

應用非同步傳輸原理在遠端輸出入之控制

Application of Asynchronous Communication in Remote I/O Control

關鍵詞

Serial Interface 串列介面

Asynchronous Communication 非同步通訊

EDIO (Exquisite Digital Inputs/Outputs) ASIC 數位輸出入控制晶片

摘要

在微控制器與周邊裝置通訊應用中，對於低成本、高速傳輸和高可靠度通訊品質的要求也不斷地成長以滿足這些應用，目前串列傳輸模式應用非常廣泛。本文說明串列傳輸原理，包含串列通訊方式與資料鏈結控制方法，並且介紹 8 種常用的串列介面，分別為 RS-232C、RS-422、RS-485、I²C、SPI、Microwire、1-Wire 及 USB。這些串列介面在速率、實體介面要求和通訊方法上都有所不同。後段說明工研院所開發之數位輸出入控制晶片 EDIO ASIC 的訊號傳輸協定、通訊速率以及其介面電路。

Abstract

In order to satisfy the application of microcontroller communication with peripheral device, low cost, high speed transmission and high quality of communication is requested constantly. At present, serial communication is very popular. This content illustrates that the principle of serial communication, include serial duplex configurations and data link control. And introduce eight kinds of serial communication interface commonly used, it is RS-232C, RS-422, RS-485, I²C, SPI, Microwire, 1-Wire and USB. The requests of these buses in the speed, physical interface and means of communication are different to some extent. Next section, explain communication protocol, transmission baud rate, and circuit of ITRI development Exquisite Digital Inputs/Outputs ASIC.

前言

微控制器是目前各種先進電子產品的核心，它需要與一個或多個周邊設備通訊，目前常用的通訊方式為並列或串列模式，而串列模式應用更為廣泛。在多種情況下需要使用串列介面，最常見的是在開發或現場使用時必須與PC進行介面通訊。多數PC帶有與周邊相接的串列介面匯流排，只有少數例外，對於必須與通用

電腦介面通訊而言，使用串列介面常常比ISA或PCI匯流排更為方便。串列通訊只需一個I/O接腳便可進行通訊，而並列通訊則需要8個或更多，串列相較於並列的主要優點是要求的線數較少，較少的線意味著所需要的控制器接腳較少。整合在一個控制器中的並列匯流排一般需要8條或更多的線，線數的多少取決於設計中地址和資料的長度，所以整合一個並列匯流排的晶片至少需要8個接腳來與外部元件結合，這增加了晶片的總體尺寸；相反地，使用串列匯流排可以將同樣的晶片整合在一個較小的封裝中，而在PCB板設計中平行匯流排需要更多的線來與其它週邊設備介面連接，因此PCB板面積更大、更複雜，也增加了硬體成本。串列資料傳輸又可分為同步與非同步二種，非同步串列傳輸由於減少了時脈信號，對於需要遠距離傳輸的裝置而言，更可降低硬體、線材等成本。

一、非同步傳輸原理

資料傳輸模式

通常資料傳輸模式可以區分為兩種，其一為並列傳輸式通訊(Parallel Communication)，另一種則為串列傳輸式通訊(Serial Communication)，這兩種不同的通訊模式以下圖表示。

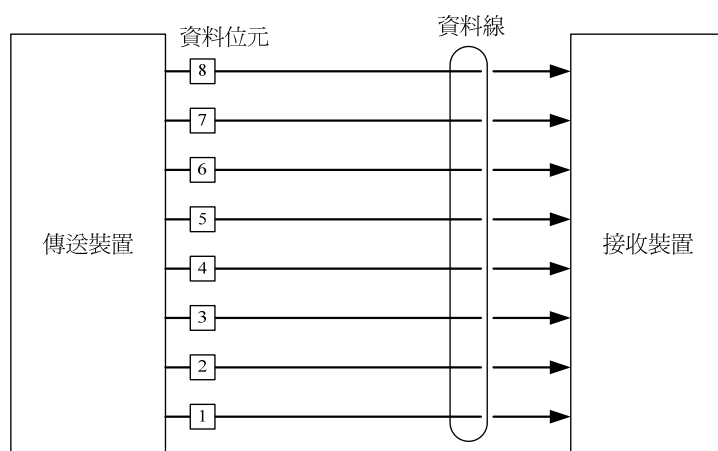


圖 1 並列資料傳輸

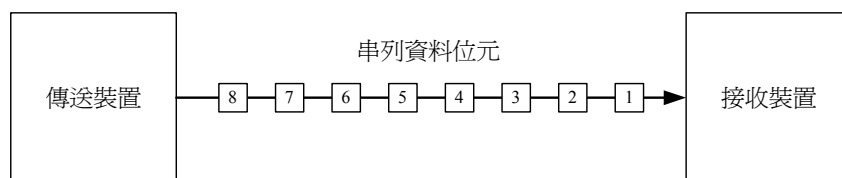


圖 2 串列資料傳輸

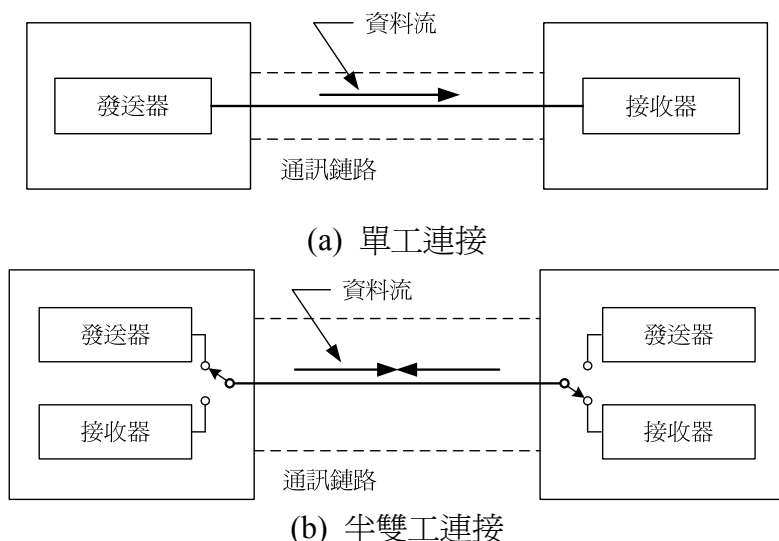
在並列資料傳送中，如圖 1 所示，最少有八個資料位元被同時從某個裝置中移動到另一個裝置中。如圖 1 所強調的，發送裝置傳送八個資料位元使用八條分開的資料線稱為資料匯流排(data bus)。以此例而言，更精確的說法是 8 位元資料

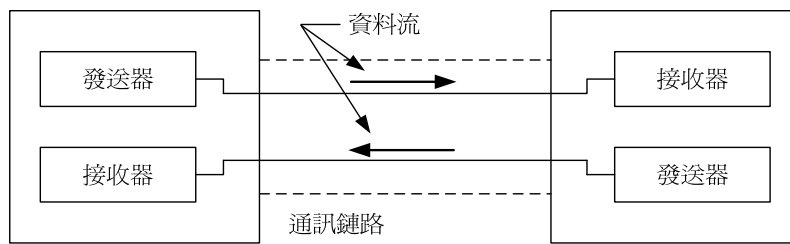
匯流排，接收裝置用同樣的八條資料線接收資料，接收裝置不需要修改或轉換就可使用這些資料。

在串列資料傳送中，如圖 2 所示，資料在同一時間內只傳送一個位元，與同一時間內傳送一組位元的並列通訊方式比較，這種通訊方式不屬於快速傳送資料的方式，而在實際應用上，也不比並列傳送快。但為什麼大家都選擇這種方式，而不選擇並列通訊呢？答案與許多其他設計硬體的問題的答案相同。使用者選擇串列資料傳送是因為硬體製作上經濟性與實用性的緣故。在兩台機器間做並列資料傳輸，至少需要包含八條資料線的電纜。對於短距離(少於 30 公尺)通訊，這種連接的方法還在合理的成本範圍內。對長距離通訊而言，使用現存的電話裝備要比重新設立並列電纜及信號放大器經濟實惠多了。

串列雙工結構(Serial Duplex Configurations)

圖 3 列出三種透過電話線連接常用的串列通訊方式。在圖 3(a)中，單工(simplex)的結構只能讓資料朝一個方向流動。所以有時也稱為單向資料匯流排(unidirectional data bus)。在圖 3(b)中，半雙工(half-duplex)結構可讓資料在二個方向中流動，也稱為雙向資料匯流排(bidirectional data bus)。半雙工結構較單工模式為優，因為資料可在二個方向流動，但是半雙工的設計僅允許資料一次朝一個方向流動，因此半雙工就是可交換方向的單工。第三種結構，如圖 3(c)稱為全雙工(full-duplex)結構(有時候只簡稱雙工)，它允許資料同時朝二個方向流動，因此全雙工即是二個單工結構，它需要雙方的通訊裝置都有全套且獨立的發送及接收能力。





(c) 全雙工連接

圖 3 資料流的方向

在串列資料傳送中，另外一個重要的因素是資料傳輸及接收的時序(timing)問題。在並列資料傳輸中，八個或更多的位元同時從起點移動到終點，不會留下任何位元組中的位元。而在串列資料傳送中，每一個資料位元循序的從起點移動到終點，這種在起點與終點之間的技術需要同步(synchronization)才能分別出位元、字元(位元組)及訊息。沒有同步的話，接收裝置將收到一連串沒有意義的信號。資料傳輸信號的同步化為將接收信號解釋成有意義的資訊所必需的。

資料鏈結控制(Data Link Control，簡稱 DLC)管理在兩個傳輸站之間同步化的問題。資料鏈結控制需要傳輸的位元、字元、或訊息與資料一起就像資料從一站走到另一站。通訊硬體或軟體增加或刪除這種資訊，而他們的操作對用戶而言是不可見的。同步時序的資訊包含在發送及接收站的硬體中。如此可保證發送站傳送的信號可以正確的被接收站辨認。在通訊過程中，1 及 0 不能被扭曲。

電腦廠高使用 DLC 的二種方法來控制電腦間資料通訊的同步化。一種為非同步(asynchronous 也稱為開始-----結束)，另一種為同步(synchronous)。資料鏈結控制中，非同步的方法在數種協定中被廣泛的應用。至於資料鏈結控制中同步的方法，可支援三種定義訊息交換的協定。這三種協定分別為二進制同步通訊(Binary Synchronous Communications，簡稱為 BSC)、同步資料鏈結控制(Synchronous Data Link Control，簡稱為 SDLC)，以及高階資料鏈結控制(High-level Data Link Control，簡稱為 HDLC)。

非同步資料正如它的名字一般，不是一種連續同步的位元流。通訊硬體會每個傳送出的位元組加上同步資訊。如圖 4 所示，每一個非同步位元組由一個開始位元當開頭，告訴接收裝置開始測量後面的資料為 1 或 0。一個在資料線上的高電壓信號永遠在開始位元之前。這種記號線(marking line)或記號狀態(marking state)提供了開始位元非常清楚的對比，讓接收裝置能偵測出一個新的開始位元。在每八個資料位元的最後一個位元稱為結束位元(stop bit)。(可能有一個、一又二分之一個或二個結束位元在每個字元的結尾。結束位元是一個高電壓的記號，它告知接收裝置一個位元組的資料已經結束。它將資料線回復到記號狀態以準備下一個

資料位元組。開始及結束位元將每個傳送位元組的 1 及 0 框起來，以清楚的將它們從資料位元組中分離出來。

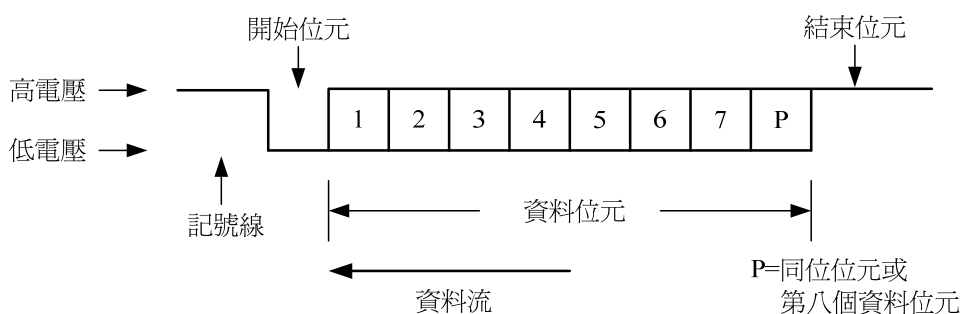


圖 4 非同步資料位元組及結構位元

為了與單獨框架(frame)的非同步位元組對比，同步資料是以區塊傳送，同步資訊在長距離通訊時則包含在傳送的區塊中，若是區域性傳送則另外提供一條資料線供同步資訊使用。圖 5 為 SDLC 協定的資料結構。這個協定在每個資料區塊的開始及結束處都提供了清楚的記號。這個協定並不為單獨的 8 位元資料做框架。事實上，同步通訊並不全然是字元導向。資料鏈結控制的同步協定可以是字元導向也可以是位元導向。字元導向協定只辨認資料位元組為特定的字元，然而位元導向協定並不將傳送的資料位元強迫指定為字元。

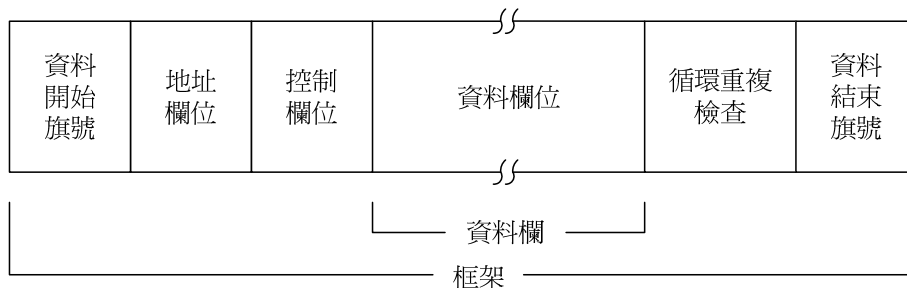


圖 5 同步資料鏈結控制(SDLC)

非同步通訊(Asynchronous Communication)

如前面所解釋的，非同步通訊為串列通訊傳送。其特性為不規則的傳送資料段。這種型式的資料鏈結控制屬於字元導向式。每一組傳送的資料位元構成一個字元且每個字元有自己的框架資料(framing data)。開始位元及結束位元將字元的資料位元框起來，其動作就如同接收裝置的同步旗標(synchronizing flags)。在收到開始位元後，接收裝置必須能夠正確地測量出全部九個位元(八個資料位元及一個結束位元)並判斷每一個為邏輯上的 1 或 0。因為這種字元接著字元重新同步化(character-by-character resynchronization)，非同步通訊容許發送器與接收器之間些微許的時序誤差。正如你所期望的，這種同步方法有助於降低成本。下面將提供

非同步資料通訊中，有關於框架位元，同位位元及資料位元更詳細的資料。

開始位元(Start Bit)

如圖 4 所示，閒置的非同步資料線保持在記號狀態，直到有資料開始在線上移動。線上電壓為邏輯 1 者代表記號信號。資料線的電壓值改變為邏輯 0 持續一個位元時間則為開始位元。開始位元信號就是告訴接收裝置"醒醒吧！"開始測量資料位元。這個信號簡單的表示字元的開始；它並不包含資訊說明後面資料的長度或型態。接收器只知道開始計算位元直到收到另外一個信號才停止。

電話線上的雜訊可能會產生與開始位元有關的問題，硬體就必須設法消除問題。線上雜訊的產生可能因電子風暴或故障的電話裝備使接收器收到一個邏輯 0 以為開始一個字元傳送。像這種錯誤解釋的雜訊會導致通訊上的錯誤，因為根本沒有資料位元跟著這個假的開始位元。非同步通訊硬體利用捕捉錯誤開始位元的方式消除這類型的通訊錯誤。通訊硬體在記號狀態時，經常檢查資料線路電壓。這種電壓檢查可讓硬體正確的測量出位元信號的期間，並且不管那些達到線上電壓卻不是開始位元期間的信號。

資料位元(Data Bits)

在通訊硬體傳送開始位元之後，就開始傳送資料位元。圖 6 說明非同步資料鏈結控制所提供的五、六、七、及八個資料位元的結構。這個圖同時也說明每種資料格式能表示字元的最大數目。資料位元所需的數目可因遠端電腦或你想傳送的資料型態而定。美國資訊交換標準碼(American Standard Code for Information Inter change，簡稱 ASCII)，是目前世界上最廣泛使用的電腦字元碼。

資料位元數目	串列資料結構	位元型式最大數目	字元最大數目	是否支援 8 位元 ASCII
5	0xxxxxP1	2^5	32	否
6	0xxxxxxP1	2^6	64	否
7	0xxxxxxxxP1	2^7	128	否
8	0xxxxxxxx1	2^8	256	是

資料結構範例：

圖 6 多種資料位元

同位位元(Parity Bit)

不管使用那種資料線路，在串列資料傳送中都常發生錯誤。因此，良好的資料通訊就需要偵測出這些錯誤的方法，非同步資料鏈結控制就使用一個同位位元來偵測錯誤。非同步通訊提供字元間斷性的傳送，而同位位元則是在字元接字元的基礎上做錯誤偵測。

除了取代傳送相同字元兩次以偵測傳輸錯誤的方式外，同位位元以單獨一個位元描述送來字元的一些資訊。非同步通訊硬體計算字元的資料位元中所有邏輯 1 的總數，然後依據資料位元總數是奇數或偶數決定同位位元的值。如果硬體的運作是使用偶數同位，則當字元中所有 1 的總和為偶數時則硬體會將同位位元設為 0，如果 1 的總數為奇數時，則同位位元設為 1。如果硬體是以奇數同位運作，則當 1 的總數為偶數時，同位位元設為 1，反之則設為 0。表 1 為決定同位位元的例子。

表 1 同位位元的決定

ASCII 字元	字元資料的位元值	偶數同位的同位位元值	奇數同位的同位位元值
A	1000001	0	1
D	1000100	0	1
F	1000110	1	0
DEL	1111111	1	0

同位錯誤偵測法是檢查單一位元錯誤(Single-bit error)的簡便方法，但它也同樣會失效。雖然硬體能標明單一位元錯誤，但卻無法標明多位元錯誤(multibit errors)，這種情況大多在資料通訊線上發現因雜訊所產生的錯誤。譬如，PC 傳送一個"A"，但因電話線上的雜訊可使另一端的 PC 收到一個"D"。在這個例子中，無論如何計算同位位元，傳輸的錯誤都不會被發現。因為使用奇同位或偶同位其結果都一樣。非同步通訊中，使用同位位元做為錯誤偵測的限制可以克服，通常可以使用包含更複雜的錯誤偵測法(計量檢查(checksum)或循環重複檢查(CRC))的檔案傳輸協定。在下一段會有更詳細的介紹。

結束位元(Stop Bits)

與非同步通訊有關的最後一個二元信號為結束位元。結束位元是跟在資料位元及同位位元之後，且構成字元傳送的結束。開始位元是構成字元的開始框架而結束位元則是構成該字元的結束框架。結束位元是具軟體選擇性的。共有 1、1.5 或 2 位元的長度。它們是邏輯 1 以保證在下一個開始位元(邏輯 0)來臨之前資料線處於記號狀態。這個在每個新字元之前的記號狀態是提供偵測每個新字元開始所必需的。如果沒有正確偵測出開始位元，就無法偵測及消除假的開始位元，也無

法達到同步化並測量資料位元。

非同步通訊的速度配合(Asynchronous Communication Speed-matching)

在串列資料傳送時，假設資料的移動是直接從一部電腦到另一部電腦儲存或顯示，但這並不是唯一的情況。如果資料能夠到另一部電腦並立刻印出、顯示或儲存，在串列資料傳送中將沒有任何的延遲及中斷。這些運作速度上不同的差異導致通訊上流量控制(flow controls)的需求。資料流量的需求與系統的選擇及通訊的速度有極大的關係。

資料流量的速度匹配技術有許多種型式，但大部份可被歸類為開關(on-off)資料流量切換或是暫時資料儲存區二類。

通訊緩衝區(Communication Buffers)

通訊緩衝區是設在隨機存取記憶體中暫時儲存資料以改變接收及處理資料速率的差異。通訊軟體在發送端及接收端都有提供緩衝區，但通訊緩衝區這個名詞通常只與接收端的緩衝區有關。下面段落中我們描述資料的流向，從電腦的磁碟檔案開始，一直到儲存到遠端電腦的磁片中結束。這可讓你深入了解發送及接收端緩衝區的運作情形。

當 PC 傳送一個檔案到遠端電腦時，它首先從自己的磁碟中讀取檔案。在這時，PC 以區塊的方式從檔案拷貝到傳送緩衝區。PC 從檔案循序的拷貝資料直到填滿傳送緩衝區，然後暫時停止拷貝，此時一個中斷驅動程序將緩衝區的內容透過串列埠送到遠端電腦。

資料在傳送緩衝區以先進先出(first-in first-out)的原則進行資料傳送；它們以從拷貝的檔案之相同順序傳出。當中斷程序減少傳送緩衝區的資料達事先訂定的低點時，PC 從檔案讀取更多的資料並放在傳送緩衝區中。新的資料會放在緩衝區剩餘資料的後面，以確保傳送順序的正確性。這個過程會持續到檔案所有資料都被傳送出去。

通訊軟體會決定傳送緩衝區的大小。如果傳送緩衝區太小，資料傳送到遠端電腦是一陣一陣的而非連續的。傳來的文字間斷的顯示或檔案傳輸太慢會讓用戶相當懊惱，但並不影響通訊的完整性，除非軟體設計者沒有提供資料到傳送緩衝區的流量控制。如果通訊軟體只簡單的將資料拷貝到緩衝區而不檢查緩衝區剩餘的空間，則傳送緩衝區很可能發生溢位(overflow)的現象。被溢出的資料永遠不會被送到遠端電腦。

雖然發送端及接收端的緩衝區大小對軟體設計都非常重要，但接收緩衝區(receive buffer)卻更為重要。如果接收緩衝區太小而且在通訊期間被填滿，則 PC

將會流失資料。當新的資料過來時，PC 就沒有地方可以放置他們。

接收端的運作與發送端極為相似。然而過程要比傳送緩衝區稍微複雜些。傳送緩衝區的大小若可配合高速電腦運作，則在低速電腦中也同樣運作的很好。但相反的，接收緩衝區就必須設計給低速電腦使用。如果電腦改以較快的速度運作，接收緩衝區仍然可以處理，而且會有比計劃中更多的剩餘空間。

接收緩衝區將通訊線路上傳來的資料暫時儲存，直到通訊軟體可以處理它們。終端機模擬器軟體通常顯示、印出、或儲存這些資料在磁碟檔案中或記憶體中。顯示或儲存資料是快速的運作，不會對接收緩衝區造成太大負荷。同時顯示及印出接收的資料則非常消耗時間，而且可能造成緩衝區溢位。當接收緩衝區溢位而導致 PC 錯失傳入的資料時，流失的資料稱為"進入位元桶中"或"掉在地上"。

中斷處理器(Interrupt Handlers)

通訊中斷處理器是資料在非同步通訊硬體或內建數據機流動的交通警察。它們控制資料從隨機存取記憶體到非同步硬體之間的流入流出。它們讓硬體達到最大的工作效率，同時它們也包含交握信號(handshake signaling)技術以保障緩衝區不會溢位。

傳送緩衝區及發送端中斷處理器可視為非同步通信硬體的傳送部份。當通訊軟體有一個字元的資料準備傳送，它會將此字元放入傳送緩衝區固定的記憶體中，如果正確的信號出現在通訊硬體上，發送端中斷處理器就將此字元帶離傳送緩衝區送到非同步硬體上。非同步硬體將此字元從並列格式轉為串列資料並送出埠。發送端中斷處理器必須在傳送緩衝區與接收裝置之間規劃好以免傳送緩衝區發生溢位。

同樣的，接收緩衝區與接收端中斷處理器為非同步通訊硬體的接收部份。當一個字元從非同步硬體接收暫存器收到後，硬體將格式從串列轉回並列然後發出中斷信號，接收端中斷處理器立刻將此字元移入接收緩衝區的記憶體中。通訊軟體看著這個緩衝區，並處理在裡面發現的資料。接收端中斷處理器必須執行的夠快才能在新的資料蓋過之前將資料從非同步硬體中取走。這個中斷處理器也需要在發送裝置與接收緩衝區之間規劃以避免發生緩衝區溢位現象。

因為發送與接收端的中斷處理是獨立運作的，所以通訊軟體必須根據用戶的輸入正確的啟動與停止它們。一旦啟動後，如果沒有正確的結束中斷處理器會產生問題。通訊軟體啟動的第一件事是定義中斷並啟動它們。當結束軟體執行的最後一件事是關掉通訊中斷處理器並釋放通訊緩衝區讓別的軟體可以使用，如果通訊軟體結束而沒有關掉中斷處理器時，它們會在通訊軟體被其他軟體取代後仍繼續執行。如果緩衝區沒有被釋放，就會減少其他軟體的可用記憶體。

實際的限制有時候使通訊軟體廠商不能提供適當的通訊緩衝區或中斷處理器處理所有速度匹配的需求。解決這個問題方法是利用別種速度匹配技術稱為資料流量管制(data flow throttling)。廠商使用不同的流量控制技術以保持資料到達率不會快過能處理的速度，而不必提供暫存資料區。最常用的流量管制為 XON/XOFF 傳輸協定。

XON/XOFF 控制(XON/XOFF Control)

XON/XOFF 資料流量控制協定是比通訊緩衝區更為積極的資料流量控制。它是主動而非被動的技術。通訊緩衝區的能力並不足夠，但 XON/XOFF 的流量控制切換使資料流為開或關以避免緩衝區溢位。

當通訊軟體或硬體在發送端及接收端提供 XON/XOFF 協定時，此協定不需要用戶的協助就可以控制資料流量的速度匹配。當暫存於通訊接收緩衝區的資料量接近緩衝區的容量時，軟體就送出 XOFF 到主電腦。主電腦收到 XOFF 後就暫時停止資料傳送，讓 PC 可以消化在接收緩衝區的資料。當緩衝區空到預設程度時，PC 軟體就送一個 XON 字元到主電腦。讓主電腦重新傳送資料，這種循環可在傳送檔案時自動重複許多次。

RTS/CTS 控制(RTS/CTS Control)

另外一種與 XON/XOFF 相似的速度匹配技術為要求發送(Request To Send 簡稱 RTS 及清除發送(Clear To Send 簡稱 CTS)協定。這種協定是在系統與數據機或其他通信裝備間的電子信號及流量控制邏輯。RTS/CTS 控制是一種主動的技術用來保持 IBM PC 系統不會侵佔到它所連接的通信硬體。當適當的使用在硬體及軟體上時，當 IBM PC 系統準備要送資料到接收裝備時，發出 RTS 信號。當接收裝備準備好接收資料時，它發出 CTS 信號。發送器在收到 CTS 信號之後才會傳送資料。因此接收裝備可以主動控制資料的接收而防止裝備被侵佔的情形。

檔案傳輸協定(File Transfer Protocol)

在電腦之間傳輸的檔案需要特殊的通信技術來偵測及更正錯誤並維持外部資料輸入至檔案中。一項簡單但耗時的檔案傳輸技術可確保檔案的完整性，就是每個檔案至少傳送兩次。你可以將檔案傳至另一部電腦兩次，並以兩個不同的檔案名稱儲存。然後你比較這兩個檔案確定它們的內容相同，如果兩個檔案的大小相同而且比較後內容相同，則你可以信賴這兩個檔案包含的資料沒有通信所產生的錯誤。如果這兩個檔案不同的話，就無法判斷那個檔案正確；也可能兩個檔案都有錯誤。使用這種技術唯一能讓你確定資料傳送是正確的方法為連續不斷的傳送同一個檔案，直到有兩份是完全相同的為止。

為了使檔案傳輸時錯誤偵測及更正的過程易於管理，通常將資料分割為數個

封包傳送。這些封包中包含了錯誤偵測的資訊可讓接收端電腦判斷是否有通信所產生的錯誤。除了將封包傳送兩次並比較兩者之外，傳送端電腦可以從欲傳送的資料導出一些資訊，並將這些資訊放入封包中。檔案傳輸協定計算一個位元組稱為計量檢查(checksum)或是計算一對位元組稱為循環重複檢查(CRC)，並將這些資訊加入封包的尾端。接收端電腦執行相同的計算並比較接收到的封包內的計量檢查或CRC值。如果比較的結果良好，資料傳輸就可以繼續下一個封包。如果比較的結果不對，則發送端就必須重新傳送資料封包。檔案傳輸協定必須重複這個過程，直到整個檔案傳送完畢。

CRC這類的封包檢查比計量檢查更為可靠，而被大部份非同步通訊所採用。CRC技術處理的方式是將封包內的資料當成一串的位元。它先工作在第一個位元組的低階位元及最終在最後一個位元組的高階位元。CCITT所建議的16位元CRC是將封包位元串除以多項式($X^{16}+X^{12}+X^5+1$)後的餘數，其中X的值為2。此項技術會偵測出所有單一以及雙位元錯誤，所有的封包錯誤為奇數位元，所有的錯誤短於16個位元，而百分之99.9的錯誤超過16位元。這個等級的錯誤偵測是較被嚴謹的商業檔案傳輸所喜愛。

二、串列介面應用

RS-232C

RS-232C 是電子工業聯盟(EIA)制定的串列通訊的介面標準，這個標準的目的是在定義資料終端設備(Data Terminal Equipment 簡稱 DTE)和資料通信設備(Data Communication Equipment 簡稱 DCE)之間的電氣特性。對 PC 用戶而言，上述二項裝備分別指 PC 及數據機。

其他有關於 RS-232C 標準應用於 PC 的項目為位元率及電纜長度。此標準提供串列資料傳輸率範圍為 0 到 20 kbps，典型的位元率是 300、1200、2400、9600、19200 等 bps。標準的電纜長度限制為 15 公尺，這通常也包含大多數 PC 通訊硬體的架構。電纜長度超過 15 公尺也可以使用(通常用在大型電腦系統)，但必須徹底測試以保證信號品質。

為保證資料能夠正確傳送以及設備控制能正確執行，信號的使用有統一的必要。RS-232C 標準提供了資料信號及控制信號的電壓範圍以滿足這項需求。這些範圍列於表 2 及圖 7。

表 2 RS-232C 交換電壓標準

交換電壓	二元邏輯狀態	信號情況	介面控制功能
正	0	Space	開



圖 7 RS-232C 信號電壓範圍

RS-232C 標準介面有 25 條信號線，4 條資料線、11 條控制線、3 條時鐘線、7 條備用和未定義線，常用的只有 9 條，它們分別是：

傳送資料(TD)

此信號是從資料終端設備(DTE)傳送資料給資料通信設備(DCE)；當沒有資料傳送時，這個電路維持在記號(mark，邏輯 1)的情況下。

接收資料(RD)

此信號是從資料通信設備(DCE)傳送資料給資料終端設備(DTE)；當沒有資料傳送時，這個電路也維持在記號(mark，邏輯 1)的情況下。

要求傳送(RTS)

此電路是用來傳送一個信號到資料通信設備(DCE)要求准許將 TD 接腳的資料送出。

清除傳送(CTS)

此電路是資料通信設備(DCE)告訴資料終端設備(DTE)已經準備好接收資料了。當此電路為 OFF 時(負電壓或是邏輯 1)，DCE 告訴 DTE 它尚未準備好接收資料。

信號接地(GND)

此電路是做為所有其他通訊電路的一項信號參考點。

載波偵測(DCD)

當接收到遠端設備正確的載波信號時，資料通信設備(DCE)傳送給資料終端設備(DTE)一個 ON 的信號。

資料終端備妥(DTR)

當資料終端設備(DTE)準備好與資料通信設備(DCE)連線時會將電路切換為 ON。

資料設定備妥(DSR)

當資料通信設備(DCE)準備好與資料終端設備(DTE)連線時會將電路切換為 ON。

振鈴指示(RI)

這個電路是用在資料通信設備(DCE)指示電話鈴聲的信號。此電路在響鈴時為 ON，鈴聲之間則為 OFF。

圖 8 為這些信號運作的流程圖。圖 9 則是 RS-232C 典型連接方式，RS-232C 為點對點通訊(一接收端、一發送端)，接收、發送端的信號是相對於信號地，其驅動器負載為 $3k\Omega\sim 7k\Omega$ 。信號的方向是從 DTE 設備的角度出發的，TD、DTR 和 RTS 信號是由 DTE 產生的，RD、DSR、CTS、DCD 和 RI 信號是由 DCE 產生的。圖 10 為 RS-232C 傳遞一個 ASCII "K" 字元碼(0x4b)的電壓準位波型圖，其中傳輸方式為 1 個開始位元、8 個資料位元及 1 個結束位元。表 3 是列出一般使用的 RS-232C 中的信號和腳位分配。

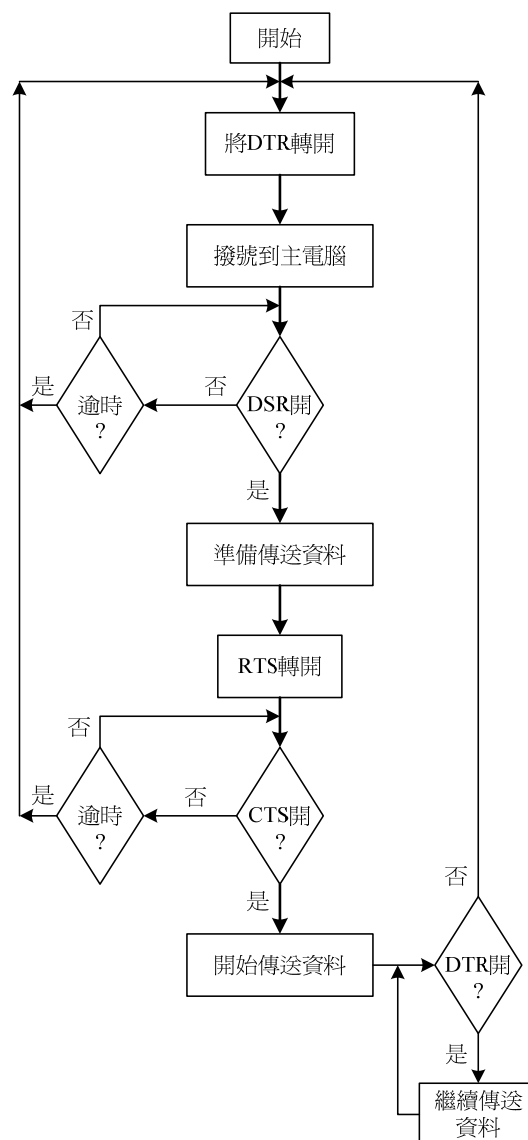


圖 8 信號運作的流程圖

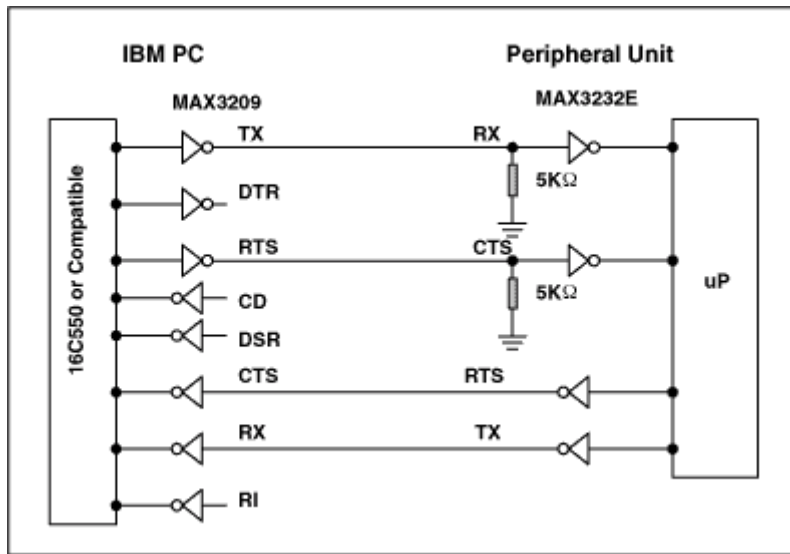


圖 9 RS-232C 典型連接方式

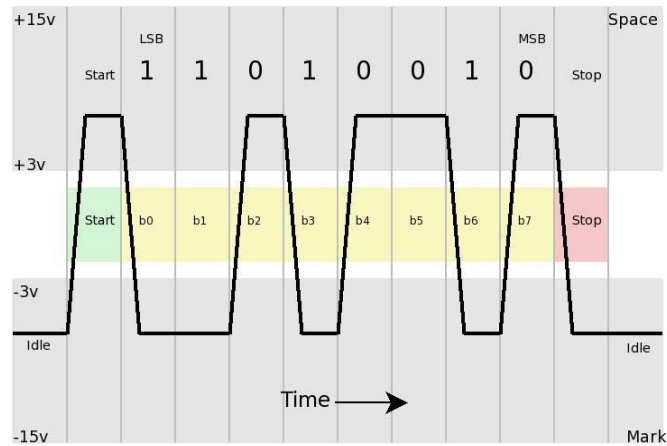


圖 10 電壓準位波形圖

表 3 一般使用的 RS-232C 中的信號和腳位分配：

Signal Type	Abbr.	Dir.	DB-25	DB-9	RJ-45
Common Ground	G	-	7	5	4
Transmitted Data	TD	Out	2	3	6
Received Data	RD	In	3	2	5
Data Terminal Ready	DTR	Out	20	4	3
Data Set Ready	DSR	In	6	6	1
Request To Send	RTS	Out	4	7	8
Clear To Send	CTS	In	5	8	7

Carrier Detect	DCD	In	8	1	2
Ring Indicator	RI	In	22	9	1

RS-422

RS-422 (EIA RS-422-A Standard) 是採用 4 線，全雙工，差分傳輸，多點通信的資料傳輸協議。圖 11 是典型的 RS-422 連接方式。實際上還有一條信號地線，共 5 條線。由於接收器採用高輸入阻抗和發送端驅動器比 RS232C 更強的驅動能力，故允許在相同傳輸線上連接多個接收節點，最多可接 10 個節點。即一個主設備(Master)，其餘為從設備(Slave)，從設備之間不能通訊，所以 RS-422 是點對多點的雙向通訊。接收端輸入阻抗為 4k，故發送端最大負載能力是 $10 \times 4k + 100\Omega$ (終接電阻)。RS-422 使用差分信號，差分傳輸使用兩根線發送和接收信號，對比 RS-232，它能更好的抗雜訊和有更遠的傳輸距離。最大電纜長度是 1200 公尺，最高傳輸速率為 10 Mbps (長度 12 公尺時) 或 100 kbps (長度 1200 公尺時)。在工業環境中更好的抗噪性和更遠的傳輸距離是一個很大的優點。目前 RS-422 的應用主要集中在工業控制環境，特別是長距離資料傳輸，如連結遠端周邊控制器或感測器。

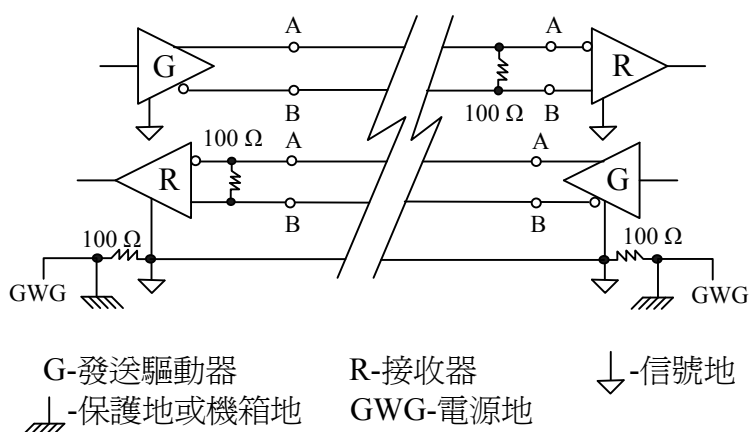


圖 11 RS-422 Master/Slave 端連接方式

RS-485

RS-485 的電氣特性規定為 2 線，半雙工，多點通信的資料傳輸協議。RS-485 (EIA-485 標準) 是 RS-422 的改進，所以 RS-485 許多電氣特性與 RS-422 相似。如都採用差動傳輸模式、都需要在傳輸線上接終接電阻等。RS-485 可以採用 2 線與 4 線模式，2 線模式可實現真正的多點雙向通信，如圖 12 所示。而採用 4 線連接時，與 RS-422 一樣只能實現點對多的通信，即只能有一個主設備，其餘為從設備，但它比 RS-422 有改進，無論 4 線還是 2 線連接模式匯流排上最多可接到 32 個設備。如圖 13 所示。RS-485 與 RS-422 的不同還在於其共模(Common-Mode)輸出電壓是不同的，RS-485 是 $-7V \sim +12V$ 之間，而 RS-422 在 $-7V \sim +7V$ 之間，RS-485 接收

端最小輸入阻抗為 $12k\Omega$ ，RS-422 為 $4k\Omega$ 。它提供高速的資料通信速率(10 公尺時 35 Mbps，1200 公尺時 100 kbps)。RS-485 和 RS-422 一樣，使用雙絞線進行差分傳輸，並且它可以進行長距離傳輸超過 1200 公尺。

RS-485出色抗噪和多設備能力，在工業應用中建立連向PC機的分散式設備網路、其他資料收集控制器、HMI或者其他操作時，串列連接會選擇RS-485。RS-485 同樣可以在一些工廠的項目控制機器上看到，來實現工廠不同樓層之間的數據通信。它可以抵抗機械設備和焊接設備的電磁干擾。

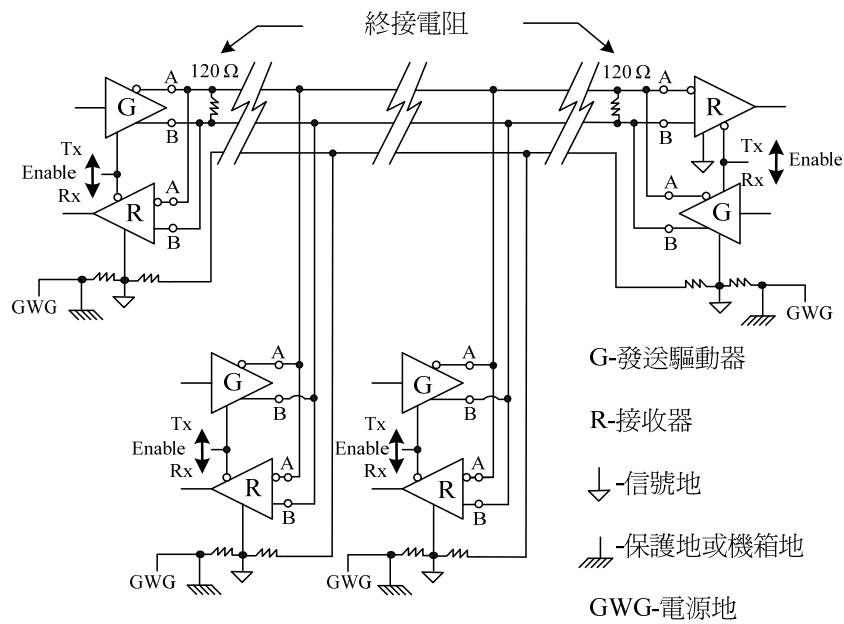


圖 12 RS-485 Master/Slave 端 2 線連接方式

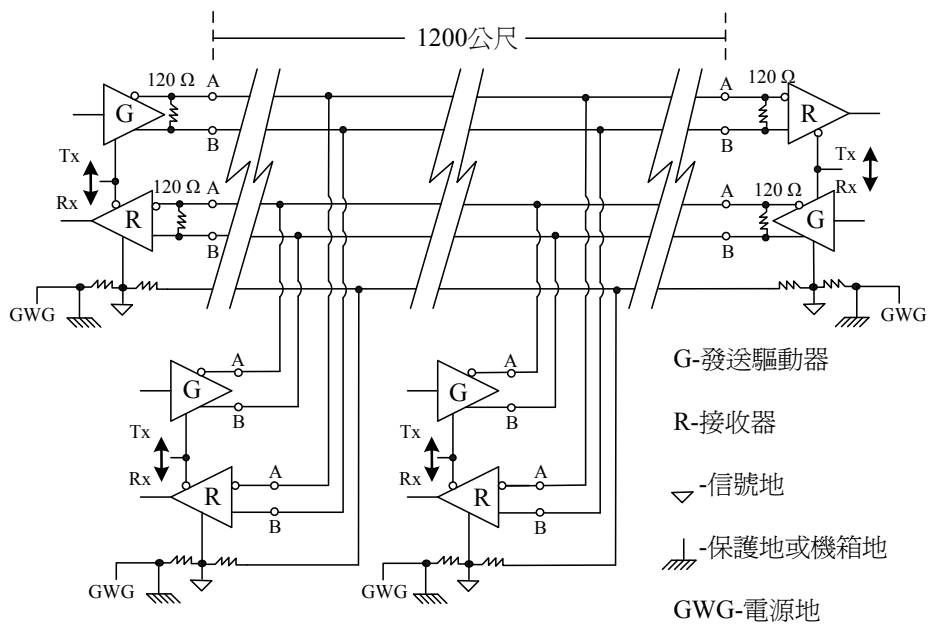


圖 13 RS-485 Master/Slave 端 4 線連接方式

I²C

內部積體電路匯流排(I²C)是由飛利浦公司在 1980 年代爲了讓主機板、嵌入式系統或手機用以連接低速週邊裝置而發展的一種專用介面。它是一種半雙工同步多主/從架構匯流排，I²C 只使用兩條雙向開放集極(Open Drain)：串列資料(SDA)及串列時脈(SCL)並利用電阻將電位上拉。I²C 允許相當大的工作電壓範圍，但典型的電壓準位爲+3.3V 或+5V。圖 14 爲 1 個主設備(微控制器)使用 I²C 匯流排連接 3 個從設備(ADC、DAC 及另一個微控制器)並且搭配升壓電阻(R_p)的架構示意圖。

I²C 的參考設計使用一個 7 位元長度的位址空間但保留了 16 個位址，所以在一組匯流排最多可和 112 個節點通訊。常見的 I²C 匯流排依傳輸速率的不同而有不同的模式：標準模式(100 kbps)、低速模式(10 kbps)，但時脈頻率可被允許下降至零，這代表可以暫停通訊。而新一代的 I²C 匯流排可以和更多的節點(支援 10 位元長度的位址空間)以更快的速率通訊：快速模式(400 kbps)、高速模式(3.4 Mbps)。雖然最大的節點數目是被位址空間所限制住，但實際上也會被匯流排上的總電容所限制住，一般而言爲 400 pF。

原始的 I²C 系統是在 1980 年代所建立的一種簡單的內部匯流排系統，當時主要的用途在於控制由飛利浦所生產的晶片。1992 年完成了最初的標準版本釋出，新增了傳輸速率爲 400 kbps 的快速模式及長度爲 10 位元的定址模式可容納最多 1008 個節點。1998 年釋出了 2.0 版，新增了傳輸速率爲 3.4 Mbps 的高速模式並爲了節省能源而減少了電壓及電流的需求。2.1 版則在 2001 年完成，這是一個對 2.0 版做一些小修正，同時也是目前的最新版本。

I²C 被應用在簡單的週邊並且其製造成本較傳輸速度更爲要求。一些常見的應用如下：

爲了保存使用者的設定而存取 NVRAM 晶片。

存取低速的數位類比轉換器(DAC)。

存取低速的類比數位轉換器(ADC)。

改變監視器的對比度、色調及色彩平衡設定(視訊資料通道)。

改變音量大小。

取得硬體監視及診斷資料，例如中央處理器的溫度及風扇轉速。

讀取即時時脈(Real-time clock)。

在系統設備中用來開啓或關閉電源供應。

I²C 的另一個強大用途在於微控制器的應用，利用兩根通用的輸入輸出接腳及軟體的規劃，可以讓微控制器控制一個小型網路。

週邊可以在系統仍然在運作的同時加入或移出匯流排，這代表對於有熱插拔需求

的裝置而言是個理想的匯排流。

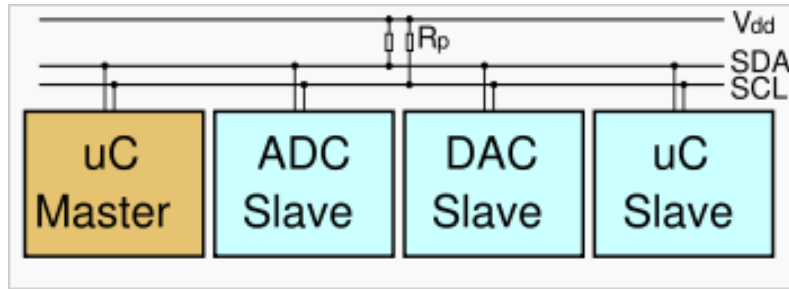


圖 14 I²C 匯流排主/從設備連接示意圖

SPI 匯流排

串列周邊介面(SPI)是由摩托羅拉開發的一種同步串列匯流排，用於該公司的多種微控制器中。如圖 15 所示，SPI 匯流排由四種信號組成，分別是主出從入(MOSI)、主入從出(MISO)、串列時脈(SCLK)及主動低電平從設備選擇(/SS)。SPI 是一種多主/從架構通訊協議，主設備與選定的從設備間使用單向 MOSI 和 MISO 線進行通訊，速率超過 1 Mbps，為全雙工模式。主設備產生一個 SCLK 脈衝，數據被同步到主設備和從設備間中。SPI 協議有四種不同的時脈類型，視 SCLK 信號的極性和相位而定。除了 1 Mbps 的速率外，SPI 還有另一優點：當僅使用一個從設備時，/SS 線路電壓會被拉低，/SS 信號可以不藉由主設備產生。如圖 16 所示，SPI 的一個缺點是每個從設備都需要獨立的/SS 線路。如果具有外部 I/O 接腳，或者外部電路板有足夠的位置配置多個選擇輸出器 IC，這並不成問題。但對於體積小、腳位數少的微控制器來說，具有多個從設備的 SPI 介面並非理想之選。

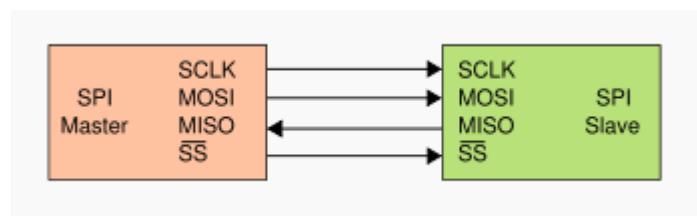


圖 15 SPI 匯流排連接示意圖

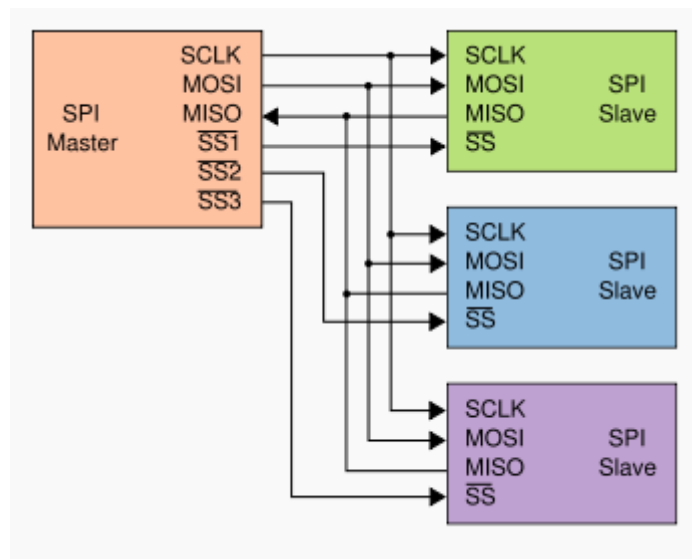


圖 16 SPI 匯流排主/從設備連接示意圖

Microwire 匯流排

Microwire 是由美國國家半導體公司開發的一種三線同步介面，用於該公司的 COP8 處理器系列產品。與 SPI 相似，Microwire 是一種主/從架構匯流排，包括主設備發出的串列資料(SO)、主設備接收的串列資料(SI)及信號時脈(SK)等三種信號，分別對應於 SPI 的 MOSI、MISO 及 SCK。此外還有一個片選信號(/CS)，其功能與 SPI 的/SS 相似。Microwire 是一種全雙工匯流排，速度可達到或超過 625 kbps(由其電容決定)。

由於 Microwire 也具有多個從設備，需要多條片選線，因此它像 SPI 一樣同時兼具優缺點。有時候，SPI 元件可工作於 Microwire 匯流排，同樣，Microwire 元件也可以工作於 SPI 匯流排，儘管只能在單元件情況下。儘管在電容配置得當且速率較低時 SPI 和 Microwire 通訊距離可長達 3 公尺，但它們通常都局限於板內通訊，距離不超過 15 公分。

1-Wire 匯流排

Dallas Semiconductor 的 1-Wire 是一種非同步主/從式匯流排，沒有用於多個主設備的協議。與 I2C 相似，1-Wire 採用半雙工通訊，在單一連線上採用一種開放集極(Open Drain)進行雙向資料傳輸。不過，1-Wire 匯流排的資料線也可以向從設備傳輸功率，儘管比較有限。1-Wire 的最高速率僅達 16 kbps，但在升壓電阻配置得當時，傳輸距離可達 300 公尺。

通用串列匯流排(USB)

USB 最初是由英特爾與微軟公司倡導發起，其最大的特點是支持熱插拔(Hot plug)和隨插即用(Plug&Play, PnP)，因此使用遠比 PCI 和 ISA 匯流排方便。USB 速

度比並列埠(Parallel Port，例如 EPP、LPT)與串列埠(Serial Port，例如 RS-232)等傳統電腦用標準匯流排快上許多。原標準中 USB 1.1 的最大傳輸頻寬為 12 Mbps，USB 2.0 的最大傳輸頻寬為 480 Mbps。近期推出的 USB 3.0 更從 480Mbps 提升到 4.8 Gbps 以上。表 4 為 USB 中的信號和腳位分配，USB 信號使用分別標記為 D+ 和 D- 的雙絞線傳輸，它們各自使用半雙工的差動信號並協同工作，可抵消長纜線的電磁干擾。圖 17 為 USB 連接器 Standard A/B 的接腳示意圖。

USB 的設計為非對稱式的，它由一個主機(host)控制器和若干通過集線器(hub)裝置以樹狀連接的裝置組成。一個控制器下最多可以有 5 級 hub，包括 hub 在內，最多可以連接 127 個裝置，而一台電腦可以同時有多個控制器。由於 USB 是一種隨插即用技術，因此在加入新的裝置時，USB 主機會自動偵測，查詢裝置的身份，並據此設定設備的驅動程式。USB 可以連接的周邊設備裝置有滑鼠、鍵盤、掃描器、數位相機、印表機、硬碟等。由於大大簡化了與電腦的連接方式，USB 也逐步取代並列埠成為印表機的主流連接方式。

現今標準中將 USB 統一為 USB 2.0，分為：

高速：傳輸速率 25 Mbps ~ 480 Mbps (最大 480 Mbps)

全速：傳輸速率 500 kbps ~ 12 Mbps (最大 12 Mbps)

低速：傳輸速率 10 kbps ~ 100 kbps (最大 1.5 Mbps)

表 4 USB 中的信號和腳位分配：

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

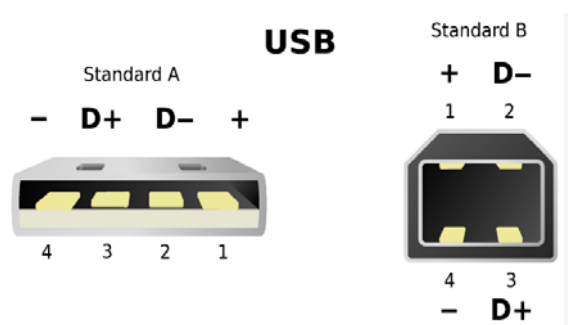


圖 17 USB 連接器 Standard A/B 的接腳示意圖

USB 為裝置和個人電腦之間提供一個價格低廉、簡單易用的連接能力。此外，

USB 提供更高速的效能、「熱插拔」功能、內建作業系統設定、一個連接埠可連接多部裝置，以及輕薄而具彈性的纜線，從而改善傳統的串列埠技術。雖然 USB 具備許多吸引人的優點，但是應用於儀器控制時卻有一些缺點。首先，USB 纜線不是工業等級的纜線，在吵雜的環境中可能會出現資料遺失的問題。再者，USB 纜線沒有栓鎖機制，纜線可以相當輕易地自個人電腦或儀器上拔除。USB 的纜線最大長度為三十公尺，包括使用線路內中繼器的情況。最後，沒有業界標準協定採用 USB 來進行儀器控制；若要控制 USB 儀器必須由儀器廠商個別進行運用。儘管 USB 有上述的缺點，但是由於目前電腦上的 USB 連接埠相當普及，加上 USB 2.0 的高速，使它成為未來儀器控制的好選擇。表 5 為常見串列介面的比較，本表顯示的是一般狀況下的最大值。

表 5 常見串列介面比較表

串列介面	格式	裝置數量 (最大值)	距離 (最大值，呎)	速率 (最大值，bps)
RS-232	非同步	2	50~100	20k
RS-422	非同步	10	4000	10M
RS-485	非同步	32	4000	10M
I ² C	同步	40	18	3.4M
SPI	同步	8	10	2.1M
Microwire	同步	8	10	2M
1-Wire	非同步	由電容與速率而定	1000	16k
USB	非同步	127	16(可透過 5 個集線器擴充到 96 呎)	1.5M、12M、480M

三、數位輸出入控制晶片 EDIO ASIC

EDIO ASIC 是工研院機械所開發的一顆專屬輸出入控制晶片，基本上它是我們 EPCIO ASIC 的精簡版本(可以有交互的替代性)，主要應用在產業機械的遠端輸出入控制上。EDIO 的全名為 Exquisite Digital Inputs/Outputs。它採用串列介面，將遠端欲受控制的輸出入點資料，透過同步全雙工方式的特殊協定格式與控制器溝通。如此一來多達 768 點的資料，在控制器端只需要少數的通信配線即可完成傳輸。除減輕系統硬體的負擔外，每 128 點(64 輸入，64 輸出)即有獨立的遠端資料擷取模組來負責資料的即時更新及傳送，一顆操作在主動(Master)模式控制器側的 EPCIO/EDIO chip，可以控制最多 6 個遠端模組上操作在被動(Slave)模式的

EPCIO/EDIO chips。

圖 18 為 CPU、EDIO-Master 與 EDIO-Slave 間信號傳遞方塊圖。圖中之被動模式遠端控制輸出埠(EDIO-Slave)是主管遠端數位之輸出入控制，所連接的 64 組輸入介面電路可接收來自外部裝置的並列輸入訊號。另外，所連接的 64 組輸出介面電路則可輸出並列訊號給外部裝置。EDIO-Slave 所接收的遠端輸入資料先以串列方式傳送到 EDIO-Master 中，EDIO-Master 再將該資料以並列方式透過 ISA Bus 傳送到主電腦 CPU 處理。經主電腦 CPU 處理後所下達的命令則依相反程序輸出遠端裝置。

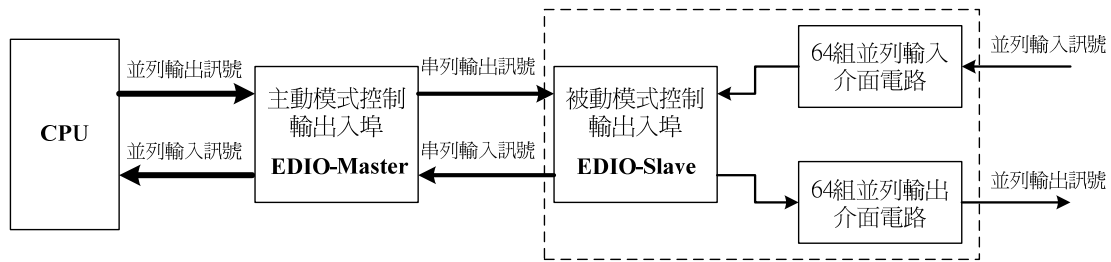


圖 18 CPU、EDIO-Master 與 EDIO-Slave 間信號傳遞方塊圖

EDIO 內建有兩組獨立的串列介面給兩個 Sets 的六個遠端輸出入模組使用，對每一個被動模式 EDIO 而言，它需要 Transmission Clock、Chip Select、Data In 和 Data Out 等四條信號線作溝通，整體的通信協定格式如圖 19 所示。

Remote IO Data Frame

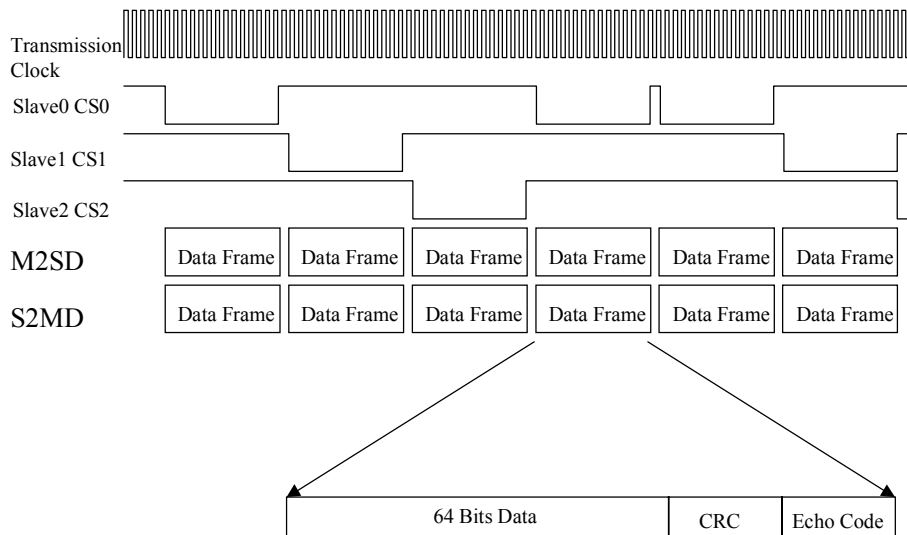


圖 19 串列介面傳輸協定

由圖上得知，Slave 0、1、2 是共用 Data In、Data Out 及 Clock 線的，而由 Chip Select 0、1、2 來做解碼。隨著傳輸 Clock 頻率動作，Data In 及 Data Out 同時被 Shift 進入及 Shift 移出兩邊的 EDIO Chip。接著 64 bits 的資料碼後，緊跟著的是 16

bits 的 CRC 碼以及 16 bits 的 Echo 碼。這些附加碼是爲了偵測傳送資料的錯誤而定的。錯誤偵測的最終目的乃是要校正傳輸的錯誤，目前 EDIO 的作法是要求信號重送，而重送的次數會累積起來，和使用者預設的最大容忍次數作比較，以便產生傳輸失敗的中斷信號。

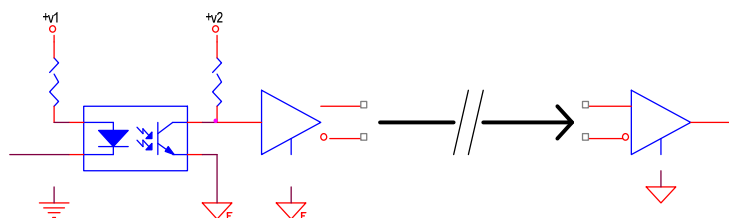
遠端模組透過 DB-9 pin 的接頭，和控制器上設於主動模式的 EDIO 溝通，使用者除了節省控制器的配線成本外，各個輸出入模組可以就近在各個致動器 (Actuator) 或感測器 (Sensor) 旁作控制，系統的穩定性得以提昇。而且可有效利用串列介面在長距離傳輸的經濟性，配合適當的電磁干擾防制措施；在設定傳輸時脈爲 1MHz 之下，我們可以得到如下的傳輸速度表現：(見表 6)

表 6 遠端輸出入模組傳輸速度

# of Slaves	Transmission Clock	Distance	Update Time
1 (64In,64Out)	1MHz	25 meters	100usec
3 (one Set)	1MHz	25 meters	300usec
6 (two Sets)	1MHz	25 meters	300usec

*EDIO can control each set independently.

圖 20 是 EDIO 串列介面所使用的標準線路。因爲採用光耦合器隔離，它所造成額外的時間延遲對於規劃傳輸 Clock 的快慢有很大關係。



differential line transceiver with isolation

圖 20 EDIO 串列介面標準電路

結語

目前工研院機械所正積極研發新一代運動控制 IC - Intelligent Motion Control IC，簡稱 IMC；IMC 爲一顆內建 CPU 的運動控制 IC，其中內部的遠端控制輸出入介面模組也在同步開發中。近年來，由於原物料不斷的上漲之下，節省硬體成本是一項必須要做的課題，因此具有精簡配線優勢的非同步串列傳輸方式是我們現在想要做的方向。當然我們還必須考慮到資料傳輸錯誤偵測的能力、傳輸速率、通訊距離等等，而且不只考慮到滿足產品的當前需要，還必須考慮到是否能用於產品的整個生命周期才是長遠的考量。

參考文獻

- [1] 于國忠(1993)(編譯), ”IBM PC 資料通信與網路應用”(第三版), 台北: 全華科技。
- [2] 徐瑞明、陳黎光(2005)(編譯), “USB2.0、Wireless USB、USB OTG 技術徹底研究”, 台北: 旗標出版。
- [3] John Patrick, ”嵌入式系統設計中串列匯流排的選擇策略”, 電子工程專輯網站, <http://www.eettaiwan.com/>。
- [4] “新型匯流排技術在儀器控制及連接領域的未來發展”, <http://www.ni.com/>。
- [5] “序列埠通訊基本概念”, <http://www.ni.com/>。
- [6] RS-232、RS-422 與 RS-485 標準及應用, <http://www.metersnet.com/>。
- [7] “Selecting and Using RS-232, RS-422, and RS-485 Serial Data Standards”, <http://www.maxim-ic.com/>。
- [8] <http://en.wikipedia.org/>。
- [9] <http://zh.wikipedia.org/>。
- [10] 馮元掄、陳文泉, ”分散式輸出入控制晶片-EDIO ASIC 介紹”, 機械工業雜誌 205 期, p.98。
- [11] 曾遠威、賴振國, “超級運動控制 ASIC-EPCIO”, 機械工業雜誌, 88 年 4 月號, P.97~110。

作者:

黃靜宜,

工業技術研究院 機械與系統研究所

智慧機械技術組 機電控制整合部