

第十一章 USB 裝置控制器

此章節說明通用序列匯流排 (USB) 協定與 PXA250 與 PXA210 應用處理器裝置控制器的特殊實作選擇。這些選擇包括端點數目、類型、功能；對 Megacell 的中斷；以及一個傳送/接收 FIFO 介面。關於 USB 標準的操作知識對於要有效率地使用此章節是很重要的。通用序列匯流排裝置控制器 (UDC) 為遵從 USB 及支援所有主機 (host) 發出的標準裝置請求。關於 USB 協定完整的說明，請參考 *Universal Serial Bus Specification, revision 1.1*。在 <http://www.usb.org> 可以取得這份文件。

通用序列匯流排 (Universal Serial Bus, USB) 是外接式匯流排的標準，可以支援到 12Mbps 的傳輸速率。一個 USB 連接埠最多可以連接 127 個裝置。可連接 USB 的裝置包括 USB 滑鼠、數據機、鍵盤或不斷電系統 (uninterruptible power supply, UPS) 的監視器。USB 匯流排支援隨插即用的技術並且支援熱插拔 (可以在不需要關閉電源的情況下插入或拔除)。USB 的標準是定義在 IEEE1384 中。

新版的 USB 2.0 速度是舊版的 40 倍，資料的傳輸率可以到達 480Mbps。當 USB 2.0 成為標準之後，原有 USB 1.1 的裝置像是鍵盤、滑鼠、搖桿或喇叭等，並不需要更換，因為他們不需要這麼快的速度就可以正常運作。而一些需要較大頻寬的裝置，像是網路攝影機或是大容量儲存系統等，就能夠充分利用 USB 2.0 的優點了。

新版本的 USB 2.0 是 USB On-the-Go (OTG)。這種規格新增了行動運算裝置，像是 PDA、行動電話、數位相機等一些主機功能 (host functionality)。OTG 不僅允許這些周邊裝置連接到 PC，同時也可以連接到其他的 USB 裝置。舉例來說，當有 OTG 支援時，數位像機的照片可以直接下載到 PC 上，直接連到印表機，或是連接到具有視訊功能的 PDA。

11.1 USB 概觀

UDC 支援 16 個端點，而且可以 12 Mbps 的半雙工速率來操作 (只可當成 slave，不可當成 host 或 hub 控制器)。UDC 支援 4 種裝置組態。組態 1、2、3 各個都支援 2 個介面。這允許主機容許實體匯流排拓撲之動態改變。組態為 USB 裝置上可用資源的特殊結合。介面為與端點有關的組合，呈現裝置的特色與功能給主機。

UDC 傳送序列資料，包含溝通協定的階層。欄位為最基本的協定。UDC 欄位包含：同步、封包識別字、位址、端點、訊框數目、資料，以及循環冗餘核對 (CRC)。欄位被結合來產生封包。一個封包的功能決定結合與組成封包的欄位數目。封包類型包含：符記 (token)、訊框的開始、資料、信號交換。封包被組合成群組來產生交易 (transaction)。交易分成四種的種類：大量 (bulk)、控制、中斷、等時性的 (isochronous)。端點 0 只用來與設定 UDC 組態的控制交易作溝通。端點 0 的職責包括：連接、指定位址、端點組態、匯流排計數、切斷連接。

UDC 使用雙埠 (dual-port) 記憶體支援 FIFO 操作。每一個 Bulk 與 Isochronous 端點 FIFO 使用雙倍緩衝區來啟動端點處理封包 (當此封包與其他封包組合時)。DMA 與 Megacell 可為填滿或是將 FIFO 清空。當接收一個封包時，會產生一個中斷或 DMA 服務請求。DMA 引擎以 32 位元組增加量來服務 UDC FIFO。當 FIFO 遭遇一個短的封包或長度為 0 的封包時，也會產生中斷。端點 0 具有一個 16 個通道長、8 位元寬的 FIFO，只可被處理器讀取或寫入。

對於端點 1 ~ 15，UDC 使用它的雙埠記憶體在確認交易錯誤時保留 Bulk OUT 交易的資料。若 Bulk OUT 交易資料為無效的，則 UDC 會傳送一個 NAK 交換信號來請求主機重送資料。軟體不會被通知 OUT 資料是無效的，直到 Bulk OUT 資料被接收及驗證。若主機傳送一個 NAK 信號來回覆 Bulk IN 資料交易，UDC 會重傳資料。因為 FIFO 會維持一份資料的拷貝，所以軟體不需要重新載入資料。

UDC 介面專用的外部腳位為 UDC+與 UDC-。USB 協定在兩個腳位使用不同的信號來作半雙工資料傳輸。當 UDC+腳位沒有驅動時，必須連接一個 1.5 k Ω 的拉起 (pull-up) 電阻至 USB 電纜的 D+訊號來將 UDC+腳位拉高。當 UDC+沒有被驅動時將其拉高可允許 UDC 為高速模式、12-Mbps 裝置，以及為資料傳輸提供正確極性。序列匯流排使用不同的信號來同時傳送多個狀態。組合這些狀態來產生傳輸資料與多種匯流排狀態，包括：閒置 (Idle)、恢復 (Resume)、封包開始 (Start of Packet)、封包結束 (End of Packet)、切斷連接 (Disconnect)、連接 (Connect)、重置 (Reset)。

11.2 裝置組態

表 11-1 為裝置組態。

表 11-1 端點組態

端點編號	類型	功能	FIFO 大小 (bytes) ×FIFO 數目
0	Control	IN/OUT	16
1	Bulk	IN	64×2
2	Bulk	OUT	64×2
3	Isochronous	IN	256×2
4	Isochronous	OUT	256×2
5	Interrupt	IN	8
6	Bulk	IN	64×2
7	Bulk	OUT	64×2
8	Isochronous	IN	256×2
9	Isochronous	OUT	256×2
10	Interrupt	IN	8
11	Bulk	IN	64×2
12	Bulk	OUT	64×2
13	Isochronous	IN	256×2
14	Isochronous	OUT	256×2
15	Interrupt	IN	8

資料流程與 USB 主機相關。IN 封包資料流程為從 UDC 至主機。OUT 封包資料流程為從主機至 UDC。

Bulk 與 Isochronous 端點之 FIFO 為雙倍緩衝區 (double-buffered)，所以當下一個封包正在組合時，可以處理此封包。當 UDC 從一個特定的端點傳輸一個 IN 封包時，Megacell 可載入相同的端點來坐下一個訊框傳輸。當 Megacell 卸下一個 OUT 端點，UDC 可繼續處理下一個進入此端點的封包。

11.3 USB 協定

在 Megacell 重置之後，或是當 USB 主機發起一個 USB 重置時，UDC 會設定所有端點的組態，並且強制使用 USB 預設位址 0。UDC 設定端點組態之後，主機會分配 UDC 一個唯一的位址。在這個點上，UDC 在主機控制下，回應使用控制交易來傳輸至端點 0 的命令。

11.3.1 信號層級

USB 使用不同的信號來作資料編碼與指示不同的匯流排情況。USB 規格與

J、K 資料狀態相關，可區別高速與低速之間的傳輸。因為 UDC 只支援 12 Mbps 速率的傳輸，只能產生索引至實際的資料狀態 0 與資料狀態 1。

藉由作 UDC+與 UDC-腳位的編碼與使用不同的資料，會表現出四個不同的狀態。其中兩個狀態用來呈現資料。「1」表示 UDC+為高電位，而 UDC-為低電位。「0」表示 UDC+為低電位，而 UDC-為高電位。其餘的兩個狀態作解碼來呈現目前 USB 的狀態。表 11-2 顯示出不同的信號如何呈現出 8 種不同的匯流排狀態。

表 11-2 USB 狀態

匯流排狀態	說明
Idle	UDC+為高電位，UDC-為低電位（與 1 相同）。
Suspend	Idle 狀態超過 3ms。
Resume	UDC+為低電位，UDC-為高電位（與 0 相同）。
Start of Packet	自 Idle 轉換為 Resume。
End of Packet	UDC+與 UDC-為低電位達 2 位元時間，之後為 1 位元時間的 Idle。
Disconnect	UDC+與 UDC-在單一結束低電位門檻之下超過 2.5 μ s。 （切斷連接為靜態匯流排狀態，當沒有裝置接到集線器埠時則會發生。）
Connect	UDC+或 UDC-為高電位超過 2.5 μ s。
Reset	UDC+與 UDC-為低電位超過 2.5 μ s。（由主機控制器重置，及由裝置控制器感測）。

主機與集線器在 D+與 D-線上具有拉下（pull-down）電阻。當裝置沒有接上電纜線，拉下電阻會引發 D+與 D-被拉下至主機或集線器的單一結束低門檻之下。這會產生一個狀態，稱為單一結束 0（single-ended zero，SE0）。當 UDC 連接至 USB 電纜線時，UDC+腳位上的拉起電阻會將 D+拉起高於單一結束高門檻層級。經過 2.5 μ s 之後，主機會偵測連線。

在主機偵測到一個連接之後，匯流排會為 Idle 狀態，因為 UDC+為高電位而 UDC-為低電位。從 Idle 狀態轉換至 Resume 狀態的匯流排會發出封包開始(SOP)的訊號。每一個 USB 封包會與啟動 1-至-0 轉換的同步欄位一同啟動（請參考第 12-9 頁，12.3.1 節「信號層級」）。在傳送完封包資料之後，匯流排會藉由拉 UDC+與 UDC-低電位 2 位元時間來發出 EOP 狀態訊號，之後為 1 位元時間的閒置（Idle）狀態。若閒置持續超過 3ms，UDC 會進入暫停（Suspend）狀態與低功率模式。host 可藉由發出重置訊號，或經由正常的匯流排活動來將匯流排轉變為恢復（resume）狀態，來從暫停狀態喚醒 UDC。在正常的操作情況下，

host 會定期發出訊框啟動 (SOF) 訊號以擔保裝置不會進入暫停狀態。

11.3.2 位元編碼

USB 使用不歸零翻轉 (NRZI) 來作個別位元的編碼。在相同的信號內時脈與資料會被編碼與傳送。資料會藉由轉換來呈現，而不是藉由訊號的狀態。藉由轉換來表示 0，而藉由無轉換來表示 1，並且會產生資料。每次發生 0 時，接收邏輯會同步化進來的資料之鮑時脈 (baud clock)。為了確保接收端會定期同步化，序列位元流內六個連續的 1 會驅動傳送端插入一個 0。此程序即為位元填充 (bit stuffing)。接收邏輯會偵測填充位元並將其自進來的資料中移除。位元填充會至少每 7 個位元就會在進來的訊號上引發一個轉換，以確定已鎖定鮑時脈。從偵測到 SOP 開始，會啟動位元填充以完成完整的封包，直到偵測到 EOP (從 Sync 欄位啟動到 CRC 欄位)。圖 11-1 為資料位元組 0b1101 0010 之 NRZI 編碼。

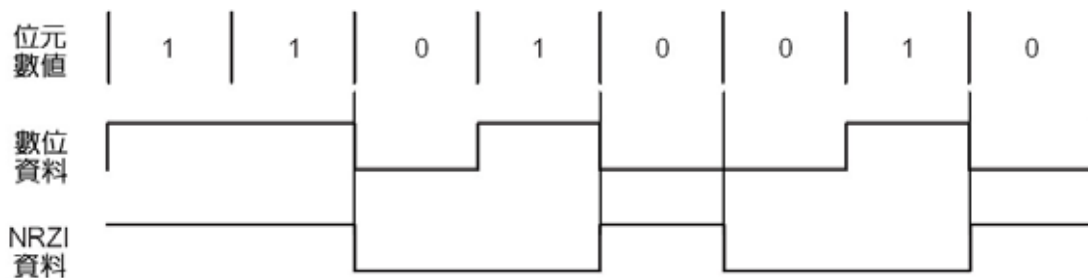


圖 11-1 NRZI 位元編碼範例

11.3.3 欄位格式

組合個別的位元為群組即稱為欄位 (field)。欄位用來構成封包 (packet)，而封包用來構成訊框 (frame) 或交易 (transaction)。具有 7 個 USB 欄位類別：同步 (Sync)、PID、位址 (Address)、端點 (Endpoint)、訊框數目 (Frame Number)、資料 (Data)、CRC。

在 Idle 狀態之前為 Sync 欄位，為每個封包的第一個欄位。Sync 欄位的第一個位元會發出 SOP 訊號至 UDC 或主機。Sync 為 8 位元寬，包含七個 0 與一個 1 (0x80)。在每個欄位內 (CRC 欄位除外)，位元會先被傳送至匯流排的 LSB 位元。

PID 為 1 位元組寬，位於 Sync 欄位之後。前四個位元包含編碼過的值，代表封包類型（Token、Data、Handshake、Special）、封包格式、與錯誤偵測的類型。後四個位元包含確認欄位，確定傳送 PID 沒有發生錯誤。確認欄位是藉由執行 PID 的 1 的補數而產生。UDC 將 PID 與 CRC 欄位做 XOR，若結果不全為 1，即表示傳輸時有錯誤發生而必須執行 USB 標準所規定的動作。

Address 與 Endpoint 欄位用來存取 UDC 的 16 個端點。Address 欄位包含 7 個位元，允許 127 個獨特的裝置置於 USB。在 USB 主機訊號重置之後，會指定 UDC 與其他所有裝置預設的位址—0。而後主機負責分配一個獨特的位址給每個匯流排上的裝置。位址會依照列舉出來的過程來分配，一次分配一個裝置。在主機分配位址給 UDC 之後，UDC 只會回應直接指向其位址的交易。在傳送每個封包時，Address 欄位會接在 PID 之後。

當 UDC 偵測到定址至自己的封包時，它會使用 Endpoint 欄位來決定要定址到哪一個 UDC 端點。Endpoint 欄位包含 4 個位元。允許執行端點 0（0000b）到 15（1111b）的編碼。端點欄位接在 Address 欄位之後。

Frame Number 為一個 11 位元寬的欄位，主機每一次傳送訊框會增加其值。當達到最大值 2047（0x7FF）時，其值會循環。SOF 封包會傳送 Frame Number 傳送，主機每隔為 1ms 會輸出此封包一次。裝置控制器使用 Frame Number 欄位來控制等時傳輸。Data 欄位用來在主機與 UDC 之間傳送封包資料。Data 欄位包含 0 到 1023 位元組。每一個位元組會先傳送 LSB。UDC 產生中斷指示發生訊框開始（Start of Frame）事件。

CRC 欄位用來偵測符記（token）與資料傳輸期間所發生的錯誤，而且應用至封包內全部的欄位（PID 欄位除外）。PID 包含自己的 4 位元的 1 的補數確認欄位來做錯誤偵測。Token 封包使用一個 5 位元 CRC（ $x^5 + x^2 + 1$ ）—稱為 CRC5，而 Data 封包使用一個 16 位元 CRC（ $x^{16} + x^{15} + x^2 + 1$ ）—稱為 CRC16。對於這兩個 CRC，在每一個封包開始時檢查員都會重置所有的 1。

11.3.4 封包格式

USB 支援四種封包類型：Token、Data、Handshake、Special。PRE（Preamble）PID 較慢速（1.5Mbps）的 USB 傳輸優先。UDC 只支援高速（12Mbps）USB 傳送。表示低速裝置的 PRE 封包與 PRE 封包之前的資料傳送會被忽略。

11.3.4.1 Token 封包類型

Token 封包位於訊框的開始，用來識別 OUT、IN、SOF 與 SETUP 處理。OUT 與 IN 訊框用來傳送資料，SOF 封包用來安排等時處理的時間，而 SETUP 封包用來做控制組態端點的傳送。Token 封包包含 Sync、PID，Address、Endpoint 與 CRC5 欄位（請參考表 11-3）。對於 OUT 和 SETUP 處理，位址與端點欄位用來選擇接收資料的 UDC 端點。對於 IN 處理，位址與端點欄位用來選擇傳送資料的 UDC 端點。

表 11-3 IN、OUT 與 SETUP Token 封包格式

8 位元	8 位元	7 位元	4 位元	5 位元
Sync	PID	Address	Endpoint	CRC5

11.3.4.2 訊框開始封包類別

SOF 為一個特別的 Token 封包類型，主機每隔一段很小的間隔（1ms +/- 0.0005ms）就會發出一個。SOF 封包包含 Sync、PID、Frame Number（每一次傳送訊框就會增加），以及 CRC5 欄位（請參考表 11-4）。SOF 封包每 1ms 會防止 UDC 進入暫停（Suspend）模式。

表 11-4 SOF Token 封包格式

8 位元	8 位元	11 位元	5 位元
Sync	PID	Frame Number	CRC5

11.3.4.3 資料封包型態

資料封包位於符號封包之後，被使用來在主機和 UDC 之間傳輸資料。PID 指示資料封包的兩種型態：DATA0 和 DATA1。使用這些資料封包以提供一個機制來確定將傳送端和接收端之間的多重交易資料順序同步化。信號交換的期間，傳送者和接收者決定哪個資料符號型態先被傳輸。當每傳輸一個子序列封包，資料封包型態就會被觸發（DATA0、DATA1、DATA0... 等等）。一個資料封包由一個 Sync、一個 PID、0~1023 位元組的資料和一個 CRC16 區域（請參考表 11-5）所組成。UDC 支援一個 Interrupt IN 資料最多可負載 8 位元組的資料、Bulk 資料最多可負載 64 位元組的資料、Isochronous 資料最多可負載 256 位元組的資料。

表 11-5 資料封包格式

8 位元	8 位元	0 ~ 1023 位元組	16 位元
Sync	PID	資料	CRC16

11.3.4.4 信號交換封包型態

信號交換封包型態由一個 Sync 和一個 PID 所組成。信號交換封包沒有 CRC，因為 PID 有自己的確認欄位。使用信號交換封包來描述資料執行狀態，包括確認資料已經成功接收、流量控制以及延遲情況。只有支援流量控制的執行可以傳回信號交換。信號交換封包的三種型態：ACK、NAK 和 STALL。ACK 指出資料封包已被接收，沒有位元填充、CRC 或是 PID 查核錯誤發生。NAK 指出 UDC 無法從主機接收資料或是無傳輸資料。STALL 指出 UDC 無法傳送或接收資料，需要主機介入來清除延遲狀況。接收單位利用刪除信號交換封包來發出位元填充、CRC 和 PID 錯誤訊號。表 11-6 說明信號交換封包的格式。

表 11-6 信號交換封包格式

8 位元	8 位元
Sync	PID

11.3.5 交易格式

封包被組合成一個群組來交易。USB 協定使用四個不同的執行格式。交易格式依照特定類型的端點而決定：大量 (Bulk)、控制 (Control)、中斷 (Interrupt) 或是等時性 (Isochronous)。端點 0 預設為控制端點，只接收控制交易。USB 所有的執行都由主機控制器開始，主機和 UDC 之間同時只能有一個方向的傳輸(半雙工)。

11.3.5.1 大量交易型態

大量交易保證在主機和 UDC 之間傳輸資料無錯誤，因為使用封包錯誤偵測和重試。當匯流排有可用的時間，主機計畫大量封包。大量交易由三個封包型態組成：符號、資料和信號交換。大量交易的八個型態是基於資料方向、錯誤和延遲情況。表 11-7 說明大量交易的型態，「封包從 UDC 傳送到主機」用粗體標示出來，而「封包從主機傳送到 UDC」則沒有。

表 11-7 大量交易格式

動作	符號封包	資料封包	信號交換封包
主機成功地從 UDC 接收資料	IN	DATA0/DATA1	ACK
UDC 暫時無法傳送資料	IN	無	NAK
UDC 端需要主機介入	IN	無	STALL
主機偵測出 PID、CRC 或位元填充錯誤	IN	DATA0/DATA1	無
UDC 成功地從主機接收資料	OUT	DATA0/DATA1	ACK
UDC 暫時無法接收資料	OUT	DATA0/DATA1	NAK
UDC 端需要主機介入	OUT	DATA0/DATA1	STALL
UDC 偵測出 PID、CRC 或位元填充錯誤	OUT	DATA0/DATA1	無
從 UDC 到主機的封包為 粗體字			

11.3.5.2 等時性交易型態

等時性交易保證固定的速率，以及主機和 UDC 之間資料傳輸錯誤容忍。在每個訊框期間，主機計畫等時性封包。USB 協定允許等時性傳輸可高達 USB 頻寬的 90%。不像大量交易，若收到毀壞的資料，UDC 將繼續處理毀壞的資料，毀壞的資料相當於訊框指示器的目前起始。等時性交易不支援信號交換或是重試功能。等時性交易由兩個封包型態組成：符號和資料。表 11-8 說明資料方向的等時性交易型態。

表 11-8 等時性交易格式

動作	符號封包	資料封包
主機從 UDC 接收資料	IN	DATA0
UDC 從主機接收資料	OUT	DATA0
從 UDC 到主機的封包為 粗體字		

11.3.5.3 控制交易型態

主機使用控制交易來組態端點且詢問它們的狀態。與大量交易相同，控制交易的起始為建立封包，接下來是選擇性的資料封包，然後是信號交換封包。控制交易預設使用 DATA0 型態傳送。表 11-9 說明控制交易的四個型態。

表 11-9 控制交易格式

動作	符號封包	資料封包	信號交換封包
UDC 成功地從主機接收控制	SETUP	DATA0	ACK
UDC 暫時無法接收資料	SETUP	DATA0	NAK
UDC 需要主機介入	SETUP	DATA0	STALL
UDC 偵測出 PID、CRC 或位元填充錯誤	SETUP	DATA0	無
從 UDC 到主機的封包為 粗體字			

為了組合控制傳輸，主機送出控制交易來通知 UDC 控制傳輸為何種型態（控制讀取或是控制寫入），此控制交易隨後接著一個或多個資料交易。建立（setup）為控制傳輸的第一個階段。裝置必須回應 ACK 或無信號交換（若資料被毀壞）。控制交易預設為使用 DATA0 傳輸，且觸發在 DATA1 和 DATA0 之間傳輸的每個子序列資料交易。使用 OUT 交易來對端點做控制寫入；使用 IN 交易做控制讀取。傳輸方向和最後的資料交易相反。使用傳輸方向來描述信號交換的狀態和功能。為了控制寫入，從 UDC 到主機的 IN 為最後交易；為了控制讀取，從主機到 UDC 的 OUT 為最後交易。最後資料交易總是使用 DATA1 傳輸，即使先前用 DATA1 傳輸。

11.3.5.4 中斷交易型態

主機使用中斷交易來詢問設備的狀態。如同大量交易，中斷交易的起始為建立封包，接下來是選擇資料封包，然後是信號交換封包。中斷交易預設為使用 DATA0 型態傳輸。表 11-10 說明中斷交易的四個型態。

表 11-10 中斷交易格式

動作	符號封包	資料封包	信號交換封包
主機成功地從 UDC 接收資料	IN	DATA0	ACK
UDC 暫時無法傳送資料	IN	無	NAK
UDC 端點需要主機介入	IN	無	STALL
主機偵測出 PID、CRC 或位元填充錯誤	IN	DATA0	無
從 UDC 到主機的封包為 粗體字			

11.3.6 UDC 設備請求

UDC 使用它的控制、狀態和資料暫存器來控制和監視端點 1~15 的傳送和接收 FIFO。Host 控制其他所有的 UDC 組態與回報使用裝置請求（如控制交易，經由 USB 送至端點 0）的狀態。端點的每個建立封包為 8 位元組長，其他詳細說明如下：

- 資料傳輸方向：主機到設備，設備到主機
- 資料傳輸型態：標準、分級(class)、製造(vendor)
- 資料接收：設備、介面、端點、其他
- 傳輸的位元組數目
- Index 或 Offset
- 值：用來通過(pass)可變大小的資料參數
- 設備請求

表 11-11 說明所有設備請求的摘要。主機設備請求的完整描述請參考 *Universal Serial Bus specification Revision 1.1*。

表 11-11 主機設備請求摘要

請求	名稱
SET_FEATURE	啟動特定的特色，例如設備遠程甦醒或端點延遲。
CLEAR_FEATURE	清除或關閉特定的特色。
SET_CONFIGURATION	為了操作，將 UDC 組態。在百萬格(Megacell)重置後或是經由 USB 發出重置訊號後使用。
GET_CONFIGURATION	UDC 目前組態傳回給主機。
SET_DESCRIPTOR	設定存在敘述元或新敘述元。存在敘述元包括：設備、組態、串流、介面和端點。
GET_DESCRIPTOR	若它存在，則傳回特定的敘述元。
SET_INTERFACE	為 UDC 的介面選擇一個交替設定。
GET_INTERFACE	傳回選擇的交替設定給特定的特色。
GET_STATUS	傳回 UDC 的狀態，包括：遠端甦醒、自行供電、資料方向、端點數目和延遲情況。
SET_ADDRESS	為了未來設備存取，設定 UDC 的 7 位元位址值。
SYNCH_FRAME	設定且描述端點的同步訊框。

使用者無請求介入，UDC 可解碼大部分的標準設備指令。以下為使用者無法使用的指令：設定位址 (Set Address)、設定特色 (Set Feature)、清除特色 (Clear Feature)、取得組態 (Get Configuration)、取得狀態 (Get Status)、取得介面 (Get Interface) 和同步訊框。使用者用 Set Configuration 和 Set Interface 指令可指示主機設定特定的組態或介面，且軟體必須採取任何必須的行動。使用者可將 Get

Descriptor 和 Set Descriptor 指令解碼。

因為無法使用 Set Feature 和 Clear Feature 指令，使用者無法將設備遠端甦醒特色指令解碼。為了解決這個問題，狀態位元 UDCCS0:DRWF 指示不管設備遠端甦醒特色沒有被啟動，UDCCS0:DRWF 為唯讀位元。當此位元設定為 1，啟動設備遠端甦醒特色；當設定為 0，關閉特色。

11.3.7 組態

對 GET_DESCRIPTOR 指令做回應，使用者設備送回一個 UDC 組態的描述。UDC 可以實體支援的資料通道頻寬比 USB 規格允許的更多。當設備回應主機時，必須詳細說明合法的 USB 組態。例如，若設備說明 6 個等時性端點（每個有 256 位元組）的組態，主機無法安排適當的頻寬且無法將 UDC 帶離組態 0。使用者設備決定哪個端點回應主機，若端點無法回應，則不使用。另一個選擇，使用等時性端點，描述封包組態的最大值小於主機 256 位元組(to describe a configuration of a packet with a maximum size less than 256 bytes to the host.為敘述)例如，若軟體回應 GET_DESCRIPTOR 指令說明端點 3 的封包等時性 IN 資料之最大值為 64 位元組，在它載入傳輸的 64 位元組之後，使用者必須設定 UDCCS3[TSP]位元。簡言之，若端點支援 128 位元組最大值封包等時性 OUT 資料，UDC 承認封包的終端，設定 UDCCS4[RPS]，中斷就會產生。

端點方向是固定的。實際上，UDC 只支援中斷端點的封包尺寸最大值為 8 位元組或更少、大量端點為 64 位元組或更少、等時性端點為 256 位元組或更少。

為了有更適合的應用處理器，UDC 支援總共四個組態。這些組態都在 UDC 裡，軟體可以有三個不同的組態，每個組態有兩個介面。組態 0 是端點 0 的預設組態，無法由任何其他的排列(arrangement)所定義。

主機完成 SET_CONFIGURATION 或 SET_INTERFACE 指令之後，軟體必須將指令解碼讓 OUT 端點淨空，且允許百萬格(Megacell)設定適當的電源/周邊組態。

11.4 UDC 硬體連結

此章節解釋各種設備如何連結 USB 介面。

11.4.1 自行供電設備

圖 11-5 說明自行供電設備如何連接 USB 介面。 0Ω 電阻為可選擇的，若沒使用到，USB D+ 必須和設備 UDC D+ 直接連結，而且，USB D- 必須和 UDC D- 直接連結。UDC D+ 和 UDC D- 腳位設計為適合 USB 電纜的阻抗，不具外部串聯電阻，為 90Ω 。為了允許較小的阻抗校正，對電路板追蹤結果做補償，建議在電路板上使用 0Ω 。

因為應用處理器的輸入腳位只可承受 3.3V，所以需要 5V~3.3V 的設備。設備可以依數個方法完成：最強壯、昂貴的方法為電源開啟重置設備，例如 MAX6348。此方法產生一個乾淨的訊號邊界，且將訊號傳輸反射最小化。稍便宜的解決方式為 3.3V 線性緩衝區，其輸入可以承受 5V。此方法無法減少訊號傳輸反射，所以軟體必須補償，利用重複地讀取 GPIO 直到它驗證穩定之後。第三個方法為訊號傳輸反射最小化電路，它可承受 5V 但是只會產生 3.3V 的訊號給 GPIO 腳位。

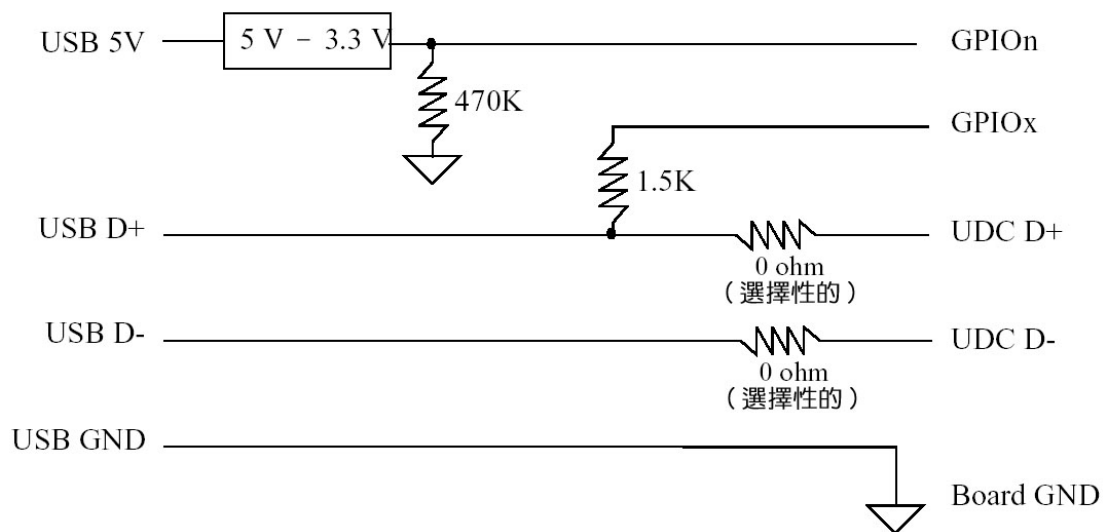


圖 11-2 自行供電設備

11.4.1.1 當 GPIOn 和 GPIOx 為不同腳位時

GPIOn 和 GPIOx 可為任何的 GPIO 腳位。要將設備從睡眠模式喚醒則 GPIOn

必須為 GPIO 腳位。重置後，GPIOx 組態為輸入，會造成 UDC+線變為浮動 (float)；GPIOn 組態成輸入，會在上升或下降邊界產生中斷。當中斷發生，軟體必須讀取 GPIOn 腳位來決定電纜是否連結。電纜有連結，GPIOn 腳位設定為 1；電纜沒有連結，設定為 0。當偵測到電纜有連結，軟體必須啟動 UDC 周邊，且以 1 驅動 GPIOx 腳位來指示主機電腦高速 USB 設備有連結。當偵測到電纜無連結，軟體必須將 GPIOx 組態成輸入，組態 GPIOn 以偵測甦醒事件，若有需要，會使無連結周邊處於睡眠模式。

若軟體必須使周邊處於睡眠模式，它會將 GPIOx 腳位組態成輸入，此會造成 UDC+線變為浮動 (floating)，看起來就好像與主機電腦沒連結。使周邊處於睡眠模式。當周邊承認睡眠模式，軟體必須以 1 驅動 GPIOx 腳位來指示主機電腦高速 USB 周邊有連結。

11.4.1.2 當 GPIOn 和 GPIOx 為同腳位

重置之後，GPIOn 組態為輸入，會在上升或下降邊界造產生中斷。當中斷發生，軟體必須讀取 GPIOn 腳位來決定電纜是否連結。電纜有連結，GPIOn 腳位設定為 1；電纜沒有連結，設定為 0。若偵測到有連結 USB，在主機電腦送出第一個 USB 指令之前，軟體必須啟動 UDC 周邊。若偵測到無連結 USB，軟體必須組態 GPIOn 腳位來偵測甦醒事件，若有需要，使周邊處於睡眠模式。

當 GPIOn 和 GPIOx 為同腳位，若設備沒有連結 USB 電纜，不用使周邊處於睡眠模式。睡眠期間，USB 控制器為重置狀態，且不會回應主機電腦。當它從睡眠模式恢復，周邊不會回應它的指定主機位址。

11.4 匯流排供電設備

應用處理器不支援匯流排供電設備，因為當主機發佈暫停，消耗會少於 500 μ A (請參考 *USB Specification, version 1.1* 的第 7.2.3 節)。應用處理器無法將其消耗的電流量限制為 500 μ A，除非進入睡眠模式。當應用處理器進入睡眠模式，它將 USB 暫存器重置，且不會回應它的指定主機位址。

11.5 UDC 操作

當接收到 USB 中斷，軟體會針對 USB ISR。USIR 為層級感應的，在離開 ISR 之前，最後一個步驟需確定清除 USIR。離開 ISR 之後，使用者應立即清除中斷

來源位元，然後清除 USIR。電源開啟或重置之後，軟體最初只會啟動 EP0 中斷。其他中斷啟動由 SET_CONFIG 指令請求。

11.5.1 狀況 1：EP0 控制讀取

1. 當軟體啟動，它會初始化軟體狀態機為 EP0_IDLE。當軟體與主機 PC 溝通時，軟體狀態機用來追蹤端點狀態。
2. 主機 PC 傳送一個 SETUP 命令。
3. UDC 產生一個 EP0 中斷。
4. 軟體決定 UDCCS0[SA]與 UDCCS0[OPR]位元的設定。這表示一個新的 OUT 封包在 EP0 緩衝區內並辨識 SETUP 交易。
5. 當設定 UDCCS0[RNE]位元（接收端不為空的）時，軟體讀取資料至緩衝區。
6. 軟體分析緩衝區內命令語法，並決定其為控制讀取。
7. 軟體開始將第一個封包資料載入 UDDR0 暫存器 FIFO，並設定內部狀態機器為 EP0_IN_DATA_PHASE。
8. 在其讀取與分析資料語法之後，軟體清除 UDCCS0[SA]與 UDCCS0[OPR]位元，並設定 UDCCS0[IPR]位元，此位元引起 UDC 在下一個 IN 時傳送資料。UDC 傳送 NAK 至此 EP 上的所有請求，直到設定 UDCCS0[IPR]位元。
9. 軟體清除 UDC 中斷位元並返回中斷服務程序。
10. host PC 發出 IN 封包，而 UDC 會傳送資料回到 host。在主機 PC 傳送 ACK 至 UDC 之後，UDC 清除 UDCCS0[IPR]位元並產生一個中斷。
11. 軟體進入 ISR 常式並檢驗其內部狀態機器。它判斷是否在 EP0_IN_DATA_PHASE 狀態內而必須傳輸更多的資料。軟體載入下個資料量時，若有需要則會設定 UDCCS0[IPR]位元，並返回中斷。內部狀態機器不會受到影響。
12. 重複步驟 10 與 11 直到傳送完所有資料或最後的資料封包為短封包。
13. 若最後的封包軟體傳送為一個短封包，則設定其內部狀態機器為 EP0_END_XFER。若最後的資料封包在 16 位元組邊界上，則軟體設定 UDCCS0[IPR]來傳送一個長度為 0 的封包而不會載入 FIFO 資料。在傳送長度為 0 的封包之後，軟體設定內部狀態機器為 EP0_END_XFER。
14. 當主機執行 STATUS OUT 階段（長度為 0 的 OUT）時，UDC 設定 UDCCS0[OPR]位元，並引發一個中斷。
15. 軟體進入 ISR 常式，並判斷是否設定 UDCCS0[OPR]位元、清除 UDCCS0[SA]位元，而其內部狀態機器為 EP0_END_XFER。軟體清除 UDCCS0[OPR]位元並傳送其內部狀態機器為 EP0_IDLE。

16. 軟體清除 UDC 中斷位元並自中斷服務常式回復。

若在這些步驟期間主機傳送另一個 SETUP 命令，則軟體必須終止第一個 SETUP 命令，並啟動一個新的命令。

11.5.2 情況 2：EP0 控制讀取與過早的狀態階段

在取得裝置敘述元（Get Device Descriptor）命令期間當主機 PC 傳送一個過早的（premature）狀態階段時，在一個列舉的週期內會發生這種情況。

1. 當軟體啟動，它會初始化軟體狀態機為 EP0_IDLE。當軟體與主機 PC 溝通時，軟體狀態機器用來追蹤端點狀態。
2. 主機 PC 傳送一個 SETUP 命令。
3. UDC 產生一個 EP0 中斷。
4. 軟體決定設定 UDCCS0[SA]與 UDCCS0[OPR]位元。這表示一個新的 OUT 封包在 EP0 緩衝區內並辨識 SETUP 交易。
5. 當設定 UDCCS0[RNE]位元（接收端不為空的）時，軟體讀取資料至緩衝區。
6. 軟體分析緩衝區內命令語法，並決定其為控制讀取。
7. 軟體開始將第一個資料封包載入 UDDR0 暫存器 FIFO，並設定內部狀態機器為 EP0_IN_DATA_PHASE。
8. 在其讀取與分析資料語法之後，軟體清除 UDCCS0[SA]與 UDCCS0[OPR]位元，並設定 UDCCS0[IPR]位元，此位元引起 UDC 在下一個 IN 時傳送資料。UDC 傳送 NAK 至此 EP 上的所有請求，直到設定 UDCCS0[IPR]位元。
9. 軟體清除 UDC 中斷位元並自中斷服務常式回復。
10. host PC 發出 IN 封包，而 UDC 會傳送資料回到 host。在 host PC 傳送 ACK 至 UDC 之後，UDC 清除 UDCCS0[IPR]位元並產生一個中斷。
11. 軟體進入 ISR 常式並檢驗其內部狀態機器。它決定在 EP0_IN_DATA_PHASE 狀態內而必須傳輸更多的資料。軟體載入下個資料量，若需要則設定 UDCCS0[IPR]位元，並自中斷回復。內部狀態機器不會受到影響。
12. 重複步驟 10 與 11 直到傳送完所有資料或最後的資料封包為短封包。
13. 當重複步驟 10 與 11 時，主機（host）會傳送一個過早的 STATUS OUT 階段，這表示主機 PC 無法接收更多的資料，而不是 IN 封包。
14. 當發生 EP0 中斷時，軟體決定設定 UDCCS0[OPR]腳位、清除 UDCCS0[SA]位元，而其內部狀態機器為 EP0_IN_DATA_PHASE。這

表示發生一個過早的 STATUS OUT。

15. 軟體清除 UDCCS0[OPR]位元並改變腳位狀態為 EP0_IDLE。軟體寫入至 UDCCS0[FTF]位元來清除任何緩衝區指標並清空傳輸 FIFO。
16. 軟體清除 UDC 中斷位元並自中斷服務常式回復。

若在這些步驟期間主機傳送另一個 SETUP 命令，則軟體必須終止第一個 SETUP 命令，並啟動一個新的命令。

11.5.3 情況 3：有或無過早狀態階段之 EP0 控制寫入

1. 當軟體啟動，它會初始化軟體狀態機為 EP0_IDLE。當軟體與主機 PC 溝通時，軟體狀態機器用來追蹤端點狀態。
2. 主機 PC 傳送一個 SETUP 命令。
3. UDC 產生一個 EP0 中斷。
4. 軟體決定設定 UDCCS0[SA]與 UDCCS0[OPR]位元。這表示一個新的 OUT 封包在 EP0 緩衝區內並辨識 SETUP 交易。
5. 當設定 UDCCS0[RNE]位元（接收端不為空的）時，軟體讀取資料至緩衝區。
6. 軟體分析緩衝區內命令語法，並決定其為控制寫入（例如：設定敘述元）。
7. 軟體設定內部為 EP0_OUT_DATA_PHASE，並清除 UDCCS0[OPR]與 UDCCS[SA]位元。
8. 為了允許過早的 STATUS IN 階段，軟體設定 UDCCS0[IPR]位元並載入一個長度為 0 的封包至傳輸 FIFO 內。
9. 軟體清除 UDC 中斷位元並自中斷服務常式回復。
10. 主機 PC 發出一個 OUT 封包，UDC 發出一個 EP0 中斷。
11. 軟體進入 ISR 常式並決定在 EP0_OUT_DATA_PHASE 狀態內設定 UDCCS0[OPR]位元、清除 UDCCS0[SA]位元。這表示有更多的資料要接收。
12. 當設定 UDCCS0[RNE]位元與清除 UDCCS0[OPR]位元時，軟體讀取資料進入緩衝區。
13. 軟體設定 UDCCS0[IPR]位元以允許過早的 STATUS IN 階段。
14. 轉體清除 UDC 中斷位元，並自中斷服務常式回復。
15. 重複步驟 11 至 14 直到接收所有資料。
16. 當重複步驟 11 至 14 時，主機會傳送一個過早的 STATUS IN 階段，這表示主機 PC 無法重送更多的資料，而不是 OUT 封包。STATUS IN 階段可能為過早或否。

17. 因為軟體載入一個長度為 0 的封包（參考步驟 8），UDC 以傳送一個長度為 0 的封包回主機 PC 來回應 STATUS IN。這會造成一個中斷。
18. 軟體進入 ISR 常式並決定在 EP0_OUT_DATA_PHASE 狀態內，且清除 UDCCS0[OPR]與 UDCCS0[IPR]位元。這表示發生 STATUS IN 階段。
19. 軟體決定在中斷之前接收多少位元組，並比較接收的位元組數目與原本的 SETUP 封包內之 wLength 欄位。若傳送的資料數目正確，軟體會分析資料並執行資料所指示的動作。若數目錯誤，則軟體清除任何緩衝區指標並忽略接收的資料。
20. 軟體改變內部狀態機器為 EP0_IDLE。
21. 軟體清除 UDC 中斷位元，並自中斷服務常式回復。

若在這些步驟期間主機傳送另一個 SETUP 命令，則軟體必須終止第一個 SETUP 命令，並啟動一個新的命令。

11.5.4 情況 4：EP0 無資料命令

1. 當軟體啟動，它會初始化軟體狀態機為 EP0_IDLE。當軟體與主機 PC 溝通時，軟體狀態機器用來追蹤端點狀態。
2. 主機 PC 傳送一個 SETUP 命令。
3. UDC 產生一個 EP0 中斷。
4. 軟體決定設定 UDCCS0[SA]與 UDCCS0[OPR]位元。這表示一個新的 OUT 封包在 EP0 緩衝區內並辨識 SETUP 交易。
5. 當設定 UDCCS0[RNE]位元（接收端不為空的）時，軟體讀取資料至緩衝區。
6. 軟體分析緩衝區內資料語法，並決定其為無資料命令。
7. 軟體執行命令，並設定內部狀態機器為 EP0_IDLE。軟體清除 UDCCS0[OPR]與 UDCCS0[SA]位元，但不可設定 UDCCS0[IPR]位元。
8. 當主機 PC 執行 STATUS IN 狀態時，UDC 會送回一個長度為 0 的封包，這表示完成一個成功的信號交換。不會產生中斷。

若在這些步驟期間主機傳送另一個 SETUP 命令，則軟體必須終止第一個 SETUP 命令，並啟動一個新的命令。

11.5.5 情況 5：EP1 資料傳送（BULK-IN）

情況 5 內的程序也可用來操作端點 6 與 11。

在情況 5 中，只有當小於 64 位元組的封包大小被傳送時，才會設定傳送短封包 (Transmit Short Packet)。若封包大小為 64 位元組，則當載入第 64 個位元組時系統會啟動 (arm)。載入第 64 個位元組與設定 UDCCS1[TSP]位元會產生一個 64 位元組的封包與一個長度為 0 的封包。

當軟體接收到 SETUP VENDOR 命令來建立一個 EP1 BULK IN 交易時，可能會執行下列兩種動作中適合所選擇的操作模型的一種：

- 設定 DMA 引擎組態並關閉 EP1 中斷，允許 DMA 引擎處理交易。
- 啟動 EP1 中斷來允許 Megacell 直接處理交易。

11.5.5.1 軟體啟動 DMA

若軟體啟動 DMA 引擎，則使用以下步驟：

1. 在 SETUP VENDOR 命令期間，軟體會啟動 DMA 引擎與遮蔽 EPI 中斷。DMA 開始位址必須對齊 16 位元組邊界。
 - a. 若封包大小為 64 位元組，軟體會傳送一個 DMA 敘述元內所有資料及設定第二個 DMA 敘述元內之 UDCCS1[TSP]位元。
 - b. 若封包大小小於 64 位元組，軟體會建立敘述元串，裡面的基數之敘述元指向資料，而偶數敘述元寫入 UDCCS1[TSP]位元。
2. 主機 PC 傳送 BULK-IN，而 UDC 傳送資料封包回主機 PC。
3. UDC 產生一個 Megacell 遮蔽的中斷。
4. DMA 引擎用資料填滿 EP1 資料 FIFO (UDDR1)，若資料封包為短封包則設定 UDCCS1[TSP]位元。
5. 重複步驟 2 到 4 直到所有的資料都被送至主機 PC。

11.5.5.2 軟體啟動 EP1 中斷

若軟體啟動 EP1 中斷以允許 Megacell 直接處理交易：

1. 在 SETUP VENDOR 命令期間，軟體會以資料填滿 EP1 資料 FIFO (UDDR1) 並清除 UDCCS1[TPC]位元。若資料封包為短封包，軟體也會設定 UDCCS1[TSP]位元。
2. 主機 PC 傳送 BULK-IN，而 UDC 傳送資料封包回主機 PC 並產生一個 EP1 中斷。
3. 軟體填滿 EP1 資料 FIFO (UDDR1)，並清除 UDCCS1[TPC]位元。若資料封包為短封包，則軟體也會設定 UDCCS1[TSP]位元。

4. 自中斷回復。
5. 重複步驟 2 到 4 直到所有的資料都被送至主機 PC。

11.5.6 情況 6：EP2 資料接收 (BULK-OUT)

此程序也可用來操作端點 7 與 12。

當軟體接收到 SETUP VENDOR 命令來建立一個 EP2 BULK OUT 交易，可能會執行下列兩種動作的中適合所選擇的操作模型的一種：

- 啟動 DMA 引擎來處理交易。
- 允許 Megacell 直接處理交易。

11.5.6.1 軟體啟動 DMA

若軟體啟動 DMA 引擎來處理交易：

1. 在 SETUP VENDOR 命令期間，軟體會建立 DMA 引擎並設定 UDCCS2[DME]位元。
 - a. 若封包大小為 32 或 64 位元組，軟體會建立敘述元串，每一個都以 32 或 64 為模數(modulo)。軟體為適當的敘述元設定中斷位元。
 - b. 若封包大小小於 32 位元組，則軟體使用中斷模式。
2. 主機 PC 傳送一個 BULK-OUT。
3. DMA 引擎從 EP2 資料 FIFO (UDDR2) 讀取資料。
4. 重複步驟 2 與 3 直到從主機讀取所有的資料。
5. 若軟體接收到 EP2 中斷，則完成下列處理：
 - a. 若 UDCCS2[RNE]被清除，而且 UDCCS2[RSP]被設定，則資料封包為一個長度為 0 的封包。
 - b. 若 UDCCS2[RNE]被設定，則資料封包為短封包，而軟體必須使用 UDCWC2 計數暫存器來從 EP2 資料 FIFO (UDDR2) 讀取適當的資料量。
 - c. 軟體清除 UDCCS[RPC]位元。：
6. 自中斷回復。

11.5.6.2 軟體允許 Megacell 處理交易

若軟體允許 Megacell 處理交易：

1. 在 SETUP VENDOR 命令期間，軟體會清除 UDCCS2[DME]位元。
2. 主機 PC 傳送 BULK-OUT，而 UDC 產生 EP2 中斷。
3. 若 UDCCS2[RNE]被清除，而且 UDCCS2[RSP]被設定，則資料封包為一長度為 0 的封包。
4. 若 UDCCS2[RNE]被設定，則軟體使用 UDCWC2 計數暫存器來從 EP2 資料 FIFO (UDDR2) 讀取適當的資料量。
5. 軟體清除 UDCCS2[RPC]位元。
6. 自中斷回復。
7. 重複步驟 2 到 6 直到從主機讀取所有的資料。

11.5.7 情況 7：EP3 資料傳送 (ISOCHRONOUS-IN)

情況 7 中的程序可用來操作端點 8 或 13。

在情況 7 中，只有當傳送的封包大小小於 256 位元組時才會設定傳送短封包。若封包大小為 256 位元組，當第 256 個位元組被載入時，則系統會預備操作。載入第 256 個位元組與設定 UDCCS3[TSP]位元會產生一個 256 位元封包與一個長度為 0 的封包。

當軟體接收到 SETUP VENDOR 命令來建立 EP3 ISOCHRONOUS IN 交易，可能會執行下列兩種動作中適合所選擇的操作模型的一種：

- 設定 DMA 引擎組態並關閉 EP3 中斷以允許 DMA 引擎處理交易。
- 發出 EP3 中斷來允許 Megacell 直接處理交易。
- 發出 SOF 中斷以訊框計數為基礎來處理交易。

11.5.7.1 軟體啟動 DMA

若軟體啟動 DMA 引擎處理交易：

1. 在 SETUPVENDOR 命令期間，軟體會啟動 DMA 引擎並遮蔽 EP3 中斷。DMA 啟動位址必須對齊 16 位元組邊界。
 - a. 若封包大小為 256 位元組，則軟體傳送 DMA 敘述元內所有的資料。

- b. 若封包大小小於 256 位元組，則軟體建立敘述元串，裡面的奇數的敘述元指向資料，偶數的敘述元寫入 UDCCS1[TSP]位元。
2. 主機 PC 傳送一個 ISON-IN，而 UDC 傳回一個資料封包至主機 PC。
3. UDC 產生 Megacell 遮蔽的中斷。
4. DMA 引擎填滿 EP3 資料 FIFO (UDDR3)，若資料封包為短封包則設定 UDCCS3[TSP]位元。
5. 重複步驟 2 到 4 直到傳送所有資料至主機。

11.5.7.2 軟體啟動 EP3 中斷

若軟體啟動 EP3 中斷來允許 Megacell 直接處理交易：

1. 在 SETUP VENDOR 命令期間，軟體填滿 EP3 資料 FIFO (UDDR3) 並清除 UDCCS3[TPC]位元。若資料封包為短封包，則軟體也會設定 UDCCS3[TSP]位元。
2. 主機 PC 傳送 ISOC-IN 命令，而 UDC 傳回資料封包給主機 PC 並產生一個 EP3 中斷。
3. 軟體填滿 EP3 資料 FIFO (UDDR3) 並清除 UDCCS3[TPC]位元。若資料封包為短封包，軟體也會設定 UDCCS3[TSP]位元。
4. 自中斷回復。
5. 重複步驟到 4 直到所有的資料都被傳送至主機 PC。

11.5.7.3 軟體啟動 SOF 中斷

若軟體啟動 SOF 中斷以訊框計數為基礎來處理交易：

1. 軟體設定 UICR3[IM3]為 1 來關閉 UDCCS3 中斷，並設定 UFNHR[SIM]為 0 來發出 UFNHR 暫存器內之 SOF 中斷。
2. 當主機 PC 傳送 SOF，UDC 會設定 UFNHR[SIR]位元，並引發一個 SOF 中斷。
3. 軟體確認 UDCCS3[TFS]來決定是否有空間給資料封包。若有空間，則軟體會用資料填滿 EP3 資料 FIFO (UDDR3) 並清除 UDCCS3[TPC]位元。若資料封包為短封包，則軟體設定 UDCCS3[TSP]位元。
4. 軟體清除 UFNHR[SIR]位元。
5. 自中斷回復。
6. 重複步驟 2 到 5 直到所有的資料都被送至主機 PC。

11.5.8 情況 8：EP4 資料接收 (ISOCHRONOUS-OUT)

情況 8 之程序也可用來操作端點 9 與 14。

當軟體接收到 SETUP VENDOR 命令來建立一個 EP4 ISOCHRONOUS OUT 交易時，可能會採取下列三個動作中較適合所選的操作模組的一個：

- 設定 DMA 引擎組態並關閉 EP4 中斷以允許 DMA 引擎處理交易。
- 啟動 EP4 中斷以允許 Megacell 直接處理交易。
- 啟動 SOF 中斷以訊框資料為基礎來處理交易。

11.5.8.1 軟體啟動 DMA

若軟體啟動 DMA 引擎，則執行以下步驟：

1. 在 SETUP VENDOR 命令期間，軟體啟動 DMA 引擎並設定 UDCCS4[DME]位元。不限制 ISO 封包大小，但封包大小為模數 (modulo) 32 的效能較高。
 - a. 若封包大小介於 32 至 256 位元組模數 32 之間，則軟體決定所需的敘述元數目並建立敘述元串。軟體為適當的敘述元設定中斷位元。
 - b. 若封包大小介於 32 位元組至 256 位元組，但非模數 32 位元組，則軟體建立一個敘述元來接收每個資料封包，而後讀取每個 UDCCS2[RSP]位元中斷剩下的資料，並建立另一個敘述元。
 - c. 若封包大小小於 32 位元組，則軟體必須使用中斷模式。
2. 主機 PC 送出 ISOC-OUT。
3. DMA 引擎自 EP4 資料 FIFO (UDDR4) 讀取資料。
4. 重複步驟 2 與 3 直到從主機讀取所有的資料。
5. 若軟體接收到 EP4 中斷，則完成以下程序：
 - a. 若 UDCCS4[RNE]被清除，而且 UDCCS4[RSP]位元被設定，則資料封包長度為 0。
 - b. 若 UDCCS4[RNE]被設定，資料封包為短封包，而軟體使用 UDCWC4 計數暫存器自 EP4 資料 FIFO (UDDR4) 讀取適當的資料量。
 - c. 軟體清除 UDCCS4[RPC]位元。
6. 自中斷回復。

11.5.8.2 軟體允許 Megacell 處理交易

若軟體允許 Megacell 處理交易：

1. 在 SETUP VENDOR 命令期間，軟體清除 UDCCS4[DME]位元。
2. 主機 PC 送出 ISOC-OUT，而 UDC 產生一個 EP4 中斷。
3. 若 UDCCS4[RNE]被清除，而且 UDCCS4[RSP]被設定，則資料封包長度為 0。
4. 若 UDCCS4[RNE]被設定，則軟體使用 UDCWC4 計數暫存器自 EP4 資料 FIFO (UDDR4) 讀取適當的資料量。
5. 軟體清除 UDCCS4[RPC]位元。
6. 自中斷回復。
7. 重複步驟 2 至 6 直到從主機讀取所有的資料。

11.5.8.3 軟體啟動 SOF 中斷

若軟體啟動 SOF 中斷以訊框計數為基礎來處理交易：

1. 軟體設定 UICR0[IM4]為 1 來關閉 UDCCS4 中斷，並設定 UFNHR[SIM]為 0 來啟動 UFNHR 暫存器內之 SOF 中斷。
2. 當主機 PC 送出 SOF，UDC 會設定 UFNHR[SIR]位元，並引發一個 SOF 中斷。
3. 若 UDCCS4[RNE] 被清除，而且 UDCCS4[RSP] 被清除，則無資料封包被接收。
4. 若 UDCCS4[RNE] 被清除，而且 UDCCS4[RSP]被設定，則資料封包長度為 0。
5. 若 UDCCS4[RNE]被設定，則資料封包為短封包，而軟體使用 UDCWC4 計數暫存器自 EP4 資料 FIFO (UDDR4) 讀取適當的資料量。
6. 軟體清除 UDCCS4[RPC]與 UFNHR[SIR]位元。
7. 自中斷回復。
8. 重複步驟至 7 直到所有的資料都被傳送至主機 PC。

11.5.9 情況 9：EP5 資料傳送 (INTERRUPT-IN)

情況 9 之程序也可用來操作端點 10 與 15。

在情況 9 中，只有當傳送的封包大小小於 8 位元組時，才會設定傳送短封包。若封包大小為 8 位元組，當載入第 8 個位元組系統會預備操作。載入第 8 個位元組與設定 UDCCS5[TSP]位元會產生一個 8 位元組封包與一個長度為 0 的封包。

當軟體接收到 SETUP VENDOR 命令來建立一個 EP5 INTERRUPT-IN 交易時，只能允許 Megacell 處理交易：

1. 在 SETUP VENDOR 命令期間，軟體會填滿 EP5 資料 FIFO (UDDR5) 並清除 UDCCS5[TPC]位元。
2. 主機 PC 傳送 INTERRUPT-IN，UDC 產生 EP5 中斷。
3. 軟體填滿 EP5 資料 FIFO (UDDR5) 並清除 UDCCS5[TPC]位元。若資料封包為短封包，軟體也會設定 UDCCS5[TSP]位元。
4. 自中斷回復。
5. 重複步驟到 5 直到傳送所有的資料至主機 PC。

11.5.10 情況 10：RESET 中斷

1. 系統重置之後，軟體以所需的數值載入暫存器。
2. 軟體設定 UDCCR[UDE]位元來啟動 UDC 並立即讀取 UDCCR[UDA]位元來判斷 USB 重置訊號目前是否在 USB 匯流排上。
 - a. 若 UDCCR[UDA]為 0，則目前有 USB 重置在匯流排上，而軟體藉由寫入 1 至 UDCCR[RSTIR]位元來清除中斷。軟體藉由清除 UDCCR[REM]位元來啟動以後的重置中斷。
 - b. 若 UDCCR[UDA]為 1，則目前沒有 USB 重置在匯流排上，而軟體藉由清除 UDCCR[REM]位元來啟動以後的重置中斷。
3. 自中斷回復。
4. 主機會觸發 USB 重置或是取消 USB 重置。
5. UDC 產生一個重置中斷。
6. 軟體會判斷是否設定 UDCCR[RSTIR]位元，並藉由寫入 1 至 UDCCR[RSTIR]位元來清除中斷。而後軟體檢查 UDCCR[UDA]位元來決定發生的重置類型：
 - a. 若 UDCCR[UDA]為 0，則會發生重置驅動。軟體自中斷回復並等待重置消除 (Reset Negation) 中斷。
 - b. 若 UDCCR[UDA]為 1，則會發生重置取消。軟體設定任何所需的初始化。
7. 自中斷回復。

11.5.11 情況 11：SUSPEND 中斷

1. 軟體啟動時，會清除 UDCCR[SRM]位元來允許 USB 暫停中斷。
2. 主機 PC 藉由停止 UDC+與 UDC-信號之行動來觸發 USB 暫停。
3. UDC 產生暫停（Suspend）中斷。
4. 軟體決定設定 UDCCR[SUSIR]位元。這表示發生 USB 暫停，而軟體會採取任何需要的動作來關閉其他週邊、清除內部緩衝區、執行電源管理，以及執行相似功能。軟體不可關閉 UDC，且當接上 USB 纜線時不可允許處理器進入 Sleep 模式。

11.5.12 情況 12：RESUME 中斷

1. 軟體啟動時，會清除 UDCCR[SRM]位元來允許 USB 恢復操作。
2. 主機 PC 藉由恢復暫停狀態中的 UDC+與 UDC-信號之行動來觸發 USB 恢復操作。
3. UDC 產生恢復（Resume）中斷。
4. 軟體判斷是否設定 UDCCR[RESIR]位元。這表示發生 USB 恢復操作，而 OS 會採取任何需要的動作來啟動其他週邊、初始化內部緩衝區、執行電源管理，與執行相似功能。

11.6 UDC 暫存器定義

所有的組態、請求/服務和狀態報告是由 USB 主機控制器控制，經由 USB 和 UDC 進行溝通。UDC 的暫存器可以控制 UDC 和軟體之間的介面。控制暫存器會啟動 UDC，遮蔽 UDC 的中斷來源。狀態暫存器指示中斷來源的狀態。16 個端點（控制、OUT、IN）都有控制或狀態暫存器。端點 0（控制）有 16x8 的資料 FIFO 位址，可以用來傳送或接收資料。端點 0 還有一個寫入計數暫存器，是用來決定 USB 主機控制器傳送給端點 0 的位元組數目。

11.6.1 UDC 控制暫存器

UDC 控制暫存器（UDCCR）包含七個控制位元：一個用來啟動 UDC、另一個用來降低活動，其他五個用來顯示狀態和相關控制功能。

11.6.1.1 UDC 啟動

UDC 啟動 (UDE) 位元用來啟動 UDC。當 UDE 設定為 1，為了 USB 串流傳送或接收會啟動 UDC；當設定為 0，關閉 UDC，而且 UDC+和 UDC-被三態化。也就是說，UDC 忽略 USB 匯流排上的所有動作。

若將 UDE 設定為 0，全部的 UDC 設計被重置。當 UDC 主動傳輸或接收檔案時，若發生重置，它會立即停止，而且在傳送或接收檔案串流 shifter 的剩餘位元會被重置。所有傳輸和接收 FIFO 裡的項目也會被重置。

11.6.1.2 UDC 有效

可讀取唯讀的 UDC 有效 (UDA) 位元來決定 UDC 目前為主動或在 USB 重置內。此位元只有當 UDC 啟動時才有效。為 0 時表示 UDC 目前正在從主機接收 USB 重置。為 1 時表示 UDC 目前為執行交易所需。

11.6.1.3 UDC 恢復 (RSM)

當 UDC 在暫停狀態，可寫入此位元以強制 UDC 進入非閒置 (non-idle) 狀態 (K state) 為時 3ms 來執行遠端喚醒操作。若主機 PC 在 3ms 內還未啟動喚醒程序，則 UDC 會回復到暫停模式。此位元為 UDC 驅動位元，並會自動清除。

11.6.1.4 回復中斷請求 (RESIR)

若清除 UDC 控制暫存器內的 SRM 位元，則會設定回復中斷請求位元，而 UDC 目前在暫停模式，且 USB 以回復信號驅動。

11.6.1.5 暫停中斷請求 (SUSIR)

當 USB 維持閒置超過 6ms 時，會設定暫停中斷請求暫存器。SUSIR 位元會保持狀態，因此軟體可決定 USB 為閒置狀態。若 SRM 為 0，設定 SUSIR 不會產生中斷，但會持續更新狀態。

11.6.1.6 暫停/回復中斷遮蔽 (SRM)

暫停/回復中斷遮蔽 (SRM) 遮蔽或啟動送至中斷控制器之暫停中斷請求。當 SRM 為 1，則遮蔽中斷，而 SUSIR 之設定不會產生中斷。當 SRM 為 0，則 USB 閒置超過 6ms 時 SUSIR 設定會產生中斷。程式化 SRM 不會影響 SUSIR

之狀態。

11.6.1.7 重置中斷請求 (RSTIR)

當主機發出重置訊號時，會設定重置中斷請求暫存器。當主機發出重置時，整個 UDC 會重置。RSTIR 位元保持其狀態，因此軟體可決定重置設計。若 REM 為 0，設定 RSTIR 不會產生中斷，但會持續更新狀態。

11.6.1.8 重置中斷遮蔽 (REM)

重置中斷遮蔽 (REM) 會遮蔽或啟動送給中斷控制器之重置中斷請求。當 REM 為 1 時，中斷會被遮蔽，而 RSTIR 設定不會產生中斷。當 REM 為 0，USB 主機控制器發出 UDC 重置時 RSTIR 設定會產生中斷。程式化 REM 不會影響 RSTIR 狀態。

附錄「UDC 控制暫存器」顯示 UDC 控制暫存器內 (UDCCR) 之位元位置。將 UDE 位元清除為 0 會關閉 Megacell 重置後的 UDC。對保留位元的寫入會被忽略，而讀取會傳回 0。

表 11-12 UDC 控制暫存器

0h 4060 0000				UDCCR				讀取 / 寫入與唯讀	
位元	31:8	7	6	5	4	3	2	1	0
	保留	REM	RSTIR	SRM	SUSIR	RESIR	RSM	UDA	UDE
重置	X	1	0	1	0	0	0	0	0

位元	名稱	說明
31:8	保留	保留給以後使用。
7	REM	重置中斷遮蔽（讀取/寫入） 0 – 啟動重置中斷。 1 – 關閉重置中斷。
6	RSTIR	重置中斷請求（讀取/寫入 1 來清除） 1 – 由主機來重置 UDC。
5	SRM	暫停/回復中斷遮蔽（讀取/寫入） 0 – 啟動暫停/回復中斷。 1 – 關閉暫停/回復中斷。
4	SUSIR	暫停中斷請求（讀取/寫入 1 來清除） 1 – UDC 自主機接收暫停信號。
3	RESIR	回復中斷請求（讀取/寫入 1 來清除） 1 – UDC 自主機接收回復信號。

2	RSM	裝置回復（讀取/寫入 1 來設定） 0 – UDC 維持暫停狀態。 1 – 強制 UDC 離開暫停。
1	UDA	UDC 主動（唯讀） 0 – UDC 目前正接收 USB 重置。 1 – UDC 目前無接收 USB 重置。
0	UDE	UDC 啟動（讀取/寫入） 0 – 關閉 UDC。 1 – 啟動 UDC。

11.6.2 UDC 端點 0 控制/狀態暫存器（UDCCS0）

UDC 端點控制/狀態暫存器包含 7 個位元，用來操作端點 0—控制端點。

11.6.2.1 OUT 封包預備（OPR）

當接收到有效的 OUT 封包至端點 0 時，UDC 會設定 OUT 封包預備位元。當設定此位元，若端點 0 發出中斷，則 USIR[IR0]位元會在 UDC 狀態/中斷暫存器內被設定。藉由寫入 1 來清除此位元。不允許 UDC 進入交易的資料階段直到清除此位元。

11.6.2.2 IN 封包預備（IPR）

若寫入至端點 0 FIFO 來傳送的資料少於 max_packet 位元組(16)，則 Megacell 會設定 IN 封包預備位元。若有 max_packet 的資料被傳送，則 Megacell 不可設定此位元。當封包成功傳送、設定 UDCCS0[FTF]位元，或接收控制 OUT 時，UDC 會清除此位元。當因成功的 IN 傳送或控制 OUT 的接收而清除此位元時，若端點 0 中斷經由 UICR0[IM0]啟動，則會設定 UDC 中斷暫存器內之 USIR0[IR0]位元。Megacell 無法清除 UDCCS0[IPR]，而讀取傳回的值總是為 0。

當軟體為 Vendor/Class 命令與控制資料命令（例如：GET_DESCRIPTOR、GET_CONFIGURATION、GET_INTERFACE、GET_STATUS、以及 SET_DESCRIPTOR）啟動狀態階段時，軟體也必須設定 IPR。傳送 FIFO 內之資料必須傳送，而且必須在 IPR 為狀態階段設定之前做中斷處理。

其他所有不具有資料階段（例如：SET_ADDRESS、SET_CONFIGURATION、SET_INTERFACE、SET_FEATURE、CLEAR_FEATURE）的 USB 標準命令之狀態階段會由 UDC 處理，而軟體不可設定 IPR。

11.6.2.3 Flush Tx FIFO (FTF)

Flush Tx FIFO 位元驅動端點 0 之傳送 FIFO 重置。當軟體寫入 1 或當 UDC 自端點 0 上的主機接收到 OUT 封包時，會設定此位元。讀取此位元總是傳回 0 值。

11.6.2.4 裝置遠端喚醒特徵 (DRWF)

主機藉由傳送設定特徵 (Set Feature) 命令或清除功能 (Clear Function) 命令來指示裝置遠端喚醒特徵的狀態。UDC 將主機傳送的命令解碼，若啟動此特徵則設定此位元為 1，而若關閉此特徵則設定此位元為 0。此位元為唯讀。

11.6.2.5 傳送停止 (SST)

當 FST 成功地強制由軟體引起的 USB 匯流排 STALL 時，則 UDC 會設定傳送停止 (sent stall) 位元。若 UDC 在 STALL 信號交換自動傳回時自主機偵測到一個違反協定行為，則不可設定此位元。在此事件中，Megacell 無介入而在主機傳送下依個 SETUP 命令之前 UDC 清除 STALL 狀態。當 USC 設定此位元，會 flush 傳送 FIFO。Megacell 寫入 1 來清除此位元。

11.6.2.6 強制停止 (FST)

強制停止位元可由 Megacell 設定來強制 UDC 發出 STALL 信號交換。UDC 發出 STALL 信號交換給目前建立控制傳送，而藉由 UDC 來清除此位元，因為端點 0 不可保持在停止狀態。

11.6.2.7 接收 FIFO 非空白 (RNE)

接收 FIFO 非空白位元指示接收 FIFO 包含未讀取的資料。為了決定 FIFO 內是否有資料，當傳送 UDCCS0[OPR]位元時必須讀取此位元。必須繼續讀取接收 FIFO 直到清除此位元或是遺失將資料。

當 UDCCS0[OPR]產生的中斷最初被服務時，若未設定 UDCCS0[RNE]，則表示接收到一個長度為 0 的 OUT 封包。

11.6.2.8 建立有效 (SA)

建立主動 (Setup Active) 位元指示目前 FIFO 內的封包為 USB 建立命令的一部份。此位元會產生中斷，並與 UDCCS0[OPR]同時變為主動。軟體必須寫入 1 來清除此位元。必須清除 UDCCS0[OPR]與 UDCCS0[SA]。

表 11-13 UDC 端點 0 控制狀態暫存器

		0h 4060 0010				UDCCS0			讀取 / 寫入	
位元		31:8	7	6	5	4	3	2	1	0
		保留	SA	RNE	FST	SST	DRWF	FTF	IPR	OPR
重置		X	0	0	0	0	0	0	0	0

位元	名稱	說明
0	OPR	OUT 封包預備（讀取/寫入 1 來清除） 1 – OUT 封包預備。
1	IPR	IN 封包預備（總是讀取 0/寫入 1 來設定） 1 – IN 封包預備。
2	FTF	Flush Tx FIFO（總是讀取 0/寫入 1 來設定） 1 – 將 Tx FIFO 之內容 flush。
3	DRWF	裝置遠端喚醒特徵（唯讀） 0 – 關閉裝置遠端喚醒特徵。 1 – 啟動裝置遠端喚醒特徵。
4	SST	傳送停止（讀取/寫入 1 來清除） 1 – UDC 傳送停止信號交易。
5	FST	強制停止（讀取/寫入 1 來設定） 1 – 強制停止信號交易。
6	RNE	接收 FIFO 非空白（唯讀） 0 – 接收 FIFO 空白。 1 – 接收 FIFO 非空白。
7	SA	建立主動（讀取/寫入 1 來清除） 1 – USB 上之建立命令為主動。
31:8	保留	保留給以後使用。

11.6.3 UDC 端點 x 控制/狀態暫存器（UDCCS_x），x = 1、6、11

UDC 端點 (x) 控制狀態暫存器包含 6 個位元，用來操作端點 (x) – Bulk IN 端點。

11.6.3.1 傳送 FIFO 服務 (TFS)

若一個或更少的資料封包留在傳送 FIFO 內，則傳送 FIFO 服務位元會變為主動。當兩個完整的資料封包留在 FIFO 內，TFS 會被清除。藉由載入資料的

64 位元組或設定 UDCCSx[TSP]來表示一個完整的資料封包。

11.6.3.2 傳送封包完成 (TPC)

當一個完整的封包傳送至主機時，藉由 UDC 來設定傳送封包完成 (transmit packet complete) 位元。當設定此位元，若啟動傳送中斷，會設定適當的 UDC 狀態/中斷暫存器內之 IRx 位元。此位元可用來使端點 (x) 控制/狀態暫存器內的其他狀態/錯誤位元生效。藉由寫入 1 來清除 UDCCSx[TPC]位元。這會清除適當的 UDC 狀態/中斷暫存器內的 IRx 位元中斷來源，但也必須清除 IRx 位元。

設定此位元不會預防 UDC 傳送下一個封包。若此位元已設定，UDC 會發出 NAK 信號交換至所有的 IN 符記，而緩衝區不會因為寫入 64 位元組或設定 UDCCSx[TSP]而驅動。

當 DMA 載入傳送緩衝區，可遮蔽 UDCCSx[TPC]產生的中斷來允許傳送資料而 Megacell 不會介入。

11.6.3.3 Flush Tx FIFO (FTF)

Flush Tx FIFO 位元驅動端點之傳送 FIFO 重置。當軟體寫入 1 至此位元，或是當主機執行 SET_CONFIGURATION 或 SET_INTERFACE 時，則會設定此 Flush Tx FIFO 位元。此位元之讀取值為 0。

11.6.3.4 傳送 Underrun (TUR)

若傳送 FIFO 遭到 underrun，則會設定傳送 underrun 位元。當 UDC 遭到 underrun 時，會傳送 NAK 信號交換至主機。UDCCSx[TUR]不會產生中斷，它只是一種狀態。藉由寫入 1 來清除 UDCCSx[TUR]。

11.6.3.5 傳送停止 (SST)

藉由 UDC 設定傳送停止位元以回應 FST 成功強制使用者引發 USB 匯流排之 STALL。當 STALL 信息交換自動傳回時，若 UDC 從主機 PC 偵測到違反協定，則不會設定此位元。在任何一個事件中，當主機傳送一個 CLEAR_FEATURE 命令時，Megacell 不會介入，而 UDC 會清除 STALL 狀態。端點操作正常繼續，且不會設定另一個 STALL 狀態，甚至當設定 UDCCSx[SST]位元時也不會。為了允許軟體繼續傳送 USB 匯流排 STALL 狀態，必須再次設定 UDCCSx[FST]位元。Megacell 寫入 1 來清除傳送停止位元。

11.6.3.6 強制停止 (FST)

Megacell 可設定強制停止 (force stall) 位元來強制 UDC 發出 STALL 信號交換至所有的 IN 符記。STALL 信號交換繼續設定直到 Megacell 設定清除特徵 (Clear Feature) 命令來清除此位元。當實際進入 STALL 狀態時，會設定 UDCCSx[SST]位元，但若設定 UDCCSx[FST]位元且 UDC 為主動時則可能會產生延遲。當設定 UDCCSx[SST]位元時，會自動清除 UDCCSx[FST]位元。為了確定在傳送清除特徵 (Clear Feature) 命令之後沒有資料傳送，且主機回復 IN 請求，軟體必須設定 UDCCSx[FTF]位元來清除傳送 FIFO。

11.6.3.7 位元 6 保留

位元 6 保留給以後使用。

11.6.3.8 傳送短封包 (TSP)

軟體使用送短封包 (transmit short packet) 位元來指示發生傳送最後的位元組至 FIFO。這表示已準備好傳送短封包或長度為 0 的封包至 UDC。若傳送 64 位元組的封包，則軟體不可設定此位元。當資料封包傳送成功，UDC 會清除此位元。

表 11-14 UDC 端點 x 控制狀態暫存器，x = 1、6、11

	0h 4060 0014				UDCCS1		讀取 / 寫入		
	0h 4060 0028				UDCCS6		讀取 / 寫入		
	0h 4060 003C				UDCCS11		讀取 / 寫入		
位元	31:8	7	6	5	4	3	2	1	0
	保留	TSP	保留	FST	SST	TUR	FTF	TPC	TFS
重置	X	0	0	0	0	0	0	0	1

位元	名稱	說明
0	TFS	傳送 FIFO 服務 (唯讀) 0 – 傳送 FIFO 沒有空間給新的資料。 1 – 傳送 FIFO 至少有空間給一個完整的資料封包。
1	TPC	傳送封包完成 (讀取/寫入 1 來清除) 0 – 錯誤/狀態位元無效。 1 – 送出傳送封包，而錯誤/狀態位元有效。
2	FTF	Flush Tx FIFO (總是讀取 0/寫入 1 來清除) 1 – 將 Tx FIFO 的內容 flush。
3	TUR	傳送 FIFO underrun (讀取/寫入 1 來清除)

		1 – 傳送 FIFO 遭到 underrun。
4	SST	傳送 STALL (讀取/寫入 1 來清除) 1 – 傳送 STALL 信號交換。
5	FST	強制 STALL (讀取/寫入) 1 – 發出 STALL 信號交換給 IN 符記。
6	—	保留 讀取總為 0
7	TSP	傳送短封包 (總是讀取 0/寫入 1 來設定) 1 – 準備好傳送短封包。
31:8	保留	保留給以後使用

11.6.4 UDC 端點 x 控制/狀態暫存器 (UDCCS_x), x = 2、7、12

UDC 端點 x 控制/狀態暫存器包含 7 個位元，用來操作端點 x—Bulk OUT 端點。

11.6.4.1 接收 FIFO 服務 (RFS)

若接收 FIFO 內具有一個完整的資料封包，且此封包已經由 UDC 做錯誤確認，

則會設定接收 FIFO (receive FIFO) 服務位元。一個完整的封包可能為 64 位元組、短封包、或長度為 0 的封包。此位元不會清除直到從兩個緩衝區讀取所有的資料。

11.6.4.2 接收封包完成 (RPC)

當接收到 OUT 封包，UDC 會設定接收封包完成 (receive packet complete) 位元。當設定此位元時，若接收中斷啟動，則會設定適當的 UDC 狀態/中斷暫存器內之 IR_x 位元。必須使用此位元來使端點 (x) 控制/狀態暫存器內之其他狀態/錯誤位元生效。狀態位元不會更新，直到設定 RPC。狀態位元保持設定直到清除 RPC。RNE 為例外，它與 RPC 一同設定，但一旦有效 FIFO 為空白時，則會清除自己。在清除 RPC 之後，下一個緩衝區將會成為主動，而狀態位元將會相應更新，包含 RPC。藉由寫入 1 來清除 UDCCS_x[RPC]位元。當此設定位元且兩個緩衝區都有未讀取的資料時，UDC 會發出 NAK 信號交換至所有的 OUT 符記。

11.6.4.3 位元 2 保留

位元 2 保留給以後使用。

11.6.4.4 DMA 啟動 (DME)

UDC 使用 DMA 啟動 (DMA enable) 來控制接收的資料中斷之時序。若設定此位元，在接收到封包結束而接收 FIFO 所剩的資料少於 32 位元組時，會觸發中斷。若未設定此位元，當接收封包結尾且所有接收的資料仍在接收 FIFO 內時，會觸發中斷。

11.6.4.5 傳送停止 (SST)

藉由 UDC 設定傳送停止位元以回應 FST 成功強制使用者引發 USB 匯流排之 STALL。當 STALL 信息交換自動傳回時，若 UDC 從主機 PC 偵測到違反協定，則不會設定此位元。在任何一個事件中，當主機傳送一個 CLEAR_FEATURE 命令時，Megacell 不會介入，而 UDC 會清除 STALL 狀態。FIFO 內任何有效的資料會保持有效，而軟體必須將其卸載。端點操作正常繼續，且不會設定另一個 STALL 狀態，甚至當設定 UDCCSx[SST]位元時也不會。為了允許軟體繼續傳送 USB 匯流排 STALL 狀態，必須再次設定 UDCCSx[FST]位元。Megacell 寫入 1 來清除傳送停止位元。

11.6.4.6 強制停止 (FST)

Megacell 可設定強制停止 (force stall) 位元來強制 UDC 發出 STALL 信號交換至所有的 IN 符記。STALL 信號交換繼續設定直到 Megacell 設定清除特徵 (Clear Feature) 命令來清除此位元。當實際進入 STALL 狀態時，會設定 UDCCSx[SST]位元，但若 UDCCSx[FST]位元被設定且 UDC 為有效時則可能會產生延遲。當設定 UDCCSx[SST]位元時，會自動清除 UDCCSx[FST]位元。為了確定在傳送清除特徵 (Clear Feature) 命令之後沒有資料傳送，且主機回復 IN 請求，軟體必須設定 UDCCSx[FTF]位元來清除傳送 FIFO。

11.6.4.7 接收 FIFO 非空白 (RNE)

接收 FIFO 非空白 (receive FIFO not empty) 位元指示未讀取的資料保持在接收 FIFO 內。當 UDCCSx[RPC]位元已設定，則必須讀取此位元來決定 FIFO 內是否有任何資料 DMA 未讀取。必須繼續讀取接收 FIFO 直到清除此位元或資料遺失。

11.6.4.8 接收短封包 (RSP)

UDC 使用接收短封包 (receive short packet) 位元來指示在主動緩衝區內目前正在讀取的已接收 OUT 封包為短封包或長度為 0。在從主動緩衝區讀取最後一個位元組並反射新的主動緩衝區狀態之後，UDC 會更新此位元。若 UDCCSx[RSP] 為 1 而 UDCCSx[RNE] 為 0，表示為一個長度為 0 的封包。若呈現出長度為 0 的封包，則 Megacell 不可讀取資料暫存器。接收下一個 OUT 封包時，會清除 UDCCSx[RSP]。

表 11-15 UDC 端點 x 控制狀態暫存器，x = 2、7、12

	0h 4060 0018	UDCCS2	讀取 / 寫入						
	0h 4060 002C	UDCCS7	讀取 / 寫入						
	0h 4060 0040	UDCCS12	讀取 / 寫入						
位元	31:8	7	6	5	4	3	2	1	0
	保留	RSP	RNE	FST	SST	DME	保留	RPC	RFS
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
0	RFS	接收 FIFO 服務（唯讀） 0－ 接收 FIFO 少於一個資料封包。 1－ 接收 FIFO 具有一個或更多的資料封包。
1	RPC	接收封包完成（讀取/寫入 1 來清除） 0－ 錯誤/狀態位元無效。 1－ 已收到接收封包而錯誤/狀態位元為有效。
2	—	保留 讀取總是為 0。
3	DME	DMA 啟動（讀取/寫入） 0－ 在接收 EOP 之後接收傳送資料中斷。 1－ 在接收 EOP 之後接收傳送資料中斷，而接收 FIFO 具有小於 32 位元組的資料。。
4	SST	傳送停止（讀取/寫入 1 來清除） 1－ 傳送 STALL 信號交換。
5	FST	強制停止（讀取/寫入） 1－ 發出 STALL 信號交換給 IN 符記。
6	RNE	接收 FIFO 非空白（唯讀） 0－ 接收 FIFO 空白。 1－ 接收 FIFO 不為空白。
7	RSP	接收短封包（唯讀）

		1 – 接收短封包並準備好讀取。
31:8	保留	保留為以後使用

11.6.5 UDC 端點 x 控制/狀態暫存器 (UDCCS_x), x=3、8 或 13

UDC 端點(x)控制狀態暫存器包含 4 個位元，用來操作端點(x)，端點(x)為一個等時性 IN 端點。

11.6.5.1 傳送 FIFO 服務 (TFS)

若傳送 FIFO 內剩下一個或更少的資料封包時，傳送 FIFO 服務位元會被設定。當 FIFO 內有 2 個完整的資料封包時，UDCCS_x[TFS]會被清除。一個完整的資料封包由載入 256 位元組或設定 UDCCS_x[TSP]來表示。

11.6.5.2 傳送封包完成 (TPC)

當傳送一個完整的封包給主機時，UDC 會設定傳輸封包完整位元。當此位元被設定時，若啟動傳送中斷，合適的 UDC 狀態/中斷暫存器的 IR_x 位元會被設定。

可以用此位元使在端點(x)控制/狀態暫存器裡的其他狀態/錯誤位元生效。利用寫入 1 到 UDCCS_x[TPC]位元來清除它。此可清除在合適的 UDC 狀態/中斷暫存器的 IR_x 位元之中斷來源，但是 IR_x 位元也必須被清除。

設定此位元不能避免 UDC 傳送到下一個緩衝區。若此位元被設定，且緩衝區已藉由寫入 64 位元組被觸發或是設定 UDCCS_x[TPC]，UDC 發佈 NAK 信號交換給所有的 IN 符記(token)。

當 DMA 被用來載入傳送緩衝區，由 UDCCS_x[TPC]產生的中斷會被遮照，而允許資料不需 Megacell 介入就可以被傳送。

11.6.5.3 Flush Tx FIFO (FTF)

Flush Tx FIFO 位元觸發端點的傳送 FIFO 重置。當軟體寫入 1 到 Flush tx FIFO 位元或當主機執行 SET_CONFIGURATION 或 SET_INTERFACE 時，此位元會被設定。位元的讀取值為 0。

11.6.5.4 傳送 Underrun (TUR)

若傳送 FIFO 的經歷為 underrun 時，傳送 underrun 位元會被設定。當 UDC 經歷為 underrun 時，UDCCSx[TUR]產生一個中斷。UDCCSx[TUR]由寫入 1 來做清除。

11.6.5.5 位元 4 保留

位元 4 保留以後使用。

11.6.5.6 位元 5 保留

位元 5 保留以後使用。

11.6.5.7 位元 6 保留

位元 6 保留以後使用。

軟體使用傳送短封包來指示資料傳輸的最後位元組已經送至 FIFO。此告訴 UDC，短封包或尺寸為 0 的封包已經準備好要被傳送。若 256 位元組的封包被傳送，軟體不必設定此位元。當資料封包成功地傳輸，此位元由 UDC 清除。

表 11-16 UDC 端點 x 控制狀態暫存器，x = 3、8、13

	0h 4060 001C	UDCCS3	讀取 / 寫入						
	0h 4060 0030	UDCCS8	讀取 / 寫入						
	0h 4060 0044	UDCCS13	讀取 / 寫入						
位元	31:8	7	6	5	4	3	2	1	0
	保留	TSP	保留	保留	保留	TUR	FTF	TPC	TFS
重置	X	0	0	0	0	0	0	0	1

位元	名稱	說明
0	TFS	傳送 FIFO 服務 (唯讀) 0 – 傳送 FIFO 沒有空間給新的資料。 1 – 傳送 FIFO 至少有空間給一個完整的資料封包。
1	TPC	傳送封包完成 (讀取/寫入 1 來清除) 0 – 錯誤/狀態位元無效。 1 – 送出傳送封包，而錯誤/狀態位元有效。
2	FTF	Flush Tx FIFO (總是讀取 0/寫入 1 來清除)

		1 – 將 Tx FIFO 的內容 flush。
3	TUR	傳送 FIFO underrun (讀取/寫入 1 來清除) 1 – 傳送 FIFO 遭到 underrun。
4	—	保留 讀取經常為 0。
5	—	保留 讀取經常為 0。
6	—	保留 讀取經常為 0。
7	TSP	傳送短封包 (總是讀取 0/寫入 1 來設定) 1 – 準備好傳送短封包。
31:8	保留	保留為以後使用。

11.6.6 UDC 端點 x 控制/狀態暫存器 (UDCCS_x), x = 4、9、14

UDC 端點 (x) 控制/狀態暫存器包含 6 個位元，用來操作端點 (x) 一等時 OUT 端點。

11.6.6.1 接收 FIFO 服務 (RFS)

若接收 FIFO 內有一個完整的資料封包，而此封包已經過 UDC 做錯誤確認，則會設定接收 FIFO 服務位元。完整的封包可能為 256 位元組、短封包，或長度為 0 的封包。不會清除 UDCCS_x[RFS]直到已從兩個緩衝區內讀取所有的資料。

11.6.6.2 接收封包完成 (RPC)

當接收到 OUT 封包，UDC 會設定接收封包完成 (receive packet complete) 位元。當設定此位元時，若接收中斷啟動，則會設定適當的 UDC 狀態/中斷暫存器內之 IR_x 位元。此位元可用來使端點 (x) 控制/狀態暫存器內之其他狀態/錯誤位元生效。藉由寫入 1 來清除 UDCCS_x[RPC]位元。

11.6.6.3 接收溢位 (ROF)

接收溢位 (overflow) 位元在適當的 UDC 狀態/中斷暫存器內 IR_x 上產生一個中斷來改變軟體以丟棄等時資料封包，因為 FIFO 緩衝區沒有空間給這些封

包。寫入 1 來清除此位元。

11.6.6.4 DMA 啟動 (DME)

UDC 使用 DMA 啟動 (DMA enable) 來控制接收的資料中斷之時序。若設定此位元，在接收到封包尾端而接收 FIFO 所剩的資料少於 32 位元組時，會觸發中斷。若未設定此位元，當接收封包結尾且所有接收的資料仍在接收 FIFO 內時，會觸發中斷。

11.6.6.5 位元 4 保留

位元 4 保留給以後使用。

11.6.6.6 位元 5 保留

位元 5 保留給以後使用。

11.6.6.7 接收 FIFO 非空白 (RNE)

接收 FIFO 非空白 (receive FIFO not empty) 位元指示未讀取的資料保持在接收 FIFO 內。當 UDCCSx[RPC]位元已設定，則必須讀取此位元來決定 FIFO 內是否有任何資料 DMA 未讀取。必須繼續讀取接收 FIFO 直到清除此位元或資料遺失。

11.6.6.8 接收短封包 (RSP)

UDC 使用接收短封包 (receive short packet) 位元來指示在主動緩衝區內目前正在讀取的已接收 OUT 封包為短封包或長度為 0。在從主動緩衝區讀取最後一個位元組並反射新的主動緩衝區狀態之後，UDC 會更新此位元。若 UDCCSx[RSP]為 1 而 UDCCSx[RNE]為 0，表示為一個長度為 0 的封包。若呈現出長度為 0 的封包，則 Megacell 不可讀取資料暫存器。接收下一個 OUT 封包時，會清除 UDCCSx[RSP]。

表 11-17 UDC 端點 x 控制狀態暫存器，x = 4、9、14

	0h 4060 0020	UDCCS4	讀取 / 寫入						
	0h 4060 0034	UDCCS9	讀取 / 寫入						
	0h 4060 0048	UDCCS14	讀取 / 寫入						
位元	31:8	7	6	5	4	3	2	1	0
	保留	RSP	RNE	保留	保留	DME	ROF	RPC	RFS
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
0	RFS	接收 FIFO 服務（唯讀） 0 – 接收 FIFO 少於一個資料封包。 1 – 接收 FIFO 具有一個或更多的資料封包。
1	RPC	接收封包完成（讀取/寫入 1 來清除） 0 – 錯誤/狀態位元無效。 1 – 已收到接收封包而錯誤/狀態位元為有效。
2	ROF	接收溢位（讀取/寫入 1 來清除） 1 – 自主機丟棄等時資料封包，因為接收端已滿。
3	DME	DMA 啟動（讀取/寫入） 0 – 在接收 EOP 之後接收傳送資料中斷。 1 – 在接收 EOP 之後接收傳送資料中斷，而接收 FIFO 具有小於 32 位元組的資料。。
4	—	保留 讀取總是為 0。
5	—	保留 讀取總是為 0。
6	RNE	接收 FIFO 非空白（唯讀） 0 – 接收 FIFO 空白。 1 – 接收 FIFO 不為空白。
7	RSP	接收短封包（唯讀） 1 – 接收短封包並準備好讀取。
31:8	保留	保留為以後使用

11.6.7 UDC 端點 x 控制/狀態暫存器（UDCCSx），x = 5、10、15

UDC 端點（x）控制狀態暫存器包含 6 個位元，用來操作端點（x）— 中斷 IN 端點。

11.6.7.1 傳送 FIFO 服務 (TFS)

若 FIFO 不包含任何資料位元組，而 UDCCSx[TSP]為設定時，則設定傳送 FIFO 服務位元。

11.6.7.2 傳送封包完成 (TPC)

當一個完整的封包傳送至主機時，藉由 UDC 來設定傳送封包完成 (transmit packet complete) 位元。當設定此位元，若啟動傳送中斷，會設定適當的 UDC 狀態/中斷暫存器內之 IRx 位元。此位元可用來使端點 (x) 控制/狀態暫存器內的其他狀態/錯誤位元生效。藉由寫入 1 來清除 UDCCSx[TPC]位元。這會清除適當的 UDC 狀態/中斷暫存器內的 IRx 位元中斷來源，但也必須清除 IRx 位元。

若此位元已設定，UDC 會發出 NAK 信號交換至所有的 IN 符記，而緩衝區不會因為寫入 8 位元組或設定 UDCCSx[TSP]而驅動。

11.6.7.3 Flush Tx FIFO (FTF)

Flush Tx FIFO 位元驅動端點的傳送 FIFO 之重置。當軟體寫入 1 至此位元，或是當主機執行 SET_CONFIGURATION 或 SET_INTERFACE 時，則會設定此 Flush Tx FIFO 位元。此位元之讀取值為 0。

11.6.7.4 傳送 Underrun (TUR)

若傳送 FIFO 遭到 underrun，則會設定傳送 underrun 位元。當 UDC 遭到 underrun 時，會傳送 NAK 信號交換至主機。UDCCSx[TUR]不會產生中斷並且只為了狀態。藉由寫入 1 來清除 UDCCSx[TUR]。

11.6.7.5 傳送 STALL (SST)

藉由 UDC 設定傳送停止位元以回應 FST 成功強制使用者引發 USB 匯流排之 STALL。當 STALL 信息交換自動傳回時，若 UDC 從主機 PC 偵測到違反協定，則不會設定此位元。在任何一個事件中，當主機傳送一個 CLEAR_FEATURE 命令時，Megacell 不會介入，而 UDC 會清除 STALL 狀態。端點操作正常繼續，且不會設定另一個 STALL 狀態，甚至當設定 UDCCSx[SST]位元時也不會。為了允許軟體繼續傳送 USB 匯流排 STALL 狀態，必須再次設定 UDCCSx[FST]位元。Megacell 寫入 1 來清除傳送停止位元。

11.6.7.6 強制 STALL (FST)

Megacell 可設定強制停止 (force stall) 位元來強制 UDC 發出 STALL 信號交換至所有的 IN 符記。STALL 信號交換繼續設定直到 Megacell 設定清除特徵 (Clear Feature) 命令來清除此位元。當實際進入 STALL 狀態時，會設定 UDCCSx[SST]位元，但若設定 UDCCSx[FST]位元且 UDC 為主動時則可能會產生延遲。當設定 UDCCSx[SST]位元時，會自動清除 UDCCSx[FST]位元。為了確定在傳送清除特徵 (Clear Feature) 命令之後沒有資料傳送，且主機回復 IN 請求，軟體必須設定 UDCCSx[FTF]位元來清除傳送 FIFO。

11.6.7.7 位元 6 保留

位元 6 保留給以後使用。

11.6.7.8 傳送短封包 (TSP)

軟體使用送短封包 (transmit short packet) 位元來指示發生傳送最後的位元組至 FIFO。這表示已準備好傳送短封包或長度為 0 的封包至 UDC。若傳送 8 位元組的封包，則軟體不可設定此位元。當資料封包傳送成功，UDC 會清除此位元。

表 11-18 UDC 端點 x 控制狀態暫存器，x = 5、10、15

	0h 4060 0024	UDCCS5	讀取 / 寫入						
	0h 4060 0038	UDCCS10	讀取 / 寫入						
	0h 4060 004C	UDCCS15	讀取 / 寫入						
位元	31:8	7	6	5	4	3	2	1	0
	保留	TSP	保留	FST	SST	TUR	FTF	TPC	TFS
重置	X	0	0	0	0	0	0	0	1

位元	名稱	說明
0	TFS	傳送 FIFO 服務（唯讀） 0 – 傳送 FIFO 沒有空間給新的資料。 1 – 傳送 FIFO 至少有空間給一個完整的資料封包。
1	TPC	傳送封包完成（讀取/寫入 1 來清除） 0 – 錯誤/狀態位元無效。 1 – 送出傳送封包，而錯誤/狀態位元有效。
2	FTF	Flush Tx FIFO（總是讀取 0/寫入 1 來清除） 1 – 將 Tx FIFO 的內容 flush。
3	TUR	傳送 FIFO underrun（讀取/寫入 1 來清除）

		1 – 傳送 FIFO 遭到 underrun。
4	SST	傳送 STALL (讀取/寫入 1 來清除) 1 – 傳送 STALL 信號交換。
5	FST	強制 STALL (讀取/寫入) 1 – 發出 STALL 信號交換給 IN 符記。
6	—	保留 讀取總為 0
7	TSP	傳送短封包 (總是讀取 0/寫入 1 來設定) 1 – 準備好傳送短封包。
31:8	保留	保留給以後使用

11.6.8 UDC 中斷控制暫存器 0 (UICR0)

UICR0 包含 8 個控制位元來啟動/關閉資料端點 0~7 的中斷服務請求。所有 UICR0 位元均重置為 1，因此在初始系統重置時不會產生中斷。

11.6.8.1 中斷遮蔽端點 x (IMx)，x = 0~7

UICR0[IMx]位元用來遮蔽或啟動對應的端點中斷請求—USIR0[IRx]。當設定遮蔽位元，中斷會被遮蔽，而不允許設定 USIR0 暫存器內對應的位元。當遮蔽位元被清除且端點內發生可中斷條件時，會設定適當的中斷位元。程式化遮蔽位元為 1 不會影響目前中斷位元目前的狀態，只會影響之後中斷位元從 0 轉變為 1。

表 11-19 UDC 中斷控制暫存器 0

0h 4060 0050		UICR0								讀取 / 寫入
位元	31:8	7	6	5	4	3	2	1	0	
	保留	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0	
重置	X	1	1	1	1	1	1	1	1	
位元	名稱	說明								
0	IM0	端點 0 之中斷遮蔽 0 – 啟動端點 0 中斷 1 – 關閉端點 0 中斷。								
1	IM1	端點 1 之中斷遮蔽 0 – 啟動傳送中斷								

		1 – 關閉傳送中斷
2	IM2	端點 2 之中斷遮蔽 0 – 啟動接收中斷 1 – 關閉接收中斷
3	IM3	端點 3 之中斷遮蔽 0 – 啟動傳送中斷 1 – 關閉傳送中斷
4	IM4	端點 4 之中斷遮蔽 0 – 啟動接收中斷 1 – 關閉接收中斷
5	IM5	端點 5 之中斷遮蔽 0 – 啟動傳送中斷 1 – 關閉傳送中斷
6	IM6	端點 6 之中斷遮蔽 0 – 啟動傳送中斷 1 – 關閉傳送中斷
7	IM7	端點 7 之中斷遮蔽 0 – 啟動接收中斷 1 – 關閉接收中斷
31:8	保留	保留給以後使用

11.6.9 UDC 中斷控制暫存器 1 (UICR1)

UICR1 包含 8 個控制位元來啟動/關閉資料端點 8 ~ 15 的中斷服務請求。UICR1 位元重置為 1，因此在初始系統重置時不會產生中斷。

11.6.9.1 中斷遮蔽端點 x (IMx)，x = 8 ~ 15

UICR1[IMx]位元用來遮蔽或啟動對應的端點中斷請求—USIR1[IRx]。當設定遮蔽位元，中斷會被遮蔽，而不允許設定 USIR1 暫存器內對應的位元。當遮蔽位元被清除且端點內發生可中斷條件時，會設定適當的中斷位元。程式化遮蔽位元為 1 不會影響目前中斷位元目前的狀態，只會影響之後中斷位元從 0 轉變為 1。

表 11-20 UDC 中斷控制暫存器 1

0h 4060 0054				UICR1				讀取 / 寫入	
位元	31:8	7	6	5	4	3	2	1	0
	保留	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8
重置	X	1	1	1	1	1	1	1	1
位元	名稱		說明						
0	IM8		端點 8 之中斷遮蔽 0 – 啟動傳送中斷 1 – 關閉傳送中斷						
1	IM9		端點 9 之中斷遮蔽 0 – 啟動接收中斷 1 – 關閉接收中斷						
2	IM10		端點 10 之中斷遮蔽 0 – 啟動接收中斷 1 – 關閉接收中斷						
3	IM11		端點 11 之中斷遮蔽 0 – 啟動傳送中斷 1 – 關閉傳送中斷						
4	IM12		端點 12 之中斷遮蔽 0 – 啟動接收中斷 1 – 關閉接收中斷						
5	IM13		端點 13 之中斷遮蔽 0 – 啟動傳送中斷 1 – 關閉傳送中斷						
6	IM14		端點 14 之中斷遮蔽 0 – 啟動接收中斷 1 – 關閉接收中斷						
7	IM15		端點 15 之中斷遮蔽 0 – 啟動傳送中斷 1 – 關閉傳送中斷						
31:8	保留		保留給以後使用						

11.6.10 UDC 狀態/中斷暫存器 0 (USIR0)

UDC 狀態/中斷暫存器 (USIR0 與 USIR1) 包含一些位元，用來產生 UDC 的中斷請求。UDC 狀態/中斷暫存器內每一個位元都 OR 在一起來產生一個中斷

請求。當 UDC 之 ISR 執行時，必須讀取 UDC 狀態/中斷暫存器來決定為何中斷會發生。USIRx 為層級感應的。離開 ISR 之前，要確定在最後一個步驟清除 USIRx

USIR0 與 USIR1 內的位元由 UDC 中斷控制暫存器 (UICR0/1) 內的遮蔽位元控制。當設定遮蔽位元時，會防止設定 USIRx 內的狀態位元。若清除特殊狀態位元的遮蔽位元，並且發生可中斷的條件，則設定狀態位元。Megacell 必須寫入 1 至欲清除的位置來清除狀態位元。UDC 之中斷請求保持主動與 USIRx 之數值不為 0 一樣長。

11.6.10.1 端點 0 中斷請求 (IR0)

若 UDC 控制暫存器內的 IM0 位元被清除，以及 UDC 端點 0 控制/狀態暫存器內的 OUT 封包預備位元被設定，而且 IN 封包預備位元被清除，或是傳送 STALL 位元被設定，則設定端點 0 中斷請求。藉由寫入 1 來清除此位元。

11.6.10.2 端點 1 中斷請求 (IR1)

若 UDC 中斷控制暫存器內的 IM1 位元被清除，以及 UDC 端點 1 控制/狀態暫存器內的 IN 封包完成 (TPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR1 位元。

11.6.10.3 端點 2 中斷請求 (IR2)

若 UDC 中斷控制暫存器內的 IM2 位元被清除，以及 UDC 端點 2 控制/狀態暫存器內的 OUT 封包完成 (RPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR2 位元。

11.6.10.4 端點 3 中斷請求 (IR3)

若 UDC 中斷控制暫存器內的 IM3 位元被清除，以及 UDC 端點 3 控制/狀態暫存器內的 IN 封包完成 (TPC) 或傳送 Underrun (TUR) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR3 位元。

11.6.10.5 端點 4 中斷請求 (IR4)

若 UDC 中斷控制暫存器內的 IM4 位元被清除，以及 UDC 端點 4 控制/狀態暫存器內的 OUT 封包完成 (RPC) 或接收溢位 (ROF) 被設定，或是 UFNHR

內的等時錯誤端點 4 (IPE4) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR4 位元。

11.6.10.6 端點 5 中斷請求 (IR5)

若 UDC 中斷控制暫存器內的 IM5 位元被清除，以及 UDC 端點 5 控制/狀態暫存器內的 IN 封包完成 (TPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR5 位元。

11.6.10.7 端點 6 中斷請求 (IR6)

若 UDC 中斷控制暫存器內的 IM6 位元被清除，以及 UDC 端點 6 控制/狀態暫存器內的 IN 封包完成 (TPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR6 位元。

11.6.10.8 端點 7 中斷請求 (IR7)

若 UDC 中斷控制暫存器內的 IM7 位元被清除，以及 UDC 端點 7 控制/狀態暫存器內的 OUT 封包完成 (RPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR7 位元。

表 11-21 UDC 狀態/中斷暫存器 0

0h 4060 0058								
USIR0								
讀取 / 寫入與唯讀								
位元	31:8	7	6	5	4	3	2	1 0
	保留	IR7	IR6	IR5	IR4	IR3	IR2	IR1 IR0
重置	X	0	0	0	0	0	0	0 0

位元	名稱	說明
0	IR0	端點 0 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 0 需要服務。
1	IR1	端點 1 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 1 需要服務。
2	IR2	端點 2 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 2 需要服務。
3	IR3	端點 3 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 3 需要服務。
4	IR4	端點 4 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 4 需要服務。
5	IR5	端點 5 之中斷請求 (讀取/寫入 1 來清除)

		1 – 端點 5 需要服務。
6	IR6	端點 6 之中斷請求（讀取/寫入 1 來清除） 1 – 端點 6 需要服務。
7	IR7	端點 7 之中斷請求（讀取/寫入 1 來清除） 1 – 端點 7 需要服務。
31:8	保留	保留給以後使用

11.6.11 UDC 狀態/中斷暫存器 1 (USIR1)

11.6.11.1 端點 8 中斷請求 (IR8)

若 UDC 中斷控制暫存器內的 IM8 位元被清除，以及 UDC 端點 8 控制/狀態暫存器內的 IN 封包完成 (TPC) 或傳送 Underrun (TUR) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR8 位元。

11.6.11.2 端點 9 中斷請求 (IR9)

若 UDC 中斷控制暫存器內的 IM9 位元被清除，以及 UDC 端點 9 控制/狀態暫存器內的 OUT 封包完成 (RPC) 或接收溢位 (ROF) 被設定，或是 UFNHR 內的等時錯誤端點 9 (IPE9) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR9 位元。

11.6.11.3 端點 10 中斷請求 (IR10)

若 UDC 中斷控制暫存器內的 IM10 位元被清除，以及 UDC 端點 10 控制/狀態暫存器內的 IN 封包完成 (TPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR10 位元。

11.6.11.4 端點 11 中斷請求 (IR11)

若 UDC 中斷控制暫存器內的 IM11 位元被清除，以及 UDC 端點 1 控制/狀態暫存器內的 IN 封包完成 (TPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR11 位元。

11.6.11.5 端點 12 中斷請求 (IR12)

若 UDC 中斷控制暫存器內的 IM12 位元被清除，以及 UDC 端點 12 控制/狀態暫存器內的 OUT 封包完成 (RPC) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR12 位元。

11.6.11.6 端點 13 中斷請求 (IR13)

若 UDC 中斷控制暫存器內的 IM13 位元被清除，以及 UDC 端點 13 控制/狀態暫存器內的 IN 封包完成 (TPC) 或傳送 Underrun (TUR) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR13 位元。

11.6.11.7 端點 14 中斷請求 (IR14)

若 UDC 中斷控制暫存器內的 IM14 位元被清除，以及 UDC 端點 14 控制/狀態暫存器內的 OUT 封包完成 (RPC) 或接收溢位 (ROF) 被設定，或是 UFNHR 內的等時錯誤端點 14 (IPE14) 被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR14 位元。

11.6.11.8 端點 15 中斷請求 (IR15)

若 UDC 中斷控制內的 IM15 位元被設定，則設定中斷請求位元。藉由寫入 1 來清除 IR15 位元。

表 11-22 UDC 狀態/中斷暫存器 1

0h 4060 005C				USIR1			讀取 / 寫入與唯讀		
位元	31:8	7	6	5	4	3	2	1	0
	保留	IR15	IR14	IR13	IR12	IR11	IR10	IR9	IR8
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
0	IR8	端點 8 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 8 需要服務。
1	IR9	端點 9 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 9 需要服務。
2	IR10	端點 10 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 10 需要服務。
3	IR11	端點 11 之中斷請求 (讀取/寫入 1 來清除) 1 – 端點 11 需要服務。
4	IR12	端點 12 之中斷請求 (讀取/寫入 1 來清除)

		1 – 端點 12 需要服務。
5	IR13	端點 13 之中斷請求（讀取/寫入 1 來清除） 1 – 端點 13 需要服務。
6	IR14	端點 14 之中斷請求（讀取/寫入 1 來清除） 1 – 端點 14 需要服務。
7	IR15	端點 15 之中斷請求（讀取/寫入 1 來清除） 1 – 端點 15 需要服務。
31:8	保留	保留給以後使用

11.6.12 UDC 訊框數目高電位暫存器 (UFNHR)

UDC 訊框數目高電位暫存器保留包含在最後接收的 SOF 封包、等時 OUT 端點錯誤狀態、SOF 中斷狀態/中斷遮蔽位元內之訊框數目最高的三個有效位元。

11.6.12.1 UDC 訊框數目 MSB (FNMSB)

UFNHR[FNMSB]為包含在最後接收的 SOF 封包內之 11 位元訊框數目的最高的三個有效位元。剩下的位元位於 UFNLR 內。此資訊用來做等時傳送。每個 SOF 時都會更新這些位元。

11.6.12.2 等時封包錯誤端點 4 (IPE4)

若端點 4 載入毀損的資料封包，則設定端點 4 等時封包錯誤 (isochronous packet error)。此狀態位元用來做端點 4 的中斷產生。為了保持同步化，當軟體服務 SOF 中斷與讀取訊框數目時，必須監視此位元。若符記封包毀損或資料封包的 PID 欄位毀損，則不會設定此位元。

11.6.12.3 等時封包錯誤端點 9 (IPE9)

若端點 9 載入毀損的資料封包，則設定端點 9 等時封包錯誤 (isochronous packet error)。此狀態位元用來做端點 9 的中斷產生。為了保持同步化，當軟體服務 SOF 中斷與讀取訊框數目時，必須監視此位元。若符記封包毀損或資料封包的 PID 欄位毀損，則不會設定此位元。

11.6.12.4 等時封包錯誤端點 14 (IPE14)

若端點 14 載入毀損的資料封包，則設定端點 14 等時封包錯誤 (isochronous packet error)。此狀態位元用來做端點 14 的中斷產生。為了保持同步化，當軟體服務 SOF 中斷與讀取訊框數目時，必須監視此位元。若符記封包毀損或資料封包的 PID 欄位毀損，則不會設定此位元。

11.6.12.5 訊框開始中斷遮蔽 (SIM)

UFNHR[SIM]位元用來遮蔽或啟動 SOF 中斷請求。當 UFNHR[SIM] = 1，則遮蔽中斷，而不允許設定 SIR 位元。當 UFNHR[SIM] = 0，則啟動中斷，且當接收端發生可中斷條件時，會設定 UFNHR[SIR]位元。設定 UFNHRSIM 為 1 不會影響 UFNHR[SIR]目前狀態，只會影響 UFNHR[SIR]之後 0 到 1 的轉換。

11.6.12.6 訊框開始中斷請求 (SIR)

若清除 UFNHR[SIM]位元並接收到 SOF 封包，則設定中斷請求位元。藉由寫入 1 來清除 UFNHR[SIR]位元。

表 11-23 UDC 訊框數目高電位暫存器

0h 4060 0060								
UFNHR						讀取		
位元	31:8	7	6	5	4	3	2	1 0
	保留	SIR	SIM	IPE14	IPE9	IPE4	3 位元訊框數目 MSB	
重置	X	0	1	0	0	0	0	0

位元	名稱	說明
2:0	FNMSB	訊框數目 MSB。 11 位元訊框數目之最高的 3 個有效位元與最後接收的 SOF 相關。
3	IPE4	等時訊框錯誤端點 4 (讀取/寫入 1 來清除) 1 – 端點 FIFO 內資料損毀之狀態指示器。
4	IPE9	等時訊框錯誤端點 9 (讀取/寫入 1 來清除) 1 – 端點 FIFO 內資料損毀之狀態指示器。
5	IPE14	等時訊框錯誤端點 14 (讀取/寫入 1 來清除) 1 – 端點 FIFO 內資料損毀之狀態指示器。
6	SIM	SOF 中斷遮蔽 0 – 啟動 SOF 中斷。 1 – 關閉 SOF 中斷。
7	SIR	SOF 中斷請求 (讀取/寫入 1 來清除)

		1 – 已接收 SOF。
31:8	保留	保留給以後使用

11.6.13 UDC 訊框數目低電位暫存器 (UFNLR)

UDC 訊框低電位暫存器為包含在最後接收的 SOF 封包內之 11 位元訊框數目的最低的 8 個有效位元。剩下的 3 個位元位於 UFNHR 內。此資訊用來做等時傳送。每個 SOF 時都會更新這些位元。

表 11-24 UDC 訊框數目低電位暫存器

0h 4060 0064		UFNLR						唯讀	
位元	31:8	7	6	5	4	3	2	1	0
	保留 d	8 位元訊框數目 LSB							
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	FNLSB	訊框數目 LSB 訊框數目之最低的 8 個有效位元與最後接收的 SOF 相關。
31:8	保留	保留給以後使用

11.6.14 UDC 位元組計數暫存器 x (UBCRx), x = 2、4、7、9、12、14

位元組計數暫存器 (byte count register) 維護 OUT 端點 (x) 內主動緩衝區剩下的位元組計數。

11.6.14.1 端點 x 位元組計數 (BC[7:0])

在讀取每個位元組之後會更新位元組計數。當軟體接收到中斷表示端點具有資料，則可讀取位元組計數暫存器來決定尚待讀取的位元組數目。輸入緩衝區內剩下的位元組數目與位元組計數+1 相等。

表 11-25 UDC 位元組計數暫存器 x，x = 2、4、7、9、12、14

	0h 4060 0068	UBCR2	唯讀						
	0h 4060 006C	UBCR4	唯讀						
	0h 4060 0070	UBCR7	唯讀						
	0h 4060 0074	UBCR9	唯讀						
	0h 4060 0078	UBCR12	唯讀						
	0h 4060 007C	UBCR14	唯讀						
位元	31:8	7	6	5	4	3	2	1	0
	保留	BC[7:0]							
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	BC	位元組計數（唯讀）。FIFO 內的位元組數目為位元組計數加 1（BC+1）。
31:8	保留	保留給以後使用

11.6.15 UDC 端點 0 資料暫存器（UDDR0）

UDC 端點 0 資料暫存器為 16 個通道 8 個位元的雙向 FIFO。當主機傳送資料至 UDC 端點 0，Megacell 會讀取 UDC 端點 0 暫存器來存取資料。當 UDC 傳送資料至主機，Megacell 會將傳送的資料寫入至 UDC 端點 0 暫存器。Megacell 在控制程序下只可在特定的點上讀取與寫入 FIFO。FIFO 資料流方向由 UDC 控制。正常情況下，UDC 為閒置狀態，等待主機傳送命令。當主機傳送命令時，UDC 會以來自主機的命令填滿 FIFO，而當命令到達時 Megacell 會從 FIFO 讀取命令。Megacell 可能寫入端點 0 FIFO 的唯一時間為來自主機的有效命令被接收之後，而且需要回應傳送時。

表 11-26 UDC 端點 0 資料暫存器

	0h 4060 0080				UDDR0		讀取 / 寫入		
位元	31:8	7	6	5	4	3	2	1	0
	保留	端點 0FIFO 之底部（對讀取）							
重置	X	0	0	0	0	0	0	0	0
	31:8	7	6	5	4	3	2	1	0
	保留	端點 0FIFO 之頂部（對寫入）							
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	DATA	端點 0 FIFO 資料之頂部/底部。 讀取 – 端點 0 FIFO 資料之底部。 寫入 – 端點 0 FIFO 資料之頂端。
31:8	保留	保留給以後使用

11.6.16 UDC 資料暫存器 x (UDDR_x), x = 1、6、11

端點 (x) 為 64 位元組長的雙倍緩衝區 bulk IN 端點。資料可經由 DMA 載入，或直接由 Megacell 寫入。由於為雙倍緩衝區，最多可能載入兩個資料封包來傳送。

表 11-27 UDC 端點 x 資料暫存器，x = 1、6、11

	0h 4060 0100	UDDR1	寫入						
	0h 4060 0600	UDDR6	寫入						
	0h 4060 0B00	UDDR11	寫入						
位元	31:8	7	6	5	4	3	2	1	0
	保留	8 位元資料							
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	DATA	目前載入的端點資料之頂部。
31:8	保留	保留給以後使用

11.6.17 UDC 資料暫存器 x (UDDR_x), x = 2、7、12

端點 (x) 為 64 位元組長的雙倍緩衝區 bulk OUT 端點。當接收到 EOP 時，UDC 會盡快產生中斷或 DMA 請求。由於為雙倍緩衝區，最多可能有兩個資料封包已準備好。經由 DMA 或直接讀取 Megacell，可能會從 UDC 移除資料。若一個封包被移除，而在此封包後的封包已被接收，則在下次傳送 OUT 封包只端點 (x) 時，UDC 會發出 NAK 給主機。NAK 狀態會適當維持直到端點 (x) UDC 內有一個完整的封包空間可使用。

表 11-28 UDC 端點 x 資料暫存器，x = 2、7、12

	0h 4060 0180	UDDR2	讀取						
	0h 4060 0680	UDDR7	讀取						
	0h 4060 0B80	UDDR12	讀取						
位元	31:8	7	6	5	4	3	2	1	0
	保留 d	8 位元資料							
重置 t	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	DATA	目前讀取的端點資料之頂部。
31:8	保留	保留給以後使用

11.6.18 UDC 資料暫存器 x (UDDR_x)，x = 3、8、13

端點 (x) 為為 256 位元組長的雙倍緩衝區等時 IN 端點。資料可經由 DMA 載入，或直接由 Megacell 寫入。由於為雙倍緩衝區，最多可能載入兩個資料封包來傳送。

表 11-29 UDC 端點 x 資料暫存器，x = 1、6、11

	0h 4060 0200	UDDR3	寫入						
	0h 4060 0700	UDDR8	寫入						
	0h 4060 0C00	UDDR13	寫入						
位元	31:8	7	6	5	4	3	2	1	0
	保留	8 位元資料							
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	DATA	目前載入的端點資料之頂部。
31:8	保留	保留給以後使用

11.6.19 UDC 資料暫存器 x (UDDR_x)，x = 4、9、14

端點 (x) 為為 256 位元組長的雙倍緩衝區等時 OUT 端點。當接收到 EOP

時，UDC 會產生中斷或 DMA 請求。由於為雙倍緩衝區，最多可能有兩個資料封包已準備好。經由 DMA 或直接讀取 Megacell，資料可能會從 UDC 移除。若一個封包被移除，而在此封包後的封包已被接收，則在下一次傳送 OUT 封包只端點 (x) 時，UDC 會發出 NAK 給主機。NAK 狀態會適當維持直到端點 (x) UDC 內有一個完整的封包空間可使用。

表 11-30 UDC 端點 x 資料暫存器，x = 4、9、14

	0h 4060 0400	UDDR4	讀取						
	0h 4060 0900	UDDR9	讀取						
	0h 4060 0E00	UDDR14	讀取						
位元	31:8	7	6	5	4	3	2	1	0
	保留	8 位元資料							
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	DATA	目前讀取的端點資料之頂部。
31:8	保留	保留給以後使用

11.6.20 UDC 資料暫存器 x (UDDR_x)，x = 5、10、15

端點(x)為為 8 位元組長的雙倍緩衝區中斷 IN 端點。資料必須經由 Megacell 直接寫入來載入。由於 USB 系統為主機初始程式模組，主機必須讀取端點 5 來決定中斷條件。UDC 不可初始交易。

表 11-31 UDC 端點 x 資料暫存器，x = 5、10、15

	0h 4060 00A0	UDDR5	寫入						
	0h 4060 00C0	UDDR10	寫入						
	0h 4060 00E0	UDDR15	寫入						
位元	31:8	7	6	5	4	3	2	1	0
	保留	8 位元資料							
重置	X	0	0	0	0	0	0	0	0

位元	名稱	說明
7:0	DATA	目前載入的端點資料之頂部。
31:8	保留	保留給以後使用

11.6.21 UDC 暫存器位置

表 11-32 為與 UDC 相關的暫存器，與用來存取的實體位址。

表 11-32 UDC 控制、資料、狀態暫存器位置

位址	名稱	說明
0h 4060 0000	UDCCR	UDC 控制暫存器
0h 4060 0004	保留	保留為以後使用
0h 4060 0008	保留	保留為以後使用
0h 4060 000C	保留	保留為以後使用
0h 4060 0010	UDCCS0	UDC 端點 0 控制/狀態暫存器
0h 4060 0014	UDCCS1	UDC 端點 1 (IN) 控制/狀態暫存器
0h 4060 0018	UDCCS2	UDC 端點 2 (OUT) 控制/狀態暫存器
0h 4060 001C	UDCCS3	UDC 端點 3 (IN) 控制/狀態暫存器
0h 4060 0020	UDCCS4	UDC 端點 4 (OUT) 控制/狀態暫存器
0h 4060 0024	UDCCS5	UDC 端點 5 (中斷) 控制/狀態暫存器
0h 4060 0028	UDCCS6	UDC 端點 6 (IN) 控制/狀態暫存器
0h 4060 002C	UDCCS7	UDC 端點 7 (OUT) 控制/狀態暫存器
0h 4060 0030	UDCCS8	UDC 端點 8 (IN) 控制/狀態暫存器
0h 4060 0034	UDCCS9	UDC 端點 9 (OUT) 控制/狀態暫存器
0h 4060 0038	UDCCS10	UDC 端點 10 (中斷) 控制/狀態暫存器
0h 4060 003C	UDCCS11	UDC 端點 11 (IN) 控制/狀態暫存器
0h 4060 0040	UDCCS12	UDC 端點 12 (OUT) 控制/狀態暫存器
0h 4060 0044	UDCCS13	UDC 端點 13 (IN) 控制/狀態暫存器
0h 4060 0048	UDCCS14	UDC 端點 14 (OUT) 控制/狀態暫存器
0h 4060 004C	UDCCS15	UDC 端點 15 (中斷) 控制/狀態暫存器
0h 4060 0050	UICR0	UDC 中斷控制暫存器 0
0h 4060 0054	UICR1	UDC 中斷控制暫存器 1
0h 4060 0058	USIR0	UDC 狀態中斷暫存器 0
0h 4060 005C	USIR1	UDC 狀態中斷暫存器 1
0h 4060 0060	UFNHR	UDC 訊框數目暫存器高電位
0h 4060 0064	UFNLR	UDC 訊框數目暫存器低電位
0h 4060 0068	UBCR2	UDC 位元組計數暫存器 2
0h 4060 006C	UBCR4	UDC 位元組計數暫存器 4
0h 4060 0070	UBCR7	UDC 位元組計數暫存器 7
0h 4060 0074	UBCR9	UDC 位元組計數暫存器 9

0h 4060 0078	UBCR12	UDC 位元組計數暫存器 12
0h 4060 007C	UBCR14	UDC 位元組計數暫存器 14
0h 4060 0080	UDDR0	UDC 端點 0 資料暫存器
0h 4060 0100	UDDR1	UDC 端點 1 資料暫存器
0h 4060 0180	UDDR2	UDC 端點 2 資料暫存器
0h 4060 0200	UDDR3	UDC 端點 3 資料暫存器
0h 4060 0400	UDDR4	UDC 端點 4 資料暫存器
0h 4060 00A0	UDDR5	UDC 端點 5 資料暫存器
0h 4060 0600	UDDR6	UDC 端點 6 資料暫存器
0h 4060 0680	UDDR7	UDC 端點 7 資料暫存器
0h 4060 0700	UDDR8	UDC 端點 8 資料暫存器
0h 4060 0900	UDDR9	UDC 端點 9 資料暫存器
0h 4060 00C0	UDDR10	UDC 端點 10 資料暫存器
0h 4060 0B00	UDDR11	UDC 端點 11 資料暫存器
0h 4060 0B80	UDDR12	UDC 端點 12 資料暫存器
0h 4060 0C00	UDDR13	UDC 端點 13 資料暫存器
0h 4060 0E00	UDDR14	UDC 端點 14 資料暫存器
0h 4060 00E0	UDDR15	UDC 端點 15 資料暫存器

問題：

1. 請簡述 UDC 資料傳送各欄位之意義？
2. 請簡述 UDC 各種資料封包類型？
3. 請簡述 UDC 各種交易種類？
4. 請簡述 UDC 訊號層級中，如何以 UDC+及 UDC-編碼，來表示各種狀態？
5. USB 傳輸過程中，所使用的位元編碼方式為何？
6. 何謂 Token 封包？可識別哪幾種不同的處理？
7. 請簡述 EP0 控制讀取的步驟。
8. 請簡述 Reset 的步驟。
9. 請簡述 Suspend 的步驟。
10. 請簡述 Resume 的步驟。