



Windows CE：將高速USB加入您的智慧型手機中

Steve Kolokowsky 與

Windows CE 能讓您把 Windows 功能輕易地放入口袋中帶著走。若要把高速 (high-speed) USB 的功能放入此平台，只要按照微軟的CE工具軟體範例做，就可以輕鬆完成。但要實際達到您原本預期的高速 USB 2.0 的頻寬，那就另當別論了。由於裝置所預期達到的效能與實際最終效能之間的差異，到底造成多少設計被捨棄或接受？別被行動系統的頻寬、下載速率、還有USB假設的設計單純性等說詞給矇騙了。在您必須為指標測試的執行速度不如預期而做出解釋之前，以下我們要討論的重要細節，都是讓您的Windows Mobile手機能否達到最佳效能的關鍵。

毫無疑問的，消費者與研發人員之間對於USB議題最大的誤解之一，同時也是全球零售商店中最常聽到的問題，就是這個USB元件的速度能多快。首先，最重要的是先瞭解USB規格中基本的速度型態：目前有線USB有三種資料傳輸速率。低速USB資料傳輸率為1.5Mbit/sec，全速USB為12Mbit/sec，而高速USB則為480Mbit/sec。大家要注意的是USB 2.0並不等於高速USB。高速USB最先是USB 2.0規格版本推出，然而該版規格同時也包含了低速及全速的資料傳輸規格。

就像任何電子系統一樣，設計人員都想獲得最佳效能。但是對於USB而言，許多設計人員在開始系統設計之初，就認為他們將達到1.5、12、或480 Mbit/sec的效能，這是很不對的假設，因為有許多原因讓您的裝置可能永遠無法使用全部頻寬。儘管現實如此，但USB仍然是市面上手機資料傳輸的最佳選擇。

事實上，選擇USB作為手機的傳輸介面有許多原因：當連接到PC埠時，USB所提供的電源能在兩小時內對一個1000 mAh容量的普通電池完成充電。USB是目前真正通用的規格，自1997年以後出廠的每部PC與麥金塔電腦都有USB功能。每個主要的作業系統也都能支援USB，即使是嵌入式的行動系統。然而，當準備要投入新的USB手機設計時，最大的考慮重點還是放在設計的簡易性（設計成本）及最後的實際效能（產品成功的因素）。

在手機上開發USB功能時，作業系統的選擇也是十分重要，因為USB效能的最佳化與您選擇的作業系統息息相關，目前市面上有許多可選用的作業系統，以智慧型手機（Smartphone）來說，此種講求高速USB速率與效能的高階手機市場目前主要採用的是Symbian作業系統。ABI Research總監Stuart Carlaw表示：「2006年Symbian在智慧型手機作業系統市場中估計有73%的佔有率，但未來由於會面臨到Linux與Windows Mobile的強勢競爭，因此我們預測，其佔有率到了2012年將會降至46%。」這表示開發人員也必須瞭解在Windows Mobile作業系統中USB的設計規格，以為將來市場變化作準備。

在桌上型PC的世界中，Windows算是最知名的具備專利的作業系統軟體。在行動運算的世界中，Windows針對此市場區塊也提出了三種解決方案：Windows CE5、Windows Embedded 6、以及Windows Mobile。簡單來說，Windows CE是屬於即時作業系統（Real Time Operating System；RTOS），可讓您運用在醫療與科學設備、工業控制器、

網際網路裝置、及其他小型電子裝置中。Windows Embedded 6則提供了更強大的功能，能運用於次世代的裝置中，並可提供有線及無線網路連線、高階的多媒體裝置、存取大量記憶資源、以及更強大的多工作業能力。最後，Windows Mobile (WM) 則是PDAs和手機的標準平台。要在上述三種系統中設計USB的功能，就必須要確實地考慮並瞭解USB的各種規格。

首先，USB就不是一種對等連線，其中有主控 (Host) 裝置及從屬 (Slave) 裝置之分。一般而言，主控裝置比從屬裝置具備更強大的功能，主控裝置控制USB匯流排上所有的動作，因此主控裝置多半建置於桌上型或筆記型電腦中。而其他可插入連結這些電腦的裝置則多屬於從屬裝置，例如滑鼠和外接式磁碟機。WM協定堆疊中具有主控端及從屬端驅動程式。由於這兩者都不能稱為USB驅動程式，因此在此特別把USB從屬端驅動程式稱為USB功能驅動程式 (USB function drivers)，簡稱USBFN。且多數的WM裝置都需要連結到電腦上，因此我們也將焦點放在USBFN上。同樣的觀念也可運用在USB主控端上。

讓我們先來看基礎的部分。Windows Mobile USB介面採取分層的模式，這樣就可以最少的工作量產生新的USB驅動程式。在USB驅動程式中，上層稱為Model Device Driver，簡稱MDD，它為USB裝置驅動程式 (如RNDIS與大量儲存裝置) 提供了一個統一介面。MDD層也為更上層的驅動程式提供裝置列舉 (enumeration)、緩衝、管線/端點轉譯等功能。此外還有Platform Dependent Driver 層，簡稱PDD，負責編程 USB 控制器中的暫存器，並且執行 MDD 標準功能呼叫與特定裝置需求之間的内容轉換工作。(如圖1)

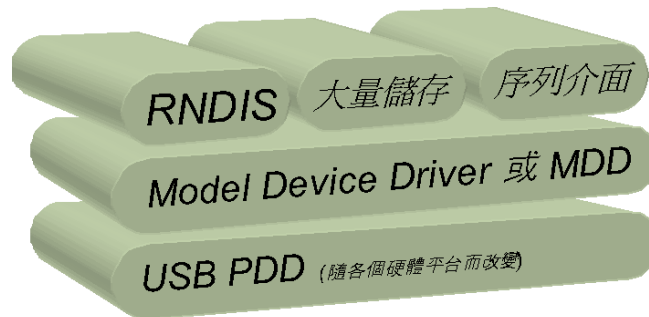


圖 1. USB Windows Mobile中的USB驅動程式分層模型

USB的PDD功能可分成四個群組，分別代表驅動程式執行週期的四個階段，如圖2所示。在DLL初始化 (initialization) 階段中，內部的機制如記憶體配置和工作建立皆在此進行。在USB裝置列舉階段中，PDD會告訴MDD關於USB硬體的功能。接著MDD會透過PDD將USB的描述元 (descriptors) 傳遞給主控端進行裝置確認。而USB資料傳輸階段則是裝置完成實際傳輸作業的階段。至於在DLL刪除 (destruction) 階段中，原先DLL所使用的作業系統資源則歸還給作業系統以供其他工作使用。

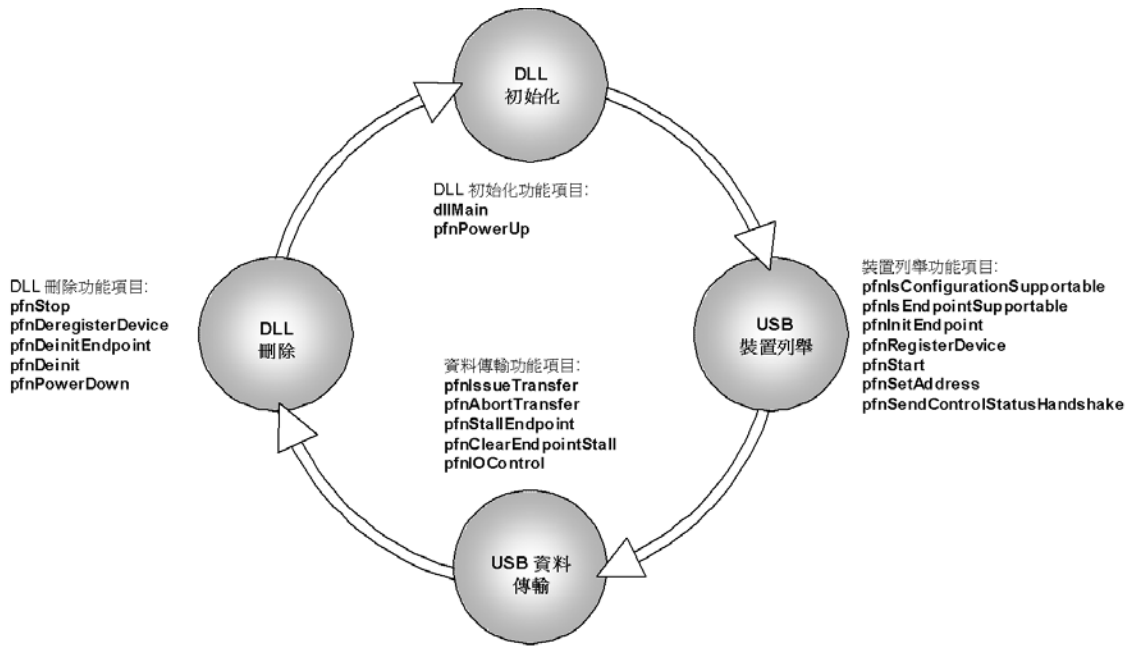


圖 2. USB Platform Dependent Driver (PDD)之工作週期與功能項目

上述PDD工作週期中所列舉的20個功能項目裡，只有一項必須修改以提昇效能，就是IssueTransfer功能項目，此功能項目是由MDD所呼叫，以啟動IN（往主控端）及OUT（從主控端）的傳輸工作。INs與OUTs呼叫IssueTransfer的方式不同，IN在執行工作時會將放滿資料的緩衝區交給PDD，一旦USB裝置上空間許可時，PDD就會把資料移到該裝置中。OUT的工作執行則是當MDD將空的緩衝區提供給PDD時才會啟動。

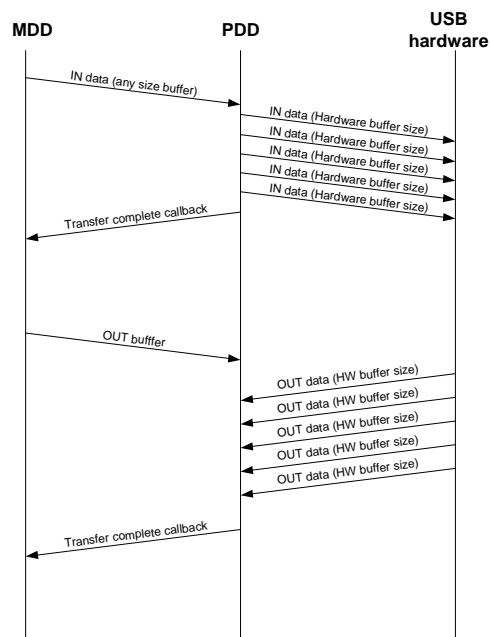


圖 3. 一般USB的IN與OUT工作順序之梯形圖

上方階梯圖3中表示了一般IN與OUT的工作順序。他們皆開始於MDD呼叫PDD_IssueTransfer功能項目，接著PDD會將一個一個符合USB規格的資料封包傳送至MDD，直到傳輸完畢為止。當傳輸完成，PDD會呼叫傳輸完成的召回功能，再由MDD轉送給用戶端的執行作業。

以上簡述的資料傳輸工作順序顯示了一項訊息：**將資料從AP搬移到USB周邊的常式執行必須要快。**在16-bit 匯流排的系統中必須重複執行此常式5,000次，才能移動10K的資料。若您的處理器對USB周邊具備DMA (Direct Memory Access；直接記憶體存取) 的功能，就可提供最快速的資料傳輸時間。若無DMA，就必須確定您的傳輸迴圈速度要夠快。我們一開始的IN資料傳輸迴圈就像圖4標示BEFORE中的樣子。

BEFORE

```

540 :   while (len-- )
541 :   {
542 :       pContext->pAntiochBaseAddress[addr] = *data_p++;
543 :   }
```

AFTER

```

WORD volatile *destPtr = &pContext->pAntiochBaseAddress[addr];
while (len>=16) // Note: Len is in WORDs, not BYTEs
{
    *destPtr = *data_p++;
    *destPtr = *data_p++;
    ... 16 times total...
    *destPtr = *data_p++;
    len -=16;
}
```

圖 4. 可最佳化資料傳輸順序工作之程式碼範例

雖然這是結構明確的程式碼，還是會編譯成11個組語層級的指令。這在高效能系統中是無法被接受的。每次都用專屬的指標取代重複的 `pContext->pAntiochBaseAddress[addr]`，如此可將迴圈縮小成10個指令；將迴圈頭段展開成多字元，可將每位元組的運算成本降到3個組語指令以下，則修改過的程式碼就如圖4的AFTER中所示。在一些處理器中，使用對應到 `data_p` 的相對位址會比每個指令累增 `data_p` 有更好的效果。在您的SOURCES檔案中加入 `"WINCECOD=1"` 的指令，就可以察看自己的程式碼。

另一項需要追蹤的地方就是您的CPU與USB周邊硬體之間所用的匯流排時脈。在我們一開始的硬體測試中，我們選擇最慢的匯流排時序，以完全確定我們能夠符合USB周邊晶片的時序需求。我們很高興且意外地發現透過記憶體管理單元的編程，就能將效能提昇至我們周邊設備所能容許的最快速度。

如果您搜尋Windows CE的文件檔案，就會發現一份標題為 “Optimizing a USB Function Driver Stack” 的文件，當中提供一個最佳化的方式—使用DMA，而該文也描述了如何配置實體緩衝區以執行DMA。但是由於匯流排寬度可能不合、訊號不相容、以及沒有針腳可用等問題，因此少有處理器/周邊組合有能力執行DMA。微軟該份文件中又指出：「由微軟所提供的USB功能用戶端驅動程式中，都未使用這項可客製化的參數。」

Cypress公司的Antioch

CYWBO124AB是一顆可執行DMA的晶片(如圖5)。這顆晶片其中一項獨一無二的優點，就是它可在儲存裝置與USB之間提供一條直接的資料路徑，而無須CPU介入處理資料傳輸的作業，如此也帶來許多提昇速率的機會，包括大量儲存裝置可直接與USB連結而無須Windows Mobile的介入，或是Windows Mobile處理器可維持對儲存裝置的控制，並且利用USB與SD/MMC卡之間的直接資料路徑作為超高速的DMA通道。這在標準的Windows協定堆疊上是相當大的改良，因為原作法需要四次移動才能將資料由USB傳送至儲存裝置上—USB到CPU、CPU到記憶體、記憶體到CPU、以及CPU到儲存裝置。

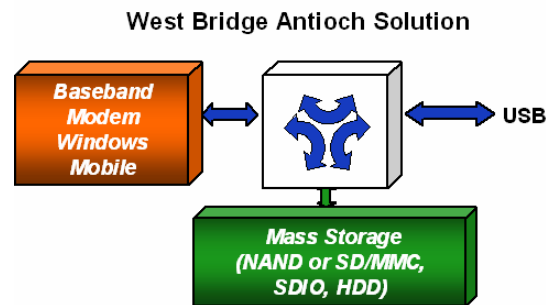


圖 5. Cypress具備直接記憶體存取 (DMA) 功能的 West Bridge Antioch

部分USB 1.1驅動程式為插斷 (interrupt) 驅動；每個封包都透過插斷服務常式 (Interrupt Service Routine ; ISR) 處理。這種方式在全速USB模式中還可接受，但在高速模式中，由於USB封包相對較短，而且匯流排速度極高，使得此法並不可行。高速USB主控端在實際應用中的傳輸速度可達40MByte/second，這代表每125微秒就會有10個長度為512-byte的封包到達。若在進出ISR時，不嘗試做環境切換 (context switch)，光是要有效率地移動這些資料就很困難。而ISR所引出的問題就是工作切換 (task switching) 的問題。即使是64KByte的封包，或多或少都會對下一個工作造成延遲，如果可以適當地傳送，其實所花的時間不到2毫秒。(如下表一)

最後，請記得USB只是日益複雜的大型系統中之一部分。當您的系統執行全速模式USB時，卻只得到300Kbytes/second的傳輸速率，您很容易會將責任歸咎於USB。然而，USB本身實際上有將近每秒1MByte的傳輸速率。用300Kbytes/second的速率傳送10MByte的檔案時雖需30秒，但其中有10秒是處理USB控制訊息，有20秒則是處理檔案系統與儲存裝置的額外資訊。如果改用高速USB，傳送10Mbytes的檔案就不用一秒。這也凸顯了一項事實，就是系統中其他部分是需要進行效能最佳化的。微軟公司提供了數種能協助您最佳化效能的工具。而在我們的經驗中，一套稱為Remote Kernel Tracker的工具對於效能改善相當有用，雖然是一種非侵入式方式，但仍可針對工作切換、臨界區間 (critical sections)、以及其他系統層級的事件提供很棒的資訊。

最佳效能檢查表:

- 字元層級 (Word-level):**
資料傳輸時快速的內部迴圈。
- 硬體層級 (Hardware level):**
依傳輸速率作最佳化的匯流排時脈。
- 無緩衝區複製 (No Buffer copies):**
不將資料從某一緩衝區複製到另一緩衝區。
- 無插斷服務常式 (No ISRs):**
無法接受ISR造成的overhead。同樣也無法接受作業系統服務。
- 無執行緒切換 (No thread switches):**
維持CPU處理直到傳輸結束。
- 關於系統:**
USB的優先順序需高於呼叫工作。

表一

瞭解驅動程式方面的運作、傳輸作業順序、以及效能最佳化的部分，都有助於工程師更進一步設計出具備理想USB效能的智慧型手機。隨著未來市場上會有更多的智慧型手機推出，改用Symbian平台以外的作業系統趨勢，以及手機不斷增加的頻寬需求，設計人員面對這些議題時，需要準備好提供快速資料傳輸的解決方案。若能仔細規劃並專注效能最佳化，設計人員研發的產品就可隨時準備好因應全球競爭最激烈的消費市場。



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.