

USB1.1 在嵌入式系統的設計與除錯

前言

USB是很常見的串列協定。USB有熱插拔的特性，當USB裝置插入主機時，主機會偵測到，並且軟體會對其作規劃，即插即用。USB的连接線供應裝置電源，可提供DC +5V的電壓，電流供應約100mA~500mA。

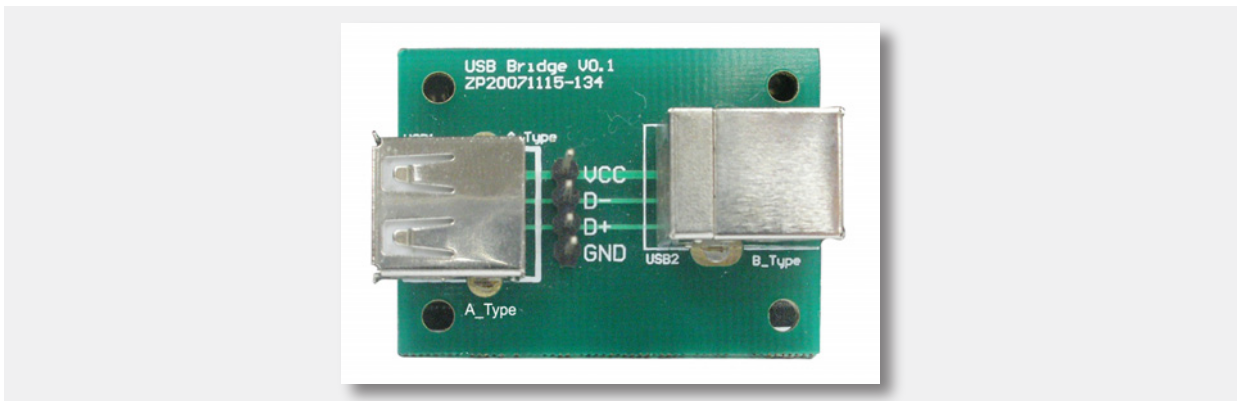
USB的连接線共有四條，其中兩條是電源與地線，另兩條是訊號線，分別是D+與D-。一般情況兩條線的訊號是相反的，若其中一條的訊號為High，則另一條的訊號就為Low。如此的差動訊號能提升訊號抗雜訊抗干擾的能力，USB就是利用D+ D-兩條訊號線的差動訊號，來達成高速傳輸的目的。

USB主機端在沒有裝置連接時，D+ D-上會有一個1.5K歐姆的電阻接地，確保這兩條線是在接地狀態。可支援至127個位址，也就是說，一個USB系統最多支援127個裝置，主機會週期性的探詢每個集線器以便得知裝置的连接狀態。以USB1.1而言，分為高速與低速兩個模式。低速傳輸率為1.5M/s，高速為12M/s。高速模式時，1秒 / 12M，每個bit的寬度約83.3ns。

封包是組成USB傳輸的最小單位。在USB傳輸時負責替電腦Host端與集線器上的每一個USB裝置傳輸溝通，而孕龍邏輯分析儀可以透過訊號抓取幫助工程師們將訊號直接轉譯解碼並顯示於軟體中，讓使用者可以直接分析USB訊號。

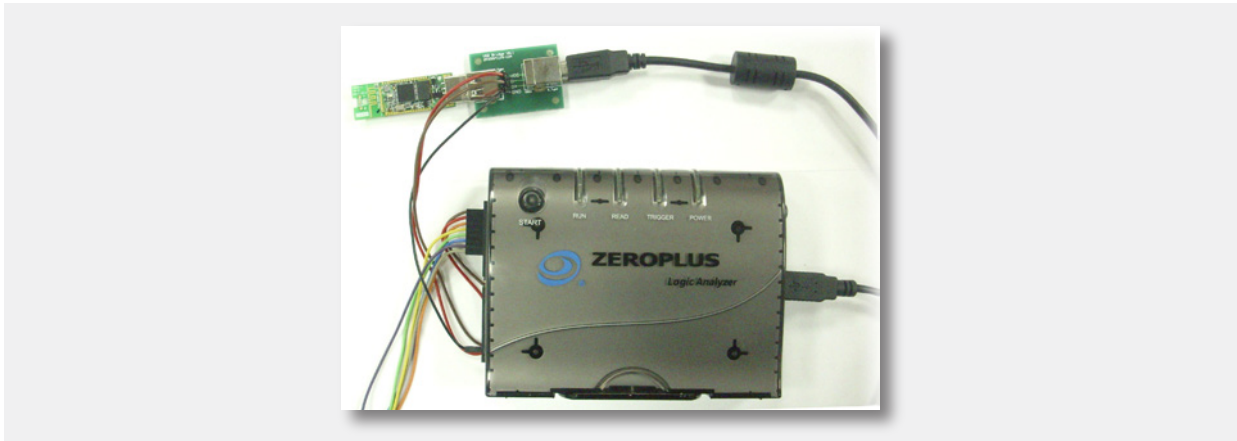
如何使用邏輯分析儀連接待測的USB裝置？

待測物(Device Under Test)為USB Bluetooth Dongle (USB1.1)，針對相同訊號使用兩部邏輯分析儀做出測量比較。孕龍邏輯分析儀針對USB1.1模組製作出相對應的測量治具能夠幫助使用者省去繁雜的接線步驟，直接將待測物連接完成。



► 圖一 USB測量治具

將治具中的A Type連接埠接上待測物，B Type連接埠透過隨附USB線接上PC USB Host，並將孕龍邏輯分析儀的A0及A1通道藉由測試線連接至USB治具上的D-、D+。連接完成圖如圖二所示。



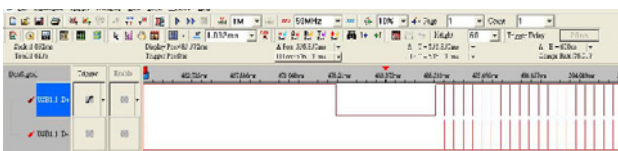
► 圖二 待測物與孕龍邏輯分析儀連接圖

使用孕龍邏輯分析儀

使用他牌邏輯分析儀

☞ 使用孕龍邏輯分析儀(已安裝USB1.1模組):

☞ 使用他牌邏輯分析儀:



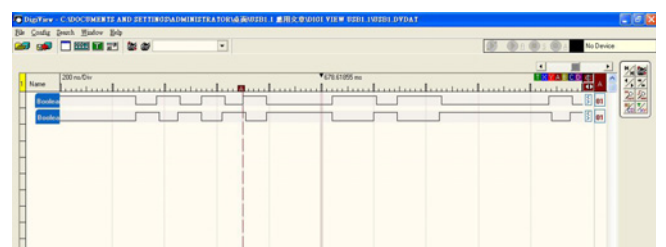
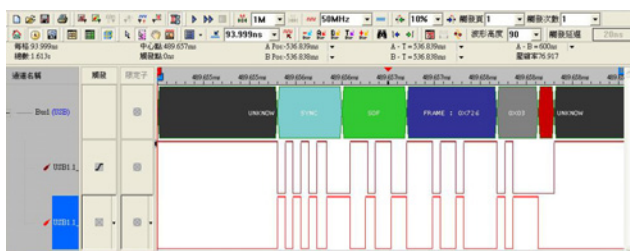
使用孕龍邏輯分析儀擷取USB Bluetooth Dongle的畫面。

使用他牌邏輯分析儀擷取USB Bluetooth Dongle的畫面。



☞ 準備分析邏輯分析儀訊號:

☞ 準備分析邏輯分析儀訊號:

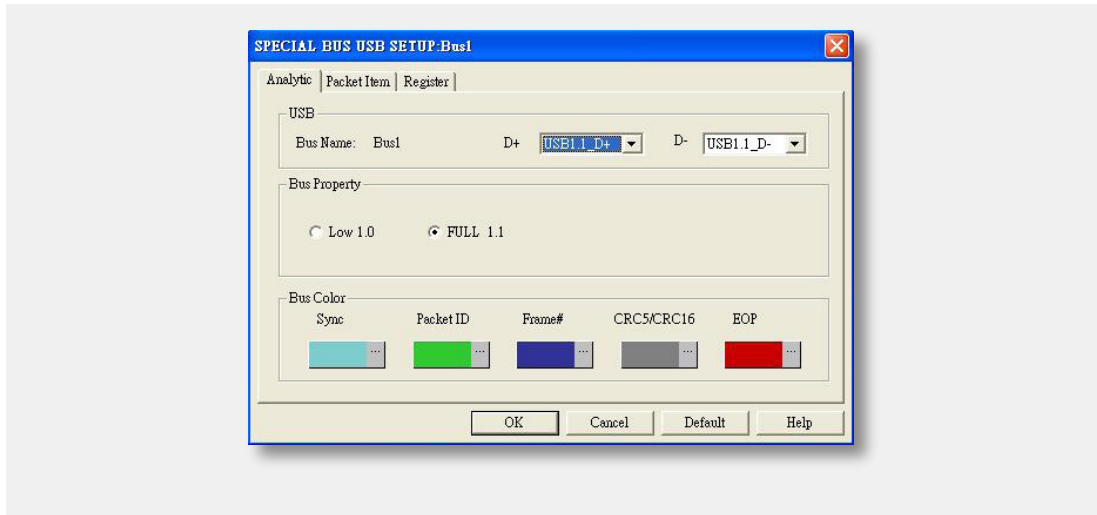


透過孕龍邏輯分析儀USB匯流排模組，可以很輕易的透過匯流排BUS知道訊號的封包架構。

他牌邏輯分析儀並無USB匯流排模組，沒有辦法針對訊號進行解碼。

USB 串列協定模組設定

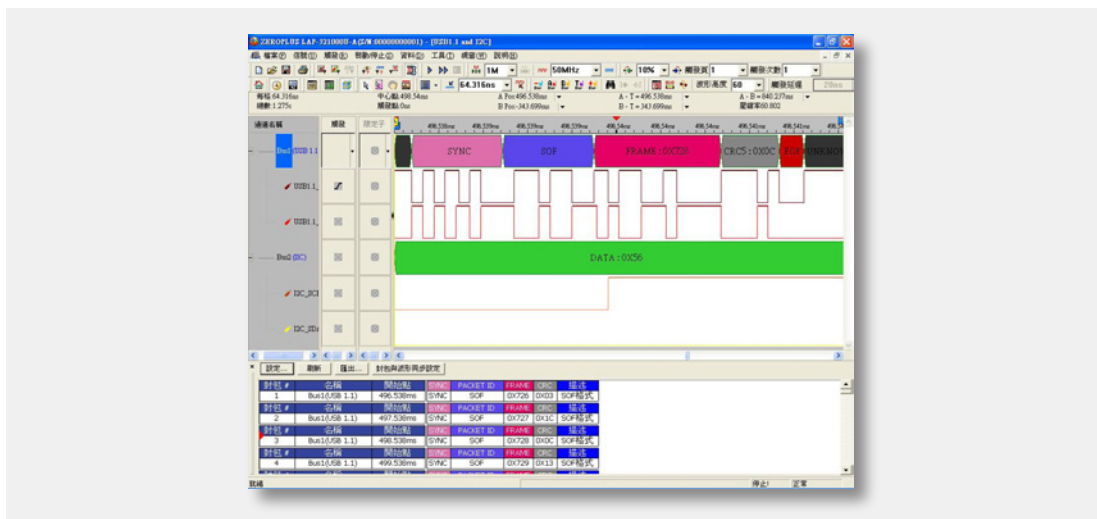
串列協定模組設定方便又快速，如圖五所示使用者只需要設定D+、D-的腳位以及訊號速度，點下OK後就可以將USB訊號解碼。



► 圖五 USB串列協定模組

串列協定封包列表

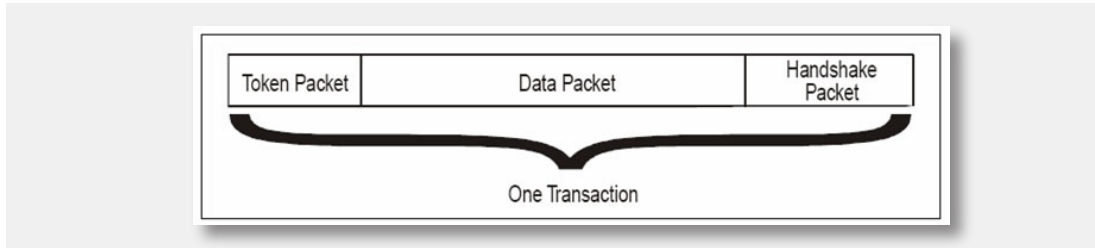
為更加方便分析以及提高效率，孕龍邏輯分析儀更加在新版軟體中加入串列協定封包列表功能，能夠將串列訊號採取封包段落顯示以直列方式把資料內容顯示出來，圖六為USB1.1訊號使用串列協定封包列表顯示的結果。



► 圖六 USB1.1訊號使用串列協定封包列表顯示的結果

USB封包簡介

封包是組成USB傳輸的最小單位。一個Transaction通常由三個封包組成，但依傳輸型態而定，一個Transaction可能包含一個、兩個、三個封包：



圖七 USB1.1封包架構概述

Token 封包：

每個Transaction以Token 封包做起始。Token 封包定義裝置、Endpoint數量，傳輸的方向。其中SOF (Start Of Frame) Token包含目前的frame數，而且會廣播 (broadcast) 給所有的full-speed裝置知道。SOF也是唯一一個不指定目標的Token。Token Packet 長度固定為4個Byte。

Data 封包：

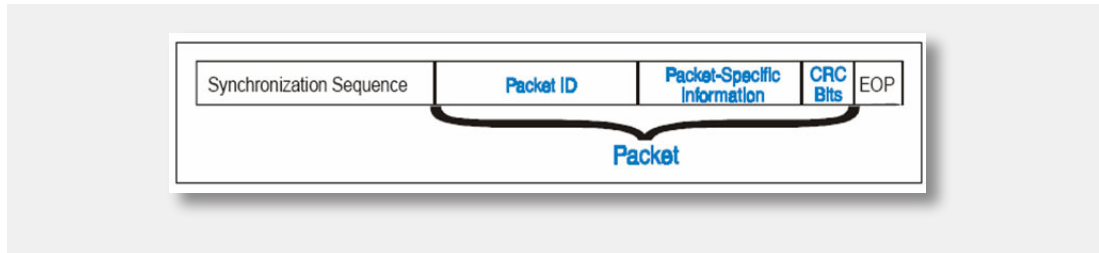
Data封包包含處理此動作的資料。一個Transaction中，Data封包最大的資料量為1023個Bytes。實際允許的最大資料量取決於傳輸狀態。

Handshake封包：

除了即時型傳輸(Isochronous)之外，所有的傳輸都保證資料的傳遞正確。Handshake封包回應資料是否正確的被收到。若執行處理動作中發生錯誤，此處理動作將重新執行。

封包格式：

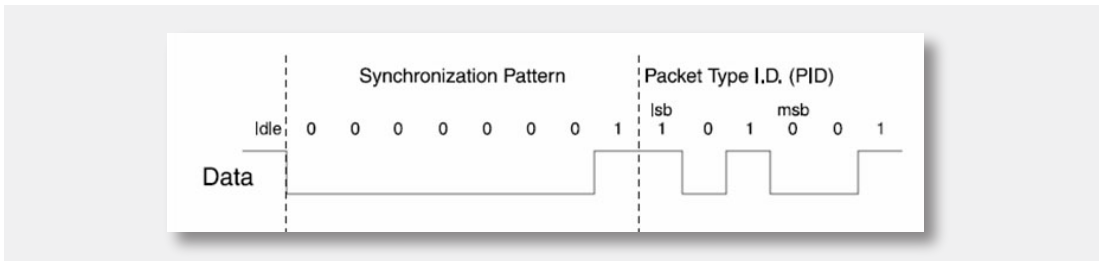
封包是執行所有處理動作的機制。圖八是USB封包的基本格式。封包緊跟在Synchronization sequence (SYNC, 同步列) 之後，Synchronization sequence 使USB裝置能跟封包中的位元速率同步。封包的Type由一個Packet ID來定義。Packet ID之後是隨著封包Type而定義的資訊，如Address或Data。最後，每個封包以一個CRC (Cyclic Redundancy Check)做結尾。CRC是用來確認資料是否正確的傳輸。每個封包始使用EOP (End Of Packet) 狀態來做辨識。



► 圖八 USB封包的基本格式

Synchronization sequence (同步序列) :

圖九是Synchronization sequence的波形。Synchronization sequence 由八個bit組成，數值是二進位的00000001，依照NRZI編碼原理，資料0代表訊號要轉換，所以前七個0之中的每個Bit都會在相對應的Bit時間中轉換，作用就好像提供了一組可用來同步的Clock。

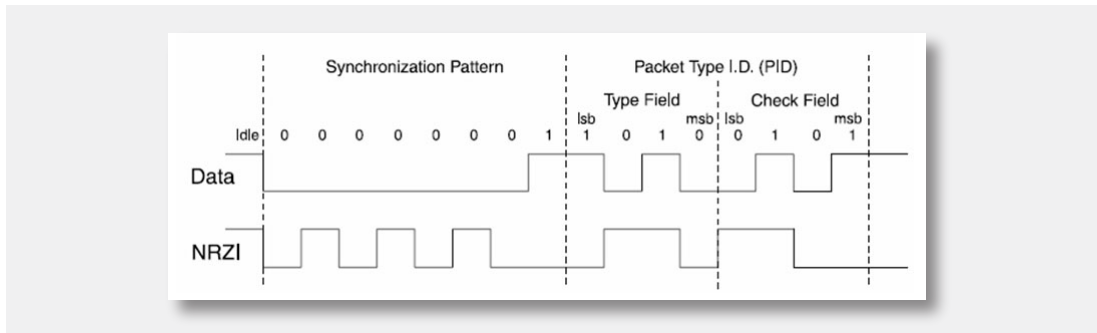


► 圖九 Synchronization sequence

Packet ID :

Packet ID定義封包的目的與內容。封包主要有三種型態：Token、Data、Handshake。其中Token封包定義傳輸的型態；Data封包接在Token之後，將傳輸的資料輸出或輸入USB裝置中；Handshake則是提供傳輸成功或失敗的訊息。

Packet ID的訊號格式如圖十。Packet ID總共有8個Bit，前四個Bit代表ID(Type Field)，後4個Bit是檢查欄位 (Check Field)，其值固定是ID的反向。如ID為二進位的1001，則檢查欄位就是0110，合起來就是10010110。



▶ 圖十 Packet ID 訊號格式

封包種類：

所有處理動作都是由一個Token封包開始。USB定義四種Token封包：SOF，IN，OUT，SETUP。圖十一是四個封包的PID與功能描述。

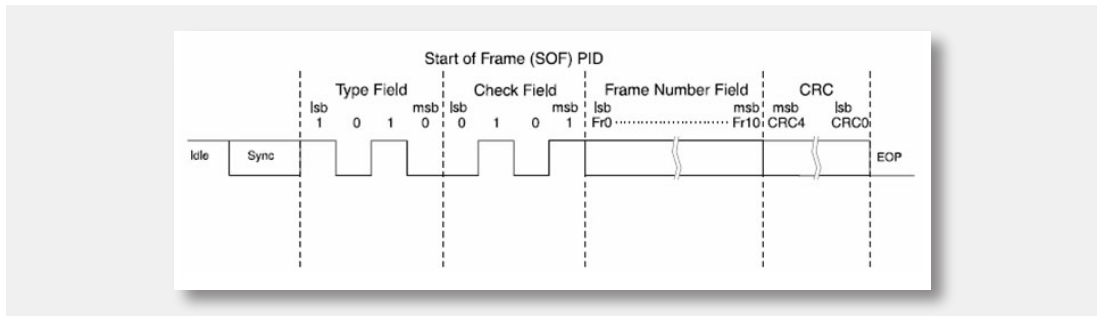
PID Type	PID Name	PID[3:0]	Description of Token Packet
Token	SOF	0101b	Contains start of frame (SOF) marker and frame number. The SOF token is used by isochronous endpoints to synchronize its transfers.
Token	SETUP	1101b	Contains USB device address and endpoint number. Transfer is from host to function for setting up a control endpoint (e.g., configuration).
Token	OUT	0001b	Contains the USB device address and endpoint number. The transfer is from host to function.
Token	IN	1001b	Contains the USB device address and endpoint number. The transfer is from function to host.

▶ 圖十一 四種Token封包狀態

SOF 封包：

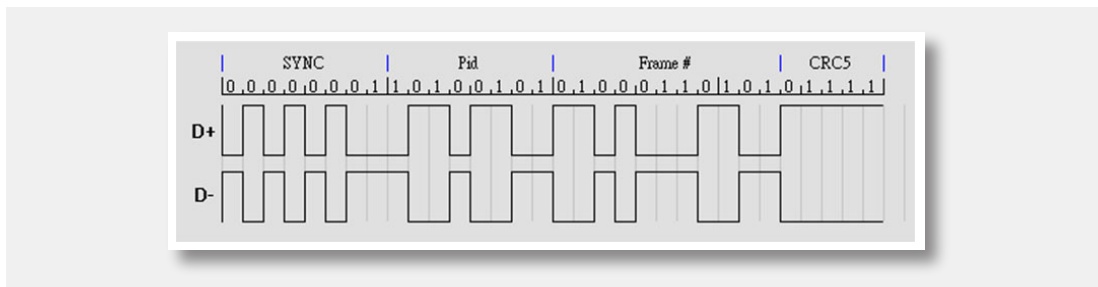
USB裝置用SOF封包來判斷訊框的起點。當每個frame開始的時候，SOF封包被送至每個全速的裝置。即時型傳輸裝置可經由SOF封包來達到同步傳輸的作用。低速裝置不支援即時型傳輸，所以SOF封包不會送至低速裝置。

如圖十二所示，SOF封包包含一個長度為11 bit的frame number(訊框號碼)，接收器使用5 bit的CRC位元來確認frame number。SOF封包不含任何資料，其傳輸也不保證一定成功。當USB裝置對此SOF封包PID檢查碼錯誤時，裝置忽略此封包；當CRC檢查錯誤時，忽略此frame number。注意PID與Frame Number都是LSB在前面MSB在後面。



▶ 圖十二 Start of Frame (SOF)

圖十三是一個SOF封包的實際波形。PID前四個 bit “0101” 就是SOF封包。PID之後的是11 bit 的Frame Number，這個例子的Frame Number是 10101100010，換算為十進位就是1378，也就是第1378個Frame。

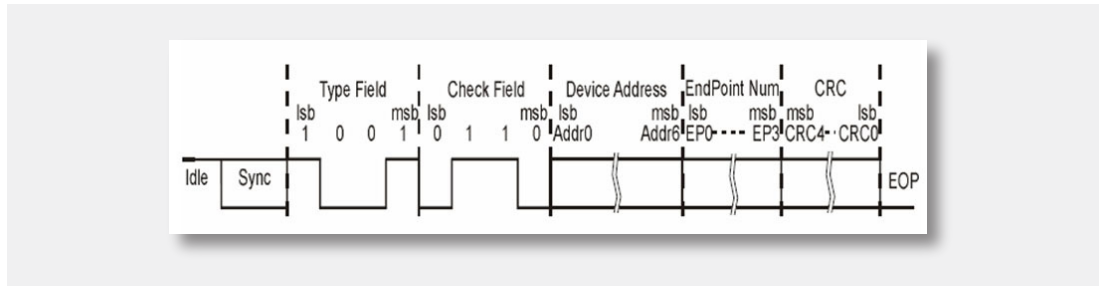


▶ 圖十三 SOF 封包實際波形

IN 封包：

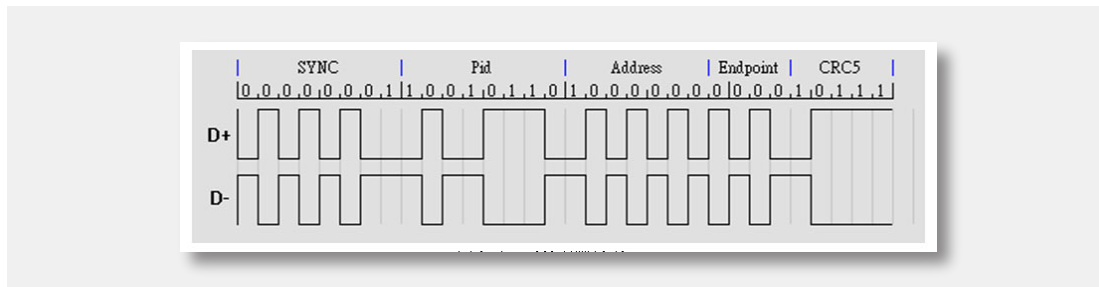
Token還有兩種封包，IN封包與OUT封包。所謂IN與OUT是指對於PC而言。IN封包是PC需要由裝置讀進來資料時使用的；OUT則是PC需要送出資料給裝置時使用的。IN封包可以使用在中斷型傳輸、巨量型傳輸、控制型傳輸的資料時期、即時型傳輸等傳輸。

圖十四是IN封包格式示意圖與實際量測圖。如圖所示，除了基本的Type欄位與Check欄位之外，IN封包包含7 bit 的Device Address與4 bit 的End Point Number。執行IN封包動作時，首先PC送出一個IN封包，接著裝置回傳一個資料封包，最後PC送出一個交握封包給Device，表示已收到資料。但在即時型傳輸中，IN處理動作不使用交握封包。IN傳輸所能傳輸的資料量取決於傳輸型態。



► 圖十四 IN封包格式示意圖與實際量測圖

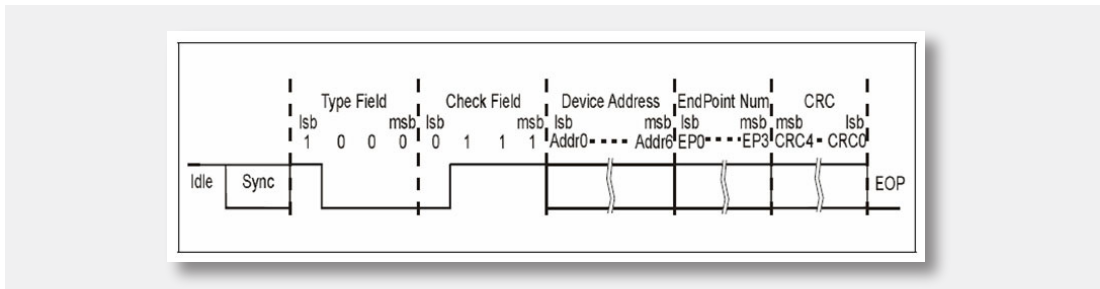
圖十五是一個實際的IN封包。PID的前4 bit 為 “1001” 就是IN封包的ID。再來是7 bit 的Address，這個例子的 Address是 “1”。再來是4 bit的Endpoint，這個例子的Endpoint是 “0”。最後是5 bit 的CRC。



► 圖十五 IN封包實際波形

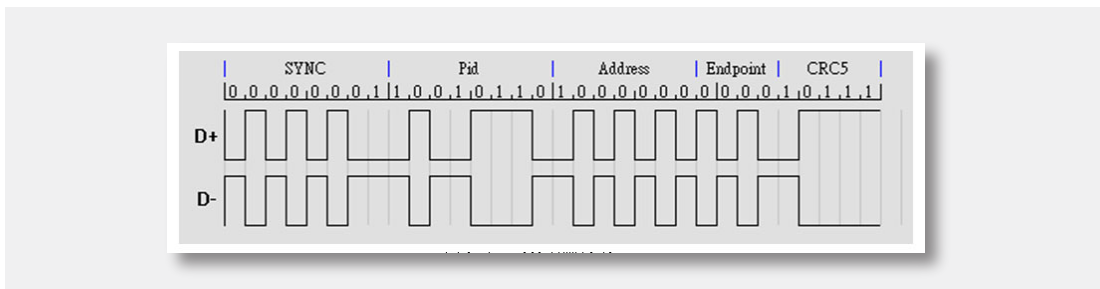
OUT 封包 :

當資料將被送出給某個Device時，系統會進行一個OUT傳輸。有三種傳輸型態用到OUT傳輸：巨量型傳輸、控制型傳輸的資料時期、即時型傳輸。OUT封包除了PID之外，其餘資料格式都與IN封包相同，一樣有7 bit 的Address，4 bit 的End-point（如圖十六所示）。



▶ 圖十六 OUT封包格式示意圖與實際量測圖

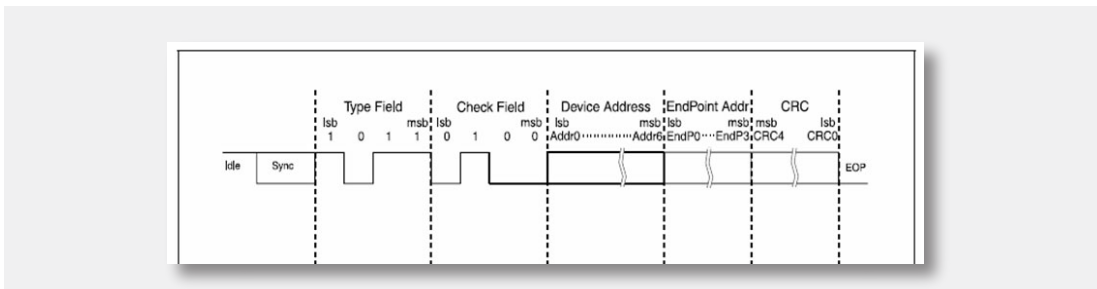
圖十七是一個實際的OUT封包。PID為“0001”就是OUT封包。這個例子的Address是“1”。再來是4 bit 的Endpoint，這個例子的Endpoint是 “0”。最後是 5 bit 的CRC。



▶ 圖十七 OUT封包實際波形

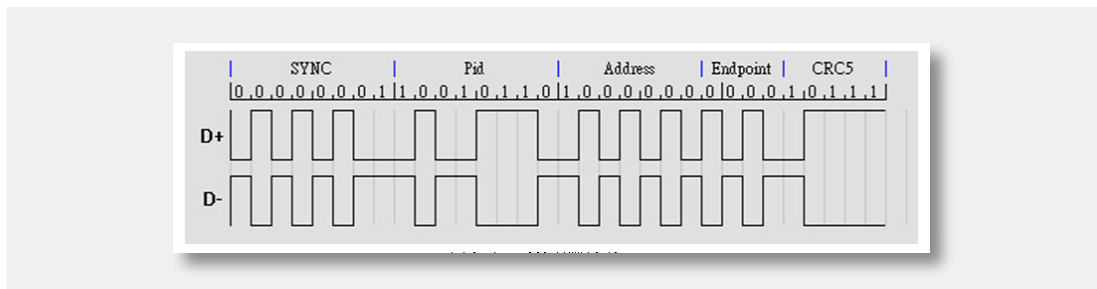
SETUP 封包：

SETUP封包只有在控制型傳輸的設定階段使用。SETUP處理動作啟動一個控制型傳輸，並且定義為設定階段。SETUP處理動作在形式上類似一個OUT動作：SETUP封包後跟著一個Data0封包與一個回覆封包。SETUP封包的目的是傳送一個要求給Device執行。下圖是SETUP封包的格式。SETUP封包除了PID之外，其餘資料格式都與IN/OUT封包相同，一樣有7 bit 的Address，4 bit 的Endpoint（如圖十八所示）。



▶ 圖十八 SETUP封包格式示意圖與實際量測圖

圖十九是一個實際的SETUP封包。PID為“1101”就是SETUP封包。這個例子的Address是“1”。再來是4 bit 的Endpoint，這個例子的Endpoint是“0”。最後是5 bit的CRC。

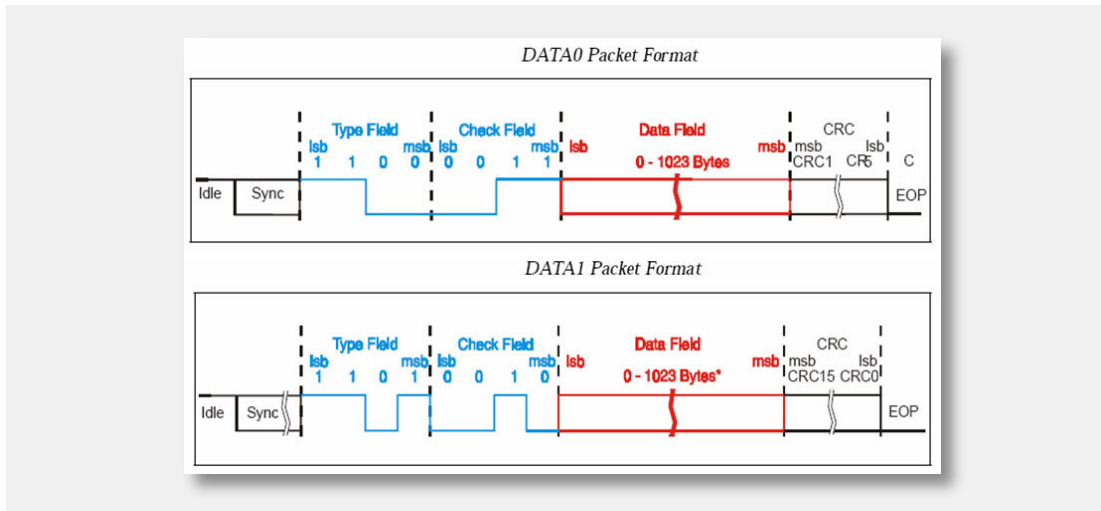


▶ 圖十九 SETUP封包實際波形

Data0 與 Data1封包：

Data封包跟在IN、OUT、SETUP三種封包之後。IN封包之後的Data封包是將資料傳入PC；OUT與SETUP之後的封包是將資料傳出給Device。傳輸資料時，Data0封包與Data1封包是交互的使用，如果此次傳輸資料是用Data0封包，則下次傳輸資料就是使用Data1封包，反之亦然。這是為了用來當作一個同步機制使用。一個很大的資料，會被切成多個小封包來傳輸，此時Data0與Data1封包交互的使用就可以當作一個同步的機制。

圖二十是Data0與Data1的封包格式。PID為“1100”就是Data0封包，PID為“1011”就是Data1封包。



► 圖二十 Data0 與 Data1 的封包格式

交握封包 (Handshake Packets) :

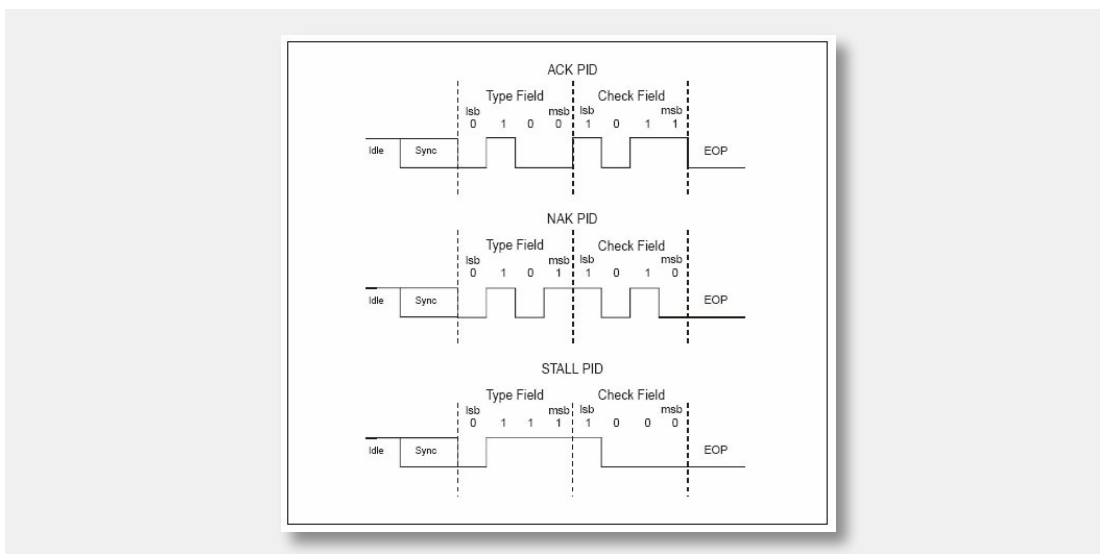
交握封包用來回報處理動作的完成狀況。交握封包有三種。

ACK封包：確認資料正確無誤的被接收。

NAK封包：告知主機現在目標裝置暫時無法接收或傳回資料。

STALL封包：無法完成傳輸，且需要軟體介入以便使得裝置能從STALL狀態復原。

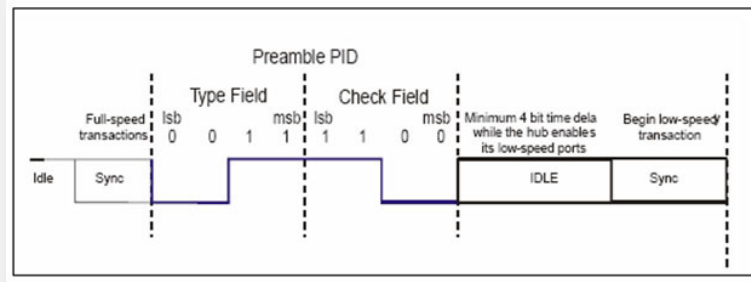
圖二十一是一三種交握封包的格式圖。交握封包沒有資料，也沒有CRC，只有SYNC與PID。ACK的PID是0010，NAK的PID是1010，STALL的PID是1110。



► 圖二十一 三種交握封包格式圖

前導封包 (Preamble Packet) :

前導封包用在低速傳輸時。在傳送一個低速封包前，一個前導封包必須被送出，主機保證前導封包之後的封包是低速傳送。圖二十二是前導封包示意圖。前導封包由一個SYNC與一個PID組成，PID是1100。有一點必須注意的是，前導封包不以EOP結尾。



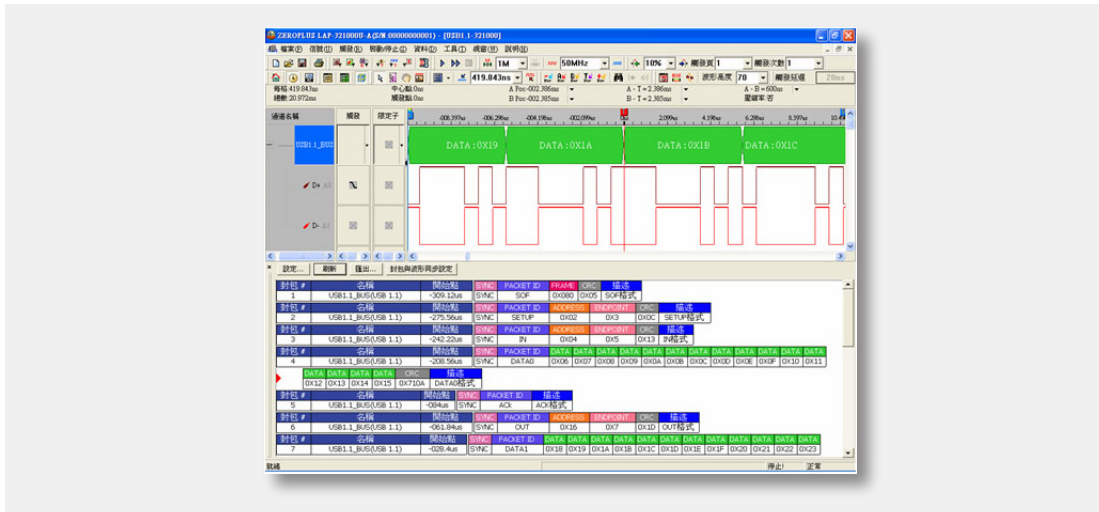
► 圖二十二 前導封包示意圖

 串列協定封包觸發

觸發功能越強大越能夠精準的觸發至正確的數值上，而串列協定封包觸發功能便是如此，僅需在設定視窗中設定串列協定中的狀態或是指定數值，透過軟體強大的功能便可正確的觸發至指定位置上。EX:設定觸發條件為USB1.1訊號指定資料為0X1B時觸發。



► 圖二十三 觸發條件設定



▶ 圖二十四 完成觸發

總結

串列協定常常被應用在嵌入式系統的設計中，但是若僅使用示波器分析這些訊號是非常花費時間的，孕龍的PC-Based邏輯分析儀可以提供工程師們更強大的觸發功能、串列協定解碼模組及便利的資料搜尋，讓工程師在開發產品時更能夠得心應手。