

有關串口卡與網絡串口服務器的特性差異

壹、前言

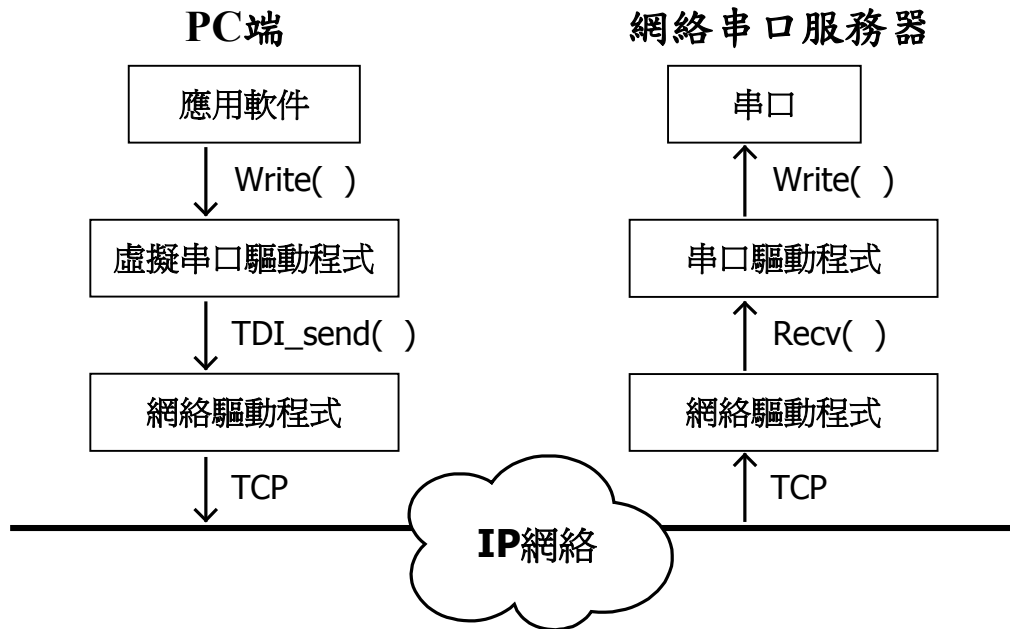
我們在 PC 上面可以有不同的方式來添加串口，每一種方式都有其不同特性。所以應用軟件的程式撰寫者，如果未考慮這些特性的差異，則相同的應用軟件在不同的串口環境可能產生不同的效果。特別是網絡發達的今天，我們大量使用網絡串口服務器，則這些特性的差異更是可能造成應用環境的失敗。本文將描述瑞旺公司有關 **APORT** 串口服務器的相關特性與考慮。

貳、多串口卡的特性

1. 我們最常看到的串口是 PC 上面的 **COM1** 及 **COM2** 串口，這是擁有固定 **I/O** 地址及 **IRQ** 位置的串口。基本上都是採用 **Windows** 作業系統本身內建的驅動程式，所以應用軟件撰寫者大都以 **COM1 / COM2** 的串口特性來考慮。而且也以可以正常工作在 **COM1 / COM2** 串口來驗證應用軟件的可使用性。我們發現 **COM1 / COM2** 的控制器是內建在主機板上，而且 **IRQ** 通常獨立而不與其他人共享，所以硬體動作相當直接，而且延時最小。
2. 我們另外常見的串口是以附加卡的方式來提供。一般而言，附加卡上面是擁有 2 個以上的串口，而且共用一個 **IRQ**。所以附加卡本身所提供的驅動程式會經由 **IRQ** 的產生，再判斷何人產生 **IRQ**（有的人可以有獨立的狀態暫存器指示何人產生 **IRQ**，有的人只好一個一個問何人產生 **IRQ**）來進行服務。聰明的您，大概可以發現附加卡的工作延時已經比 **COM1 / COM2** 要長些。這就要看有多少人來共用 **IRQ**，以及指示提出 **IRQ** 需求的人有否快速被服務的機制來決定延時長短。因此原本針對 **COM1 / COM2** 串口特性撰寫應用軟件的情況，可能在此就有些差異而會出現問題。例如不同的串口被服務的延時就不同，當第一個串口提出 **IRQ** 需求可能都會被先問到，而先服務。可是最後一個串口可能最後被問到，當被服務時已經延時很久。如果延時超過假設時間，則可能就出問題。
3. 現在我們再來看 **USB** 外接串口轉換盒的特性。由於 **USB** 匯流排是一種半雙工的工作模式，所以一切的動作都要以 **PC** 為核心，固定時間來輪詢 **USB** 設備有需要進行資料交換否。所以串口的送收動作是要經由 **USB** 封包來進行，於是我們只能等待每一次 **USB** 輪詢時刻才可以進行。所以資料由應用軟件想要傳送到真正由串口傳送的延時是相當大的。

參、網絡串口服務器的特性

1. 網絡串口服務器是利用本地的 CPU 來對串口進行資料送收動作。
2. 這些串口資料再依據某些規則來打包成 TCP 封包而在 IP 網絡傳輸到 PC。



3. PC 上面的虛擬串口驅動程式，則隨時待命解析這些 TCP 封包的資料與應用軟件進行資料交換。
4. 現在我們來觀察應用軟件要傳送一筆資料要多少延時。首先應用軟件的命令行提出串口資料寫出需求，虛擬串口驅動程式接受此需求而打成 TCP 封包要求網絡卡送出。當網絡串口服務器收到此 TCP 封包則解開由串口送出。
5. 在上述過程中，我們會有許多因素決定真正串口資料被送出的時間點。是由應用軟件交給虛擬串口驅動程式就是工作完成呢？還是虛擬串口驅動程式打成 TCP 封包要求網絡卡驅動程式送就是工作完成呢？網絡卡把 TCP 封包送出取得對方確認就好呢？還是串口服務器收到 TCP 封包就轉給串口控制器送出才算完成。
6. 由於 TCP 封包傳送是一種連接性網絡模式，一定會保證傳送到對方，所以何時會把 TCP 封包真正傳到對方是無預期的情況。可能在 10ms 內就送到對方，可能 200ms 送到對方，也可能 10 秒送到，也可能永遠送不到（因會中間有路不通，只好等路通）。

7. 現在您可以發現利用網絡串口服務器進行的串口資料傳輸動作是無法確認延時的時間。有時快有時慢。如果應用軟件仍然以 COM1 / COM2 的資料傳輸延時來考慮，則可能在網絡串口服務器上會出現問題。
8. 例如，我們由 PC 送出一個控制碼“A”給對方，要求回答設備現況。傳統上考慮到對方在收到命令後，可能要花費 10ms 來準備回送資料，所以應用軟件可能在寫出控制碼“A”的動作後，等待最長 2 倍於 10ms 的時間來取得資料。可是現在這個命令要經由網絡串口服務器送出，卻在 200ms 之後才真正送到對方，則開始出現混亂現象。應用軟件超過時間等不到資料則再發出命令來問，對方收到命令是在 200ms 後就回答，而應用軟件收到的回覆資料可能是 400ms 之前的命令。可見所有的現實情況與一開始基於 COM1 / COM2 動作的環境完全不同，所以對話情況是一團亂。

肆、常見問題與設定修改

1. 為何應用軟件已經收到回傳說資料已經寫出，可是對方串口尚未看到資料？
 - ⇒這是因為串口驅動程式回答應用軟件資料已經寫出，可是真正的資料尚要等待 TCP 封包送到網絡串口服務器才會由串口送出。此時間是不可預期，但是我們可以針對一些已經確認的延時量進行補償。
 - ⇒例如我們可以由 PC 先向網絡串口服務器進行 ping 測試，以了解由 PC 到網絡串口服務器的網絡途徑要花多少時間。則我們可以在 TPORT 驅動程式進行 TPORT_AP 設定時把“TCP delay time”的參數以“ms”為單位給予指定。這是我們已知由 PC 傳送 TCP 封包到網絡串口服務器所需要的時間。一般而言，如果未經過路由器來轉發封包則此值不會太大，則我們給“0”。但如果有經過路由器則可能有 200ms 等級（ping 的環回時間），則給予其值以補償之。亦即虛擬串口驅動程式會在收到應用軟件的寫出命令後，至少等此延時設定後才回答完成傳送動作。
 - ⇒對於傳送延時不在乎的應用（例如檔案傳送）則我們“TCP delay time”參數要設成“0”才可以有最高傳輸效率。
 - ⇒對於傳送延時很在乎的應用（例如對話工作模式，採用一問一答方式，則回答的等待時間就很重要。因為真正問話到達對方的時間就影響到我們可以等多久。是問話太晚到達或問話未送達就會有不同解讀及應對方式及後果），我們一定要調整“TCP delay time”的設定值以避免發生邏輯處理認知上的錯誤，而造成應用軟件的錯誤。

2. 為何應用軟件傳送資料的等效速度比起相同格式在 COM1 / COM2 上面慢很多？
 - ⇒這是因為虛擬串口驅動程式在收到應用軟件寫出命令時，會根據 TPORT_AP 有關該串口“TCP delay time”的設定來回答傳送完畢。
 - ⇒如果“TCP delay time”未設成“0”，則代表我們每次寫出命令的資料量，要分攤“TCP delay time”的時間才可以繼續送資料。如果分攤的時間大於真正資料傳送在串口的時間，則等效的速度就會變慢。
 - ⇒對於不必強調資料真正送出串口的時間點之應用軟件而言，我們一定要把“TCP delay time”參數設成“0”，則我們的虛擬串口驅動程式就會以最大速度來傳送。而真正的效率限制就變成 TCP 封包何時可以由 PC 送出及到達網絡串口服務器。
3. 如果我的應用軟件在傳輸資料過程中，網絡忽然斷線了，則可能應用軟件就咬死了。
 - ⇒這是因為 TCP 封包傳輸的特性是一定會把資料傳到對方。由於網絡斷線了，當然 TCP 封包傳不到對方而無法結束工作。此時如果您的應用軟件又是設成一定要等待資料寫出的命令要回答完成工作，才可以繼續，則當然應用軟件就會咬死在此。
 - ⇒所以您的應用軟件一定要有逾時設定機制在寫出命令上，如此才不會有咬死問題。
4. 我的應用軟件無法收到一筆完整資料而被分成數筆接收。
 - ⇒這是因為網絡串口服務器在收到串口資料時會希望以最快速度送回 PC。於是在等待接收資料的過程，在被通知有串口資料到達的第一時間就會被打成 TCP 封包來送回 PC。而每一個 TCP 封包被傳送的時間又不確定，所以就出現一筆串口資料被打成多個 TCP 封包到 PC。
 - ⇒對於某些應用軟件而言，他們希望一次就可以接收完整的資料，可是被分成數次資料來接收，將造成問題。此時我們必須把串口設成無流控狀態。當我們把串口設在有流控狀態（例如 RTS/CTS 硬流控或 XON/XOFF 軟流控）代表我們很在意串口資料的傳輸效率。於是網絡串口服務器會在任何時刻被通知有串口資料到達時就立刻把串口資料打成 TCP 封包來傳給 PC。如此可以避免資料堆積在網絡串口服務器，而且也可以保證 PC 在最快時間收到資料。

⇒當我們把串口設成無流控狀態，代表雙方有自己的方式來對話，則我們的網絡串口服務器會在串口接收資料量大於 200byte 或者超過 20ms 沒有資料再來的情況下把串口資料打成 TCP 封包送到 PC。如此則 PC 上面的應用軟件可以收到一筆完整的對話資料（一般 MODBUS 對話格式適合此設定）。

5. 我的 PC 在使用某串口後停止該串口的使用，但是其他 PC 要用該串口卻不能使用。

⇒這是因為我們的 TPORT 虛擬串口驅動程式是採用 TCP 連接方式，而且一連接就一直佔住。於是雖然 PC1 在使用串口後不再使用，可是網絡串口服務器卻一直維持與 PC1 的連接狀態。於是 PC2 想要使用串口時，網絡串口服務器卻不接受 PC2 的 TCP 連接。這是因為我們希望 PC1 在與網絡串口服務器連接後可以一直保持連接，則 PC1 隨時要使用串口都可以使用。

⇒對於希望在不同時刻不同 PC 都有機會使用網絡串口服務器的串口之應用環境，我們會建議安裝 RPORT 虛擬串口驅動程式。這個驅動程式會在 PC1 想用串口時才與網絡串口服務器進行 TCP 連接。如果目前無人使用，則會連接成功而完成串口使用需求。如果目前有人使用，則連接不會成功，當然串口也無法使用。當 PC1 不想再用串口時，我們會中斷 TCP 連接，於是在 10 秒鐘內網絡串口服務器就會知道此中斷連接狀態，而允許別人使用。此時 PC2 也可以使用串口了。

⇒由於 RPORT 驅動程式是在要使用串口時才開始進行 TCP 連接，所以無法保證自己想要使用就可以用到。不像 TPORT 驅動程式只要用過串口就一直保持連接隨時可以再用。因此使用者一定要依據自己的需求來安裝 TPORT 或 RPORT 驅動程式。