeout；若選擇錄音，最長可錄 30 秒，亦可按一數字鍵提前結束。錄音完畢系統會自動 play 一次。

5.OHDETECT.EXE

ohdetect.exe 是偵測 remote 掛斷的相關演式程式。下面是在 DOS 命令提示執行本程式的方法：

drive>ohdetect /me|/moh[b|r][other option..]
[voice filename]

在啓動 ohdetect.exe 可由/me 或/moh 選擇一種工作模式；如果是使用/me，則在電話兩端接通後， ohdetect.exe real time 地將 energy 的變化輸出至螢幕；如果選用/moh，則本 demo program 是在執行 remote 掛斷偵測的工作模式。如果在**[**voice filename**]**欄有指明一語音檔的路徑名稱，則在電話接通後，VP-894 會撥放該語音內容。以下列出 ohdetect.exe 可使用的 options 的說明：

/me	監視 energy 的變化
/moh	偵測 remote 掛斷
b	偵測忙音
r	偵測聒噪音
/pg=?	設定 play gain (由-10 至 10)
/rg=?	設定 record gain (由-10 至 10)

如果在使用本程式偵測 remote 掛斷發生錯誤時，請參考手冊及 ohdetect.c 中 comment 說明修改 ohdetect.c

6.CALLOUT.EXE

callout.exe 是撥號後執行 CPM 偵測之示範程式。在 DOS 命令行啓動方式如下:

drive>callout [option(s)] phonenumber [count]

[option(s)]
/rg=? 設定 RecordGain (由-10 至 10)
/ot=? 設定 off threshold (由 0 至 15)
/area=? 指明 remote 長途區域碼
/me 監視 energy 變化

phone number 指出要撥號的遠端，如有指明 count，則 callout.exe 由 phone number 所指明的號碼開始，連續撥出 count 通。本程式 default 是由分機撥至外線 **(**即 LineToPBX 設定爲 TRUE**)**。其由系統內 INSTALL 的 VP-894 中最小卡號之 channel 0 執行撥號，其他 channel 則執行放音 **(**在現目錄須有 Temp.voc **)**。 例如只有一卡，其卡號爲 2，則執行撥號的 channel 是 8，如要改變撥號的 channel，在 channel 號碼前導以/，例如要用卡上第二線撥號，則須在 callout 後加/2。使用本程式偵測 CPM 如有不正確的情形發生，可參考手冊中 "VP-894 之 CPM 工作原理" 說明以及 callout.c 中的 comment 修改 CPM 參數。 **(**callout.exe 在判斷到 remote answer 後，須由 keyboard 鍵入任何一字，才會繼續下一通的撥號**)**

7.INTRLINK.EXE

本程式示範 SetInterLink() function 的使用方式。測試者可由任一 channel 撥入電話，然後依照語音提示輸入受話方的電話號碼；輸入完畢，系統會利用成對的另一 channel 撥出所指定的號碼，在受話方 answer 後，系統即會連接此成對的兩端電話迴路。執行本程式前，請先確定以下三個語音檔已存放在現目錄中--WAIT.VOC，QTELNUM.VOC 及 FAIL.VOC。測試者可依以下命令格式啓動 intrlink.exe:

intrlink [option(s)]
[option(s)]
/pg=?　設定 play gain (由-10 至 10)
/rg=?　設定 record gain (由-10 至 10)
/ot=?　設定錄音之 off Threshold (由 0 至 15)
/busy=?　設定判斷 remote hung-up 的忙音
　　Threshold 參數(0 至 21)
/roar=?　設定判斷 remote hung-up 的聒噪音
　　Threshold 參數(由 0 至 21)
/pbx　設定 VP-894 爲連接在 PBX 的分機線上
/me　設定 monitor energy 爲 TRUE

4-5、問題與解答

1.問題： Install 894 adapter 後系統無法開機或螢幕出現奇怪的字元或圖案 。

答： ① 請參考第二章 **(**894 硬體設定說明**)** 調整 S1－6～S1－8 以改變 SRAM 的 address 設定；894 adapter 佔用 PC A000～EFFF 中 32K 的連續位置，由於在這塊區域中網路卡、視訊卡、EMS 卡或 SCSI 等很可能均需佔用其中一部份位置，所以必須小心設定勿使 894 adapter RAM 佔用的位置與其他硬體介面卡重疊。

② 系統無法開機的另一種可能原因是由於 Install 多片 894 而 PC power 供電不足所導致。可嘗試減少 PC 內 install 的 894 卡數量，看看問題是否能解決。建議當使用多片 894 時，最好使用供電較大的工業用 PC。

2.問題： Install 894 adapter 後，螢幕不穩定、跳動。

答：見前一項回答②。

3.問題：Install DRV894.EXE 時，driver report 找不到卡 。

答： 請先參考問題 1.的回答① 檢查是否有硬體介面卡記憶體重疊的問題，如果確定沒有，請檢查系統內是否有 install memory management **(**例如 386MAX 等**)** 或 EMS emulator 軟體 **(**例如 DOS utility--EMM386**)**。因爲前者會將 extended memory remapping 至 640～1024K 之間，以增加在 DOS 系統下的可用記憶體空間，而後者則將 extended memory 模擬成 expand memory，故就如同 EMS 卡一般，且在 AOOO～EFFF 間須佔用 64K 記憶體作爲 page frame。因此如果系統內已使用這兩種軟體，亦須避免記憶體重疊問題。

4.問題：Install DRV894.EXE 時，僅找到一部份卡。

答：請檢查每一片卡是否 RAM address 皆調在同一個位置，且使用相同的 IRQ。

5.問題: Install DRV894.EXE 時，report 找不到 IRQ。

答： 請檢查是否與其他介面卡使用的 IRQ 有衝突，並確認每一片 894 IRQ 均設定同一 IRQ。另多片 VP-894 的 dip switch S1－1 僅一片可設爲 on。

6.問題: 無法正確偵測 Pulse 撥號 。

答： VP-894 在理想環境下均能正確完成 Pulse 撥號解碼功能，但在現實環境下，牽涉線路、話機、外界噪音等因素，故會產生誤差。建議在以下之條件下使用 Pulse 撥號解碼功能，應有較高之辨識率。

① 限制非長途線路使用

② 儘可能使用外線，避免使用內線分機

③ 避免在嘈雜環境下使用

④ 避免使用免持聽筒話機

五、CLIPPER 語言介面程式(DOS 版)說明

在本節中將針對 VP894 Clipper API 的使用，提供一般概括性地解說。基於此一 API 大多數的定義 (如常數、功能函式等)均與 C 語言的 API 相同，因此本章內將不再贅述僅說明與其相異之處。若您在閱讀本文件遇有任何項目、專有名稱不明瞭或是有任何 API 的函數未在此說明者，請參考前章 C 語言 API 的說明。本 API 與 Clipper vesion5.x 或以上版本相容。在 API 的磁片上有以下的檔案

.\api894.ch	included file
.\api894.lib	library
.\demo.rmk	make file for demo programs
.\src\test.prg	
.\src\energy.prg	
.\src\callout.prg	
.\src\beeper.prg	
.\src\dtmf.prg	
.\src\ohdetect.prg	
.\src\demo.prg	
.\src\depulse.prg	
.\src\hookstat.prg	
.\src\utility.prg	
.\bin\demo.exe	
.\bin\test.exe	
.\bin\dtmf.exe	
.\bin\energy.exe	
.\bin\hookstat.exe	
.\bin\callout.exe	
.\bin\beeper.exe	
.\bin\ohdetect.exe	
.\bin\depulse.exe	

其中 API894.CH 是一個定義所有在使用 API 時可能參考到的常數或巨集呼叫的含入檔,所以你必須在所有可能使用到 API 函數的應用程式的原始檔案之開頭處使用#include 命令將此檔案含入。

在\clipper\src 的子目錄中包含了數個範例程式，示範 API 函數的使用方法。你可以直接執行在\clipper\bin 目錄內，這些範例程式的可執行版本，看看這些範例程式是如何工作的,您亦可使用 demo.rmk 重新編譯連結這些範例程式。在你要重建這些程式之前，請先確定 API894.CH API894.LIB 及 DEMO.RMK 與這些範例程式是拷貝在同一個目錄中。您可以在 DOS 的命令提示後輸入以下的命令格式來重建範例程式

rmake demo /F <DEBUG="T">

RMAKE 是由 CLIPPER 所提供的一個專案維護的公用工具程式，有關此公用程式的使用方法及其命令列參數的說明，請自行參考 clipper 的程式設計手冊。如果您希望在產生的執行檔中包含除錯的訊息，在編輯，連結完成後，可以使用 clipper 的除錯器來追蹤程式的執行，您可在 RMAKE 的命令行後加上 DEBUG="T"。

5-1、與 C 語言 API 之差異

主要造成C語言API及clipper API兩者不一致處的原因有以下兩點:

① 變數的資料類別表示方法

在 clipper 中並沒有同等於 C 語言中可將互相相關而不同類型資料組合在一起的 struct 宣告，因此在需要傳送大量資料的功能函式，使用陣列來替代 C 語言的 struct。

② 程式執行時期的記憶體管理

在每一個使用 clipper 編譯產生的執行檔，包含了一個虛擬記憶體管理器負責爲應用程式管理在執行時期所使用的記憶體。它是利用類似於多工作業系統所慣常使用的方法來提供比實際記憶容量更多的記憶體給應用程式，也就是使用一部份硬碟空間當作第二儲存記憶體，在必要時把最近最少參考到的資料置換至第二儲存空間，而將讀寫速度較快的記憶體空出來給其他目前正使用的資料，在這個執行過程中常會造成記憶體區塊間的縫隙。因此爲了能夠有效地利用記憶體，虛擬記憶體管理器有時還必須搬移記憶體區塊，使被已配置記憶體區塊分割得肢離破碎的 free 記憶體得以合併，因此應用 程式在執行時期所產生的大多數記憶體物件均不會一直停留在固定的位置。在 VP-894API 中有些事件驅動的多工函式，是直接讀取由應用程式傳入參數記憶體位置的內容來執行工作，而非在呼叫時製造另一副本，因此虛擬記憶體對資料的搬移會造成 VP894 驅動程式參考到錯誤資料的嚴重後果 **(**因爲事件驅動的多工函式是在背景執行，儘管程式的執行權已回到應用程式，仍不意味多工函式已執行結束**)**。幸而 clipper 另外有一個固定記憶體配置器可以幫助我們解決這個問題，不過這又造成應用程　　式必須留意釋放這些配置的固定記憶體的負擔，有關此詳細的說明，請參考 5-3 節中 GetDTMF() 及 ReadReceivedDTMF()函數的說明。

5-2 API 的傳回值

VP-894 clipper API 中增加一個傳回值 ERR894_NO_FIXED_HEAP_AVAIL 表示已無固定的記憶體可供配置，其它 API 傳回值的定義及說明，請參考前面第三　章軟體發展介面程式大要中－錯誤碼常數定義中之　說明。

5-3、功能函數介紹

1.GetCtrlParam(ChNum，VP894CPB)

目的：取得所指定 channel 目前的控制參數設定

傳入參數：

① ChNum
channel 號碼

② VP894CPB
這是一個包含 21 個元素的陣列，用來儲存由驅動程式所傳回指定 channel 的控制參數。在 C 語言的 API 中這項傳入參數是一個指到結構類型 typeCPB 的指標，除了第一個陣列元素外，其它的陣列元素均對應於 type-CPB 的一個結構成員。以下是他們之間的對應關係：

VP894CPB[2]	OffHookDelay
VP894CPB[3]	OnHookDelay
VP894CPB[4]	FlashTime
VP894CPB[5]	PulseMake
VP894CPB[6]	PulseBreak
VP894CPB[7]	PulsePostDigitPause
VP894CPB[8]	ToneDuration
VP894CPB[9]	InterTonePause
VP894CPB[10]	OutsideLineAccess
VP894CPB[11]	RingsToAnswer
VP894CPB[12]	WaitAnswerDuration
VP894CPB[13]	InterDigitPause
VP894CPB[14]	NoSignalTimeOut
VP894CPB[15]	MaxSilence (obsolete，but reserved for compatible)
VP894CPB[16]	MaxRoarDuration (obsolete，but reserved for compatible)
VP894CPB[17]	PlayGain
VP894CPB[18]	RecordGain
VP894CPB[19]	PlayMode

VP894CPB[20] RecordMode
VP894CPB[21] OffThreshold

因爲 DialMode，LineToPBX TriggerMode，DetectRoaringRemoteHangUp，DetectBusy-RemoteHangUp，DetectRemoteHangUpWhen- Recording，DetectRemoteHangUpAlways，StopOperationRemoteHangUp，MonitorDTMF，MonitorEnergy，DetectVoltReverse- RemoteHangUp，DetectSilentWhenRecording，LearnPulse 及 MonitorPulse 等在 typeCPB 結構中的 bit field 成員宣告均只用一個位 元即可表示其所有有效值，因此我們將這些成員全部組合在 VP894CPB 陣列的第一個元素中，並且在 API894.CH 含入檔定義了一組常數 來代表這些 bit field 成員變數，下表是它們的對應關係：

alternates	bit fields in typeCPB
CPF_DIAL_PULSE	DialMode
CPF_LINE_TO_PBX	LineToPBX
CPF_MONITOR_LOCAL_PHONE_STATUS	TriggerMode
CPF_DETECT_HOWL_RHU	DetectRoaring-RemoteHangUp
CPF_DETECT_BUSY_RHU	DetectBusyRemote-HangUp
CPF_DETECT_RHU_WHEN_RECORDING	DetectRemoteHang-UpWhenRecording
CPF_DETECT_RHU_ALWAYS	DetectRemoteHang UpAlways
CPF_STOP_OPERATION_RHU	StopOperation-RemoteHangUp
CPF_MONITOR_DTMF	MonitorDTMF
CPF_MONITOR_ENERGY	MonitorEnergy
CPF_DETECT_VOLT_REVERSE_RHU	DetectVoltReverse-RemoteHangUp
CPF_DETECT_SILENT_WHEN_RECORDING	DetectSilentWhen-Recording
CPF_LEARN_PULSE	LearnPulse
CPF_MONITOR_PULSE	MonitorPulse

你可以使用加法運算將意欲 enable 功能的定義常數累加起來，然後指定給 VP894CPB[1]。例如假設你希望某一 VP894 的 channel 具備以下的功能特性：

(a) pulse 撥號
(b) 電話接線是 PBX 的分機線
(c) 偵測振鈴
(d) 偵測聒噪音，並視爲遠端掛斷的訊號
(e) 在與遠端接通全程中，均需監視對方是否掛斷
(f) 在偵測到遠方掛斷後，自動終止目前的工作，使 channel 回到 idle 的狀態
(g) 當偵測到一個有效的 DTMF 輸入時，產生 EVT_DETECT_DTMF 事件
(h) 不要產生週期性的 EVT_REPORT_ENERGY 事件。

你必須以底下的 statment 來指定 VP894CPB[1]的值：

```
VP894CPB[1]=CPF_DIAL_PULSE+；
CPF_LINE_TO_PBX+；
CPF_DETECT_HOWL_RHU+；
CPF_DETECT_RHU_ALWAYS+；
CPF_STOP_OPERATION_RHU+；
CPF_MONITOR_DTMF
```

typeCPB 內每一個結構成員的用途及設定方法，請參考第四章 C 語言 API 的說明。

傳回值：0 代表成功，否則傳回錯誤碼

傳回事件類別：無

2.SetCtrlParam(ChNum，VP894CPB

目的：設定所指定 channel 的控制參數

傳入參數：

① ChNum
channel 號碼

② VP894CPB
請參閱 GetCtrlParam()功能函數的第二個參數說明

傳回值：0 代表成功，否則傳回錯誤碼

傳回事件類別：EVT_EOP_NORMAL

3.GetDTMF(ChNum，Count，AcceptDTMF，ExitDTMF)

目的：由所指定的 channel 接收一串 DTMF 輸入

傳入參數：

① ChNum
channel 號碼

② Count
指明要接收的 DTMF 個數

③ AcceptDTMF
定義那些 DTMF 輸入是可接受的

④ ExitDTMF
定義那些 DTMF 輸入可以用來在收到所指定 DTMF 個數之前，終止本函數的工作

傳回值：0 代表成功，否則傳回錯誤碼

傳回事件類別：EVT_EOP_NORMAL
EVT_DTMF_INTERCEPT
EVT_TIME_OUT

說明：

如果本函式傳回 0 表示成功，API library 內部會透過 Clipper 的固定記憶體配置器獲得長度 conut + 1bytes 的暫存緩衝區用來儲存接收到的 DTMF 碼。應用程式在接收到以上各代表本函式工作結束的事件後,呼叫 GetReceivedDTMF()就可獲得所接收到的 DTMF 字串內容。而 GetReceivedDTMF()函數則會將暫存緩衝區內的 DTMF 字串拷貝至應用程式呼叫此函數時所提供的另一變數的參考位置，之後再自動將此先前配置得來的固定記憶體緩衝區釋還。因此應用程式必須負責在 GetDTMF()被成功呼叫並執行結束後，呼叫 GetReceivedDTMF()函數，且不論傳回的事件類型是 EVT_TIME_OUT 而接收到 DTMF 字串內容是無用的，才可以使 API library 有機會釋回所配置的記憶體。你如果在 GetDTMF()被成功呼叫，執行 StopCh()，則是一個例外情形； API 內部會在執行

StopCh()前檢查是否有任何記憶體物件與該 channel 相關連，若有則會自動釋放該記憶體，因此可以無須再呼叫 GetReceivedDTMF()來完成 clean up。請同時參考第四章、C 語言 API 中 AcceptDTMF 及 ExitDTMF 的說明。

4.ReadReceivedDTMF(ChNum，Buffer)

目的：用來讀取指定的 channel 所接收到的 DTMF 字串

輸入參數：

① ChNum
channel 號碼

② Buffer
用來儲存所接收到的 DTMF 字串

傳回值：0 代表成功，否則傳回錯誤碼

傳回事件類別：無

說明：本功能函數執行完成後，在返回應用程式前，會將與指定 channel 相關連所配置的固定記憶體緩衝區釋還，因此若以同一 channel 連續呼叫本函數會獲得 ERR894_INVALID_CHANNEL 的錯誤碼，以指明記憶體緩衝區內容已無效 **(**如果以一不存在設備的 channel 號碼呼叫本函式，亦會傳回此結果碼**)**。

5. GetEvent(Event)

目的：獲得下一個事件訊息

輸入參數：

① Event

這是一個包含 3 個元素的陣列，用來儲存由 API 傳回的事件訊息。各個陣列元素內容所代表的意義，條列如下：

Event[1] 發出此事件的設備 channel 號碼
Event[2] 事件的類別
Event[3] 其它額外的資料

傳回值：0 代表事件佇列是空的，非 0 值表示有一新的事件訊息被傳回

說明：Event[3]中的內容，依照 Event[2]內不同的事件類別，而有相異的定義：

Event[2]內的事件類別	Event[3]的定義
EVT_DTMF_INTERCEPT 或 EVT_DETECT_DTMF	代表 DTMF 的 ASCII 碼
EVT_DETECT_PULSE	代表 pulse 的 ASCII 碼
EVT_PCMIO_ERROR	DOS 的系統錯誤碼
EVT_INTRN_QUEUE_OVERFLOW	ERR894_EVENT_QUEUE_OVERFLOW 或 ERR894_PCMIO_QUEUE_OVERFLOW

如果 Event[2]內的事件類別是列在上表以外的其它事件，Event[3]的內容則無定義。完整的各種事件類別說明請參考前面第三章中事件定義。

6.InsertEvent(Event)

目的：將一事件排程至事件佇列中

傳入參數：

① Event
請參考 GetEvent()函數中的說明

傳回值：0 代表成功，否則傳回錯誤碼

傳回事件類別：無

說明：VP894 API 保留事件號碼 160 至 255 給應用程式定製自己的 private 事件類別。

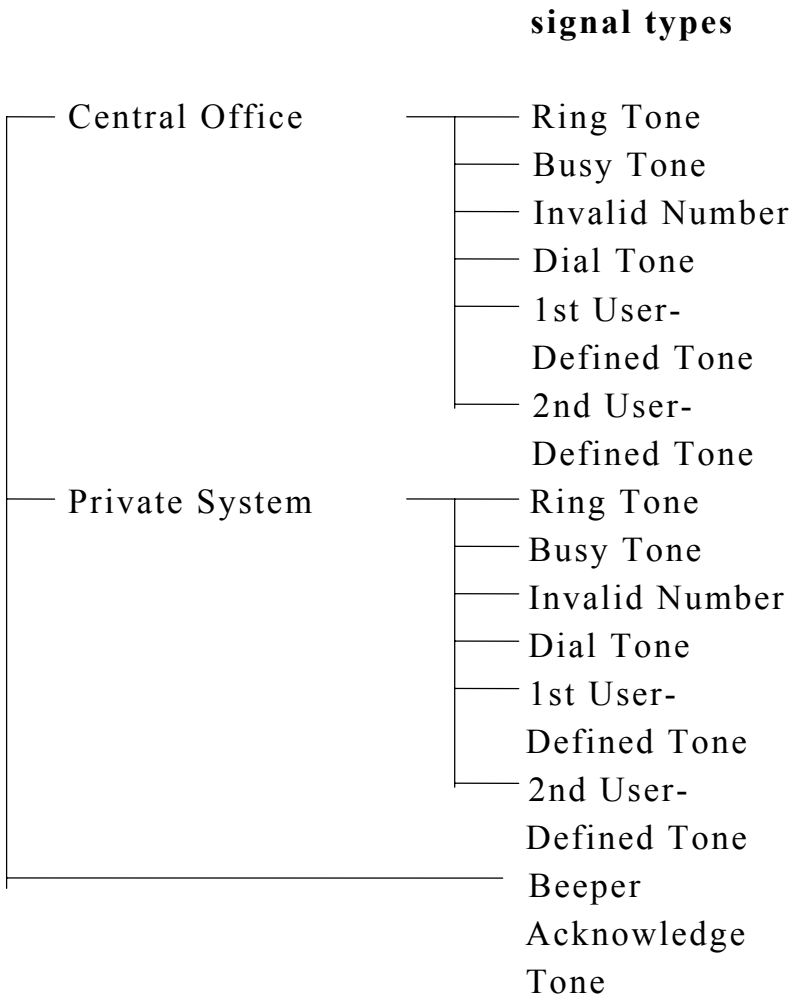
7.GetCPMParam(CPMParam)

目的：讀取目前的 CPM 參數設定

輸入參數：

① CPMParam

這是一個用來儲存由驅動程式所傳回 CPM 參數的陣列。當我們要撥號呼叫遠方時，可能會獲得以下結果中的任何一個一無撥號音、遠端佔線中、持續響鈴而遠端一直未接聽或遠端接答等；爲了使 VP894 API 中有關 call 的函式能夠知道撥號的結果，VP894 adapter 上的 micro processor 小心地監視線上能量的變化，以辨別出撥號後，線上所傳回訊號音的 on-off duration 與驅動程式 CPM 參數設定值比對，以找出相符者。因爲各種不同的交換機系統之訊號音的 on-off duration 都不同，所以我們提供這組參數讓應用程式可根據不同的使用環境作調整修正。在 call 的操作中，線上可能出現的訊號音不外以下四種： 撥號音(dial tone)、回鈴音(ring back tone)、忙音(busy tone)或是無效號碼(invalid number)，除此之外，我們另外多提供了兩組使用者可自行定義的訊號作爲擴充之用。再者、因爲連接在 VP-894 上的電話線可能是來自電信局的局線或私人交換機(PBX)的分機線，所以我們以 Central Office 及 Private System 兩組來定義個別不同的 6 種訊號。最後我們以一組訊號參數來定義在撥呼叫器 (beeper or B.B.call)號碼和訊息之間由電信局送出的 acknowledge tone (go ahead tone) 使 VP894 能具備撥叫呼叫器自動化的功能。結果所產生的資料結構如下表示：



而每一種訊號，我們是以以下幾項參數來定義其特徵：

(a) WaveCount--指明要以多少個 duty cycles 來作爲這個訊號的特性描述**(**訊號 1 次 on-off 變化稱爲 1 個 duty cycle**)**

(b) Type--指明這個訊號是否是週期性變化的訊號

(c) RecognizeCycles--指明這個訊號必須持續出現幾次才被承認

(d) Wave[] [4]--用來定義每一個 duty cycle 的 on-off duration 及誤差值 **(**tolerance**)**

在第四章的 C 語言的 API 中、我們是定義了一個 struct 資料類別來儲存這些參數，但就如同在前言中所提及的在 Clipper 中必須使用陣列替代之。在此陣列中每連續 27 個元素爲一組定義一個訊號的特徵 **(**每一個訊號最多只可定義 6 組不同的 duty

cycle)，因爲總共有 13 個不同的訊號要分別描述之，所以這個陣列必須包含 351 個元素。下表是列出陣列中每一組 27 個元素所代表的訊號：

central office
CPMParam[1] -CPMParam[27] ring tone
CPMParam[28] -CPMParam[54] busy tone
CPMParam[55] -CPMParam[81] invalid number
CPMParam[82] -CPMParam[108] dial tone
CPMParam[109]-CPMParam[135] 1st user-defined tone
CPMParam[136]-CPMParam[162] 2st user-defined tone

private system
CPMParam[163]-CPMParam[189] ring tone
CPMParam[190]-CPMParam[216] busy tone
CPMParam[217]-CPMParam[243] invalid number
CPMParam[244]-CPMParam[270] dial tone
CPMParam[271]-CPMParam[297] 1st user-defined tone
CPMParam[298]-CPMParam[324] 2st user-defined tone

beeper
CPMParam[325]-CPMParam[351] beeper
acknowledge tone

對於每一組 27 個元素中，各別所表示的意義則列於下表中：

CPMParam[n+0] Wave Count
CPMParam[n+1] Type
CPMParam[n+2] Recognize Cycle(s)
CPMParam[n+3] Wave[0] Silence
CPMParam[n+4] Wave[0] None Silence
CPMParam[n+5] Wave[0] Silence Tolerance
CPMParam[n+6] Wave[0] None Silence Tolerance
CPMParam[n+7] Wave[1] Silence
CPMParam[n+8] Wave[1] None Silence
CPMParam[n+9] Wave[1] Silence Tolerance
CPMParam[n+10] Wave[1] None Silence Tolerance
:
(omitted)
:

CPMParam[n+23] Wave[6] Silence
CPMParam[n+24] Wave[6] None Silence
CPMParam[n+25] Wave[6] Silence Tolerance
CPMParam[n+26] Wave[6] None Silence Tolerance

在上表內，陣列索引的表示式中，n 代表每一個訊號類別在陣列中的起始元素號碼。有關這些 CPM 參數如何控制影響 VP-894CPM 的動作的詳細說明，請參考附錄的 CPM 原理解析。

傳回值：0 代表成功，否則傳回錯誤碼

傳回事件類別：無

8.SetCPMParam(CPMParam)

目的：設定新的 CPM 參數值

輸入參數：

① CPMParam

請參閱 GetCPMParam()函數中，輸入參數的說明

傳回值：0 代表成功，否則爲錯誤碼

傳回事件類別：無

9.GetHungUpParam(HungUpParam)

目的：取得目前遠方掛斷偵測的參數

輸入參數：

① HungUpParam

這是一個具有 27 個元素的陣列，用來儲存由驅動程式所傳回的遠方掛斷偵測參數。這些參數是描述在遠端掛斷電話後，線上所出現訊號的特徵，以作爲判斷遠端是否掛斷電話的依據。在第四章 C 語言 API 中這些參數是被組合在結構類別 typeHungUpParam 內，下表是各陣列元素與 typeHungUpParam 結構成員互相對應的對照表：

HungUpParam[1]	MinBusyDuration
HungUpParam[2]	MinRoarDuration
HungUpParam[3]	BusyThreshold
HungUpParam[4]	RoarThreshold
HungUpParam[5]	MinSilentDuration
HungUpParam[6]	SilentThreshold
HungUpParam[7]	Busy.Varieties
HungUpParam[8]	Busy.Cadence[0].Silence
HungUpParam[9]	Busy.Cadence[0].NoneSilence
HungUpParam[10]	Busy.Cadence[0].Silence-Tolerance
HungUpParam[11]	Busy.Cadencd[0].None-SilenceTolerance
HungUpParam[12]	Busy.Cadence[1].Silence
HungUpParam[13]	Busy.Cadence[1].NoneSilence
HungUpParam[14]	Busy.Cadence[1].Silence-Tolerance
HungUpParam[15]	Busy.Cadence[1].None-SilenceTolerance
:	
(omitted)	
:	
HungUpParam[24]	Busy.Cadence[4].Silence
HungUpParam[25]	Busy.Cadence[4].NoneSilence
HungUpParam[26]	Busy.Cadence[4].Silence-Tolerance
HungUpParam[27]	Busy.Cadence[4].None-SilenceTolerance

有關每一個 type HungUpParam 結構成員的完整說明請參閱前面第三章節軟體發展介面程式大要中的電話掛斷參數定義及本函式在第四章 C 語言 API 中的說明。

傳回值：0 代表成功，否則傳回失敗的錯誤碼

傳回事件類型：無

10.SetHungUpParam(HungUpParam

目的：設定新的遠方掛斷偵測參數

輸入參數：

① HungUpParam

請參考 GetHungUpParam()函數中，輸入參數的說明

傳回值：0 代表成功，否則傳回失敗的錯誤碼

傳回事件類別：無

六、Visual C 語言介面程式 (WINDOWS 版)說明

由於 Windows 是一個多工的作業環境，因此 VP-894 Device driver for Windows 不像在 DOS 作業系統下的 TSR 驅動程式，僅能支援一個 application 工作。VP-894 的 Windows driver 可以同時支援多個不同的 application 或是同一個 application 的多個 instance 工作，最多可以有 16 個視窗同時使用 VP-894 的設備。因爲前述作業環境上的差異，使得 DOS 及 Windows 下的 API 在使用上有些不同,以下列出此二者之間主要的差異:

(1) 在 Windows 的 API 中刪除 GetEvent()及 FlushEvent()二項 function，我們直接使用每個 Windows application 的 message queue 作爲 device driver 及 application 之間的溝通橋樑。

(2) Init894()返回時會傳回一個 instance number **(**此 instance number 不同於 windows instance，它是 VP-894 device driver 內部用來辨別呼叫它的 application**)**，大部份的 API function 在使用時均需傳入此一 instance number 做爲 device driver 驗證呼叫者之用。

(3) 因爲 VP-894 device driver 本身是一個 dynamic linking library 而不是一個 task，因此它不像其它在 Windows 下執行的 applications 一樣，可以直接使用到一些 Windows 的 resource，所以它必須借用呼叫它的 application 的 handle 向 Windows 要求服務。通常第一個 application 是被用來替 VP-894 device driver 維護這些 resources。如果這個幫 VP-894 device driver 維護 resources 的 application 結束 **(**呼叫 Close894()**)**，VP-894 會送出一個 EVT_SET_ CONTROL 的事件通知其他在系統內使用 VP-894 設備的 instance 接手管理 device driver 的 Windows resources **(**呼叫 AcceptControl()來完成**)**，所以 application 必須留意這個 event,並做合適的回應， 否則 VP-894 device driver 將無法再正常工作。在使用 setup diskette install 完 VP-894 API，在 Windows 的 system 子目錄下會有一個 VP894.DLL，這個就是構成 VP-894 驅動程式的核心，application 只要在程式內連結 VP-894 API 目錄下 VP894.L1B 這個 import library，就可使用 VP-894 此設備的功能。除非本文件中特別提出指明，否則有關 VP-894 API 的一切定義在 DOS 及 Windows 下均同，請一併參考第四章 DOS API 的說明。

6-1、與 DOS 版 API 差異說明

6-1-1、增加的 Event Types

1.EVT_SET_CONTROL

就如前所述的，當 application 在結束時呼叫 Close894()，device driver 會 post 這個 event，在系統內其它有用到 VP-894 設備的 application 必須呼叫 AcceptControl()funoction，否則 VP894.DLL 將無法再正常地對 application 提供服務。

2.EVT_REQUEST_ADAPTER

當系統內的一個 application 如果需要使用更多的 VP-894 設備時可以呼叫 Request- Adapter()向其它 application 發出這個事件，其它 application 在收到這個 event，如果可能的話，可以呼叫 ReleaseAdapter()釋放已不再使用的 VP-894 設備。

3.EVT_FREE_ADAPTER

當 application 收到這個 event，表示其它 application 已釋放一些 VP-894 設備，如果需要的話，可以呼叫 AddAdapter()把這些閒置的設備納入使用。當一個 application 結束執行時，無須使用 ReleaseAdapter()釋放使用的 VP-894 設備，但必須呼叫 Close894()，以便 device driver 完成必要的 house keeping 工作。

以上三個 event 的 Issuer 無定義，Event Data 在 EVT_FREE_ADAPTER 及 EUT_REQUEST_ ADAPTER 中是指釋放或要求的卡號，而對於 EVT_SET_CONTROL 是無定義。

6-1-2 傳回錯誤碼

除了原來在 DOS API 中的定義外，Windows API 增加了以下的錯誤碼：

1.ERR894_INSTALL_NONE
找不到 VP894.DLL

2.ERR894_IRQ_NOT_FOUND
找不到 VP894 的 IRQ 設定

3.INCOMPATIBLE_FIRMWARE
系統內有不相容的 firmware 版本

4.ERR894_WINDOWS_ERROR_ALLOC_SELECTOR
無法向 Windows 配置 free selector 使用

5.ERR894_WINDOWS_ERROR_SET_SELECTOR
無法設定 selector base

6.ERR894_NO_MORE_ADAPTER
沒有閒置的 VP-894 設備可使用

7.ERR894_BAD_WINDOWS_HANDLE
傳給 VP-894 API 的 Windows handle 是無效的

8.ERR894_REGISTER_MESSAGE
無法向 Windows 註冊 VP894 的 message

9.ERR894_SET_TIMER
無法 install 系統 timer

10.ERR894_BAD_FILE_NAME
指定的聲音檔案不存在

11.ERR894_RECORD_FILE
無法 create 所指定的聲音檔

12.ERR894_NO_MORE_INSTANCE
無法建立更多的 VP894 instance

13.ERR894_INVALID_OWNER

呼叫 API function 的 application 並未擁有所指定的 channel

14.ERR894_INVALID_HANDLE

無效的 VP-894 instance 號碼

15.ERR894_UNKNOWN_ERROR

不明原因的錯誤

6-1-3、Event 的傳遞方式

VP-894 設備的 event 是直接透過 application 的 message queue 來傳遞，在 VP894.DLL POST 階段會向 Windows 註冊一個 VP894 的 message， application 可呼叫 Query894() 獲得這個 message 的 identifier。在 application 由 message queue 所得到的 message，屬於 VP-894 的 message 它的 word 參數存放的是 VP-894 的 event type，double word 參數存放的是這個 event 資料的位置 **(**此 event 資料爲一個 typeEvent 的結構，被定義在 API894.H 中**)**。由於每一個 Windows applocation 預設的 message queue 大小只有 8，爲了避免在多線同時工作時 application 來不及處理完所有的 message，而造成 message queue overflow 的狀況，請在 create window 之前使用 SetMessageQueue() 增大 message queue 的空間。

6-2、功能函數介紹

1.Init894(HWND hWnd，LPINT Instance，LPINT OwnerAdapter，DWORD Owner)

目的：向 VP894 device driver 要求配置設備，並做初始化

參數：

hWnd：要求配置設備的 Windows´handle

Instance：爲一整數型態的指標，用來存放 VP894 device driver 返回給 application 的 instance identifier，這個 instance identifier 是 application 隨後呼叫 VP-894 API 其它 function 時，device driver 用來驗證呼叫者合法性之用

OwnerAdapter：爲一含有 16 個元素的整數陣列啓始位置用來傳回配置給 application VP-894 設備的資料，這 16 個元素中任何一個內容不爲 0 代表該卡號之 VP-894 已配置給此呼叫的 application，0 則否。例如：OwnerAdapter[1]， OwnerAdpter[5]，OwnerAdapter[10]中若爲非 0，則表示 application 擁有卡號 1，5 及 10 的使用權

Owner：指明要求配置的卡號。在 API894.H 中定義有 OWNER_0~OWNER_31 及 OWNER_ALL 常數代表各 VP-894 之卡號(目前僅 OWNER_0~OWNER_15 有效) 可用 operator bitwise OR 結合數個不同卡號。OWNER_ALL 是指明所有目前閒置中的 VP-894 設備

傳回值：0 代表成功，否則爲錯誤碼

Event Type：無

2.Close894(int inst)

目的：application 在結束前用來通知 device driver 收回配置給它的設備，並作必要的 house keeping 工作

參數：

inst：這是先前呼叫 Init894()時，由 device driver 返回用來辨識此 application 的 instance identifier

傳回值：0 代表成功，否則爲錯誤碼

Event type：無

3.OpenVoiceFile(LPSTR fileName，long startpos，UINT fuMode)

目的：開啓檔案

參數：

fileName：要開啓的檔案路徑名稱

startpos：檔案開啓後，檔案指標要設定的位置

fuMode：指明檔案的開啓方式及屬性。本函式是呼叫 Windows API 中的 OpenFile()來開啓檔案，因此本參數的定義即同於 OpenFile()第三個參數的定義，請參考 Windows SDK 的說明

傳回值：同於 OpenFile()的傳回值

Event Type：無

4.CloseVoiceFile(int inst，int ChNum)

目的：關閉檔案

參數：

inst：VP-894 的 instance identifier

chNum：先前指定錄音或放音的 channel 號碼

傳回值：0 代表成功，否則爲錯誤碼

Event Type：無

5.AcceptControl(int inst)

目的：承認 VP-894 內部 Windows resource 的管理權

參數：

inst：VP-894 的 instance identifier

傳回值：0 代表成功，否則表示其它 application 已代爲接管。其實 application 可以無須理會本 function 的傳回值，因爲成功與否均不影響 application 的執行方式，但是 application 在收到 EVT_SET_ CONTROL 時一定要呼叫本函式

Event Type:無

6.Query894(LPINF0894 lpInfo，LPINT Installed)

目的：獲得有關 VP-894 設備的相關訊息

參數：

lpInfo：本參數是用來存放由 device driver 返回的 VP-894 設備訊息，它是一個指到定義在 API894.H 中的結構指標 --INFO894，以下是這個結構中各個成員的說明：

verMajor -- VP894.DLL 的主版本號碼
verMinor -- VP894.DLL 的次版本號碼
Msg -- VP894 向 Windows 註冊的 message ID，application 利用這個 ID 來分辨所收到的 message 是否來自於 VP-894
IRQ -- P894 的 IRQ 號碼設定
SRAM -- VP894 shared buffer 的位置

Installed：爲一含有 16 個元素的整數陣列啓始位置，用來傳回系統內所有的 VP894 設備。安裝組態這 16 個元素中任何一個內容不爲 0 時，表示系統內有 install 相對卡號的 VP894 設備

傳回值：無

Event Type：無

7.RequestAdapter(int Instance，DWORD owner)

目的：向 VP894.DLL 要求配置 VP894 設備

參數：

Instance:VP894 的 instance identifier

owner:同於 Init894()中的參數 Owner

傳回值:0 代表成功，否則爲錯誤碼

Event Type:無

8.ReleaseAdapter(int inst，DWORD owner)

目的：釋放 VP-894 設備

參數：

inst：VP894 的 instance identifier

owner：要釋放的設備，定義同於 Init894()中的參數 owner

傳回值：0 代表成功，否則爲錯誤碼

Event Type：無

9.GetFreeAdapter(LPINT freeAdapter)

目的：獲得目前系統中閒置的 VP894 設備訊息

參數：

freeAdapter：是一含有 16 個元素的整數陣列的啓始位置，用來存放由 device driver 傳回系統內閒 置的 VP894 設備資料，如果某個元素內容爲非 0 的值，代表相對應的卡號設備是尚未被配置的

傳回值：爲一個 DWORD 其定義同於 Init894()中的參數 owner，可直接用來設定呼叫 RequestAdapter()或 AddAdpter()中的第二個參數

Event Type：無

10.AddAdapter(int inst，DWORD owner)

目的：向 VP894.DLL 要求增加配置 VP894 設備

參數：

inst：VP894 的 instance identifier

owne：要增加的設備，定義同於 Request- Adapter()中的參數 owner，但指定的設備編號(卡號）必須是目前閒置的，要獲知有那些 VP-894 設備成功地配置給呼叫的 application，必須利用 function Get894Owner()

傳回值：0 代表成功，否則爲錯誤碼

Event Type：無

11.GetIncompatible(LPSTR incompatible)

目的：取得不相容 firmware 版本的 VP894 設備的資料

參數：

incompatible：爲一含有 16 個元素的字元陣列的啓始位置，用來存放由 device driver 返回系統內 firmware 版本不相容的 VP894 設備訊息，任一元素內容如爲非 0 的值，代表相對卡號的設備爲不相容的

傳回值：0 代表沒有不相容的設備，非 0 則否

Event Type：無

12.Get894Version(int AdptrNum)

目的：取得指定卡號的 firmware 版本號碼

參數：

AdptrNum：VP894 的卡號

傳回值：爲一整數型態，HIBYTE 是主版本號碼 LOBYTE 爲次版本號碼

Event Type：無

13.Valid894Handle(int inst)

目的：驗證所指定的 instance handle 是否是一合法的 VP894 instance identifier

參數：

inst：要檢查的 instance handle

傳回值：0 代表這個 instance handle 是無效的，否則是一個有效 VP894 instance identifier

14.Test894Owner(int inst，int AdptrNum)

目的：驗證指定的 VP894 instance 是否擁有所指定卡號的設備

參數：

inst：VP894 的 instance identifier

AdptrNum：要驗證的設備卡號

傳回值：0 代表此 instance 沒有所指定卡號設備的使用權，非 0 則代表有所指定卡號設備的使用權

Event Type：無

15.Get894Owner(int inst，LPINT installed)

目的：取得所指定 VP894 instance 擁有的 VP894 設備資訊

參數：

inst：VP894 的 instance identifier

installed：爲一含有 16 個元素的整數陣列啓始位置，用來傳回指定 VP894 instance handle 所擁有的 VP894 設備。這 16 個元素中任何一個內容不爲 0 時，表示此 VP894 instance 擁有該相對卡號的設備

傳回值：爲一個 DWORD，其定義同於 Init894()中的參數 Owner，任一 bit 爲 1 時，代表所指定 VP894 instance 擁有該 bit 代表的 VP894 設備

Event Type：無

七、Visual Basic 語音介面程式 (WINDOWS 版) 說明

此介面程式可在 Windows 3.1 或 Windows 95 上執行，並且與 Visual Basic 2.00 (含) 以上相容。請注意由於 VP894CC.VBX 只是 VP894.DLL 的一個 function wrapper，因此在 Windows\System 目錄下必須同時安裝這兩個 dynamic link libraries 才能開始使用。

7-1、VP-894 的 Custom Control 說明

要使用 VP894CC 這個 custom control，你必須拉下 VB 的 File 選單，然後選擇 Add File..這個功能選項，在 Add File 的 dialog box 選擇 P894CC.VBX。　如果以上的步驟均正確無誤，在 VB 的 ToolBox 中即會顯示這個代表 VP894 custom control 的 button bitmap，另外，你也必須使用 Add File 這個功能選項加 VP894INC.BAS 這個 code module，此 module file 中定義了與 VP894CC 相關的 global constant，user-defined data type 及 API function prototype declaration。被放置到 Form 的每一個 VP894 control 各代表著一片 VP894 adapter，並以此兩種圖案代表該 control 所象徵的 VP894 adapter 目前的狀態；在 design time 時前者表示該 VP894 設備是 free，後者表示該 VP894 設備不存在或已配置給其他 Task；在 run time 時，前者表示該 VP894 設備已配置給擁有該 VP894CC 的 application，後者同於 design time 的定義(在 run time 時，必須 VP894CC 的 environment property 中的 Visible At Runtime 爲 TRUE 時，才會被顯示，請參考後面有關 environment 的說明)。接著我們將分別以下面數節來說明 VP894CC 的特點及其使用方法。

7-2、VP-894CC.VBX 的功能屬性(Properties)

在說明 VP894CC 各個 property 時，將仿照 VB 的手冊，以下面的圖案代表此 property 在 design time 及 run time 時的 accessibility：

● **read / write**

○ **read only**

○ **write only**

○ **not available**

7-2-1、Status

design time ○

run time ○

Status 是一個代表此 VP894 control 目前狀態的 integer，它的值的範圍是由 0-2。 2 表示此 control 所代表的 VP894 adapter 已被其他 Task 所配置使用；1 表示此 control 所代表的 VP894 adapter 未 installed 於系統內；0 在 design time 時，表示此 control 所代表的 VP894 adapter 是 free，在 run time 時，則表示此 control 所代表的 VP894 adapter 已被配置給此 control 使用。**(**亦即此 control 可使用它所代表的 VP894 adapter 的所有功能**)**

7-2-2、AdapterNumber

design time ●

run time ○

AdapterNumber 是一個 integer，其值的內容是此 control 所代表的 VP894 卡號，在 run time 時 VP-894 custom control 即是利用此值向 VP894.DLL 要求配置 VP894 設備。

7-2-3、CtrlParam

design time ●

run time ○

當你在 design time 時，如果在 properties box 中選擇了本 property，VP894 custom control 會 popup 一個 dialog box，顯示此 control 所代表的 VP894 device 在 run time 初始化時的 default channel control parameters **(**此 channel control parameter 的各項資料說明， 請參閱第三章 3-3 控制參數定義說明**)**，你可以依照需要修改其中的內容。此 property 僅在 VP894CC 初始化時設定其 channel control parameter，如果你必須於執行時期動態地改 變其設定請使用 VP894CC 的 API function- SetCtrlParamProp()。
(請參閱以下 VP894CC API function 的說明**)**

7-2-4、FlowEditor

design time ●

run time ○

此 property 是控制 VP894CC 執行流程的核心樞紐。在 design time 時，如果你在 properties box 選擇了本 property，VP894 custom control 會 pop up 一個 dialog box，你可以在這個 dialog box 上增加、刪除、編輯或修改這個 VP894CC 的 telecommunication flow control。在 VP894CC 的 flow control 中，我們是以 process 作爲一個執行單元，目前的版本中共定義了以下 14 種不同的 process types：
1.Query 2.Record 3.Call 4.Stop Channel
5.Set Control Parameter 6.Interlink 7.Flash 8.Pick Up
9.Hang Up 10.Ringing 11.Local Phone Picked Up
12.Local Phone Hung Up 13.Remote Hang Up
14.Learning Pulse Success。這些 processes 各有不同的 properties 定義此 process 如何被執行，如果粗略的區分，可以將這些 properties 概分爲兩類:**(**一**)**共通的 property **(**二**)**不同 process 所特有的 property。以下是共通的 properties 的說明：

7-2-4-1、流程單元(Process)之共通屬性(Common Properties)

1.ID

代表此 process 的 ID code，這個 ID code 也被用來在任何必須輸入 next step 的 process properties 的 edit box 中，指明此 process 的下一執行步驟，例如共通 properties 中的 Default Next Step。

2.Type

本 process 的 process type

3.Name

這個 process 的名稱。本項 property 僅是供 application programmer 輔助記憶明瞭本 process 的工作內容或用途，你可輸入任何 ASCII 字元(不包含 null)

4.Default Next Step

用來指明此 process 正常結束後，應該執行下一 process 的 ID code。如果此欄爲 blank 或-1，代表本 process 執行完畢後，進入 idle 狀態 (如果此 process 的 properties 中有其它 next step 的 properties，則所謂 process 的正常結 束因不同的 process type 而有不同的解釋，例如 process Query、call 等，請參閱各個不同 process type 的說明)

5.Fire Event(僅 Ringing，Local Phone Picked Up，Local Phone Hung Up，Remote Hang Up 及 Learning Pulse Success 才具 有本 property)

用來指明當 VP894CC 偵測到 Ringing，Local Phone Picked up，Local Phone Hung Up，Remote Hang Up 或 Pulse Learning Success 時，是否要 Fire VB Event 通知 VB application **(**Ringing， PhonePickedUp，PhoneHungUp，RemoteHangUp 或 LearnPulseSuccess**)**

6.Fire Pre-Execution Event(Ringing，Local Phone Picked Up *Local Phone Hung Up，Remote Hang Up 及 Pulse Learning Success 無此 property)*

設定 VP894CC 於執行本 process 前，是否要 Fire VB Event-PreExecuted 通知 VB application。

7.Fire Post-Execution Event(Ringing，Local Phone Picked Up ，Local Phone Hung Up， Remote Hang Up 及 Pulse Learning Success 無此 property)

設定 VP894CC 在執行完本 process 後，是否要 Fire VB Event - Post Executedxxx **(**xxx 代表不同的 process types**)** 通知 VB application。

7-2-4-2、特定 Process 專用屬性(Exclusive Properties)

在接下來的文件內容中，我們將依各種不同的 process type 分別說明其使用用途及 process-depended 的 properties：

1.Query

本 process 是用來播放一個或數個 voice files，並接收 remote 的電話按鍵輸入。你如果用 Flow Editor 的"New" push- button 產生此種 process，在 Windows 的桌面上會多顯示兩個 dialog box - Prompt List 和 Next Step Table。Prompt List 是用來設定這個 Query Process 要放音的檔案，檔案的播放順序是依照 Prompt List 中位置順序而定，你如果要改變目前的播放順序，可以先將 mouse 指在要改變位置的聲音檔上，按下 mouse 的左鍵，然後拖移 mouse，直到該聲音檔移至要放置的位址，才放開左鍵；Next Step Table 是用來設定此 process 若接收到其 property - Exit DTMF 中，所指定的 DTMF 碼時，而必須執行的步驟及 remote 端超過等待輸入時間時所必須執行的 process。Query 除了上述兩個 dialog box 中的 property 及 process 的 common properties 外，尚有下列三種 properties：

① Exit DTMF

設定以何者電話按鍵輸入提前終止此 Query 等待接收一串按鍵輸入，例如，假設此 Query 要接收 6 個 0～9 的數字按鍵，如果 Exit DTMF 設定了'*' 或'#'當作終止鍵，則當 remote 端輸入此兩個按鍵中的任何一個，不論 remote 端先前是否已輸入足夠的 DTMF 碼，本 process 均會結束，並且依照 Next Step Table DTMF * 或 DTMF # 中設定的 process ID，執行下一個 process。你可以直接在 Exit DTMF 的 edit control 中輸入你的設定，或者使用旁邊的電話圖案。 **(**電話圖案上的 Keypad 如果是 pressed down，代表該鍵是 enable 的**)**

② Accepted DTMF

設定那些電話按鍵是可接受的。你可以直接在 Accepted DTMF 的 edit control 中輸入你的選擇，或者使用旁邊的電話圖案。

③ DTMF Amount

指明此 Query 要接收電話按鍵的總數。

附註：在 Prompt List 中的檔案路徑由 VP894CC property Environment 中的 Prompt Directory 所指定，你如果必須播放不同路徑的 voice file 或在 Run Time 時，設定或更改 Prompt List 的內容，可使用 VP894CC 的 API function SetQueryProp()；此 process 的 Default Next Step 的定義是在執行結束，沒有收到任何 Exit DTMF，而且在 Time out 前接收到 DTMF Amount 所指明的按鍵輸入個數。

2.Record

Record 是用來作錄音。你如果用 Flow Editor 的 "New"push-button 產生此種 process，在 windows 的桌面上會另外再顯示一個 dialog box - Next Step Table **(**有關此 dialog box 的說明，請參考 Query 的描述**)**。Record 的 process- depended properties，除了 Next Step Table，還包含下列 3 項：

① Exit DTMF

請參閱 Query 中 Exit DTMF 的說明。

② Length

指定錄音的長度。

③ File Name

指定錄音的檔案名稱。

附註：File Name 所指明的錄音檔的路徑由 VP894CC property - Environment 中的 Record Directory 所指定;如果必須使用不同的路徑或於 Run Time 時，設定或修改 Record 的 properties，可使用 VP894CC 的 API function - SetRecordProp()；此 process 的 Default Next Step 的定義是在工作結束後，沒有接收到任何 Exit DTMF。

3.Call

Call process 是用來 call 外線,分機電話或 Beeper。你如果用 Flow Editor 的"New" push-button 產生此種 process，在 windows 的桌面上會另外再顯示一個 ialog box - More Call Properties。Call 的 process-depended properties 包含下列數項：

① Called Type
指定此 process 要 call 外線，分機電話或 beeper。

② Number
指定要 call 的號碼。

③ Message
如果 called type 設定爲 beeper，此項爲 beeper 撥通後，要傳送的 message。**(**如果有 prolog 或 epilog digit， 例如'#'必須包含在其中**)**

④ Called Failure Next Step
指明 call 如果失敗，必須執行的 process ID。

⑤ Don't Execute CPM
Don't Wait Dial Tone
Detect Invalid Number
Detect 1st User-Defined Signal
Detect 2nd User-Defined Signal
如果 Called Type 爲外線或分機電話，這幾項是指定撥號的模式。**(**請參考本手冊第四章 C 語言介面程式中 CallRemote()的描述**)**

附註：Default Next Step 是此 process 執行成功**(**被呼叫者接聽或 beeper 撥號成功**)**；如果必須於 run time 設定或修改 call 的 properties， 可使用 VP894CC API function- SetCallProp()。

4.Stop Channel

終止目前執行的工作。

5.Set Control Parameter

此 process type 是用來在 run time 時， 改變某些 channel 的控制參數
(VP894CC property-CtrlParam 是用於 run time 的初始化**)**。有關此 process 的
process- depended properties 的說明，請參考 CtrlParam。

附註：此 process type 在 run time 是屬於靜態設定**(**即其 process-depended properties 是於 design time 已指定好的**)**，如果你必須在 run time 時動態設定 channel 的控制參數，請使用 VP894CC API function SetCtrlParamProp()。

6.Interlink

此 process type 是用來設定相鄰奇偶成對的兩個 channel 的電話介面是否相互連接。

7.Flash

此 process type 是使 VP894 做一個
on hook-off hook 的動作 **(**本動作在交換機內是爲了轉接的目的**)**，其 on hook-off hook 之間的 duration 由 Flash Time 決定。

8.Pick Up

本 process type 是使 VP894 佔線。

9.Hang Up

本 process type 是使 VP894 掛斷電話。

10.Ringing

本 process type 僅是一個 notification，它並沒有執行任何有關 VP894 實際的硬體動作。在 VP894 偵測到有電話 call in 時，VP894CC 根據此 process 的 property-Fire Event，決定是否要 Fire VB Event-Ringing，通知 VB application，並且依照 Default Next Step 的設定，執行下一步驟。

11.Local Phone Picked Up

本 process 是一個 notification。在 VP894 偵測到串接電話的話筒被拿起時， VP894CC 根據此 process 的 property- Fire Event，決定是否要 Fire VB Event-PhonePickedUp，通知 VB application，並且依照 Default Next Step 的設定，執行下一步驟。

12.Local Phone Hung Up

本 process 是一個 notification。在 VP894 偵測到串接電話的話筒被掛上時， VP894CC 根據此 process 的 property- Fire Event，決定是否要 Fire VB Event-PhoneHungUp，通知 VB application，並且依照 Default Next Step 的設定，執行下一個步驟。

13.Remote Hang Up

本 process 是一個 notification。在 VP894 偵測到遠端掛上電話時，VP894CC 根據此 process 的 property-Fire Event，判斷是否要 Fire VB Event-RemoteHangUp，通知 VB application，並且依照 Default Next Step 的設定，執行下一個步驟。

14.Learning Pulse Success

本 process 是一個 notification。Ctrl- Param 中的 Learn Pulse 必須 checked，你才可在 Flow Editor 產生此種 processtype。在 VP894 成功完成 pulse learning 後，VP894CC 根據此 process 的 property-Fire Event，決定是否要 Fire VB Event-LearnPulseSuccess 通知 VB application，並且依照 Default Next Step 的設定，執行下一個步驟。**(**有關 pulse learning 程序的詳細說明，請參閱第四章 4-2 節內相關說明**)**

7-2-5、Environment

design time ●

run time ○

你如果在 properties box 中選擇了這個 property，VP894CC 會 popup 一個 dialog box 顯示此 VP894 control 的 environment options。這些 environment options 包含下列幾項：

1.Prompt Directory

設定提示語的儲存路徑。

2.Record Directory

設定錄音檔案的儲存路徑。

3.Auto-Executed Step After Enable

指定當 VP894CC 的 property-Active 被設爲 TRUE 時，VP894 必須自動執行的 process ID。在 run time 時，如須動態更改自動執行的 process ID 可使用 VP894CC 的 property- AutoExecProc。

4.Visible At Runtime

設定在執行時期是否要顯示此 VP894 control 的 bitmap 及狀態。

7-2-6、Active

design time ○

run time ●

本 property 是一個 Boolean integer array，共有 4 個成員，分別用來代表及設定此 control 的四個 VP894 channels 的工作狀態。當 Active 設定爲 False 時，其對應的 VP894 channel 即進入 idle 的狀態；反之，若 Active 設定爲 TRUE，且 Active 狀態是由 False 改爲 TRUE， AutoExecProc 中若爲一有效的 process ID， VP894 即由 AutoExecProc 所指定的 process 開始執行。

7-2-7、AutoExecProc

design time ○

run time ●

本 property 是一個 integer array，共有 4 個成員，分別用來代表及設定此 control 的四個 VP894 channels 的自動執行 process ID。**(**請一併參閱 VP894CC property-Active 的說明**)**

7-2-8 CPMParam

design time ●

run time ○

你如果在 properties box 中選擇了這個 property，VP894CC 會 pop up 一個 dialog box，顯示此 VP894 Control 在 run time 初始化時的 default CPM parameters **(**此 CPM parameters 的各項定義，請參考第五章 5-3 節中有關 GetCPMParam()的說明**)**，你可以依照需要更改其中的內容。

7-2-9、HungUpParam

design time ●

run time ○

你如果在 properties box 中選擇了這個 property，VP894CC 會 pop up 一個 dialog box，顯示此 VP894 control 在 run time 初始化時 的 default 偵測遠端掛斷的參數 **(**偵測遠端掛斷的參數的各項定義，請參考第五章 5-3 節中 GetHungUpParam() 的說明**)**，你可以依照需要，修改其中的內容。

7-2-10、PromptDir

design time ◯

run time ◯

這個 property 是一個字串，儲存該 VP-894CC 預設的提示語的所在路徑名稱。
(其內容同於 Environment 中的 Prompt Directory**)**

7-2-11、RecordDir

design time ◯

run time ◯

這個 property 是一個字串，儲存該 VP-894CC 預設的錄音檔案的輸出路徑名稱。
(其內容同於 Environment 中的 Record Directory**)**

7-3、VP894CC.VBX 的事件(Events)說明

所有的 VP894CC events 均會傳入 ChNum 及 idCurProc 這兩個參數,前者代表 fire 這個 event 的 channel number,這個 channel number 的定義不同於其它 API (DOS-C，Clipper，Windows)或第五章 5-1 節中的定義，它的值的範圍是由 0 至 3，分別代表此 VP894 control 的 0～3 ports，你如果要 MAP 此 ChNum 至實際的 channel number，可以此 VP894 control 的 AdapterNumber 乘以 4，加上 ChNum 即得；後者表示 fire 這個 event 時，VP894CC 正執行的 process。在 design time 時，使用 Flow Editor 是設計 run time 的靜態 process，亦即這些 process 的內容及相互間的關聯性是預先訂製好的，在 application 的 life time 是固定不變的。你如果在程式的執行過程中，必須依不同的外在環境所改變的狀況，而動態地改變 VP894CC 的工作流程，可以利用 VP894CC 的 API function 及改變 idCurProc 這個參數的內容，也就是說你如果在 Exit event procedure 之前更改了 idCurProc，當控制權回到 VP894CC (亦即 Exit event procedure)，VP894CC 即會執行 idCurProc 內所指定的 process(其中 event Detect-DTMF 及 DetectPulse 爲例外狀況，你無法在這兩個 event procedure 中以更改 idCurProc 來改變 VP894CC 的執行程序)。會影響 VP894CC execution flow 的因素有三：依照其優先順序分別爲

1.VP894CC property Active 由 False 改爲 True 自動執行的程序
2.在 run time 改變 idCurProc
3.用 Flow Editor 設計的靜態 process 內 Next Step 的設定

因此假設如果以上三個狀況同時發生，VP894CC 會執行 AutoExecProc 內所指定的 process。某些 events 除了傳入以上兩個參數外，還會傳入其他的參數，下面在描述各個 event 時，會一併提出說明。

1.DetectDTMF

你如果在 CtrlParam 中 checked MonitorDTMF **(**即 enable monitor DTMF 的功能**)**，當 VP894CC 偵測到有 DTMF 輸入時，即會 fire 這個 event。

其它參數：

① DTMF

VP894CC 偵測到的 DTMF 碼

2.PhoneHungUp

當 VP894CC 偵測到串接的電話話筒被掛上時，而且 process-Local Phone Hung Up 中的 property-Fire Event 爲 TRUE，即會 fire 這個 event。

其它參數：

① idPreProc

VP894CC 執行本 process 前，執行的 process ID。

3.PhonePickedUp

當 VP894CC 偵測到串接的電話話筒被拿起，而且 process-Local Phone Picked Up 中的 property- Fire Event 爲 TRUE 時，即會 fire 這個 event。

其它參數：

① idPreProc

VP894CC 執行本 process 前，所執行的 process ID。

4.PostExecutedCall

發生在 call process 執行完後，且
call process property-Fire Post-Execution Event 設定爲 TRUE。

其它參數：

① Ret894Evt

VP894.DLL 傳回的 event type，可能的 event type 有 EVT_NO_DIAL_TONE，EVT_CPM_COMPLETE
(有關 VP894 的 event type 說明，請參閱第三章 3-1 節**)**

② CallResult

如果 Ret894Evt 爲 EVT_CPM_COMPLETE， CallResult 可能爲下列任一結果－
CPMR_NO_ANSWER、CPMR_BUSY、CPMR_INVALID_NUM、CPMR_USER_DEFINED1、CPMR_USER_DEFINED2、CPMR_NO_SIGNAL、CPMR_ANSWER、CPMR_NO_ RINGBACK、CPMR_CALL_BEEPER_SUCCESS **(**請參考 VP894 第三章 3-1 節中 EVT_CPM_COMPLETE**)**，如果 Ret894Evt 爲 EVT_NO_DIAL_TONE，則 Call Result 無定義。

5.PostExecutedFlash

發生在 Flash process 執行完後，且
Flash process property-Fire Post-Execution Event 設定爲 TRUE。

6.PostExecutedHangUp

發生在 Hang Up process 執行完後，且
Hang Up process property-Fire Post-Execution Event 設定爲 True。

7.PostExecutedInterlink

發生在 Interlink process 執行完後，且
Interlink process property-Fire Post-Execution Event 設定爲 True。

8.PostExecutedPickUp

發生在 Pick Up process 執行完成後，且
Pick Up process property-Fire Post-Executed Event 設定爲 True。

9.PostExecutedQuery

發生在 Query process 執行完成後，且
Query process property-Fire Post-Execution Event 設定爲 True。

其它參數：

① Ret894Evt
EVT_EOP_NORMAL，EVT_DTMF_INTERCEPT，或
EVT_TIME_OUT **(**請參閱第三章 3-1 節中相關說明**)**

② ExitDTMF
如果 Ret894Evt 爲 EVT_DTMF_INTERCEPT，則本參數內包含終止此 Query process 的 DTMF 輸入，否則無定義

③ RxDTMFs
此 Query process 所接收到的 DTMF 輸入

10.PostExecutedRecord

發生在 Record process 執行完成後，且
Record process property-Fire Post-Execution Event 設定爲 True。

其它參數：

① Ret894Evt
EVT_EOP_NORMAL 或 EVT_DTMF_INTERCEPT

② ExitDTMF
如果 Ret894Evt 爲 EVT_DTMF_INTERCEPT，則本參數內包含終止此 Record process 的 DTMF 輸入，否則無定義。

11.PostExecutedSetCtrlParam

發生於 Set Control Parameter process 執行完成後，且 Set Control Parameter process property- Fire Post-Execution Event 設定爲 True。

12.PostExecutedStopCh

發生於 Stop Channel process 執行完成後，且
Stop Channel process property-Fire Post-Execution Event 設定爲 True。

13.PreExecuted

發生在任何 process 被執行前，且該 process 的 property-Fire Pre-Execution Event 設定爲 True。

14.Ringing

當 VP894CC 偵測到有電話 call in，而且 process-Ringing 中的 property-Fire Event 爲 True 時，即會 fire 這個 event。

其它參數：

① idPreProc

VP894CC 執行本 process 前，執行的 process ID。

15.DetectPulse

你如果在 CtrlParam 中 checked "Monitor Pulse"，當 VP894CC 偵測到有 pulse 撥號輸入時，即會 fire 這個 event。

其它參數：

① Pulse

VP894CC 偵測到的 pulse 撥號碼

16.LearnPulseSuccess

你如果在 CtrlParam 中 checked "Learn Pulse"，當 VP894CC 成功完成 pulse learning 的程序，即會 fire 這個 event。

其它參數：

① idPreProc

VP894CC 執行本 process 前，執行的 process ID

17.RemoteHangUp

當 VP894CC 偵測到遠端掛斷電話,而且 process -Remote Hang Up 中的 property-FireEvent 爲 TRUE 時，即會 fire 這個 event。

其它參數：

① idPreProc
VP894CC 執行本 process 前，執行的 process ID

② Ret894Evt
EVT_LINE_SILENT，EVT_LINE_ROARING_REMOTE _HANG_UP，EVT_LINE_VOLT_REVERSE_REMOTE_ HANG_UP 或 EVT_LINE_BUSY_REMOTE_HANG_UP **(**請參考第三章 3-1 節中這些 event type 的解釋**)**

7-4、API 程式說明

7-4-1、功能函數說明

VP894CC 另外提供一組 API function (function 的 prototype 宣告在 VP894INC.BAS 中),目的是使 VB 的 application 可以在 run time 時， 動態地更改 process 的 property，以下是這些 function 的說明：

1.GetQueryProp(VP894，idProc，QueryInfo)

取得所指定的 process 的 properties information。

輸入參數：

① VP894
VP894CC 的 control object

② idProc
要被讀取 properties information process ID

③ QueryInfo
這是一個 User-Defined type-QueryProp 的變數 (variable) 用來儲存要讀取的 properties information

2.SetQueryProp(VP894，ChNum，PromptList，QueryInfo)

設定下一個執行 process 爲 Query 及指定 Query 的 properties。

輸入參數：

① VP894
VP894CC 的 control object

② ChNum
被設定更改的 channel number。此 channel number 爲此 VP894CC 上的 channel number(0～3)，不同於第四章 4-1 節中的定義

③ PromptList
此項輸入參數可爲 Null pointer 或一字串指明此 Query 要放音的檔案。如果設定爲 Null pointer，則此 Query process 僅接收遠端按鍵輸入，而不放音；如果要放音的檔案超過一個以上，每一個檔案的路徑全名必須以 str$(0)分隔開。例如，假設你有以下三個檔案要播放— C:\PATH1\VOICE1，D:\PATH2\VOICE2，E:\PATH3\VOICE3，PromptList 必須如下設定:PromptList = "C:\PATH1\VOICE1"+ Str$(0)+"D:\PATH2\VOICE2"\Str$(0)+" E:\PATH3\VOICE3"+Str$(0) 如果檔案名稱內未包含路徑，VP894CC 以 Environment 中的 Prompt Directory 的路徑，當作該檔案的路徑。

④ QueryInfo
這是一個 User-Defined type-QueryProp 的變數，指明要更改的 Query properties 內容

3.GetRecordProp(VP894，idProc， RecordInfo)

取得所指定的 process 的 properties information。

輸入參數：

① VP894
VP894CC 的 Control object

② idProc
要被讀取 properties information 的 process ID

③ RecordInfo
這是一個 User-Defined type-RecordProp 的變數，用來儲存要讀取的 properties information

4.SetRecordProp(VP894，ChNum，FileName，RecordInfo)

設定下一個執行 process 爲 Record 及指定 Record 的 properties。

輸入參數：

① VP894
VP894CC 的 control object

② ChNum
被設定更改的 channel Number

③ FileName
指定錄音的檔案名稱

④ RecordInfo
這是一個 User-Defined type-RecordProp 的變數，指明要更改的 Record properties 內容

5.GetCallProp(VP894，idProc，CallInfo)

取得所指定的 process 的 properties information。

輸入參數：

① VP894
VP894CC 的 control object

② idProc
要被讀取 properties information 的 process ID

③ CallInfo
這是一個 User-Defined type-CallProp 的變數，用來儲存要讀取的 properties information

6.SetCallProp(VP894，ChNum，CalledNumber，CallInfo)

設定下一個執行 process 爲 call 及指定 call 的 properties。

輸入參數：

① VP894
VP894CC 的 control object

② ChNum
被設定更改的 channel number

③ CalledNumber
指定被呼叫號碼的字串，如果 called type 爲 beeper，則此項參數包含被呼叫的 beeper 號碼及要傳遞的 message，中間以一 Str$(0)分隔

④ CallInfo
這是一個 User-Defined type-CallProp 的變數，指明要更改的 call properties 內容

7.GetCtrlParamProp(VP894，idProc， CtrlParam)

取得所指定的 process 的 properties information。

輸入參數：

① VP894
VP894CC 的 control object

② idProc
要被讀取 properties information 的 process ID

③ CtrlParam
這是一個 User-Defined type- SetCtrlParamProp 的變數，用來儲存要讀取的 properties information。

8.SetCtrlParamProp(VP894，ChNum， CtrlParam)

設定下一個執行 process 爲 Set Control Parameter 及指定 Set Control Parameter 的 properties。

輸入參數：

① VP894
VP894CC 的 control object

② ChNum
被設定更改的 channel number

③ CtrlParam
這是一個 User-Defined type- SetCtrlParamProp 的變數，指明要更改的 Set Ctrl Param properties 內容。

7-4-2、可由使用者自訂的資料型態說明

結構(1)

```
Type QueryProp
  AcceptDTMF As Integer
  ExitDTMF As Integer
  DTMFAmount As Integer
 NextStepForDTMFIntercept (0 To 15) As
 Integer
 NextStepForTimeOut As Integer
End Type
```

結構成員說明：

① AcceptDTMF：

設定可以接受的電話按鍵輸入。這個 integer 中的每一個 bit 各代表 16 個 DTMF 碼中的一個，你可以使用 VP894INC.BAS 中已定義好的 global constant (comment "DTMF Mask Definition") — E_DTMF0～E_DTMF9，E_DTMF_ASTERISK，E_DTMF_POUND 等，來設定這個 Item。例如你如果希望 Query process 接受"0"，"1"，"2" 的按鍵輸入，可指定 AcceptDTMF = E_DTMF0 or E_DTMF1 or E_DTMF2。

② ExitDTMF：

指明以那些按鍵輸入終止 Query process。設定方法請參閱 AcceptDTMF。

③ DTMFAmount：

指明此 Query process 要接收幾個電話按鍵輸入。

④ NextStepForDTMFIntercept (0 To 15)：

這個 integer array 中的每一個成員各儲存一個 process ID，代表 16 個可做爲 Exit DTMF 下一步驟的一個，它們之間的對應關係如下表：

NextStepForDTMFIntercept	DTMF 碼
(0)	0
(1)	1
(2)	2
(3)	3
(4)	4
(5)	5
(6)	6
(7)	7
(8)	8
(9)	9
(10)	A
(11)	B
(12)	C
(13)	D
(14)	*
(15)	#

⑤ ***NextStepForTimeOut***

指明此 Query process 如果 time out 所必須執行的 process。

結構(2)

```
Type RecordProp
  ExitDTMF As Integer
    Length As Integer
    NextStepForDTMFIntercept (0 TO 15) As
  Integer
End Type
```

結構成員說明：

① ExitDTMF：

指明以那些按鍵輸入終止 Record process。設定方法請參考 User-Defined type - QueryProp 中的成員 AcceptDTMF。

② Length：

設定錄音的時間長度，單位爲秒。如果本 Item 設定爲 0，代表沒有錄音長度限制。

③ NextStepForDTMFIntercept (0 To 15)：

請參考 User-Defined type - QueryProp 中的成員 NextStepForDTMFIntercept 的說明。

結構(3)

```
Type CallProp
   Mode As Integer
   NextStepForCallFailure As Integer
    TargetType As Integer
End Type
```

結構成員說明：

① Mode：

如果此 Call process 是要 call 外線或分機電話，則本 Item 是指定此 Call process 的工作模式，其定義如本手冊中第四章、API 函數的 CallLocal()或 CallRemote()的第三個參數 Mode。

② NextStepForCallFailure：

指定如果 call 失敗所必須執行的 process。

③ Target Type：

指定要 call 的是外線，分機電話或 beeper。在 VP894INC.BAS 中已定義此三種不同 target type 的 global constant (comment"called target type")

結構(4)

```
Type SetCtrlParamProp
    fModified As Integer
    OffHookDelay As Integer
    OnHookDelay As Integer
    InterdigitPause As Integer
    FlashTime As Integer
    WaitAnswerDuration As Integer
    NoSignalTimeOut As Integer
    RingsToAnswer As Integer
    OffThreshold As Integer
    PlayMode As Integer
   PlayGain As Integer
   RecordMode As Integer
   RecordGain As Integer
End Type
```

結構成員說明：

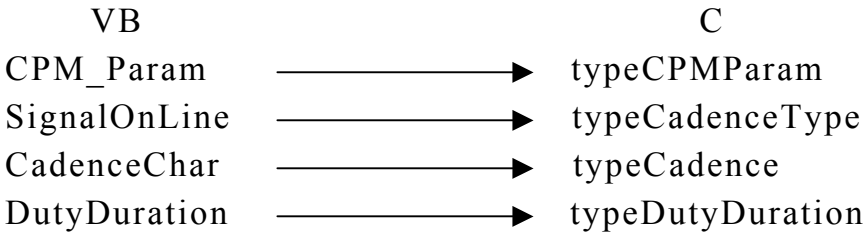
fModified 爲一個 bits Mask，用來指明那些 channel control parameters 要被修改，在 VP894INC.BAS 已定義了一組 global constant 來代表要修改的項目，例如如果你要設定下列幾項 channel control parameter-OnHookDelay，FlashTime，PlayMode 及 RecordGain，可指定 fModified = modON_HOOK_DELAY or modFLASH_TIME or modPLAY_MODE or modRECORD_GAIN，其它資料成員 **(**OffHookDelay，OnHookDelay，InterdigitPause....等**)** 的定義請參閱第三章 3-3 節中相關說明。

結構(5)

```
Type DutyDuration
   OffDuty As Integer
   OnDuty As Integer
   OffDuty Tolerance As Integer
   OnDuty Tolerance As Integer
End Type

Type CadenceCher
   Feature As Integer
   RecognizeCycle As Integer
  WaveDuration(0 To 5) As DutyDuration
End Type

Type SignalOnLine
   Ring As Cadencehar
   Busy As CadenceChar
   InvalidNum As CadenceChar
   DialTone As CadenceChar
   UserDefined1 As CadenceChar
   UserDefined2 As CadenceChar
End

Type CPM_Param
   CentralOffice As SignalOnLine
   PrivateSystem As SignalOnLine
   Beeper As CadenceCher
End Type
```

結構成員說明：

以上幾個 User-Defined data type 是用來定義 VP894 CPM 參數，它們和 C 語言 API 中與 CPM 設定相關聯的結構變數，有以下相同的對應關係：



其中必須注意的是，因爲在 typeCadence 中的兩個成員 WaveCount 及 Type 是被宣告爲位元變數，在 Visual Basic 的 User-Definedata type 中，並未提供相同的宣告語法，所以在 CadenceChar 中是以成員 Featrue 來代表 WaveCount 及 Type 的組合，因此假設如困要指明 CPM 參數內的某一訊號是 CYCLIC_WAVE，且一次週期性變化包含 3 個不同的 duty duration 的組合，則可指定 Featrue=3+CYCLIC_WAVE。其它的結構成員說明及設定方式，請參閱本手冊的 CPM 原理中的解釋。

結構(6)

```
Type HungUpBusy
    Varieties As Integer
    WaveDuration(0 To 4) As DutyDuration
End Type

Type HungUpParam
    Busy As HungUpBusy
    MinBusyDuration As Integer
    MinRoarDuration As Integer
    Busy Threshold As Integer
    Roar Threshold As Integer
   MinSilentDuration As Integer
   Silent Threshold As Integer
End Type
```

結構成員說明：

以上兩個 User-Defined data type 是用來定義 VP894 的偵測遠端掛斷訊號的參數，它們和 C 語言 API 中相關聯的結構變數，有以下相同的對應關係：

VB		C
HungUpParam	⟶	typeHungUpParam
HungUpBusy	⟶	typeHungUpBusy

有關結構成員的說明及設定方式，請參閱 C 語言 API 的說明。

7-5、VP894CC.VBX 應用注意事項

1. 每一個 VP894 control 僅代表一片 VP894 adapter，因此如果你的 application 中支援三片 VP894 adapter，你就必須在你的 Form 內放置三個 VP894 control。又因每一個 control 的 properties 是 private，你改變其中一個 VP894 control 的 process flow，channel control parameter 或 environment options 等，僅改變了該 VP894 control 的 property，而並不影響其它的 VP894 control，所以你如果要產生一個 application 包含多個 property 完全一致的 VP894 control，你可以先在 Form 上放置一個 VP894 control，customize 它的 properties，完成之後，利用 Edit menu 的 copy 功能，將這個 VP894 control 拷貝至 clipboard，然後利用 paste 功能產生新的 VP894control。如果這些 VP894 control 還必須共享 event procedure 的程式碼，(亦即這些 VP894 control 有相同的 execution flow)，你必須選擇產生 control array。未來作程式修改時，也必須採用相同的方式，先刪除其它 VP894 control，僅留下一個作修改，修改完成後，才利用 Edit 的 copy paste 產生其它 control。

2. 在你的 VP894 application 中不可以使用 InputBox 或 MsgBox 兩個 function。因爲這兩個 function 會 block 其它 message 的傳遞，直到你 close 這兩個 dialog box。如果在你使用這兩個 function 與使用者對談時，VP894.DLL 有 message 要傳送給 VP894CC 的 control，這些 message 將無法成功地到達 VP894CC 的 control。但你仍然可以使用 form 來產生 dialog box (因爲使用 Form 產生 dialog box 並無以上問題)。

3. 當你在 VB 的整合環境中除錯，如果程式的執行因爲你所設定的中斷而進入 VB 的中斷模式，此時 VP894.DLL 或 VP894CC.VBX 所產生的任何 event (message) 均會被 VB 爲了同步及 event serialization 的理由而被 block out，因此當你由中斷點繼續往下執行，極可能發現有許多 channel 不再工作 (因爲 VP894 的運作亦是採用 event-driven scheme，倚靠 message 的緊密傳遞，達成 process 與 process 之間的連接) 這是正常現象。你可利用剩下仍正常運作的 channel(s)繼續除錯 (通常會僅剩一線仍正常工作) 或 estart program。

7-6 問題與解答

1.問題：在 Visual Basic 內發生無法載入 VP894CC.VBX 的錯誤。

答：① 請檢查你所使用的 Visual Basic 版本是否爲 2.00 或 2.00 以上。

②請排除所有硬體衝突的可能性。你可以利用在 "VP894CC Demo"program group 中的 VP894 device diagnosis 來確定 VP894.DLL 是否能找到系統內的 VP-894 設備？ **(**如果你在 install VP894CC 並未選擇讓 setup 爲你產生 "VP894CC Demo" program group，你可直接使用 program manager 或 file manager 的 Run 命令啓動在你 install VP894CC 目錄下的可執行程式—chkadptr.exe**)** 如果 VP894 的設備不能被偵測到 ，則表示有硬體相衝突的狀況發生，你可以參考本手冊第二章，改變 VP-894 的硬體設定，或者更改其它硬體設備的設定。

2.問題：啓動 VP894CC 的 demo 程式後，demo 程式對所有有關 VP-894 設備的功能均無反應。

答：請確定所執行 demo 程式內的 VP894 control 的 property-AdapterNumber 是否與你 PC 系統內 install 的 VP-894 設備的硬體組態相符。在 VP894CC install disk 上所有的 demo 程式內的 VP894 control 的 property-AdapterNumber，預設均爲 0，如果在你 PC 系統內的 VP894 實體卡號設定不爲 0，你可以在 Visual Basic 內更改 demo 程式的 VP894 control，使其 property-AdapterNumber 與實體設備相符，重新 Make EXE 檔即可。

3.問題：執行 VP894CC 的 demo 程式時，在流程處理 Query process 均無語音被播放出來。

答： 請確定所執行的 demo 程式內的 VP-894 control property-environment 的 Prompt Directory 設定，是否符合你 install VP894CC demo 程式的路徑。如果你在 install VP-894CC 時接受 setup 所建議的 default destination path，則 Prompt Directory 的設定即與實際提示語音檔案所儲存的路徑相符，否則你必須更改所有 demo 程式的 VP894 control 的 Prompt Directory 。

4.問題：如果當使用 API function-SetQueryProp()更改 process 的 properties 時，Visual Basic 顯示一個 error message box“This channel is not exist...”，但是所傳給 SetQueryProp()的 channel number 是肯定正確的?

答: 請檢查傳給 SetQueryProp()function 的第 3 個參數 PromptList 的資料類型是否被明確地宣告爲 String。 因爲我們爲了能讓 client application 在呼叫 API function SetQuery- Prop()時能夠指定 PromptList 爲 Null(亦即指定此 Query Process 僅接收 DTMF，而不播放任何語音)，所以在 VP-894INC.BAS 中宣告 SetQueryProp()的 prototype 時，其參數列上的 PromptList 是宣告爲 ANY。在正常情形 SetQueryProp()被呼叫時，應該是以 4 bytes 的遠程指標傳送 PromptList 的地址給 VP894 CC.VBX，如果 PromptList 是被明確地宣告爲 String，Visual Basic 會正確地傳遞此項參數給 VP894CC.VBX，但是如果 PromptList 沒有被明確地宣告爲 String 的資料類型，Visual Basic 是以 default(內定)的資料類型 - Variant 視之，因此在 SetQueryProp()被呼叫時，PromptList 是被轉換爲 2 bytes 的 integer 傳送給 VP894CC.VBX，如此即造成 VP894CC.VBX 所接收到的參數位元組總數少於正常的數目，於是 SetQueryProp()在 interpreting 參數 ChNum 時，被 shift 到參數 VP894 control object 上，而認爲 application 所傳送的 ChNum 有誤。有關正確 的實作，請參考 Depulse 的 demo 程式。

八、VP-894 相關介面卡應用說明

8-1、EX-24 電話交換卡

EX-24 是一具類比交換功能的卡片，與 VP-894 搭配使用後可使最多 24 條電話線中的任意兩線在維持原有通話品質的條件下，利用系統軟體控制進行搭接。與 VP-894 本身只能進行固定配對的電話轉接功能相比，EX-24 的使用更具有彈性，因而擴大了 VP-894 的應用層面。此外，由於 EX-24 任意交換的功能並不限於成對的連接，使得三方或多方通話的應用亦成爲可能。

8-1-1、EX-24 硬體說明

1.EX-24 具備 24 組類比信號彼此互相交換之能力，可與 6 片 VP-894 配合使用。卡上有 6 個連接器(JM1---JM6)與對應之 VP-894 相接。

2.系統架構的方式是將最多 6 片 VP-894 連接至 EX-24 上，則 24 線中任意一線可與相鄰的 23 線中任意一線相連接。

3.EX-24 有 6 個 I/O PORT 可供選擇，指撥開關 (S1)爲供 I/O PORT 之設定，方式如下：

S1								I/O
1	2	3	4	5	6	7	8	PORT
OFF	ON	ON	OFF	OFF	ON	ON	OFF	260H
OFF	ON	ON	OFF	OFF	ON	OFF	OFF	268H
OFF	ON	ON	OFF	OFF	OFF	ON	OFF	270H
OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	278H
OFF	ON	OFF	ON	ON	ON	ON	OFF	280H
OFF	ON	OFF	ON	ON	ON	OFF	OFF	288H
OFF	ON	OFF	ON	ON	OFF	ON	OFF	290H

8-1-2、介面發展程式(API)DOS 版說明

1.說明：

本驅動程式目前提供 C 語言介面及 CLIPPER 5.01 介面，請使用 MSC 5.0 以上版本或 TURBO C 2.0 以上版本及 BORLAND C++ 2.0 以上版本進行連結(LINK)使用

2.函數功能摘錄：

① EX_INITIAL() ：初始化 EX24，並設定 I/O PORT
② EX_OPENALL()：切斷所有的連接
③ EX_CONNECT()：搭接左右兩側所指定的兩線
④ EX_DISCON() ：切斷已搭接的某兩線
⑤ EX_STATUS() ：取得左側某線的連接狀態

3.各函數使用說明：

① void EX_INITIAL(int port)

功能：(1) 將 EX24 所有連接閘門打開**(**切斷所有的連接**)**
(2) 設定 EX24 所使用的 I/O PORT

傳入值：
int port：I/O 的值
(0x260，0x268，0x270， 0x278，0x280，0x288，0x290**)**
(For Clipper-608，616， 624，632，640，648， 656**)**

傳回值：無

附註：應用程式中只須在開始時呼叫一次本函數，即可重複使用下列函數。

② void EX_OPENALL(void)

功能：切斷所有連接

傳入值：無

傳回值：無

③ int EX_CONNECT(int Lline，int Rline)

功能：將 Lline 及 Rline 互相連接

傳入值：
int Lline：左側線號 (0 - 23)
int Rline：右側線號 (0 - 23)

傳回值：0 表成功
1 表失敗 (Lline = Rline)

④ *int EX_DISCON(int Lline，int Rline)*

功能：將 Lline 及 Rline 的連接切斷

傳入值：
int Lline：左側線號 (0 - 23)
int Rline：右側線號 (0 - 23)

傳回值：0 表成功
1 表失敗 (Lline = Rline)

⑤ *int EX_STATUS(int Lline)*

功能：取得左側某線的連接狀態

傳入值：
int Lline：左側線號 (0 - 23)

傳回值: -1 表無連結
n 表正與第 n 線相連

8-1-3、介面發展程式(API)WINDOWS 版說明

1.說明：

本驅動程式乃以 DLL 模式存在於 WINDOWS 中，依照各種語言的宣告及呼叫 DLL 的方式進行呼叫 DLL 的名稱爲 EX24.DLL。

2.函數功能摘錄：

① EX_INITIAL()　：初始化 EX24，並設定 I/O PORT
② EX_OPENALL()：切斷所有的連接
③ EX_CONNECT()：搭接左右兩側所指定的兩線
④ EX_DISCON()　：切斷已搭接的某兩線
⑤ EX_STATUS()　：取得左側某線的連接狀態

3.各函數使用說明：

① void EX_INITIAL(int port)

功能：(1) 將 EX24 所有連接閘門打開
(2) 設定 EX24 所使用的 I/O PORT

傳入值：
int port：I/O 的值
(0x260，0x268，0x270， 0x278，0x280，0x288，0x290**)**
(For Visual Basic &H260，&H268，&H270，&H278，&H280，&H288，&H290**)**
(For Clipper-608，616， 624，632，640，648，656**)**

傳回值：無

附註：應用程式中只須在開始時呼叫一次本函數，即可重複使用下列函數。

② void EX_OPENALL(void)

功能：切斷所有連接

傳入值：無

傳回值：無

③ int EX_CONNECT(int Lline，int Rline)

功能：將 Lline 及 Rline 互相連接

傳入值：
int Lline:左側線號 (0 - 23)
int Rline:右側線號 (0 - 23)
Lline != Rline

傳回值：0 表成功
1 表失敗 (Lline = Rline)

④ *int EX_DISCON(int Lline，int Rline)*

功能：將 Lline 及 Rline 的連接切斷

傳入值：
int Lline：左側線號 (0 - 23)
int Rline：右側線號 (0 - 23)
Lline != Rline

傳回值： 0 表成功
1 表失敗 (Lline = Rline)

⑤ *int EX_STATUS(int Lline)*

功能：取得左側某線的連接狀態

傳入值：
int Lline：左側線號 (0 - 23)

傳回值：-1 表無連結
n 表正與第 n 線相連

8-1-4、示範程式說明

1.XD894.EXE 說明

本程式用以表現 VP-894 與 EX24 交換卡共同使用時，如何完成一個簡單的電話轉接系統，其程式概念如下**:**

① 將 VP-894 的偶數 CH**(**0，2，4…**)**定義爲主叫方語音卡
② 將 VP-894 的奇數 CH**(**1，3，5…**)**定義爲轉接用語音卡**(**註**)**
③ 主叫方撥入系統後，系統應答並要求輸入欲轉接的電話號碼。待主叫方完成輸入後，由轉接用語音卡撥出轉接號碼
④ 下令 EX24 將雙方接通
⑤ 偵測主叫方線路是否出現掛斷信號，若是則下令 VP-894 掛斷電話，並令 EX24 切斷雙方連線

執行語法：Drv894 (Enter)
XD894 (Enter)

註： 本程式並未寫成任意交換功能，測試時僅能以固定配對進行。配對方式類似 VP-894 之內部轉接，即 CH0 對 CH1，CH2 對 CH3….依此類推。

2.X3D894.EXE(FOR DOS)說明

本程式用以示範如何使用 VP-894 與 EX24 達成多方通話的功能，其程式設計概念如下：

① 將 VP-894 0～5 號卡均設定在等待電話撥入的狀態

② 任何一線有電話撥入時，即令其 pick up

③ 令 EX24 將該線與其他已撥通之 channel 接通，達到多方通話的目的

執行方式：DRV894 (Enter)
X3D894 (Enter)

3.WINDOWS VISUAL BASIC 示範程式說明

本程式用以展示在 WINDOWS 環境中如何以 VISUAL BASIC 使用 VP894 與 EX24 達成多方通話的功能，其程式設計概念與 X3D894.EXE 相同。

執行方式：① 啓動 VISUAL BASIC 之 IDE
② Load \EX24\WIN 中之 X3D894.VBP
③ RUN (註)

註：執行本程式前，請將 WIN 中之 EX24.DLL COPY 至 WINDOWS 的\SYSTEM 子目錄中。

8-1-5、問題與解答：

1.問題:插入 EX24 交換卡後，電腦無法開機

答：此現象有可能是卡未插好或電腦上有另一介面卡與 EX24 使用同一 I/O PORT 請查明並重新設定後再開機。 **(**請注意 EX24 使用 BASE 到 BASE+7 八個 I/O PORT**)**

2.問題:執行程式但 EX24 不動作

答：此現象應是 EX24 交換卡上設定的 I/O PORT 位置與軟體不符合引起，若經查兩者 I/O PORT 相同則可能是 EX24 硬體功能失效，請速與本公司連絡。

8-2、EX-2424 電話交換卡

EX-2424 與 EX-24 功能近似(請參考 EX-24 說明),不同點爲 EX-2424 內部分成兩個不同的 Group，同一 Group 內的信號無法交換，祗有不同 Group 間之線路才能進行轉接。此外，EX-2424 無法達成類似 EX-24 的多方通話功能

8-2-1 EX-2424 交換卡硬體說明

1.EX-2424 具備內部兩組各 24 線彼此之間類比信號交換能力，可與 12 片 VP-894 配合使用。左邊連接器編號爲 JM1---JM6，右邊連接器編號爲 JW1---JW6，各經由連接器與對應之 VP-894 相接。

2.系統架構的方式是將 VP-894 分成左右兩組每組最多 6 片，分別與 EX2424 連接，則 EX-2424 左側 24 線中任一線可與右側 24 線中任何一線連接。

3.EX-2424 有 6 個 I/O PORT 可供選擇,指撥開關(S1)爲供 I/O PORT 之設定，方式如下：

S1								I/O
1	2	3	4	5	6	7	8	PORT
OFF	ON	ON	OFF	OFF	ON	ON	OFF	260H
OFF	ON	ON	OFF	OFF	ON	OFF	OFF	268H
OFF	ON	ON	OFF	OFF	OFF	ON	OFF	270H
OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	278H
OFF	ON	OFF	ON	ON	ON	ON	OFF	280H
OFF	ON	OFF	ON	ON	ON	OFF	OFF	288H
OFF	ON	OFF	ON	ON	OFF	ON	OFF	290H

8-2-2、介面發展程式(API)DOS 版說明

1.說明：

本驅動程式目前提供 C 語言介面及 CLIPPER 5.01 介面，請使用 MSC 5.0 以上版本或 TURBO C2.0 以上版本及 BORLAND C++ 2.0 以上版本進行連結(LINK)使用

2.函數功能摘錄：

① EX_INITIAL() ：初始化 EX2424，並設定 I/O PORT
② EX_OPENALL()：切斷所有的連接
③ EX_CONNECT()：搭接左右兩側所指定的兩線
④ EX_DISCON() ：切斷已搭接的某兩線
⑤ EX_STATUS() ：取得左側某線的連接狀態

3.各函數使用說明：

① void EX_INITIAL(int port)

功能：(1) 將 EX2424 所有連接閘門打開 **(**切斷所有的連接**)**
(2) 設定 EX2424 所使用的 I/O PORT

傳入值：
int port：I/O 的值
(0x260，0x268，0x270， 0x278，0x280，0x288， 0x290**)**
(For Clipper- 608，616， 624，632，640，648，656**)**

傳回值：無

附註：應用程式中只須在開始時呼叫一次本函數，即可重複使用下列函數。

② void EX_OPENALL(void)

功能：切斷所有連接

傳入值：無

傳回值：無

③ void EX_CONNECT(int Lline，int Rline)

功能：將 Lline 及 Rline 互相連接

傳入值：
int Lline:左側線號 (0 - 23)
int Rline:右側線號 (0 - 23)

傳回值：無

④ *void EX_DISCON(int Lline，int Rline)*

功能：將 Lline 及 Rline 的連接切斷

傳入值：
int Lline：左側線號 (0 - 23)
int Rline：右側線號 (0 - 23)

傳回值：無

⑤ *int EX_STATUS(int Lline)*

功能：取得左側某線的連接狀態

傳入值：
int Lline：左側線號 (0 - 23)

傳回值：-1 表無連結
n 表正與第 n 線相連

8-2-3 介面發展程式(API)WINDOWS 版說明

1.說明：

本驅動程式乃以 DLL 模式存在於 WINDOWS 中，請依照各種語言的宣告及呼叫 DLL 的方式進行呼叫 DLL 的名稱爲 EX2424.DLL。

2.函數功能摘錄：

① EX_INITIAL() ：初始化 EX2424，並設定 I/O PORT
② EX_OPENALL()：切斷所有的連接
③ EX_CONNECT()：搭接左右兩側所指定的兩線
④ EX_DISCON() ：切斷已搭接的某兩線
⑤ EX_STATUS() ：取得左側某線的連接狀態

3.各函數使用說明：:

① void EX_INITIAL(int port)

功能：(1) 將 EX2424 所有連接閘門打開
(2) 設定 EX2424 所使用的 I/O PORT

傳入值：
int port：I/O 的值
(0x260，0x268，0x270，0x278，0x280，0x288，0x290**)**
(For Visual Basic &H260， &H268，&H270，&H278，&H280，&H288，&H290**)**
(For Clipper-608，616， 624，632，640，648， 656**)**

傳回值：無

附註：應用程式中只須在開始時呼叫一次本函數，即可重複使用下列函數。

② void EX_OPENALL(void)

功能：切斷所有連接

傳入值：無

傳回值：無

③ void EX_CONNECT(int Lline，int Rline)

功能：將 Lline 及 Rline 互相連接

傳入值：
int Lline：左側線號 (0 - 23)
int Rline：右側線號 (0 - 23)

傳回值：無

④ *void EX_DISCON(int Lline，int Rline)*

功能：將 Lline 及 Rline 的連接切斷

傳入值：
int Lline:左側線號 (0 - 23)
int Rline:右側線號 (0 - 23)

傳回值：無

⑤ *int EX_STATUS(int Lline)*

功能：取得左側某線的連接狀態

傳入值：
int Lline:左側線號 (0 - 23)

傳回值：-1 表無連結
n 表正與第 n 線相連

8-2-4、示範程式說明

1.XXD894.EXE 說明

本程式用以表現 VP-894 及 EX2424 交換卡共同使用時，如何完成一個簡單的電話轉接系統，其程式概念如下：

① 將 VP-894 卡號 0 - 5 定義爲主叫方語音卡
② 將 VP-894 卡號 6 - 11 定義爲轉接用語音卡**(**註**)**
③ 主叫方撥入系統後，系統應答並要求輸入欲轉接的電話號碼。待主叫方完成輸入後，由轉接用語音卡撥出轉接號碼
④ 下令 EX2424 將雙方接通
⑤ 偵測主叫方線路是否出現掛斷信號，若是則下令 VP-894 掛斷電話，並令 EX2424 切斷雙方連線

執行語法：Drv894 (Enter)
XXD894 (Enter)

註： 本程式進行轉接時，會由卡號 6 的 VP-894 開始按未使用的 port 順序做爲與主叫方的連接路徑。故如僅使用兩條外線測試時，建議第一條外線連至第一片 VP-894**(**卡號 0**)**的任意 port，第二條外線連至 CH24**(**卡號 6 的第一個 port**)**

2.WINDOWS VISUAL BASIC 示範程式說明

本程式用以展示在 WINDOWS 的 VISUAL BASIC 環境下如何同時使用 VP-894 及 EX2424，其程式設計概念與 XXD894.EXE 相同，請參考前一節。

執行方式： ① 啓動 VB 之 IDE
② LOAD EX2424\WIN 內之 EX2424.VBP
③ RUN (註)

註：執行本程式前，請將\WIN 中之 EX2424.DLL COPY 至 WINDOWS 的\SYSTEM 子目錄中

8-2-5、問題與解答

1.問題:插入 EX2424 交換卡後，電腦無法開機

答：此現象有可能是卡未插好或電腦上有另一介面卡與 EX2424 使用同一 I/O PORT 請查明並重新設定後再開機。 **(**請注意 EX2424 使用 BASE 到 BASE+7 八個 I/O PORT**)**

2.問題:執行程式但 EX2424 不動作

答：此現象應是 EX2424 交換卡上設定的 I/O PORT 位置與軟體不符合引起，若經查兩者 I/O PORT 相同則可能是 EX2424 硬體功能失效，請速與本公司連絡。

附錄 A、通信小辭典

與 VP-894 相關的電腦語音應用均爲與電話系統 **(**無論是電信局的大型系統或中小型商用交換機**)** 結合，因而程式設計師在進行語音系統的規劃時，必須對各種不同情況的電話線路信號音有所概念，進而對應有操作方式有所瞭解。以下針對一些基本的通信專有名詞予以說明：

送話端(CALLING PARTY)：拾起話筒按鍵，撥出電話號碼到電信局的這一端。

受話端(CALLED PARTY)：聽到話機響鈴聲，拾起話筒，回應來電的這一端。

脈衝撥號(PULSE DIALING)：舊型撥盤式電話的撥號方式，撥號時，話機以機械性地 on-off 電話迴路來送出代號數字的脈衝。如撥 1 時送出 1 個脈衝，撥 0 時送出 10 個脈衝。

複頻(DTMF)碼：話機按鍵上 0～9，＊，＃，12 個鍵所送出的碼的格式。爲新式話機使用的撥號方式。 **(**DTMF 爲 Dual-Tone Multi-Frequency，即雙音複頻之簡稱**)**

回鈴音(RINGBACK TONE)：在送話端完成撥號動作時，如果受話端爲空閒狀況，電信局會送出回鈴音給送話端。此種回鈴音一般有單響式**(**SINGLE RING**)**及雙響式**(**DOUBLE RING**)**兩種。

響鈴音：在送話端完成撥叫動作後，電信局會送出此一振鈴**(**RING DOWN**)**訊號，使話機發出可聞之響鈴。

忙音(BUSYTONE):：當送話端撥叫一個正在使用中的受話端時，電信局會送出忙音訊號給送話端。

聒噪音(ROARING TONE)：當完成通話後的兩端，若有一端忘了掛上電話，此時有些**(**並非所有**)**電信局會送出聒噪音做爲提醒。此聒躁音通常是非常吵雜且高分貝的雜音，並連

續送出一段時間(幾分鐘)後才停止。

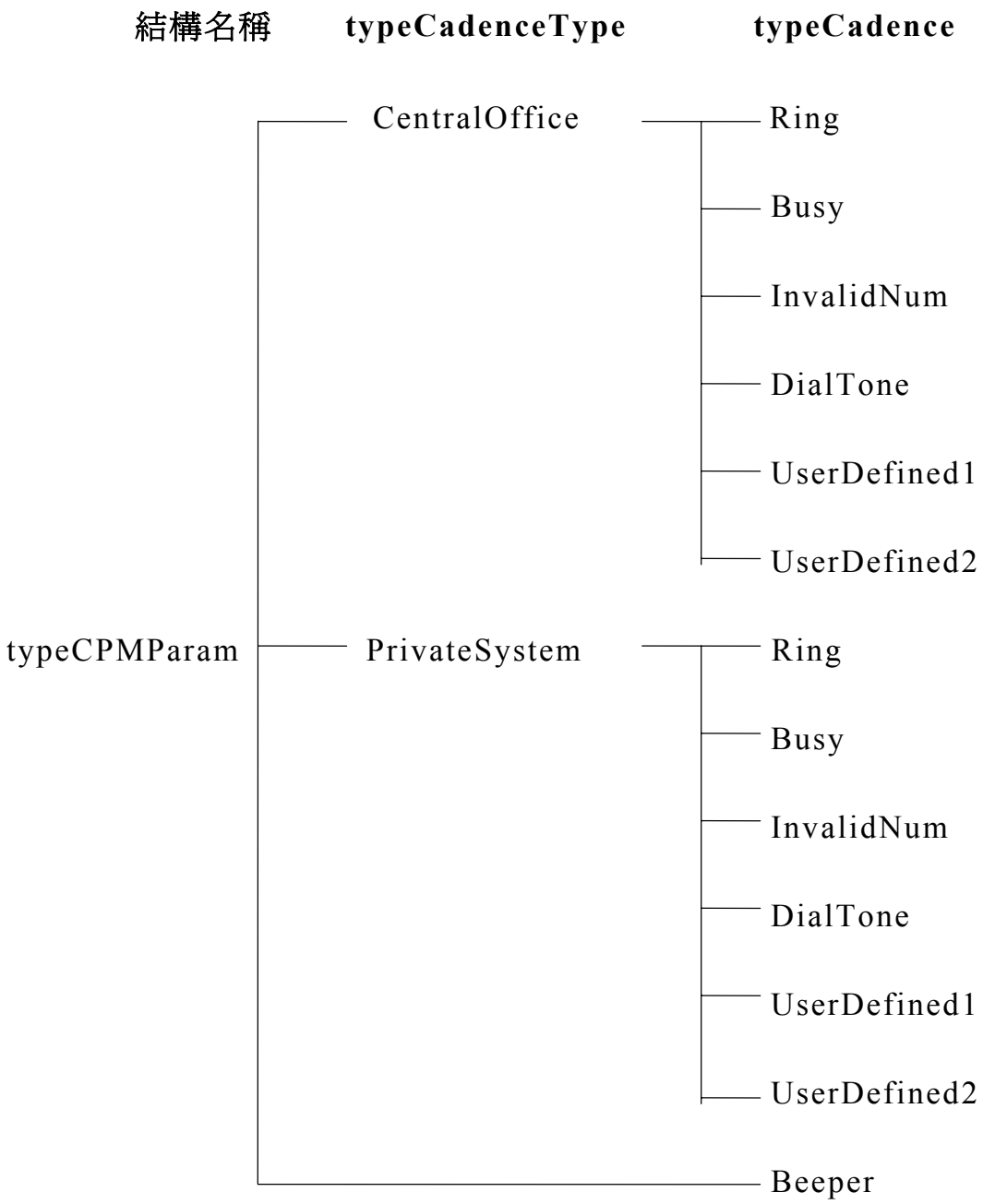
其他信號音：例如空號音 (代表交換機內無此分機)以及少數特殊信號音等。其規格須由交換機廠商提供。

佔用外線碼：如 VP-894 是連接公司內部的交換機的一個分機，則在使用外線時要先撥通一個代碼(如 9 或 0)，並等待外線撥號音的出現才能正式開始撥號。

轉接鍵(FLASH)：按放電話 HOOK 的動作鍵(有的話機本身具備此功能鍵)，做爲轉接其他分機的前置動作，以毫秒千分之一秒爲單位，一般規格爲 600 毫秒 (即 0.6 秒)。唯不同交換機之規格殊不一致，某些情況下有的交換機甚至要求二次轉接動作。

附錄 B、CPM 原理解析

這一節簡單地概述我們如何利用 VP-894 所產生的 energy record 來完成 CPM (Call Progress Monitor)的工作，及 Application programmer 如何正確地設定 CPM 的參數 (CPM 參數的設定直接影響 CPM 偵測的正確率及靈敏度)。CPM 的參數可分爲兩個 groups，一組是紀錄局線(外線)的，另一組是紀錄 PABX 內線的 (分別由結構 typeCPMParam 中成員 CentralOffice 及 PrivateSystem 紀錄，另一成員 Beeper 中則紀錄呼叫器 AcknowledgeTone 的訊號特徵)。而這兩組資料結構又可劃分爲 ring(回鈴)，Busy (忙音)，InvalidNum (空號)，DialTone (撥號音) UserDefined1(使用者定義信號 1)及 UserDefined2 (使用者定義信號 2)。於次頁以樹狀圖表示資料結構：



由圖中我們可看出 Ring，Busy...等六種訊號成員及 Beeper 同屬於結構類別 typeCadence，此結構定義出各個訊號特徵如何被量化紀錄，及 VP-894 是如何辨認出一個特殊的訊號。以下是 typeCadence 的原型定義及各結構成員的說明：

```
struct _tagCadence {
        unsigned char          WaveCount :3;
        unsigned char          Type      :1;
        unsigned char                    :4;
        unsigned char          RecognizeCycle;
        typeDutyDuration       Wave[6];
       }typeCadence;
```

我們先來看看 Wave[ ]，Wave[ ]是一個包含 6 個元素的陣列，而每個元素的資料類別是預先定義的結構 typeDutyDuration，下面是它的原型定義：

```
structb _tagDutyDuration {
        unsigned  Silence;
        unsigned  NoneSilence;
        unsigned  SilenceTolerance;
        unsigned  NoneSilenceTolerance;
       }typeDutyDuration;
```

由上面的結構定義可知，我們是紀錄訊號的 Duty Duration 也就是訊號的 on 或 off 持續時間。其中 SilenceTolerance 及 NoneSilenceTolerance 是分別　紀錄 off、on 的容許誤差。這四個成員的單位是 8ms。我們稱訊號一次 on、off 變化的時間爲一個 Duty Cycle，因此 typeCadence 對每一種訊號最多可紀錄 6 個連續 Duty Cycle。至於某一段 Duration 的訊號是 on 或 off 並不是以絕對的 energy value 來決定， 而是以相對於前一段 energy 大小來判定 (此乃因各　PABX 或各地局線訊號大小皆不相同)。而在實際應用上 VP-894 監測每 8ms 取樣進來的 energy value，並將 energy 變化在最大最小差異小於 4 時視爲同一段　on 或 off 的訊號來分割 (如果變化的 Duration 不超過 4 者則被視同雜訊而消除) 然後將分段出來的訊號 Duration 與 ring、busy 等六種訊號參數設定作比較找出相符者。

typeCadence 中另一成員 WaveCount 指出這個訊號參數包含有多少個 duty cycle (最大爲 6)；如果這個訊號是週期性循環出現的 (例如 ring 或 busy) 則 Type 可指定爲 CYCLIC_WAVE 否則須設定爲 NONE_CYCLIC_WAVE；最後一個成員參數 Recognize- Cycle 是針對 Type 爲 CYCLIC_WAVE 的訊號，它指示 VP-894 須連續偵測到幾個 cycle 相符的訊號，才判定是接收到此種訊號。以下是幾個實際的應用例子：

1.typeCPMParam CPMParam; int Result;

```
CPMParam.PrivateSystem.Busy.Type = CYCLIC_WAVE;
CPMParam.PrivateSystem.Busy.WaveCount = 1;
CPMParam.PrivateSystem.Busy.RecognizeCycle = 2;
CPMParam.PrivateSystem.Busy.Wave[0].Silence = 58;
CPMParam.PrivateSystem.Busy.Wave[0].NoneSilence = 67;
CPMParam.PrivateSystem.Busy.Wave[0].SilenceTolerance = 9;
CPMParam.PrivateSystem.Busy.Wave[0].NoneSilence- Tolerance =
                                                    12;
CPMParam.PrivateSystem.Ring.Type = CYCLIC_WAVE;
CPMParam.PrivateSystem.Ring.WaveCount = 2;
CPMParam.PrivateSystem.Ring.RecognizeCycle = 1;
CPMParam.PrivateSystem.Ring.Wave[0].Silence = 30;
CPMParam.PrivateSystem.Ring.Wave[0].NoneSilence = 67;
CPMParam.PrivateSystem.Ring.Wave[0].SilenceTolerance = 6;
CPMParam.PrivateSystem.Ring.Wave[0].NoneSilence- Tolerance =
                                                    12;
CPMParam.PrivateSystem.Ring.Wave[1].Silence = 255;
CPMParam.PrivateSystem.Ring.Wave[1].NoneSilence = 67;
CPMParam.PrivateSystem.Ring.Wave[1].SilenceTolerance = 45;
CPMParam.PrivateSystem.Ring.Wave[1].NoneSilence- Tolerance =
                                                    12;
if (Result = SetCPMParam (&CPMParam))
FatalError (Result);
```

本段程式是示範設定內線訊號 busy on duty duration 爲 536±96ms，off duty duration 爲 464±72ms;而回鈴爲雙響回鈴 **(**double ring**)**，第一個 cycle 的 on duty duration 爲 536±96ms，off duty duration 爲 240±48ms，第二個 cycle 的 on duty duration 爲 536±96ms，off duty duration 爲 2040±360ms。

2.type CPMParam CPMParam; int Result;

```
CPMParam.CentralOffice.Busy.Type = CYCLIC_WAVE;
CPMParam.CentralOffice.Busy.WaveCount = 1;
CPMParam.CentralOffice.Busy.RecognizeCycie = 2;
CPMParam.CentralOffice.Busy.Wave[0].Silence = 57;
CPMParam.CentralOffice.Busy.Wave[0].NoneSilence = 68;
CPMParam.CentralOffice.Busy.Wave[0].SilenceTolerance = 9;
CPMParam.CentralOffice.Busy.Wave[0].NoneSilence- Tolerance =
                                                    12;
```

```
CPMParam.CentralOffice.Ring.Type = CYCLIC_WAVE;
CPMParam.CentralOffice.Ring.WaveCount = 1;
CPMParam.CentralOffice.Ring.RecognizeCycle = 1;
CPMParam.CentralOffice.Ring.Wave[0].Silence = 246;
CPMParam.CentralOffice.Ring.Wave[0].NoneSilence = 129;
CPMParam.CentralOffice.Ring.Wave[0].SilenceTolerance = 40;
CPMParam.CentralOffice.Ring.Wave[0].NoneSilence- Tolerance =
                                                    20;
if (Result = SetCPMParam(&CPMParam))
FatalError (Result);
```

以上程式片段是示範設定局線訊號 busy on duty duration 爲 544± 96ms.off duty duration 爲 456 ± 72ms；而回鈴爲單回鈴音 **(**single ring**)**，on duty duration 爲 1032±160ms.而 off duty duration 爲 1968±320ms.**(**忙音要連續偵測到兩次，才被確認**)**

上面兩段程式所示範的 ring back 和 busy 訊號參數設定，皆是針對週期性循環出現的訊號，如果是非週期性的訊號 **(**例如局線的 dial tone 是一個持續固定的頻率。呼叫器的 acknowledge tone 是一個長約一秒的 tone**)** 除了 typeCadence 中的 Type 必須設定爲 NONE_CYCLIC_WAVE 外，Wave[ ]最後一個 Duty cycle 的 Silence 須視狀況設爲 0 或其它值。例如局線的 dial tone 因爲是一個持續的固定頻 率所以其 Silence 必須設定爲 0 以下是 dial tone 及 Beeper acknowledge tone 的一個設定例子。

3.typeCPMParam CPMParam; int Result;

```
CPMParam.ContralOffice.DialTone.Type = NONE_CYCLIC_WAVE;
CPMParam.ContralOffice.DialTone.WaveCount  = 1;
CPMParam.ContralOffice.DialTone.Wave[0].Silence = 0;
CPMParam.ContralOffice.DialTone.Wave[0].NoneSilence = 125 ;
CPMParam.Beeper.Type = NONE_CYCLIC_WAVE;
CPMParam.Beeper.WaveCount = 1;
CPMParam.Beeper.Wave[0].Silence = 30;
CPMParam.Beeper.Wave[0].NoneSilence = 125;
CPMParam.Beeper.Wave[0].NoneSilenceTolerance = 38;
if (Result = SetCPMParam(&CPMParam))
FatalError (Result);
```

前面的例子是當偵測到連續一秒的 on duty duration 即判定是 dial tone；如果在出現 on duty duration 1000±304ms 後出現 off duty duration 240ms 則

判定偵測到呼叫器的 acknowledge tone。就如在 GetEnergy( )中所提到的 energy value 會隨著線路信號的強弱及 RecordGain 的調整而有所不同，依照本公司在開發過程中所測試的數據顯示 RecordGain 的設定範圍在 2 ～-4 之間能有極高的正確率及極佳的靈敏度 **(**RecordGain 大於 2 時， energy value 的變化呈不規則或大小震盪超出 4 的設定值；當 RecordGain 小於-4 時，energy value 太小或爲 0**)** 以下是本公司在實驗中藉外線電話爲對象，每一種 RecordGain 設定均取樣一百通電話號碼所得到的結果： **(**在 RecordGain=0 條件下實際測試得到的 energy report 值，請參考附錄 C **)**

	正確率
RecordGain = -4	99%
RecordGain = -2	99%
RecordGain = 0	99%
RecordGain = 2	93%

由上表可歸納出調高 RecordGain 會使正確率降低，但對於判斷受話方 answer 的靈敏度則會增加**(**RecordGain -4 及 -2 中有一通的不正確是判斷受話方 answer 不良**)**，所以若 Application 必須在 RecordGain 設定爲 2 ～ -4 的範圍之外工作時請在執行 CallRemote()，CallLocal() 及 CallBeeper()之前，務須將 RecordGain 重新設定在 2 ～-4 之間。若在使用 Device Driver 內 CPM 參數的預設值執行 Call out 時，經常發生判斷錯誤的情況時，則必須呼叫 SetCPMParam()修改預設的 CPM 參數。VP-894 SDK 磁片上的 demo 程式 energy.exe 可以幫助你正確地設定 CPM 參數。它的使用格式如下：

>energy　電話號碼　(> 檔案名稱)

括號內的命令是 option，這是使用 DOS 的 redirection 的功能把 energy 資料的輸出送到檔案中。如果括號內的命令省略，則所有的輸出將被送到 stdout。

附錄 C、VP-894 的 LINE BUSY，RINGBACK 及 ANSWER 的能量偵測記錄

九、Visual C 32 Bit 語言介面程式說明

VP-894 Windows 32 位元版的驅動程式包含 ring 0 的 VP894.VxD 及 ring 3 的 VP894_32.DLL 兩部份，因爲 VP894_32.DLL 是透過 exported function 與 application cooperate，所以基本上只要 program developing language 有支援可以呼叫 DLL 的 exported function，而且 application 可以 retrieve Windows 的 scheduled message，都可以利用 VP-894 Win32 device driver 所提供的 API 來發展語音通信的應用程式。VP-894 的 Win32 API 的介面定義，在使用上幾乎完全與 Win16 的 API 相同，唯一不同的地方是 Win32 的 API 增加了兩個 application 在啓動及終止前必須呼叫的 function Init894Driver ( )和 Close894Driver( )。本單元僅列出與 VP-894 Win16 API 相異之處，其它部份請參考第六章 Win16 的 API 說明。

1.Init894Driver ( )

輸入參數：無

傳回值：0 代表失敗，指示向 VP894_32.DLL 註冊的 client application 已經額滿(同時可以有 16 個 execution 可以註冊成爲 VP894_32 的 client application)；否則傳回其註冊的序號。

功能描述：Application 利用本函式向 VP894_32.DLL 註冊成爲其 client application，並且讓 VP894_32.DLL 執行其必要的初始化程序。本功能必須在使用其它 API function 之前呼叫。

2.Close894Driver ( )

輸入參數：無

傳回值：0 代表失敗，非 0 值表示成功

功能描述：Application 若成功地利用 Init894Driver()function 成爲 VP894_32.DLL 的 client application 後，在程式結束執行前必須呼叫本函式註銷其註冊，讓 device driver 完成 free system resource 及其它必要的 clean up 工作。

十、Visual Basic OCX 語言介面程式說明

此介面程式只在 Windows 95 上執行，並且與 Visual Basic 5.00 (含) 以上相容。請注意由於 VP894CC.OCX 只是 VP894_32.DLL 的一個 function wrapper，因此在 Windows\System 目錄下必須同時安裝 VP894CC.OCX，VP894_32.DLL 及 VP894.VXD 才能開始使用。

10-1、VP-894 的 Active X Custom Control 說明

要使用 VP894CC 這個 active X control，你必須拉下 VB 的 Project 選單，或直接在 ToolBox 上以滑鼠右鍵叫出 pull-up dialog 然後選擇 Components 個功能選項，就可在 Component 的 dialog box 選擇 VP894CC.OCX。如果以上的步驟均正確無誤，在 VB 的 ToolBox 中即會顯示這個代表 VP894 active X control 的 button bitmap，另外，你也必須使用 Project/Add Module 這個功能選項加 VP894INC.BAS 及 MEMORY.BAS 這個兩個 code module，在 VP894INC. BAS module file 中定義了與 VP894CC 相關的 globalconstant，user-defined data type， 而 MEMORY.BAS 中放了一些 user-defined data type 和 OLE Handle 之間的轉換函數。被放置到 Form 的每一個 VP894 control 各代表著一片 VP894 adapter，並以此兩種圖案代表該 control 所象徵的 VP894 adapter 目前的狀態；在 design time 時前者表示該 VP894 設備是 free，後者表示該 VP894 設備不存在或已配置給其他 Task；在 run time 時，前者表示該 VP894 設備已配置給擁有該 VP894CC 的 application，後者同於 design time 的定義。接著我們將分別以下面數節來說明 VP894CC 的特點及其使用方法。

10-1-1、模組介紹

Active X custom control 介面程式的 method 在被呼叫時，所使用的參數都必須是標準型態，所以我們自訂型態的參數都必須轉換成 OLE Handle 才能做資料交換，爲減輕程式設計者的負擔，我們提供一個已經寫好的
Visual Basic Module 提供一些 OLE Handle 和自訂型態參數之間的轉換函數，這些函數共可分爲兩類：一類爲記憶體配置函數，第二類爲記憶體釋放函數，以下就是這些函數的介紹：

1.記憶體配置函數

① Function QueryPropToHandle (Prop As QueryProp)As Long

此函數將會配置一塊記憶體，放置所傳入之 QueryProp 參數，並將此記憶體的 Handle 傳回，如果此函數記憶體配置失敗，將會傳回 0 表示工作失敗。程式設計者可依此類推其他所有的 RecordPropToHandle，CallPropToHandle，SetCtrlPropToHandle，ChParamToHandle，CPMParamToHandle 函數，除了傳入的參數不同，其工作原理理完全相同。

② Function GetHandle(MemSize As Long)as Long

由於在比較簡單的設定中，呼叫 OCX method 時並不需要更改 methold 參數中所需要的自訂型態參數，所以一般我們只要先 GetHandle()取得一 OLE Handle 再用此 Handle 先做 GetXXX method 再做 SetXXX method 即可，但謹記程式設計者不論用 GetHandle( )或其他的 XXXPropToHandle( ) module function 取得的任何 OLE Handle，再不用時請務必把它用 FreeHandle( ) 或 HandleToXXXProp( ) module function 把它釋放掉。

2.記憶體釋放函數

① FreeHandle(Handle)

本函數可將傳入之不論是用 GetHandle( )或其他的 XXXPropToHandle( ) module function 取得的任何 OLE Handle 所指向的記憶體釋放掉。

② Public Function HandleToQueryProp(ByRef Prop As QueryProp，ByVal MemHandle As Long) As Boolean

此函數將會傳入之 MemHandle 所指之記憶體，複製到所傳入之 QueryProp 參數，並將此記憶體釋放掉，如果此函數一切工作正常，本函數將會傳回 TRUE 表示成功，否則將會傳回 FALSE 表示工作失敗。程式設計者可依此類推其他所有的 HandleToRecordProp，

HandleToCallProp， HandleToSetCtrlProp，HandleToChParam，HandleToCPMParam 除了傳入的參數不同，其工作原理完全相同。

10-1-2、Active X 與 VBX 差異說明

1.Active X custom control 介面程式只能在 Windows 95 上執行，並且與 Visual Basic 5.00 **(**含**)** 以上相容，適用於 32 位元的 Windows 程式。而 VBX 適用於 Windows31 或 95 之 16 位元程式。

2.VBX 版本中有提供一些 API 在 application 可以在 run time 時，動態地更改 process 的 property，OCX 中不再提供 API，而改以 METHOD function 提供相同的功能，method 和 API 之間比較大的差異有兩個：

① 呼叫的方法不同

method function 同 property function 般是對 control object 做呼叫，而不是像 API function 般對整個 Active X 做呼叫。

② 自訂型態參數傳遞的方法不同

由於程式與 Active X control 之間的自訂型態參數傳遞只能使用 OLE handle。所以所有自訂型態參數在傳遞前都必須做轉換。**(**轉換的函數可參考 MEMORY.BAS**)**。

3.在 VBX 版本中，都是以 process id 的數字做所有 API 中的 process 設定或 Event 中的 process notification。爲方便程式設計修改，在 Active X control 的設計上，所有 method 中有關於指定 process 做設定的參數都改爲字串，程式設計者可以選擇用 process id 數字字串或 process name 做設定，而 Event 中有關於 current process 或 previous process 的 notification 也都包含 process id 及 process name 兩種供程式設計者選擇使用。

4.在 VBX 版本中,你如果要在程式的執行過程中,必須依不同的外在環境所改變的狀況,而動態地改變 VP894CC 的工作流程,必需利用 VBX 的 API function 及在 Event handler 改變 idCurProc 這個參數的內容, 在此部份 Active X 則以 method function SetNextStep( )取代,也就是說你如果在 Exit event procedure 之前執行了 SetNextStep( ) method,當控制權回到 VP894CC(亦即 Exit event procedure), VP894CC 即會執行 SetNextStep( )所指定的 process。

5.Active X custom control 中錄放音的檔案路徑和 VBX 時不同,OCX 中全部用絕對路徑來設定音檔的位置,原本 VBX 版本中的 PromptDir 和 RecordDir property 在 OCX 中便不再提供。

6.在 VBX 版本中,SetCtrlParamProp( ) API 可以設定部分的 Channel 參數,Active X Control 中爲相容 VBX 的 API,所以也提供相同的 method 供程式設計者使用,但爲使程式設計者可以在 run-time 時設定所有的參數,在 Active X 版本中有另外提供兩個 method: GetChParam( ) 及 SetChParam( )可讀寫 channel control parameter 所有的參數。

10-2、VP894CC.OCX 的功能屬性(Properties)

在說明 VP894CC 各個 property 時，將仿照 VB 的手冊，以下面的圖案代表此 property 在 design time 及 run time 時的 accessibility：

● **read / write**

○ **read only**

○ **write only**

○ **not available**

在使用 VP894CC Active X Control 做設計時，可以屬性頁設定該 Control 的一些重要參數，叫出屬性頁的方法有兩種：一種是在設計模式下可以滑鼠右鍵點擊 Control，並在彈出視窗中選取 property，即可進入屬性頁視窗或是在設計模式下，property window 中選 custom，也可進入屬性頁視窗。

10-2-1、Status

design time ○

run time ○

Status 是一個代表此 VP894 control 目前狀態的 integer，它的值的範圍是由 0-2。2 表示此 control 所代表的 VP894 adapter 已被其他 Task 所配置使用;1 表示此 control 所代表的 VP894 adapter 未 installed 於系統內； 0 在 design time 時，表示此 control 所代表的 VP894 adapter 是 free，在 run time 時，則表示此 control 所代表的 VP894 adapter 已被配置給此 control 使用。**(**亦即此 control 可使用它所代表的 VP894 adapter 的所有功能**)**

10-2-2、AdapterNumber

design time ●

run time ○

AdapterNumber 是一個 integer，其值的內容是此 control 所代表的 VP894 卡號，在 run time 時
VP-894 active X control 即是利用此值向 VP894_32.DLL 要求配置 VP894 設備，在 design time 時使用者可以直接在 ToolBox 中更改本項參數改變該 Control 所指向的 VP894 卡片，如果所設定的 Control 不存在，Control 圖形將會變成打叉的圖形。

10-2-3、CtrlParam

design time ●

run time ○

本 property page 是用來設定此 control 所代表的 VP894 device 在 run time 初始化時的 default channel control parameters **(**此 channel control parameter 的各項資料說明，請參閱第三章 3-3 控制參數定義說明**)**，你可以依照需要修改其中的內容。此 property 僅在 VP894CC 初始化時設定其 channel control parameter，如果你必須於執行時期動態地改變其設定請使用 VP894CC METHOD function-SetChParamProp( )或 SetCtrlParamProp( )。**(**請參閱以下 VP894CC METHOD function 的說明**)**

10-2-4、FlowEditor

design time ●

run time ○

此 property page 是控制 VP894CC 執行流程的核心樞紐。共包含了兩部分：

1.Auto-Executed Step After Enable

指定當 VP894CC 的 property-Active 被設爲 TRUE 時，VP894 必須自動執行的 process ID。在 run time 時，如須動態更改自動執行的 process ID 可使用 VP894CC 的 property-AutoExecProc。

2.Process Editor

在 design time 時，你可以在這個 dialog box 上增加、刪除、編輯或修改這個 VP894CC 的 telecommunication flow control。在 VP894CC 的 flow control 中，我們是以 process 作爲一個執行單元，目前的版本中共定義了以下 14 種不同的 process types：

1.Query 2.Record 3.Call 4.Stop Channel
5.Set Control Parameter 6.Interlink
7.Flash 8.Pick Up 9.Hang Up 10.Ringing
11.Local Phone Picked Up 12.Local Phone Hung Up
13.Remote Hang Up 14.Learning Pulse Success。這些 processes 各有不同的 properties 定義此 process 如何被執行，如果粗略的區分，可以將這些 properties 概分爲兩類：

(一)共通的 property
(二)不同 process 所特有的 property。

以下是共通的 properties 的說明：

10-2-4-1、 流程單元(Process)之共通屬性(Common Properties)

1.ID

代表此 process 的 ID code，這個 ID code 也被用來在任何必須輸入 next step 的 process properties 的 edit box 中，指明此 process 的下一執行步驟，例如共通 properties 中的 Default Next Step。

2.Type

本 process 的 process type

3.Name

這個 process 的名稱。本項 property 僅是供 application programmer 輔助記憶明瞭本 process 的工作內容或用途，你可輸入任何 ASCII 字元(不包含 null)

4.Default Next Step

用來指明此 process 正常結束後，應該執行下一 process 的 ID code。如果此欄爲 none 或空白或-1，代表本 process 執行完畢後，進入 idle 狀態 (如果此 process 的 properties 中有其它 next step 的 properties，則所謂 process 的正常結束因不同的 process type 而有不同的解釋，例如 process Query、call 等，請參閱各個不同 process type 的說明)

5.Fire Event(僅 Ringing，Local Phone ，Picked Up，Local Phone Hung Up，Remote Hang Up 及 Learning Pulse Success 才具有本 property)

用來指明當 VP894CC 偵測到 Ringing，Local Phone Picked up，Local Phone Hung Up，Remote Hang Up 或 Pulse Learning Success 時，是否要 Fire VB Event 通知 VB application (Ringing ，PhonePickedUp，PhoneHungUp， RemoteHangUp 或 LearnPulseSuccess)

6. Fire Pre-Execution Event(Ringing， Local Phone Picked Up，Local Phone，Hung Up，Remote Hang Up 及 Pulse Learning Success 無此 property)

設定 VP894CC 於執行本 process 前，是否要 Fire VB Event-PreExecuted 通知 VB application。

7. Fire Post-Execution Event(Ringing， Local Phone， Picked Up，Local Phone， Hung Up， Remote Hang Up 及 Pulse Learnin Success 無此 property)

設定 VP894CC 在執行完本 process 後， 是否要 Fire VB Event - Post Executedxxx **(**xxx 代表不同的 process types**)** 通知 VB application。

10-2-4-2、 特定 Process 專用屬性(Exclusive Properties)

在接下來的文件內容中，我們將依各種不 同的 process type 分別說明其使用用途及 process-depended 的 properties：

1.Query

本 process 是用來播放一個或數個 voice files，並接收 remote 的電話按鍵輸入。你如果用 Flow Editor 的"New" push- button 產生此種 process，在 Windows 的桌面上會多顯示兩個 dialog box - Prompt List 和 Next Step Table。Prompt List 是用來設定這個 Query Process 要放音的檔案，檔案的播放順序是依照 Prompt List 中位置順序而定，你如果要改變目前的播放順序，可以先將 mouse selection 指在要改變位置的聲音檔上，可以上箭號或下箭號按鈕改變聲音檔的播放順序；Next Step Table Dialog 是用來設定此 process 若接收到其 property - Exit DTMF 中，所指定的 DTMF 碼時，而必須執行的步驟及 remote 端超過等待輸入時間時所必須執行的 process。Query 除了上述兩個 dialog box 中的 property 及 process 的 common properties 外，尚有下列三種 properties：

① Exit DTMF

設定以何者電話按鍵輸入提前終止此，假設此 Query 要接收 6 個 0～9 的數字按鍵，如果 Exit DTMF 設定了'*'或'#'當作終止鍵，則當 remote 端輸入此兩個按鍵中的任何一個，不論 remote 端先前是否已輸入足夠的 DTMF 碼，本 process 均會結束，並且依照 Next Step Table DTMF * 或 DTMF #中設定的 process ID，執行下一個 process。你可以直接在 Exit DTMF 的 edit control 中輸入你的設定，或者使用旁邊的電話圖案。**(**電話圖案上的 Keypad 如果是 pressed down，代表該鍵是 enable 的**)**

② Accepted DTMF

設定那些電話按鍵是可接受的。你可以直接在 Accepted DTMF 的 edit control 中輸入你的選擇，或者使用旁邊的電話圖案。

③ DTMF Amount

指明此 Query 要接收電話按鍵的總數。

附註：在 Prompt List 中的檔案路徑和 VBX 時不同，OCX 中只能用絕對路徑來設定因檔的位置在 Run Time 時你如果必須播放不同路徑的 voice file 或，設定或更改 Prompt List 的內容，可使用 VP894CC 的 method function SetQueryProp()；此 process 的 Default Next Step 的定義是在執行結束，沒有收到任何 Exit DTMF，而且在 Time out 前接收到 DTMF Amount 所指明的按鍵輸入個數。

2.Record

Record 是用來作錄音。你如果用 Flow Editor 的 "New"push-button 產生此種 process，在 windows 的桌面上會另外再顯示一個 dialog box - Next Step Table(有關此 dialog box 的說明，請參考 Query 的描述)。Record 的 process-depended properties，除了 Next Step Table，還包含下列 3 項：

① Exit DTMF

請參閱 Query 中 Exit DTMF 的說明。

② Length

指定錄音的長度。

③ File Name

指定錄音的檔案名稱。

附註：同 Query，File Name 所指明的錄音檔的路徑必須是絕對路徑；如果必須使用不同的路徑或於 Run Time 時，設定或修改 Record 的 properties，可使用 VP894CC 的 METHOD function-SetRecordProp( )；此 process 的 Default Next Step 的定義是在工作結束後，沒有接收到任何 Exit DTMF 時所做的 process ID。

3.Call

Call process 是用來 call 外線，分機電話或 Beeper。你如果用 Flow Editor 的"New" push-button 產生此種 process，在 windows 的桌面上會另外再顯示一個 dialog box - More Call Properties。Call 的 process-depended properties 包含下列數項：

① Called Type
指定此 process 要 call 外線，分機電話或 beeper。

② Number
指定要 call 的號碼。

③ Message
如果 called type 設定爲 beeper，此項爲 beeper 撥通後，要傳送的 message。**(**如果有 prolog 或 epilog digit，例如'#'必須包含在其中**)**

④ Called Failure Next Step
指明 call 如果失敗，必須執行的 process ID。

⑤ Don't Execute CPM Don't Wait Dial Tone Detect Invalid Number Detect 1st User-Defined Signal Detect 2nd User-Defined Signal
如果 Called Type 爲外線或分機電話，這幾項是指定撥號的模式。**(**請參考本手冊第四章 C 語言介面程式中 CallRemote( )的描述**)**

附註：Default Next Step 是此 process 執行成功**(**被呼叫者接聽或 beeper 撥號成功**)**；如果必須於 run time 設定或修改 call 的 properties，可使用 VP894CC METHOD function-SetCallProp( )。

4.Stop Channel

終止目前執行的工作。

5.Set Control Parameter

此 process type 是用來在 run time 時，改變某些 channel 的控制參數 **(**VP894CC property-CtrlParam 是用於 run time 的初始化**)**。有關此 process 的 process-depended properties 的說明，請參考 CtrlParam。

附註：此 process type 在 run time 是屬於靜態設定 **(**即其 process - depended properties 是於 design time 已指定好的**)**，如果你必須在 run time 時動態設定 channel 的控制參數，請使用 VP894CC METHOD function SetCtrlParamProp( )。

6.Interlink

此 process type 是用來設定相鄰奇偶成對的兩個 channel 的電話介面是否相互連接。

7.Flash

此 process type 是使 VP894 做一個 on hook-off hook 的動作**(**本動作在交換機內是爲了轉接的目的**)**，其 on hook-off hook 之間的 duration 由 Flash Time 決定。

8.Pick Up

本 process type 是使 VP894 佔線。

9.Hang Up

本 process type 是使 VP894 掛斷電話。

10.Ringing

本 process type 僅是一個 notification，它並沒有執行任何有關 VP894 實際的硬體動作。在 VP894 偵測到有電話 call in 時，VP894CC 根據此 process 的 property-Fire Event，決定是否要
Fire VB Event-Ringing，通知 VB application，並且依照 Default Next Step 的設定，執行下一步驟。

11.Local Phone Picked Up

本 process 是一個 notification。在 VP894 偵測到串接電話的話筒被拿起時，VP894CC 根據此 process 的 property-Fire Event，決定是否要 Fire VB Event-PhonePickedUp，通知 VB application，並且依照 Default Next Step 的設定，執行下一步驟。

12.Local Phone Hung Up

本 process 是一個 notification。在 VP894 偵測到串接電話的話筒被掛上時，VP894CC 根據此 process 的 property-Fire Event，決定是否要 Fire VB Event-PhoneHungUp，通知 VB application，並且依照 Default Next Step 的設定，執行下一個步驟。

13.Remote Hang Up

本 process 是一個 notification。在 VP894 偵測到遠端掛上電話時，VP894CC 根據此 process 的 property-Fire Event，判斷是否要
Fire VB Event-RemoteHangUp，通知 VB application，並且依照 Default Next Step 的設定，執行下一個步驟。

14.Learning Pulse Success

本 process 是一個 notification。Ctrl-Param 中的 Learn Pulse 必須 checked，你才可在 Flow Editor 產生此種 process type。在 VP894 成功完成 pulse learning 後，VP894CC 根據此 process 的 property-Fire Event，決定是否要 Fire VB Event- LearnPulseSuccess 通知 VB application，並且依照 Default Next Step 的設定，執行下一個步驟。**(**有關 pulse learning 程序的詳細說明，請參閱第四章 4-2 節內相關說明**)**

10-2-5、Active

design time ○

run time ●

本 property 是一個 Boolean integer array，共有 4 個成員，分別用來代表及設定此 control 的四個 VP894 channels 的工作狀態。當 Active 設定爲 False 時，其對應的 VP894 channel 即進入 idle 的狀態；反之， 若 Active 設定爲 TRUE，且 Active 狀態是由 False 改爲 TRUE，AutoExecProc 中若爲一有效的 process ID，VP894 即由 AutoExecProc 所指定的 process 開始執行。

10-2-6、AutoExecProc

design time ○

run time ●

本 property 是一個 integer array，共有 4 個成員，分別用來代表及設定此 control 的四個 VP894 channels 的自動執行 process ID。**(**請一併參閱 VP894CC property-Active 的說明**)**

10-2-7、CPMParam

design time ●

run time ○

本 property page 是用來設定此 Control 在 run time 初始化時的 default CPM parameters **(**此 CPM parameters 的各項定義，請參考第五章 5-3 節中有關 GetCPMParam( )的說明**)**，你可以依照需要更改其中的內容。

10-2-8、HungUpParam

design time ●

run time ○

本 property page 是用來設定此 control 在 run time 初始化時的 default 偵測遠端掛斷的參數 **(**偵測遠端掛斷的參數的各項定義，請參考第五章 5-3 節中 GetHungUpParam( )的說明**)**，你可以依照需要，修改其中的內容。

10-2-9、ToneDetectParam

design time ●

run time ○

本 property page 是用來設定此 Control 在 run time 初始化時的 default tone 偵測參數，你可以依照需要，修改其中的內容。

10-2-10、Available

design time ○ → read-only

run time ● → read-only

本 property 是針對 VP894J 卡的使用者所設計，因爲 VP894J 只有兩個 Channel，所以在對 Channel 下指令時，應先以本 Property 檢查該 Channel 是否存在，本 property 是一個唯讀的 Boolean integer array，共有 4 個成員，分別用來代表及設定此 control 的四個 VP894 channels 的存在狀態。當該 Channel 的 Available 狀態爲 False 時，表示其對應的 VP894 channel 即不存在；反之，該 Channel 的 Available 狀態爲 True 時，其對應的 VP894 channel 即可正常使用。

10-3、VP894CC.OCX 的事件(Events)說明

所有的 VP894CC events 均會傳入 ChNum，idCurProc 及 CurProcName 這三個參數，前者代表 fire 這個 event 的 channel number，這個 channel number 的定義不同於其它 METHOD(DOS-C，Clipper，Windows)或第五章 5-1 節中的定義，它的值的範圍是由 0 至 3，分別代表此 VP894 control 的 0～3 ports，你如果要 MAP 此 ChNum 至實際的 channel number，可以此 VP894 control 的 AdapterNumber 乘以 4，加上 ChNum 即得；後者表示 fire 這個 event 時，VP894CC 正執行的 process 的 ID 及名稱。在 design time 時，使用 Flow Editor 是設計 run time 的靜態 process，亦即這些 process 的內容及相互間的關聯性是預先訂製好的，在 application 的 life time 是固定不變的。你如果在程式的執行過程中，必須依不同的外在環境所改變的狀況，而動態地改變 VP894CC 的工作流程，可以利用 VP894CC 的 METHOD function SetNextStep()，也就是說你如果在 Exit event procedure 之前執行了 SetNextStep() method，當控制權回到 VP894CC(亦即 Exit event procedure)，VP894CC 即會執行 SetNextStep() 所指定的 process(其中 event Detect-DTMF 及 DetectPulse 爲例外狀況，你無法在這兩個 event procedure 中以執行 SetNextStep()來改變 VP894CC 的執行程序)。會影響 VP894CC execution flow 的因素有三：依照其優先順序分別爲 (1)VP894CC property Active 由 False 改爲 True 自動執行的程序 (2)在 run time 執行 SetNextStep( ) (3)用 Flow Editor 設計的靜態 process 內 Next Step 的設定因此假設如果以上三個狀況同時發生，VP894CC 會執行 AutoExecProc 內所指定的 process。某些 events 除了傳入以上兩個參數外，還會傳入其他的參數，下面在描述各個 event 時，會一併提出說明。

1.DetectDTMF

你如果在 CtrlParam 中 checked MonitorDTMF
(即 enable monitor DTMF 的功能**)**，當 VP894CC 偵測到有 DTMF 輸入時，即會 fire 這個 event。

其它參數：

① DTMF

VP894CC 偵測到的 DTMF 之 ASCII 碼

2.PhoneHungUp

當 VP894CC 偵測到串接的電話話筒被掛上時，而且 process-Local Phone Hung Up 中的 property-Fire Event 爲 TRUE，即會 fire 這個 event。

其它參數：

① idPreProc
VP894CC 執行本 process 前，執行的 process ID。

② PreProcName
VP894CC 執行本 process 前，執行的 process Name。

3.PhonePickedUp

當 VP894CC 偵測到串接的電話話筒被拿起，而且 process-Local Phone Picked Up 中的 property- Fire Event 爲 TRUE 時，即會 fire 這個 event。

其它參數:

① idPreProc

VP894CC 執行本 process 前，所執行的 process ID。

② PreProcName

VP894CC 執行本 process 前，執行的 process Name。

4.PostExecutedCall

發生在 call process 執行完後，且 call process property-Fire Post-Execution Event 設定爲 TRUE。

其它參數：

① Ret894Evt

VP894_32.DLL 傳回的 event type，可能的 event type 有 EVT_NO_DIAL_TONE，EVT_CPM_ COMPLETE **(**有關 VP894 的 event type 說明，請參閱第三章 3-1 節**)**

② CallResult

如果 Ret894Evt 爲 EVT_CPM_COMPLETE，CallResult 可能爲下列任一結果－
CPMR_NO_ANSWER、CPMR_BUSY、PMR_INVALID_NUM、 CPMR_USER_DEFINED1、CPMR_USER_DEFINED2、CPMR_NO_SIGNAL、CPMR_ANSWER、CPMR_NO_RINGBACK、CPMR_CALL_BEEPER_SUCCESS**(**請參考 VP894 第三章 3-1 節中 EVT_CPM_COMPLETE**)**，如果 Ret894Evt 爲 EVT_NO_DIAL_TONE，則 Call Result 無定義。

5.PostExecutedFlash

發生在 Flash process 執行完後，且 Flash process property-Fire Post-Execution Event 設定爲 TRUE。

6.PostExecutedHangUp

發生在 Hang Up process 執行完後，且 Hang Up process property-Fire Post-Execution Event 設定爲 True。

7.PostExecutedInterlink

發生在 Interlink process 執行完後，且 Interlink process property-Fire Post-Execution Event 設定爲 True。

8.PostExecutedPickUp

發生在 Pick Up process 執行完成後，且 Pick Up process property-Fire Post-Executed Event 設定爲 True。

9.PostExecutedQuery

發生在 Query process 執行完成後，且 Query process property-Fire Post-Execution Event 設定爲 True。

其它參數**:**

① Ret894Evt

EVT_EOP_NORMAL，EVT_DTMF_INTERCEPT，或 EVT_TIME_OUT**(**請參閱第三章 3-1 節中相關說明**)**

② ExitDTMF

如果 Ret894Evt 爲 EVT_DTMF_INTERCEPT，則本參數內包含終止此 Query process 的 DTMF 輸入，否則無定義

③ RxDTMFs

此 Query process 所接收到的 DTMF 輸入

10.PostExecutedRecord

發生在 Record process 執行完成後，且 Record process property-Fire Post-Execution Event 設定爲 True。

其它參數：

① Ret894Evt
EVT_EOP_NORMAL 或 EVT_DTMF_INTERCEPT

② ExitDTMF
如果 Ret894Evt 爲 EVT_DTMF_INTERCEPT，則本參數內包含終止此 Record process 的 DTMF 輸入，否則無定義。

11.PostExecutedSetCtrlParam

發生於 Set Control Parameter process 執行成後，且 Set Control Parameter process property-Fire Post-Execution Event 設定爲 True。

12.PostExecutedStopCh

發生於 Stop Channel process 執行完成後，且
Stop Channel process property-Fire Post-Execution Event 設定爲 True。

13.PreExecuted

發生在任何 process 被執行前，且該 process 的 property-Fire Pre-Execution Event 設定爲 True。

14.Ringing

當 VP894CC 偵測到有電話 call in，而且 process-Ringing 中的 property-Fire Event 爲 True 時，即會 fire 這個 event。

其它參數：

① idPreProc

VP894CC 執行本 process 前，執行的 process ID。

② PreProcName

VP894CC 執行本 process 前，執行的 process Name。

15.DetectPulse

你如果在 CtrlParam 中 checked "Monitor Pulse"，當 VP894CC 偵測到有 pulse 撥號輸入時，即會 fire 這個 event。

其它參數：

① Pulse
VP894CC 偵測到的 pulse 撥號碼

② PreProcName
VP894CC 執行本 process 前，執行的 process Name。

16.LearnPulseSuccess

你如果在 CtrlParam 中 checked "Learn Pulse"，當 VP894CC 成功完成 pulse learning 的程序，即會 fire 這個 event。

其它參數：

① idPreProc
VP894CC 執行本 process 前，執行的 process ID。

② PreProcName
VP894CC 執行本 process 前，執行的 process Name。

17.RemoteHangUp

當 VP894CC 偵測到遠端掛斷電話，而且 process -Remote Hang Up 中的 property-FireEvent 爲 TRUE 時，即會 fire 這個 event。

其它參數：

① idPreProc

VP894CC 執行本 process 前，執行的 process ID 。

② PreProcName

VP894CC 執行本 process 前，執行的 process Name。

③ Ret894Evt

EVT_LINE_SILENT，EVT_LINE_ROARING_REMOTE_HANG_UP，EVT_LINE_VOLT_REVERSE_REMOTE_HANG_UP 或 EVT_LINE_BUSY_REMOTE_HANG_UP **(**請參考第三章 3-1 節中這些 event type 的解釋**)**

18.LineMute

如果在 VP894CC 的 property-CtrlParam 有 check Monitor Tone 的選擇，則線上能量小於 ToneThreshold 的持續時間較 ToneMonitorParam 中 MinOffDuration 的設定時間爲長或相等時，VP894CC 會 fire 本事件通知 client application。

19.TonePresence

如果在 VP894CC 的 property-CtrlParam 有 check Monitor Tone 的選擇,則線上能量大於或等於 ToneThreshold 的持續時間超過 ToneMonitor-Param 中 MinOnDuration 的設定時間，VP894CC 會 fire 本事件通知 client application。

10-4、METHOD 程式說明

10-4-1、功能函數說明

VP894CC 的 METHOD function 和 VP894CC.VBX 的 API 部分類似，其中比較大的差異有兩個：

1.呼叫的方法不同

method function 同 property function 般是對 control object 做呼叫,而不是像 API function 般對整個 Active X 做呼叫。

2.自訂型態參數傳遞的方法不同

由於程式與 Active X control 之間的自訂型態參數傳遞只能使用 OLE handle，所以所有自訂型態參數在傳遞前都必須做轉換。**(**轉換的函數可參考 MEMORY.BAS**)**

method function的目的是使VB的application可以在run time 時，動態地更改 process 的 property，以下是這些 method function 的說明：

1.GetQueryProp(ProcString，QueryInfo)

取得所指定的 process 的 properties information。

輸入參數：

① ProcString
要被讀取 properties information process ID 字串或 process 名稱字串。

② QueryInfo
這是一個 OLE HANDLE 指向 User-Defined type-QueryProp 的變數**(**variable**)**用來儲存要讀取的 properties information。

2.SetQueryProp(ChNum，PromptList，QueryInfo)

設定下一個執行 process 爲 Query 及指定 Query 的 properties。

輸入參數：

① ChNum

被設定更改的 channel number。此 channel number 爲此 VP894CC 上的 channel number**(**0～3**)**，不同於第四章 4-1 節中的定義。

② PromptList

此項輸入參數可爲 Null pointer 或一字串指明此 Query 要放音的檔案。如果設定爲 Null pointer，則此 Query process 僅接收遠端按鍵輸入，而不放音；如果要放音的檔案超過一個以上，每一個檔案的路徑全名必須以空格**(**str$**(32))**分隔開。例如，假設你有以下三個檔案要播放—C:\PATH1\VOICE1，D:\PATH2\VOICE2，E:\PATH3\VOICE3，PromptList 必須如下設定：

PromptList="C:\PATH1\VOICE1"+Str$**(32)**+"D:\PATH2\VOICE2"\Str$**(32)** +"E:\PATH3\VOICE3"+Str$**(**0**)**。

③ QueryInfo

這是一個 OLE HANDLE 指向 User-Defined type-QueryProp 的變數，指明要更改的 Query properties 內容。

3.GetRecordProp(ProcString，RecordInfo)

取得所指定的 process 的 properties information。

輸入參數：

① ProcString

要被讀取 properties information 的 process ID 字串或 process 名稱字串。

② RecordInfo

這是一個 OLE HANDLE 指向 User-Defined type-RecordProp 的變數，用來儲存要讀取的 properties information。

4.SetRecordProp(ChNum，FileName， RecordInfo)

設定下一個執行 process 爲 Record 及指定 Record 的 properties。

輸入參數：

① ChNum
被設定更改的 channel Number

② FileName
指定錄音的檔案名稱

③ RecordInfo
這是一個 OLE HANDLE 指向
User-Defined type-RecordProp 的變數，指明要更改的 Record properties 內容。

5.GetCallProp(ProcString，CallInfo)

取得所指定的 process 的 properties information。

輸入參數：

① ProcString

要被讀取 properties information 的 process ID 字串或 process 名稱字串。

② CallInfo

這是一個 OLE HANDLE 指向 User-Defined type-CallProp 的變數，用來儲存要讀取的 properties information。

6.SetCallProp(ChNum，CalledNumber，CallInfo)

設定下一個執行 process 爲 call 及指定 call 的 properties。

輸入參數：

① ChNum

被設定更改的 channel number

② CalledNumber

指定被呼叫號碼的字串，如果 called type 爲 beeper，則此項參數包含被呼叫的 beeper 號碼及要傳遞的 message，中間以一空白**(**Chr$**(32))**分隔。

③ CallInfo

這是一個 OLE HANDLE 指向 User-Defined type-CallProp 的變數，指明要更改的 call properties 內容。

7. GetCtrlParamProp(ProcString， CtrlParam)

取得所指定的 process 的 properties information。

輸入參數：

① ProcString

要被讀取 properties information 的 process ID 字串或 process 名稱字串。

② CtrlParam

這是一個 OLE HANDLE 指向 User-Defined type-SetCtrlParamProp 的變數，用來儲存要讀取的 properties information。

8.SetCtrlParamProp(ChNum，CtrlParam)

設定下一個執行 process 爲 Set Control Parameter 及指定 Set Control Parameter 的 properties。

輸入參數：

① ChNum

被設定更改的 channel number

② CtrlParam

這是一個 OLE HANDLE 指向 User-Defined type-SetCtrlParamProp 的變數，指明要更改的 Set Ctrl Param properties 內容。

9. GetCPMParam(Param)

取得目前 VP894 的 CPM 設定參數。

輸入參數：

① Param

這是一個 OLE HANDLE 指向 User-Defined Type-CPMParam 的變數，用來儲存所讀取的 VP894 CPM 參數。

10.SetCPMParam(Param)

設定新的 CPM 參數。

輸入參數：

① Param

這是一個 OLE HANDLE 指向 User-Defined type-CPMParam 的變數，指明新的 CPM 參數設定。

11.GetRemoteHungUpParam(Param)

取得目前 VP894 遠端掛斷訊號偵測參數設定。

輸入參數：

① Param

這是一個 OLE HANDLE 指向 User-Defined type-HungUpParam 的變數，用來儲存所讀取的偵測參數設定。

12.SetRemoteHungUpParam(Param)

設定新的 VP894 遠端掛斷訊號偵測參數。

輸入參數：

① Param

這是一個 OLE HANDLE 指向 User-Defined type-HungUpParam 的變數，用來指明新的偵測參數。

新增 method functions：

爲使符合 VBX 使用者的需求，所以在 OCX 的 method 設計上使所有可以改變的參數都有相對應的 method 在 run-time 時可供讀寫。

13. GetChParam(Param)

取得目前 VP894 控制參數設定。

輸入參數：

① Param

這是一個 OLE HANDLE 指向 User-Defined type-ChParam 的變數，用來儲存所讀取的偵測參數設定。

14.SetChParam(Param)

設定目前 VP894 控制參數設定。

輸入參數：

① Param

這是一個 OLE HANDLE 指向 User-Defined type-ChParam 的變數，用來儲存所讀取的偵測參數設定。

15.SetNextStep(ChNum，NextStepString)

取代 VBX 時直接在 Event handler 中更改 idCurProc 而在 run-time 動態改變程式流程的 method，也就是說你如果在 Exit event procedure 之前執行了 SetNextStep( ) method，當控制權回到 VP894CC **(**亦即 Exit event procedure**)**， VP894CC 即會執行 SetNextStep( ) process **(**其中 event Detect-DTMF 及 DetectPulse 爲例外狀況，你無法在這兩個 event procedure 中以執行 SetNextStep( ) VP894CC 的執行程序**)**。

① ChNum

被設定更改的 channel number。

② NextStepString

用所希望之下一 Process 的 id 字串或所希望之下一 Process 的名稱。

16.GetFlowProp(Param，bQueryOrGet)

系統內部使用。

17.SetFlowProp(Param)

系統內部使用

18. GetToneDetectParam(Param)

取得目前 VP894CC 對電話線路上 tone 偵測的參數設定。

輸入參數：

① Param

這是一個 OLE HANDLE 指向 User-Defined type-ToneMonitorParam 的變數，用來儲存所讀取的 tone 偵測參數設定。

19.SetToneDetectParam(Param)

設定新的 VP894CC 電話線路 tone 偵測的參數。

輸入參數：

① Param

這是一個 OLE HANDLE 指向
User-Defined type-ToneMonitorParam 的變數，
用來指明新的 tone 偵測參數設定

10-4-2、可由使用者自訂的資料型態說明

結構(1)

```
Type QueryProp
    AcceptDTMF As Integer
    ExitDTMF As Integer
    DTMFAmount As Integer
    NextStepForDTMFIntercept (0 To 15) As
    Integer
    NextStepForTimeOut As Integer
End Type
```

結構成員說明：

① AcceptDTMF：

設定可以接受的電話按鍵輸入。這個 integer 中的每一個 bit 各代表 16 個 DTMF 碼中的一個，你可以使用 VP894INC.BAS 中已定義好的 global constant (comment "DTMF Mask Definition") — E_DTMF0～E_DTMF9，E_DTMF_ASTERISK，E_DTMF_ POUND 等，來設定這個 Item。例如你如果希望 Query process 接受"0"，"1"，"2"的按鍵輸入，可指定 AcceptDTMF = E_DTMF0 or E_DTMF1 or E_DTMF2。

② ExitDTMF：

指明以那些按鍵輸入終止 Query process。設定方法請參閱 AcceptDTMF。

③ DTMFAmount：

指明此 Query process 要接收幾個電話按鍵輸入。

④ NextStepForDTMFIntercept (0 To 15)：

這個 integer array 中的每一個成員各儲存一個 process ID，代表 16 個可做爲 Exit DTMF 下一步驟的一個，它們之間的對應關係如下表：

NextStepForDTMFIntercept	DTMF 碼
(0)	0
(1)	1
(2)	2
(3)	3
(4)	4
(5)	5
(6)	6
(7)	7
(8)	8
(9)	9
(10)	A
(11)	B
(12)	C
(13)	D
(14)	*
(15)	#

⑤ ***NextStepForTimeOut***

指明此 Query process 如果 time out 所必須執行的 process。

結構(2)

```
Type RecordProp
   ExitDTMF As Integer
   Length As Integer
   NextStepForDTMFIntercept (0 TO 15) As
   Integer
End Type
```

結構成員說明：:

① ExitDTMF：

指明以那些按鍵輸入終止 Record process。設定方法請參考 User-Defined type - QueryProp 中的成員 AcceptDTMF。

② Length：

設定錄音的時間長度，單位爲秒。如果本 Item 設定爲 0，代表沒有錄音長度限制。

③ NextStepForDTMFIntercept (0 To 15)：

請參考 User-Defined type - QueryProp 中的成員 NextStepForDTMFIntercept 的說明。

結構(3)

Type CallProp
Mode As Integer
NextStepForCallFailure As Integer
TargetType As Integer
End Type

結構成員說明：

① Mode：
如果此 Call process 是要 call 外線或分機電話，則本 Item 是指定此 Call process 的工作模式，其定義如本手冊中第四章、API 函數的 CallLocal()或 CallRemote()中的第三個參數 Mode。

② NextStepForCallFailure：
指定如果 call 失敗所必須執行的 process。

③ Target Type：
指定要 call 的是外線，分機電話或 beeper。在 VP894INC.BAS 中已定義此三種不同 target type 的 global constant (comment"called target type")。

結構(4)

```
Type SetCtrlParamProp
   fModified As Integer
   OffHookDelay As Integer
   OnHookDelay As Integer
   InterdigitPause As Integer
   FlashTime As Integer
   WaitAnswerDuration As Integer
   NoSignalTimeOut As Integer
   RingsToAnswer As Integer
   OffThreshold As Integer
    PlayMode As Integer
    PlayGain As Integer
   RecordMode As Integer
   RecordGain As Integer
End Type
```

結構成員說明:

fModified 爲一個 bits Mask，用來指明那些 channel control parameters 要被修改，在 VP894INC.BAS 已定義了一組 global constant 來代表要修改的項目，例如如果你要設定下列幾項 channel control parameter-OnHookDelay，FlashTime，PlayMode 及 RecordGain，可指定 fModified = modON_HOOK_DELAY or modFLASH_TIME or modPLAY_MODE or modRECORD_GAIN，其它資料成員 **(**OffHookDelay，OnHookDelay， InterdigitPause....等**)**的定義請參閱第三章 3-3 節中相關說明。

結構(5)

```
Type DutyDuration
   OffDuty As Integer
   OnDuty  As Integer
   OffDutyTolerance As Integer
   OnDutyTolerance As Interger
End Type

Type CadenceChar
   Feature As Integer
   RecognizeCycle As Integer
   WaveDuration(0 To 5) As DutyDuration
End Type

Type SignalOnLine
   Ring As CadenceChar
   Busy As CadenceChar
   InvalidNum As CadenceChar
   DialTone As CadenceChar
   UserDefined1 As CadenceChar
   UserDefined2 As CadenceChar
   End Type

 Type CPM_Param
   CentralOffice As SignalOnLine
   PrivateSystem As SignalOnLine
   Beeper As CadenceChar
End Type
```

結構成員說明:

以上幾個 User-Defined data type 是用來定義 VP894 CPM 參數，它們和 C 語言 METHOD 中與 CPM 設定相關聯的結構變數，有以下相同的對應關係：

VB		C
CPM_Param	⟶	typeCPMParam
SignalOnLine	⟶	typeCadenceType
CadenceChar	⟶	typeCadence
DutyDuration	⟶	typeDutyDuration

其中必須注意的是，因爲在 typeCadence 中的兩個成員 WaveCount 及 Type 是被宣告爲位元變數，在 Visual Basic 的 User-Defined data type 中，並未提供相同的宣告語法，所以在 CadenceChar 中是以成員 Feature 來代表 WaveCount 及 Type 的組合，因此假設如果要指明 CPM 參數內的某一訊號是 CYCLIC_WAVE，且一次週期性變化包含 3 個不同的 duty duration 組合，則可指定 Feature=3+CYCLIC_WAVE。其它的結構成員說明及設定方式，請參閱本手冊的 CPM 原理中的解釋。

結構(6)

```
Type HungUpBusy
   Varieties As Integer
   WaveDuration(0 TO 4) As DutyDuration
End Type

Type HungUpParam
   Busy As HungUpBusy
   MinBusyDuration As Integer
   MinRoarDuration As Integer
   BusyThreshold As Integer
   RoarThreshold As Integer
   MinSilentDuration As Integer
   SilentThreshold As Integer
End Type
```

結構成員說明：

以上兩個 User-Defined data type 是用來定義 VP894 的偵測遠端掛斷訊號的參數，它們和 C 語言 METHOD 中相關聯的結構變數，有以下相同的對應關係：

VB		**C**
HungUpParam	⟶	typeHungUpParam
HungUpBusy	⟶	typeHungUpBusy

有關結構成員的說明及設定方式，請參閱 C 語言 METHOD 的說明。

結構(7)

```
Type ToneMonitorParam
    ToneThreshold As Integer
    MinOnDuration As Integer
    MinOffDuration As Integer
End Type
```

結構成員說明:

這個 User-Defined data type 是用來定義 VP894 對線上 tone 訊號偵測的參數。

① ToneThreshold

指明被認可視爲 tone 的能量程度

② MinOnDuration

指明 VP894 必須偵測到線上能量變化大於或等於 ToneThreshold 的持續時間超過或等於本設定時間，VP894CC 才 fire VB event - TonePresence，通知 client application。

③ MinOffDuration

指明 VP894 必須偵測到線上能量變化小於 ToneThreshold 的持續時間超過或等於本設定時間，VP894CC 才 fire VB event - LineMute，通知 client application。

10-5、VP894CC.OCX 應用注意事項

(1) 每一個 VP894 control 僅代表一片 VP894 adapter，因此如果你的 application 中支援三片 VP894 adapter，你就必須在你的 Form 內放置三個 VP894 control。又因每一個 control 的 properties 是 private，你改變其中一個 VP894 control 的 process flow，channel control parameter 或 environment options 等，僅改變了該 VP894 control 的 property，而並不影響其它的 VP894 control，所以你如果要產生一個 application 包含多個 property 完全一致的 VP894 control，你可以先在 Form 上放置一個 VP894 control，customize 它的 properties，完成之後，利用 Edit menu 的 copy 功能，將這個 VP894 control 拷貝至 clipboard，然後利用 paste 功能產生新的 VP894 control。如果這些 VP894 control 還必須共享 event procedure 的程式碼，(亦即這些 VP894 control 有相同的 execution flow)，你必須選擇產生 control array。未來作程式修改時，也必須採用相同的方式，先刪除其它 VP894 control，僅留下一個作修改，修改完成後，才利用 Edit 的 copy paste 產生其它 control。

(2) 在你的 VP894 application 中不可以使用 InputBox 或 MsgBox 兩個 function。因爲這兩個 function 會 block 其它 message 的傳遞，直到你 close 這兩個 dialog box。如果在你使用這兩個 function 與使用者對談時，VP894_32.DLL 有 message 要傳送給 VP894CC 的 control，這些 message 將無法成功地到達 VP894CC 的 control。但你仍然可以使用 form 來產生 dialog box(因爲使用 Form 產生 dialog box 並無以上問題)。

(3) 當你在 VB 的整合環境中除錯，如果程式的執行因爲你所設定的中斷而進入 VB 的中斷模式，此時 VP894_32.DLL 或 VP894CC.OCX 所產生的任何 event (message) 均會被 VB 爲了同步及 event serialization 的理由而被 block out，因此當你由中斷點繼續往下執行，極可能發現有許多 channel 不再工作 (因爲 VP894 的運作亦是採用 event-driven scheme，倚靠 message 的緊密傳遞，達成 process 與 process 之間的連接) 這是正常現象。你可利用剩下仍正常運作的 channel (s)繼續除錯 (通常會僅剩一線仍正常工作) 或 restart program。

10-6、問題與解答

1.問題:在 Visual Basic 內發生無法載入 VP894CC.OCX 的錯誤。

答：① 請檢查你所使用的 Visual Basic 版本是否爲 5.00 或 5.00 以上。

② 請排除所有硬體衝突的可能性。你可以利用在 "VP894CC Demo"program group 中的 VP894 device diagnosis 來確定 VP894_32.DLL 是否能找到系統內的 VP894.VXD 及 VP-894 設備？ **(**如果你在 install VP894CC 並未選擇讓 setup 爲你產生 "VP894CC Demo" program group，你可直接使用 program manager 或 file manager 的 Run 命令啓動在你 install VP894CC 目錄下的可執行程式—chkadptr.exe**)**如果 VP894 的設備不能被偵測到 ，則表示有硬體相衝突的狀況發生，你可以參考本手冊第二章，改變 VP-894 的硬體設定，或者更改其它硬體設備的設定。

2.問題：啓動 VP894CC 的 demo 程式後，demo 程式對所有有關 VP-894 設備的功能均無反應。

答：請確定所執行 demo 程式內的 VP894 control 的 property-AdapterNumber 是否與你 PC 系統內 install 的 VP-894 設備的硬體組態相符。在 VP894CC install disk 上所有的 demo 程式內的 VP894 control 的 property-AdapterNumber，預設均爲 0，如果在你 PC 系統內的 VP894 實體卡號設定不爲 0，你可以在 Visual Basic 內更改 demo 程式的 VP894 control，使其 property-AdapterNumber 與實體設備相符，重新 Make EXE 檔即可。

3.問題：執行 VP894CC 的 demo 程式時，在流程處理 Query process 均無語音被播放出來。

答：請確定所執行的 demo 程式內的 VP-894 control property-environment 的 Prompt Directory 設定，是否符合你 install VP894CC demo 程式的路徑。如果你在 install VP-894CC 時接受 setup 所建議的 default destination path，則 Prompt Directory 的設定即與實際提示語音檔案所儲存的路徑相符，否則你必須更改所有 demo 程式的 VP894 control 的 Prompt Directory。

4.問題：如果當使用 METHOD function-SetQueryProp( )更改 process 的 properties 時，Visual Basic 顯示一個 error message box "This channel is not exist..." ，但是所傳給 SetQueryProp( )的 channel number 是肯定正確的?

答：請檢查傳給 SetQueryProp( )function 的第 3 個參數 PromptList 的資料類型是否被明確地宣告爲 String。 因爲我們爲了能讓 client application 在呼叫 METHOD function SetQuery-Prop( )時能夠指定 PromptList 爲 Null (亦即指定此 Query Process 僅接收 DTMF，而不播放任何語音)，所以在 VP-894INC.BAS 中宣告 SetQueryProp( )的 prototype 時，其參數列上的 PromptList 是宣告爲 ANY。在正常情形 SetQueryProp( )被呼叫時，應該是以 4 bytes 的遠程指標傳送 PromptList 的地址給 VP894CC.OCX，如果 PromptList 是被明確地宣告爲 String， Visual Basic 會正確地傳遞此項參數給 VP894CC.OCX，但是如果 PromptList 沒有被明確地宣告爲 String 的資料類型，Visual Basic 是以 default(內定)的資料類型 -Variant 視之，因此在 SetQueryProp( ) 被呼叫時，PromptList 是被轉換爲 2 bytes 的 integer 傳送給 VP894CC.OCX，如此即造成 VP894CC.OCX 所接收到的參數位元組總數少於正常的數目，於是 SetQueryProp( )在 interpreting 參數 ChNum 時，被 shift 到參數 VP894 control object 上，而認爲 application 所傳送的 ChNum 有誤。有關正確的實作，請參考 Depulse 的 demo 程式。