



Common USB Development Mistakes – You Don't Have To Make Them All Yourself!

USB 設計常見的錯誤

作者 (Steve Kolokowsky 與 Trevor Davis)

Executive Summary

11 years have passed since USB was first introduced. Believe it or not, people are finding new and innovative ways to use this protocol every single day. They are also making the same mistakes every single day on thousands of designs worldwide! When will learn from one another? How about right now? This article will discuss the top 10 biggest USB mistakes when designing with USB. It will provide specific technical details for planning for, and preventing, common USB errors.

USB 已經問世十一年了，然而，您相信嗎？人們每天都還是不斷發現創新的方法來利用這套通訊協定；相對地，他們也是每天在成千種的設計方案中，重複著相同的錯誤！何不現在就從錯誤中學習正確的觀念呢？有趣的是，USB 和研發人員所熟悉的其他協定有著眾多相似之處。然而，這比工程師在面對所熟悉的 PS/2 與 RS-232 領域時，一再犯下相同錯誤的程度還要不可思議。此外，面對嚴格的標準規範，工程師所犯的錯誤可能會造成不符 USB 規範的代價。這些錯誤主要可分成五種類型：速度、電源、訊號品質、軟體、以及規格符合。(如圖 1 所示)

圖 1：USB 設計最常見的「致命錯誤」

USB設計最常見的「致命錯誤」

- 1 速度**
 - 假設你的應用能實際可達480Mbit/sec
 - 無法預測系統瓶頸點
- 2 電源**
 - 瞭解匯流排的電源值(500mA, 100mA, 500uA等限制)
 - D+的逆流電源 (back-power)
 - 集線器—無法為下行資料流連接埠供應全部電壓
 - 匯流排供電的集線器—在高/低電壓間切換
- 3 訊號品質**
 - 多組裝置共用D+/D-訊號
 - 資料路徑中的切換
- 4 軟體**
 - 不使用現有的類別驅動程式(class drivers)
 - 未能從USB規格制定論壇(I/F)取得製造商ID (Vendor ID, VID)
- 5 規格符合**
 - 在規格測試之前，沒有先進行原型測試
 - FCC/VCCI (EMI)—未能分離屏蔽 / 接地

速度

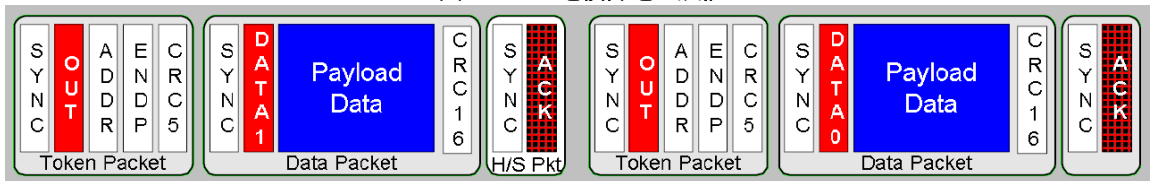
毫無疑問的，USB裝置的速度不僅是消費者與研發業者之間最常發生的爭議；且是在遍佈全球的通路賣場中最常被問及的問題。目前有線USB裝置共具備三種資料傳輸速度：低速USB(Low Speed USB)的傳輸資料速度為1.5 Mbits/sec；全速USB(Full Speed USB)的傳輸資料的速度則為12 Mbit/sec；而高速 USB(High Speed USB)的傳輸資料的速度為480 Mbit/sec。但有一點千萬不要搞混了—那就是**USB 2.0並不等於高速(High Speed)USB**。高速USB最初是以USB 2.0的規格版本發表的，而該規格也同時適用於低速與全速USB的資料傳輸速度。

就和任何電子系統一樣，設計人員都會希望獲得最佳效能。然而對 USB 而言，許多設計人員在設計之初就相信他們會從系統中得到所有 1.5 Mbit/sec、12 Mbit/sec、或 480 Mbit/sec 的效能。老實說，這是一個很糟糕的假設。因為有很多原因將會造成您的裝置無法使用全部的頻寬。首先，多個使用者可共享 USB 匯流排。即使您使用主機板上不同的連接埠，您也有可能同時與匯流排上的其他裝置共用一個主機控制器 (host controller)，也就是說，您的裝置與其他所有裝置共用 USB 匯流排頻寬。

其次，USB 封包格式化協定會把較長的資料分成數個 512-byte 的封包。每個封包的起始頭 (header) 負責辨識封包內容，而封包尾端則有 CRC，負責確定資料的完整性。每個封包傳送後都需要由接收端回覆 ACK(如圖 2 所示)。訊框起始 (Start of Frame, SOF) 封包則是每 125 uSec

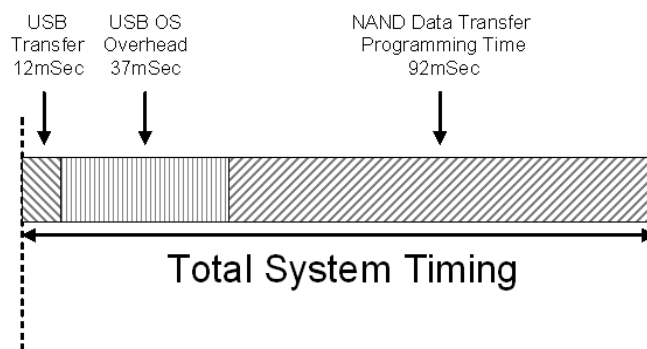
(microframe) 傳送一次，以維持匯流排上的時序。在此因素影響下，理論上 USB 的最大頻寬是每 microframe 可允許 13 個連續封包傳送，相當於每秒 53,248,000 bytes 的傳輸速率。然而由於一般主機控制器在每個 microframe 僅能接收 10 個連續封包，或傳送 8 個連續封包，因此現有主機控制器無法提供理論上的速度極限值。

圖 2：USB 連續傳送之訊框



無法預測系統瓶頸點：許多高速系統都在這項議題上遭遇挫折。Cypress 最近分析了將資料寫入 NAND Flash 的系統。這個系統透過 USB 傳送資料，之後將資料先暫放於緩衝區 (buffer)，接著再寫入快閃記憶體中。每個封包都分成三種時間元素：1) USB 傳輸所需時間，2) 主要作業系統 (Operating System) 頭段時間，以及 3) NAND Flash 韌體的編程時間。圖 3 為 128Kbyte 資料區塊經過即時效能分析後的時序圖：

圖 3：系統瓶頸範例



工程師幾乎花費所有時間，利用加快 USB 介面晶片的波形，以嘗試降低 USB 傳輸所需的時間，直到將時間切割成不同區段。一旦他們了解效能其實是決定於所使用的 NAND Flash 可編程韌體，才能夠藉由減少 NAND 的 overhead，來大幅提昇效能。在大多數的系統中，高速 USB 並不會是瓶頸所在。因此，設計人員必須仔細地檢查整個系統，以確保他們有足夠的頻寬空間，來達成他們想要的系統速度。

電源

USB 匯流排電源：根據 USB 的規格，USB 裝置可以是由「匯流排供電」(Bus-powered)、由 USB 電源線提供電源，也就是「自行供電」(Self-powered)、或是由電池或外接電源供電。最理想的 USB 供電方式之一是由匯流排供電，如此一來就不需要再外接電源！然而利用 USB 匯流排供電

也意味著用戶將受到 USB 規格中 500uA、100mA、與 500mA 的電流限制。但是許多設計人員卻沒有仔細地留意這些限制，因此造成他們的設計無法符合匯流排供電的規格。

- 500uA – 由主機端供應電源，但是當 USB 沒有執行傳輸動作時，您的裝置必須是處於 USB 暫停 (suspend) 的狀態。在此狀態下，您只能從 VBUS 獲得 500uA 的電流。這個狀態是爲了當 PC 處於暫停模式時，能將最小的電流輸出降至最低。
- 100mA – USB 具備高電源 (500mA) 與低電源 (100mA) 兩個連接埠。低電源連接埠常用於匯流排供電型的集線器中，而且它是承接 500mA 的電源，再分配給下端連接埠各 100mA 的電流。當您剛接上 USB 裝置時，此裝置在收到來自主機端的 SET_CONFIGURATION 訊息之前，是無法判讀它是在何種連接埠上，因此會將電流限制在 100mA。這表示您的裝置必須在非常低的電源模式下，在 USB 上進行裝置列舉的作業，直到收到 SET_CONFIGURATION 訊息指示，才能夠切換至高電源模式。這在高速 USB 上原本是非常困難的事，這情況直到 2004 年 Cypress 發表 FX2LP 晶片後才得以解決。
- 500mA – 這是 USB 規格中所允許的最大電源輸出值。

實際設計測試應由系統工程師主導，以確保裝置在由匯流排供電的運作模式下所需要的不同電源值，否則，您就得爲新的 USB 系統額外購買昂貴的外接電源。

逆流電源 (back-power)：「自行供電」的 USB 裝置也有本身電源的問題。由於這類裝置有獨立的電源供應器，因此它們可能在主機端關閉時，仍處於開機的狀態。這種情況可能會讓 D+ 訊號有小幅電壓上升，而造成 USB 裝置偵測的部分會緩慢地對整個主機系統充電，進而干擾系統的啓動。自行供電的 USB 裝置（包括由電池供電的裝置）必須要直接由 VBUS 消除該電壓上升的問題，或是透過軟體控制，利用 VBUS 感應器將裝置關閉。

訊號品質

共用 D+/D-訊號：爲了降低時間、流程與成本，有些產品會試著在多個裝置之間共用 USB 的訊號線路。例如：USB 型的基座 (docking station) 可能允許一個軟碟機或一個 DVD 播放機插入儲存插槽中。這兩種裝置間共用 USB 線路，就可藉由降低集線器連接埠的數目，進而節省成本。然而，如果沒有完全瞭解系統中的裝置特性，要利用這種方法就會相當的困難。在三態 (tri-state) 編排中 (如圖 4 的設計選擇二)，匯流排上的另一裝置會增加 USB 線路上的電容值，而連結至該裝置也會造成訊號反射，干擾高速 USB 的運作。另外，在開關型編排中 (如圖 4 的設計選擇一)，開關本身會對 USB 線路造成額外的電容值與電阻值，這將會減緩 USB 線路上訊號的上升/下降時間，並且造成 USB 訊號眼 (eye) 縮小。

圖 4：基座中 D+/D-訊號共用的設計選擇

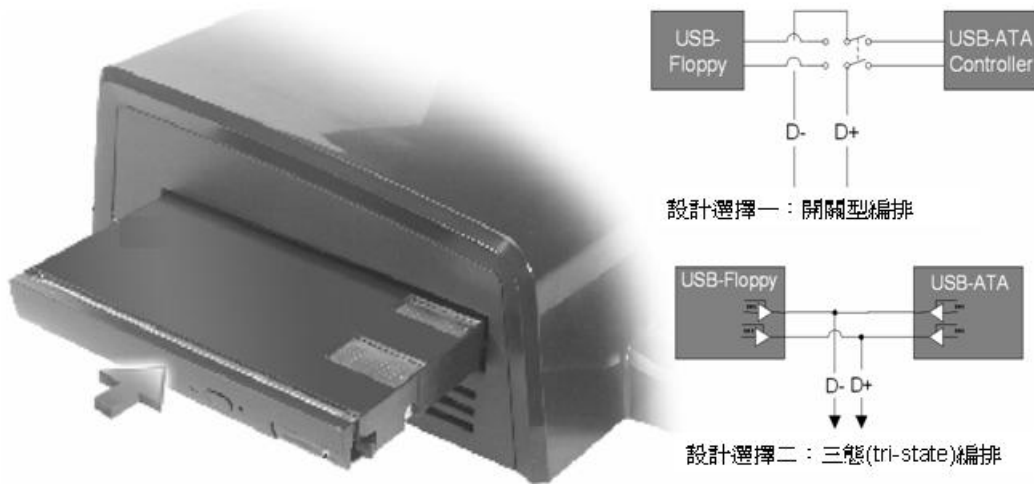
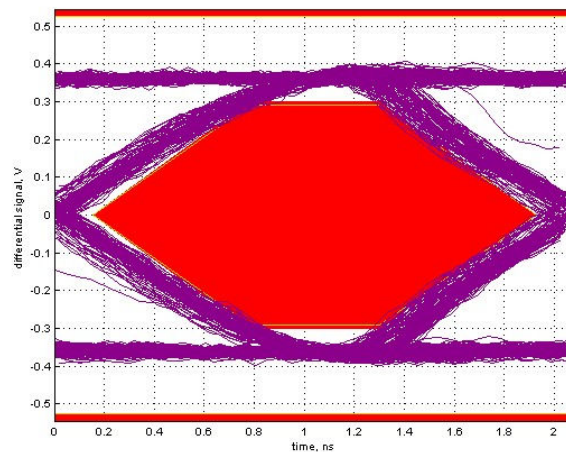


圖 5 顯示一般 USB 裝置訊號以 10 pF/10 Ohm 負載，傳輸經過開關後的訊號眼圖。在單一訊號眼圖中，紅色區域內應該不會有干擾存在。而 USB 訊號共用線路成功的關鍵就是必須維持開關的低負載，並且採用具備高訊號轉換速率的晶片。

圖 5：共用訊號線路所造成的額外訊號干擾



軟體

不使用現有的類別驅動程式：類別驅動程式（class drivers）是 USB 系統環境中非常重要的一部分。這些驅動程式是由主要作業系統所提供，因此無須另外進行開發。USB 的類別是由 Device Working Groups 所定義，這是一個在 USB 的號召之下，所組成的自發性團體，其目的在創造裝置之間標準的溝通語言。目前市面上的 USB 裝置類別包括各類 HID 裝置（如：滑鼠、鍵盤、或其他控制器等）、大量儲存裝置（如：各式磁碟機）、通訊裝置（如：數據機、網路介面卡）、音頻信號、影像、以及靜態影像（如：相片與掃描器）。

若您的裝置完全符合現有的類別架構，那方法很簡單，只要到 www.usb.org 下載類別定義，並以此進行設計即可。然而，若您的裝置並非完全符合現有的類別，該裝置還是可用的，例如：Microsoft 就採用靜態影像類別(Still Image Class)，設計新的媒體裝置通訊協定 (Media Transfer Protocol, MTP) 類別。另外，人機介面裝置並不一定要連接到與使用者互動的介面上，它也可以在不用另外開發新驅動程式的情況下，與程式中的溫度計、壓力感測器、幫浦控制器等裝置連結。

雖然藉由一個類別驅動程式就能夠完成設計，許多公司基於某種原因還是會想嘗試開發專用驅動程式，或以高價委外，由外部設計公司開發驅動程式。類別驅動程式不僅能夠排除設計風險、成本與時程等問題，也可避免許多除錯與複雜度的問題。

未能從 USB 規格制訂者論壇(Implementer's Forum, I/F)取得製造商 ID (Vendor ID, VID)：每個 USB 裝置都有一組獨一無二的辨識碼，讓作業系統可藉以找出正確的裝置驅動程式。辨識碼的第一部分是由 USB 規格制訂者論壇 (www.usb.org) 所指定的 16 位元 VID 數值；而第二部分則是由產品製造商所指定的 16 位元數值，稱為產品 ID(Product ID, PID)。(如表 1 所示)

表 1: VID 與 PID 數值

USB 製造商 ID – 由 USB 規格制訂者論壇所指定	產品 ID – 由製造商所指定
16 位元	16 位元

若廠商與設計人員事先規劃，並且能事先與 USB 規格制訂者論壇溝通，要取得 VID 並且指定 PID 其實是相當簡單的程序。然而，每年仍有數以千計的裝置已完成韌體軟體的設計工作，但因無法取得適當的辨識碼，而造成設計延遲的情況發生。

規格符合

在規格測試之前，沒有先進行原型測試：產品必須通過USB規格測試，才能合法使用各類USB標章(如圖6所示)。USB規格制訂者論壇負責規格測試，以確保所有用戶都能對USB產生良好的使用者經驗。由於所有USB廠商都相互依賴，以建立良好的商譽，因此這點相當重要。若用戶對於某個USB裝置產生不好的經驗，這將會大幅降低其再多花心力去使用其他裝置的意願。

圖6：USB認證標章

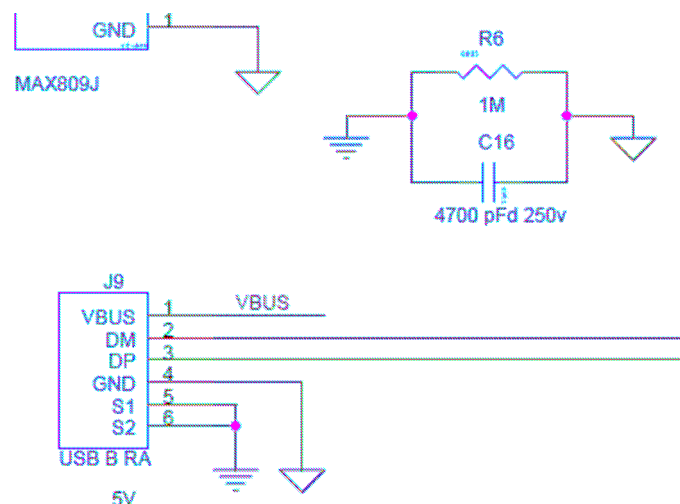


研發廠商可能無法負擔規格測試流程中一些需要昂貴高階測試設備的步驟。然而，許多裝置無法通過規格測試的原因，則是因為一些廠商沒有檢查到一些簡單的項目。因此，在裝置送測之前，您必須至少先進行下列的測試：

- USB Command Verifier Tool第9章的測試。這個程式可驗證您的裝置是否可以處理可能由主機端送出的最重要的SETUP指令（請參考<http://www.usb.org/developers/tools/>網站）
- 電源測試：暫停（suspend）電流、湧入（inrush）電流、尚未配置（unconfigured）電流

電磁干擾（EMI）：要介紹良好的電磁干擾設計技術可以另外寫成一篇文章了！最常見、也是最容易解決的 EMI 錯誤，就是嘗試利用系統的接地面作為 USB 纜線的屏蔽。這會讓雜訊引入接地面上，進而越過裝置屏蔽(請參考圖 7)。此外，這樣的雜訊也會透過相當於 2 公尺長的天線四處散播！

圖 7：CypressFX2LP 開發公板內含分離式屏蔽接地的圖表



現在您應該知道問題之所在了。數以百計的設計者每年絞盡腦汁，就是為了解決上述的問題。從別人的錯誤中學習，您就可以省下許多耗費精力、重複設計、以及時程延遲的壓力。但最重要的是，請繼續保留您的創造力，因為這個世界需要更多創新、精緻、有趣的 USB 相關裝置。



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
<http://www.cypress.com>

© Cypress Semiconductor Corporation, 2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™, Programmable System-on-Chip™, and PSoC Express™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.