

跨時代的發明—Wireless USB Hub

在科技如此進步的時代，搭上無線網路普及智慧居家的盛起，人人一台平板電腦、智慧型手機，甚至是智慧手錶等，已是現代人常見的生活樣態。加上近年來物聯網概念興起，如何建立各智慧裝置間的通訊，藉以開發更多智慧的應用，成為炙手可熱的議題。智慧裝置數量的成長是如此之迅速，我們必須承認，以無線通訊取代傳統的有線通訊已是不可逆的趨勢。因此，無線通訊的建立與應用，如 Bluetooth、Wi-Fi 或其他新興的通訊技術，將是使智慧應用成為可能的關鍵推手。在這個時代的浪潮中，我們選擇的切入點是將舊有的有線傳輸介面無線化。USB 是當今最為普及的有線傳輸介面，若是我們將其無線化，將使更多裝置可以直接使用 USB 裝置。我們甚至可以使 USB 裝置不再只是提供給單一使用者使用，讓更多使用者可共享資源。當然，這樣的技術同時也保留了 USB 隨插即用的特色。

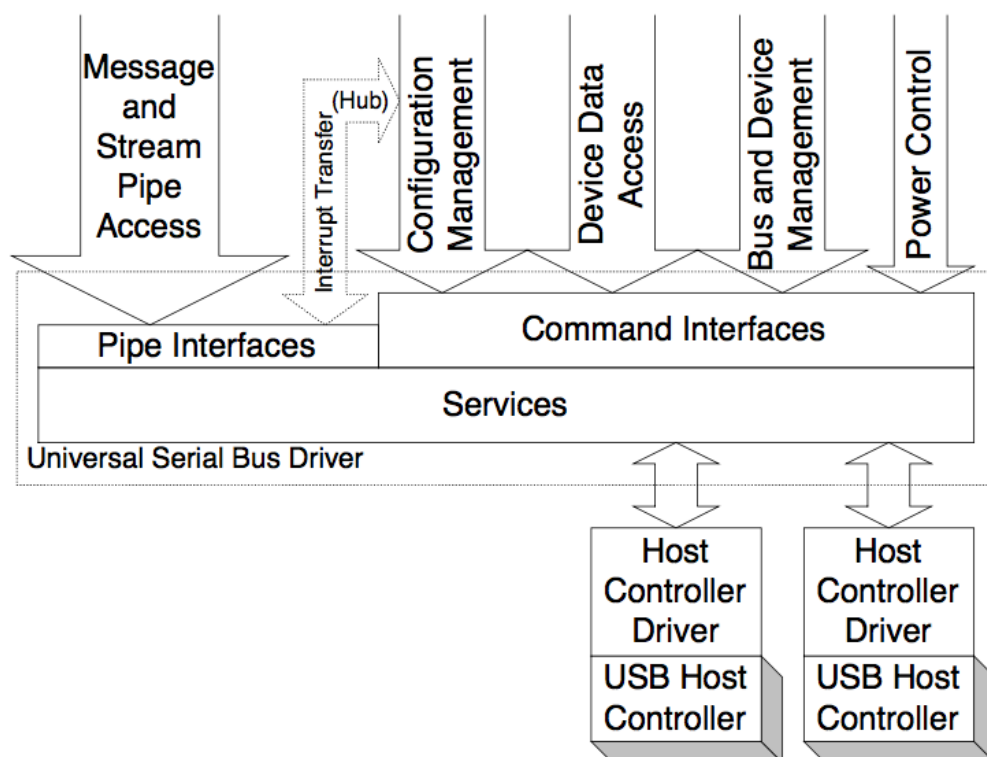
我們將整合且提供一個平台，讓 USB 介面周邊產品（如隨身碟、鍵盤、滑鼠等）達到共享且容易使用。不但可使容量小的手持裝置可達到容量擴充的功能外，也能解決 USB 周邊產品相容性問題。讓更多裝置只要透過無線網路，便可讓多人共享資源，無需先前註冊等麻煩手續。

我們將利用 Raspberry Pi 作為 USB Hub，也是將 USB 傳輸無線化的通訊中心。這個 USB Hub 使用 USB/IP 將原本 USB 裝置的周邊溝通與資料傳輸方式，改成網

路封包的格式以無線傳輸。如此便可經由網路協定，輕易將裝置公布共享。我們將提供使用者程式，讓使用者可輕鬆使用。我們的終極目標是希望能夠將更多普及的有線傳輸介面（如 HDMI、VGA）無線化，達到更多方便的應用。

實作方式—USB/IP

首先，我們先介紹 USB 傳輸協定。USB 驅動程式的軟體機制基本上分成兩部分：Command mechanism 及 Pipeline mechanism。前者允許客戶端控制 USB 裝置，並存取其預設通道（default pipe）；而後者則是用來管理檔案的處理與傳輸。



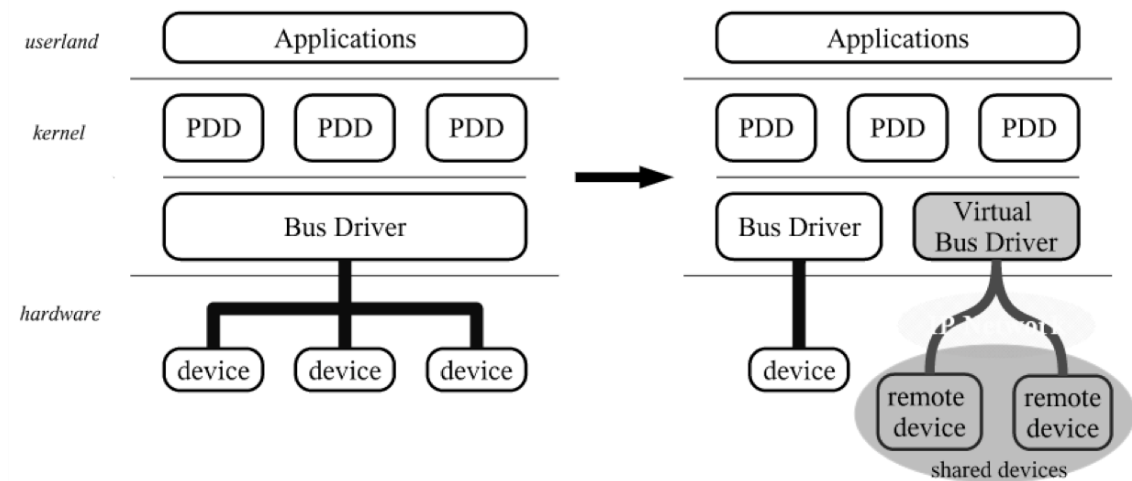
From Universal Serial Bus 2.0 Protocol

而 USB 驅動程式又可以分成 Per-Device Driver（PDD），Core Driver，及 Host Controller Driver（HCD）。

PDD 用於在應用程式或作業系統對 USB 裝置做出 I/O 的請求時，把該請求轉換成一連串的命令，即 URB。PDD 不與 USB 的硬體層溝通，而只是給出裝置位址 (device address)、端點位址 (endpoint address) 以及要使用何種傳輸方式 (Transfer Type) 的資訊。Core Driver 用於 USB 裝置的動態配置。舉例來說，USB 的 Enumeration 就是由 Core Driver 完成的。他提供了一個往上給 PDD 及往下到 Host Controller Driver 的介面。HCD 則是從 Core Driver 去接收 URB，必且把他切碎成更小的請求，稱作 Transfer Descriptor (TD)。這些 TD 會根據其傳輸方式而被排程，並且連結到適當的 frame list。

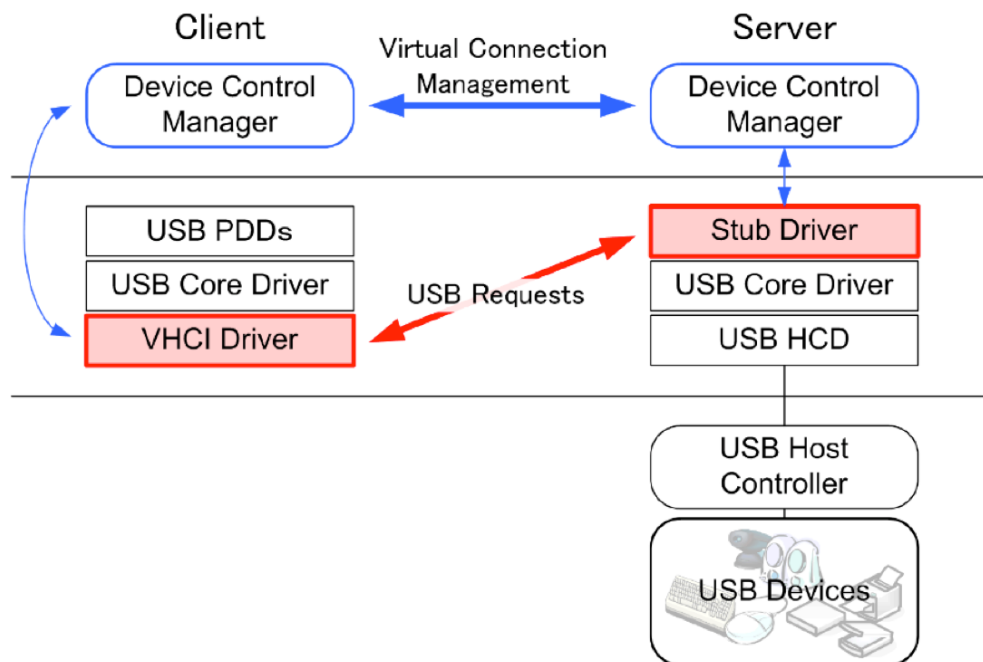
有了上面的基礎知識，我們現在可以開始介紹如何將 USB 傳輸無線化。我們所使用的方法是 USB/IP¹。USB/IP 的做法是 IP Encapsulation，先做出一個稱為 Virtual Host Controller Interface (VHCI) 的 Virtual Bus Driver，使得分享的裝置能夠享有足夠的網路透明度 (network transparency)。其想法是使得分享的裝置能夠直接由主機 (host) 端的驅動程式控制，主機端的作業系統及其上的應用程式並不會察覺這個連線與有線的 USB 傳輸有何不同，會直接把透過網路連接的 USB 裝置當作以傳輸線連結本地端。

¹ T. Hirofuchi, E. Kawai, K. Fujikawa, H. Sunahara, *USB/IP-aPeripheral Bus Extension for Device Sharing over IP Network*



From USB/IP

VHCI 的功能與 HCD 雷同，他可以處理 URB 並把它轉成 USB/IP request block (RB)，並且送到遠端的 USB 裝置。然後做一個 stub driver 在遠端的裝置上，功能類似 PDD，用於解譯遠端丟過來的 USB/IP RBs，並把它交給 USB 裝置。整個流程圖如下：



From USB/IP

如此便可以讓客戶端以為被插了一個 USB 裝置，而遠端的 USB 裝置也以為它插在一個功能完整的作業系統或應用程式上。

討論與未來工作

從 USB/IP 實作的過程中我們掌握了 USB 資料如何轉換為網路封包並透過 TCP/IP 傳送，並透過具體的應用來實現了遠端使用 USB 裝置的可能性。雖然已經成功將 USB 傳輸無線化，但其中仍舊存在許多問題。以下分別從幾個不同的面向作出討論。

使用者操作性

現行在主機端與客戶端兩端皆須事先安裝 USB/IP 驅動程式並將其啟動才能使用的方式顯然不夠直覺，應該加以改進。例如事先寫好建立連線、自動辨識主機端是否插入裝置的自動執行腳本，當有客戶端出現時能自動詢問、建立連線，讓使用者能直接透過本地裝置的介面操作主機端的設備，不須多輸入指令。

技術面

連線速度

USB/IP 透過無線網路建立連線的方式是否夠穩定？連線速度的延遲是否會使特定種類 USB 裝置在使用上產生延遲甚至無法使用？答案顯然是肯定的，例如使

用網路攝影機時需要較大的流量與頻寬，系統如何判別連線狀況並提醒使用者？

資料安全

透過網路傳輸的 USB 資料如何加密或確保資料安全性？又或是同時間只容許一個主機端與一個客戶端一對一使用？

商品化可能問題

價格

現行運作方式主機端需要網路連線設備與 linux 嵌入式系統，也需要足夠且穩定的電源供應來應付多樣化、同時接入的 USB 裝置，考慮到實際商品功能，價格可能偏高。

跨平台

目前的傳輸僅限於同樣使用 Linux kernel 的系統之間。若要商品化，能夠跨平台與裝置，讓手機或平板皆能連上 USB hub 才可能具備吸引市場的亮點。

延伸應用與想像

假使 USB 裝置的訊號能成功轉換為 TCP/IP 封包，其他連接埠的訊號是否可能類推適用？例如音訊的輸出輸入、VGA、HDMI 等等，也有可能轉換為網路封包，讓沒有這些連接埠的裝置也能使用？