

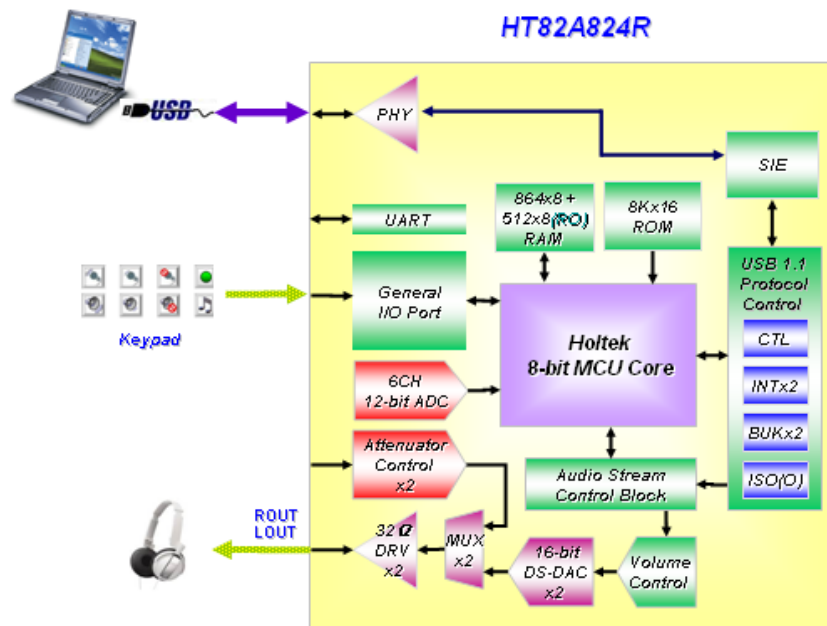
HT82A824R 在 USB Speaker 之應用說明

文件編碼：HA0277T

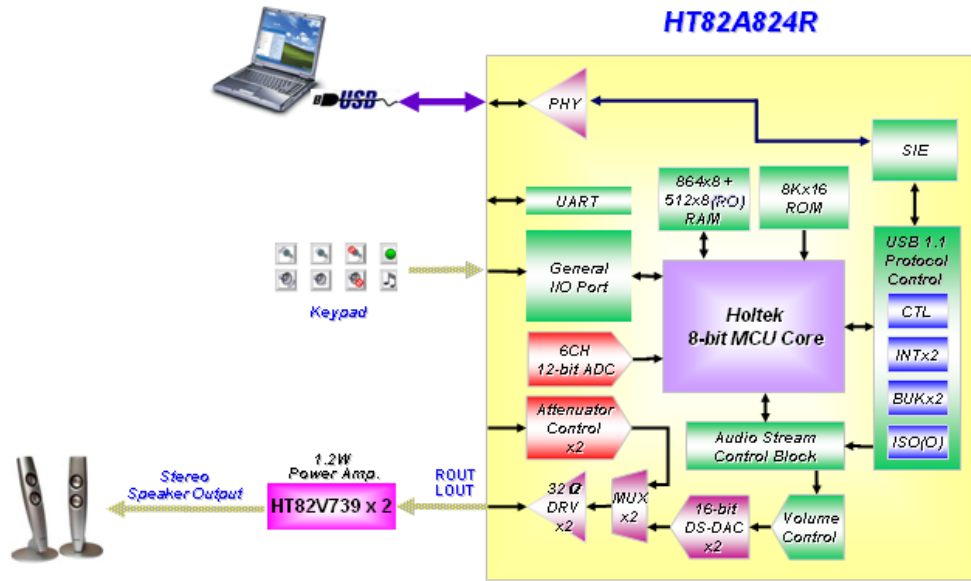
簡介

HT82A824R 為 Enhanced USB Speaker MCU，本檔將介紹如何使用 HT82A824R 設計 USB Speaker 相關產品，也探討 USB Audio Class 的規範與程式架構說明。

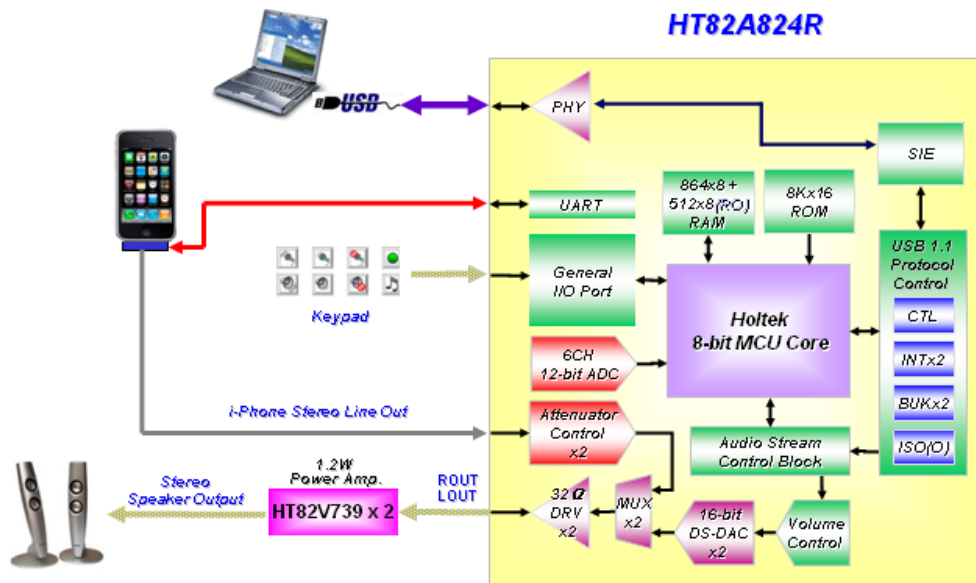
HT82A824R USB Headphone應用方塊圖



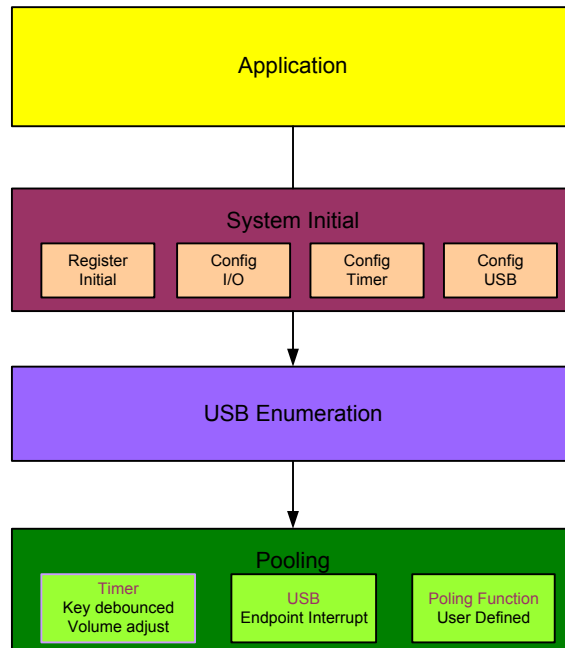
HT82A824R USB Speaker 應用方塊圖



HT82A824R USB/i-Phone Docking Speaker 應用方塊圖



Holtek USB Audio 程式架構



Register Initial：暫存器初始化

Config I/O：I/O 初始化

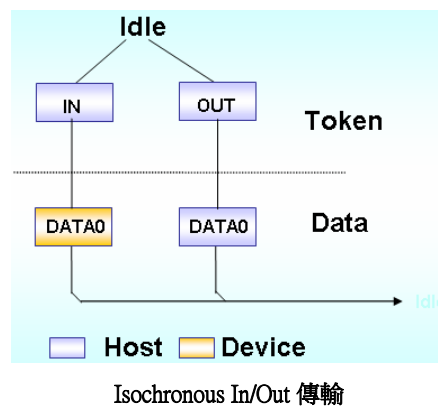
Config Timer：Timer 相關暫存器初始化

Config USB：USB 暫存器初始化，USB 暫存器設定完，USB Function 即會開始動作

USB Enumeration：此步驟為 USB 做列舉的動作

當 USB 完成列舉的動作後，有三個小模組 (Timer 中斷、USB Endpoint 中斷、Pooling Function) 是使用者可以自行修改並作運用的。

HT82A824R 之 USB Audio 使用 Full Speed Isochronous 傳輸模式，每個 Frame 為 1ms (USB 規範 $1ms \pm 500ns$)，根據不同的頻率會傳送不同大小的語音資料量，例如 48kHz/16-bit/2-CH 時，1ms 的資料量為 192 Byte，以此為例，盛群 USB Phone MCU 內部有兩塊 Ping-Pong 結構的 Buffer，各為 192 Bytes，總共為 384 Bytes，以 SOF (Start Of Frame) 做為切換兩個 Buffer 的控制信號。由 Isochronous In/Out 傳輸的圖可以發現 USB 的 Isochronous 傳輸不需要 ACK 信號，與 USB 的其他三種傳輸 Control/Interrupt/Bulk 是不同，也就是說，就算是資料有錯，也不會重傳。



HT82A824R 有一個 Isochronous Out 端點 (Endpoint)。Isochronous Out 是用來將 PC 的撥放媒體的語音資料經由 USB Port 傳給裝置，語音資料設定為 PCM 格式，假設是在 48kHz/16-bit/2-CH 情況下，一個 Frame 傳 192 Bytes 資料，所以此端點的 Ping-Pong Buffer 大小總共為 384 Bytes，在描述元 (Descriptor) 關於撥放語音格式的定義如下文字描述所示，在此描述元定義了撥放頻率、一個 Frame 的資料量、端點號碼與端點的方向等；HT82A824R 的 Endpoint Number 設定為 2；如下的描述語言為支援 48kHz/16-bit/2-CH 與 44.1kHz/16-bit/2-CH 兩種格式。

HT82A824R Isochronous 端點都是由硬體來實現，對於寫軟體的工程人員，不必耗費大量資源來處理喇叭的語音資料，可以大幅結省軟體開發的時間。如果客戶需要分析 USB Audio 的數位元資料，可經由 HT82A824R 內部硬體 DMA 的方式把資料放置在 RAM Bank 1~4，再由軟體分析 USB Audio 的資料內容。

```

format_type_descriptor:
    ;support 48K/44.1K
    DW 0240EH          ;descriptor type(CS_INTERFACE) , size of descriptor (add 44.1 KHz)
    DW 00102H          ;FormatType(FORMAT_TYPE_I) , descriptorSubType(FORMAT_TYPE)
    DW 00202H          ;SubFrameSize(2 byte per slot) , number of channel(2 channels)
    DW 00210H          ;SamFreqType(support 2 type) , BitSolution(16 bits)(add 44.1 KHz)
    DW 03F80H          ;Sample Frequency(48000 Hz)
    DW 000BBH          ;
    DW 03F44H          ;Sample Frequency(44100 Hz)
    DW 000ACH          ;
end_point_descriptor:
    DW 00509H          ;descriptor type(END_POINT) , size of descriptor
    DW 00902H          ;endpoint attributes(adaptive,isochronous) , endpoint2(out direction)
    DW 000C0H          ;maxPacketSize(192 bytes)
    DW 00001H          ;Refresh(0) , Interval(1ms)
    DW 03F00H          ;index string of this descriptor
    
```

Microsoft® WHQL Audio Requirement

要取得微軟的 USB Audio Device Logo，在 Audio 這個部份，必需符合下列兩個規範內容：

- 取樣頻率：USB Audio device 輸出的取樣頻率必需支援 44.1k 與 48k (或其整數倍頻率)
- 輸出的 Audio 品質必需符合如下的規格 (USB Speaker)

For premium desktop implementations:

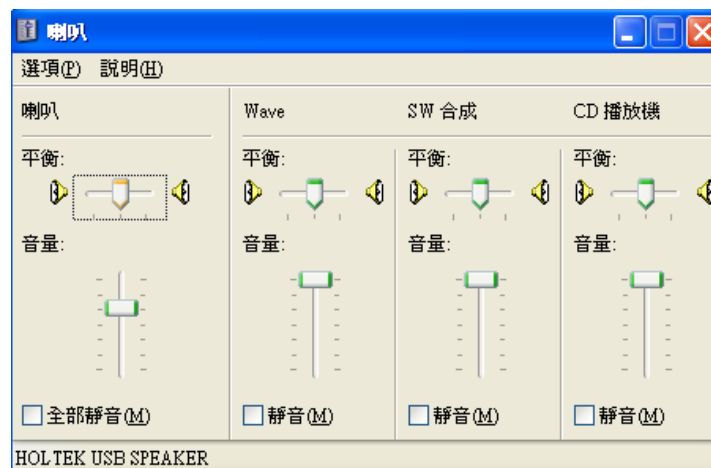
Device Type	Requirement	Value	Frequency range at 48kHz and above
Analog Speaker Output Jack (Example: 125mW into 8 Ohm load)	THD+N	$\geq 75\text{dB FS}$	[20Hz, 20kHz]
	Dynamic range with signal present	$\geq 90\text{dB FS A-weight}$	[20Hz, 20kHz]
	Magnitude Response	$\leq \pm 2.5\text{dB ripple}$ (.5dB peak to peak delta), 1dB at upper band edge, 3dB at lower band edge	[20Hz, 20kHz]
	Sampling frequency accuracy	0.02%	
	Line output cross-talk	$\geq 60\text{dB}$	[20Hz, 15kHz]
	Full scale output voltage	$\geq 1\text{Vrms}$	
	Dynamic range with signal present during system activity	$\geq 90\text{dB FS A-weight}$	[20Hz, 20kHz]
Interchannel phase delay	30 degrees or 12.5 microseconds, whichever is greater	[20Hz, 20kHz]	

注意事項：HT82A824R 不支援同時有 Speaker 與耳機插孔產品的 WHQL Audio 認證。支援耳機插孔輸出的產品測試 WHQL Audio 時需要測試 Audio Fidelity。其中 Render Power Transition Test 測試項目會 Fail 因而導致 WHQL 測試結果 fail。若產品不支援 jack 插孔輸出則無此測項。

USB音效裝置說明

音量控制

喇叭的音量控制：在微軟作業系統有一個音量控制的視窗，當使用者拉動此控制鍵時，USB 會透過端點 0 (Control Endpoint) 下一個 SET_CUR 的咨求函數來調整 (SET_CUR 咨求函數內容如下表格所示) 音量，PC 下過來的音量設定值會反應到 IC 的 Digital Volume Control 暫存器 (USVC)，因此就能改變音樂撥放時的聲音大小。



下表為喇叭的音量控制咨求函數，由 SET_CUR 的 wIndex 欄位來決定是要對喇叭還是麥克風做音量控制，wIndex=0x0200 表示要對喇叭做音量控制。

Offset	Field	Size	Value	Description
0	bmRequestType	1	0x21	
1	bRequest	1	0x01	SET_CUR
2	wValue	2	0x0200	DAC : VOLUME_CONTROL Master Channel
4	wIndex	2	0x0200 0x0600	D/A Feature Unit ID: 0x02(Lineout Lch and Rch Volume) Lower Byte : Audio Control Interface(0x00)
6	wLength	2	0x0002	Volume Control

SET_CUR 咨求函數內容

Offset	Field	Size	Value	Description
0	bMute	2	0xYYYY	The value is set by host

SET_CUR 咨求函數送出 2-Bytes 的資料

在對喇叭進行音量控制時，咨求函數後會跟隨 2-Bytes 的資料，這兩個 Bytes 資料就是音量大小的值，下表的喇叭音量控制的高位元組，是根據 IC 的音量控制暫存器所定義的設定範圍。此外，在匯流排列舉 (Bus Enumeration) 過程中，作業系統會以 SET_CUR 的咨求函數來設定喇叭的初始音量值，一般而言會設在中間值。

Volume Value	USB Audio Class Format
+6.0 dB	0x0C00
+5.5 dB	0x0B00
↓	↓
+0.5 dB	0x0100
+0.0 dB	0x0000
-0.5 dB	0xFF00
-1.0 dB	0xFE00
↓	↓
-12.0 dB	0xE800
-13.0 dB	0xE700
-14.0 dB	0xE600
↓	↓
-32.0 dB	0xC800

喇叭音量控制的高位元組(低位元元組皆為 00H)

Transaction	F	SETUP	ADDR	ENDP	T	D	TP	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
8108	S	0xB4	7	0	0	H->D	C	I	0x01	0x0200	0x0200	2	0x4B	00027.0482 2730
Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC6	EOP	Idle	Time Stamp				
16908	-->	S	00000001	0xB4	7	0	0x16	233.330 ns	100.000 ns	00027.0482 2730				
Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp					
16909	-->	S	00000001	0xC3	21	01 00 02 00 02 00	0xAD01	233.330 ns	383.320 ns	00027.0482 2910				
Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp							
16910	<--	S	00000001	0x4B	233.330 ns	355.450 μs	00027.0482 3427							
Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp					
8132	S	0x87	7	0	1	00 00	0x4B	261.000 μs	00027.0485 2254					
Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp					
8161	S	0x96	7	0	1	0 bytes	0x4B	731.467 μs	00027.0487 2914					

喇叭音量設定 USB 封包圖

下表為喇叭的靜音控制咨求函數，由 wIndex 欄位來決定是要對喇叭還是麥克風作音量控制，wIndex=0x0200 表示要對喇叭作靜音控制，接下來主機會送 1-byte 資料，如果值是 0x01 代表要靜音。

Offset	Field	Size	Value	Description
0	bmRequestType	1	0x21	
1	BRequest	1	0x01	SET_CUR
2	wValue	2	0x0100	MUTE_CONTROLCHANNEL_0
4	wIndex	2	0x0200 0x0600	Mute for Lineout Volume I interface 0 Mute for MIC Recording Volume I interface 0
6	wLength	2	0x0001	

靜音控制的咨求函數

Offset	Field	Size	Value	Description
0	bMute	1	0x01 0x00	TRUE (靜音) FALSE

靜音控制咨求函數送出 1-Byte 的資料

在匯流排列舉過程中，作業系統會先來讀取 Audio Device 音量的參數 (最大音量、最小音量、變化單位元)，當作業系統從裝置讀取這些參數後，才能對裝置做正確音量控制，讀取關於音量咨求函數與裝置的回傳值說明如下表;從回傳值的表格中可以得知，喇叭的音量最大為+6 dB，喇叭音量最小為-32dB，以上所談到的這些值，可以依不同音效裝置的需求而改變。

Offset	Field	Size	Value	Description
0	bmRequestType	1	0xA1	
1	bRequest	1	0x81 0x82 0x83 0x84	GET_CUR GET_MIN GET_MAX GET_RES
2	wValue	2	0x0200	VOLUME_CONTROLMaster CH
4	wIndex	2	0x0200 0x0600	Lineout Volume I interface 0 MIC Recording Volume I interface 0 Lower Byte : Audio Control Interface (0x00)
6	wLength	2	0x0002	Volume Control

讀取音量參數的咨求函數

bRequest	wValue	wIndex	wVolume	說明
0x81	0x0200	0x0200	0xYYYY	讀取裝置目前的喇叭音量
0x82	0x0200	0x0200	0xE000	讀取裝置目前的喇叭音量的最小音量(-32 dB) .
0x83	0x0200	0x0200	0x0C00	讀取裝置目前的喇叭音量的最大音量(+6 dB) .
0x84	0x0200	0x0200	0x0100	讀取裝置的喇叭音量變化單位元 (1)

讀取喇叭音量參數咨求函數的回傳值範例

Transaction	F	SETUP	ADDR	ENDP	T	D	TP	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
6748	S	0xB4	7	0	0	D->H	C	I	0x82	0x0200	0x0200	2	0x4B	00026.7572.2352
Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp				
13942	-->	S	00000001	0xB4	7	0	0x16	233.330 ns	100.000 ns	00026.7572.2352				
Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp					
13943	-->	S	00000001	0xC3	A1 B2 00 02 00 02 02 00	0x0F14	233.330 ns	349.990 ns	00026.7572.2532					
Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp							
13944	<--	S	00000001	0x4B	233.330 ns	457.400 μs	00026.7572.3047							
Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp					
6799	S	0x96	7	0	1	00 B0	0x4B	13.917 μs	00026.7576.0491					
Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp					
6800	S	0x87	7	0	1	0 bytes	0x4B	00026.7576.1326						

讀取喇叭的最小音量之 USB 封包圖

Transaction	F	SETUP	ADDR	ENDP	T	D	TP	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
6854	S	0xB4	7	0	0	D->H	C	I	0x84	0x0200	0x0200	2	0x4B	00026.7588.2182
Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp				
14162	-->	S	00000001	0xB4	7	0	0x16	233.330 ns	100.000 ns	00026.7588.2182				
Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp					
14163	-->	S	00000001	0xC3	A1 84 00 02 00 02 02 00	0x9614	233.330 ns	383.330 ns	00026.7588.2362					
Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp							
14164	<--	S	00000001	0x4B	233.330 ns	455.867 μs	00026.7588.2679							
Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp					
6905	S	0x96	7	0	1	00 01	0x4B	13.833 μs	00026.7592.0231					
Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp					
6906	S	0x87	7	0	1	0 bytes	0x4B	518.317 μs	00026.7592.1061					

讀取喇叭的音量變化單位元之 USB 封包圖

由於 HT82A824R 支援 48K / 44.1K 兩種頻率，因此當 PC 在撥放不同的取樣頻率的音源檔內容時，會下一個設定頻率 SET_FREQ 的咨求函數來改變 Device 的輸出頻率設定，在 HT82A824R 的 MODE_CTRL 暫存器 bit3 可以設定輸出頻率是 48K 還是 44.1K。

Offset	Field	Size	Value	Description
0	bmRequestType	1	0x22	
1	bRequest	1	0x01	SET_FREQ
2	wValue	2	0x0100	DAC : VOLUME_CONTROL/Master Channel
4	wIndex	2	0x0200	D/A Feature Unit ID : 0x02
6	wLength	2	0x0003	Volume Control

SET_FREQ 咨求函數內容

Offset	Field	Size	Value	Description
0	bFreq	3	0xYYYYYY	The value is set by host 48K : 0x00BB80 44.1K : 0x00AC44

SET_FREQ 咨求函數送出 3-Bytes 的資料

程式範例:

```

(1)
;support 48K/44.1K ,2007-09-10
Request_TYPE22: ;22 01 00 01 02 00 03 00
    clr wdt
    ;set report
    MOV A,FIFO_REQUEST
    XOR A,SET_CUR
    SNZ Z
    JMP Request_TYPE22_End
    MOV A,FIFO_wIndexL
    XOR A,02H ;Endpoint 2
    SNZ Z
    JMP Request_TYPE22_End
    MOV A,FIFO_wLengthL
    XOR A,03H ;3-bytes data
    SNZ Z
    JMP Request_TYPE22_End
;Setting Frequency 48K or 44.1K
    mov a,22h
    mov nCmdIndex1,a
    jmp USB_EP0_ISR_END
Request_TYPE22_End:
    JMP SendStall0

(2)
Set_Sampling_Frequency:
;00BB80H=48k
;00AC44H=44.1K
    MOV A,FIFO_out1
    XOR A,80H
    SZ Z
    JMP Freq_48K_BYTE_1_2
    MOV A,FIFO_out1
    XOR A,44H
    SZ Z
    JMP Freq_44_1K_BYTE_1_2
    JMP ;Set_Sampling_Frequency_End

Freq_48K_BYTE_1_2:
    MOV A,FIFO_out2
    XOR A,0BBH
    SNZ Z
    JMP Set_Sampling_Frequency_End
    MOV A,FIFO_out3
    XOR A,00H
    SNZ Z
    JMP USB_EP0_OUT_TOKEN_End
    ;Set Sampling Frequency=48KHz
    CLR MODE_CTRL.FREQ
    JMP ;Set_Sampling_Frequency_End
    
```

```

Freq_44_1K_BYTE_1_2:
    MOV A,FIFO_out2
    XOR A,0ACH
    SNZ Z
    JMP          ;Set_Sampling_Frequency_End
    MOV A,FIFO_out3
    XOR A,00H
    SNZ Z
    JMP USB_EP0_OUT_TOKEN_End
                                ;Set Sampling Frequency=44.1KHz
    SET MODE_CTRL.FREQ
    JMP          ;Set_Sampling_Frequency_End
    
```

要改變電腦 USB 音效裝置的音量大小，除了直接用 Windows 上的內建應用程式音量控制改變音量外，另外一種方式也可以用 USB 規範中的 HID (Human Interface Device)，直接在使用者的裝置端以按鍵的方式，直接改變音量，要用此方法在裝置端改變音量，必須撰寫正確的 HID 類別與報告描述語言，以下就以喇叭的音量控制來作說明。首先在如下的 HID 類別 Interface2_descriptor 得知，是使用 ReportID 1 是用來傳輸 Media Key，也就是 Volume Increment / Decrement、Mute 等功能；而 ReportID 3 是用來傳輸其他類型的資料，也就是使用者自定的功能。當有使用 Report ID 時的報告描述元的長度為 DW 0003FH (63 個 Bytes)。HID_end_point_descriptor 是在 Interface2 宣告一個端點描述元，此端點描述為輸入端點 1，而主機來詢問我們傳送資料的時間間隔為 DW 03F30H 的 30H=48ms。由於使用同一個輸入端點 1，所以當要傳送 Media Key 或其他類型的資料時，裝置必須用不同的 Report ID 來區分他們，這樣才會知道裝置送什麼樣的資料給 PC，這就是使用 Report ID 的原因。

描述語言範例:

```

Interface2_descriptor:
    HID_class:
        DW 00409H ;INTERFACE descriptor , Size of this descriptor
        DW 00002H ;Index of this string , index of this interface
        DW 00301H ;HID , 1 endpoint
        DW 00000H ;Unused , Non-Boot Device
        DW 03F00H ;null string
    HID_Desc:
        DW 02109H ;HID , Size of this descriptor
        DW 00110H ;HID spec rev #1.10
        DW 00100H ;bNumDescriptor , bCountryCode
        DW 03F22H ;Report Descriptor
                                ;Use ReportID , Report ID 1 = Volume HID control
                                ;Report ID 3 = Transform Other Data
    IF UseReportID
        DW 0003FH ;63 bytes
    ELSE
        DW 0001FH ;31 bytes
    ENDIF
    
```

```

HID_end_point_descriptor:
    DW 00507H ;Endpoint descriptor , Length of this descriptor
    DW 00381H ;Interrupt , Endpoint 1 In direction
    DW 00008H ;wMaxPacketSize = 8 Bytes
    DW 03F30H ;48ms Interval
end_config_desc_table:
    
```

hid_report_desc_table 的標籤是 Media Key 的完整報告描述元。一個 Report ID 佔用一個 Byte，現在來說明 Media Key 的報告描述元；在 USB HID Usage Tables 規範中可以查詢，Volume Increment 的 Usage ID 是 E9、Volume Decrement 的 Usage ID 是 EA 及 Mute 的 Usage ID 是 E2，而前面的 09 代表後面的 Usage ID 是一個 Byte，如果後面的 Usage ID 是 2 個 Bytes，則 Usage ID 前面的 Byte 為 0A。值 DW 00175H 所表示的意思是所輸入的 Media Key 佔多少位元，以 00175H 為例子，它表示一個 Key 佔用一個位元。第二個值為 DW 00295H 表示的意思是上面使用到幾個 Bit，由於上面使用到 Volume Increment / Decrement 二種功能，所以在這裏為 00295H，佔用二個 Bit。由上說明我們使用 Media Key，目前使用了三個鍵，而這三個鍵分別為 Volume Increment / Decrement / Mute，這三個鍵在一個 Byte 裏佔了三個 Bit，其功能與位元對應關係如下表所示。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Unused	Unused	Unused	Unused	Unused	Mute	Volume Decrement	Volume Increment

而且先宣告的功能是先佔用低的位元。以我們例子來看，上面先宣告 Volume Increment，所以它在位元 0，而 Volume Decrement 則在位元 1，以此類推。此時，有五個位元未使用到 (Unused) 也必須將報告描述元完整描述，構成一完整位元組。

我們在傳送 Media Key 的第一個 Byte 為 Report ID=1，即 01H，而第二個 Byte 則為我們在報告描述元所描述的按鍵功能，舉例來說，若為靜音時，則第二個 Byte 則為 04H，若為音量增加鍵則第二個 Byte 為 01H，若為音量減少鍵則第二個 Byte 為 02H。

描述語言範例:

```

hid_report_desc_table:
    DW 00C05H          ;//Usage Page(Consumer)
    DW 00109H          ;//Usage Page(Consumer Control)
    DW 001A1H          ;//Collection(Application)

    IF UseReportID
    DW 00185H          ;//Report_ID(01)
    ENDIF
    DW 00015H          ;//Logic Minimum(0)
    DW 00125H          ;//Logic Maximum(1)
    DW 03F09H          ;//Usage(Volume Increment)
    DW 03FE9H
    DW 03F09H          ;//Usage(Volume Decrement)
    DW 03FEAH
    DW 00175H          ;//Report Size(1) : Data Length (1)bit
    DW 00295H          ;//Report Count(2): Number of Data(INC,DEC)
    DW 02A81H          ;//Input(Data,Variable,Absolute,No Wrap,No Preferred)
    DW 03F09H          ;//Usage(Mute)
    DW 03FE2H
    DW 00195H          ;//Report Count(1)
    DW 02E81H          ;//Input(Data,Variable,Relative,No Wrap,No preferred)
    DW 00595H          ;//Report Count(5)
    DW 00181H          ;//Input(Constant)
    DW 03FC0H          ;//End Collection
end_hid_report_desc_table:
    
```

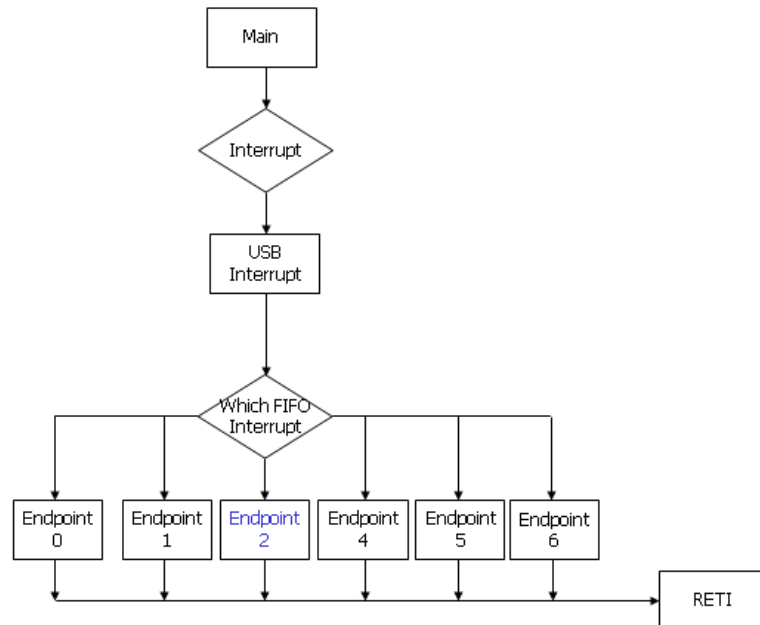
盛群USB Audio韌體技術

盛群 USB Audio MCU 的 SIE 是盛群自行設計的,其設計理念除了相容性外,也考慮到讓 USB 韌體工程師更容易撰寫 USB 底層的程式,現就針對比較重要的部份以文字說明配合流程圖如下:

- USB 中斷處理流程

當 USB SIE 的某一個端點有資料需傳輸時,會產生 USB 中斷,進入中斷副程式後再由暫存器判斷是由那個端點產生的中斷,然後再進入相對應的程式碼處理。

說明: Endpoint 2 為 Isochronous out 端點



程式範例:

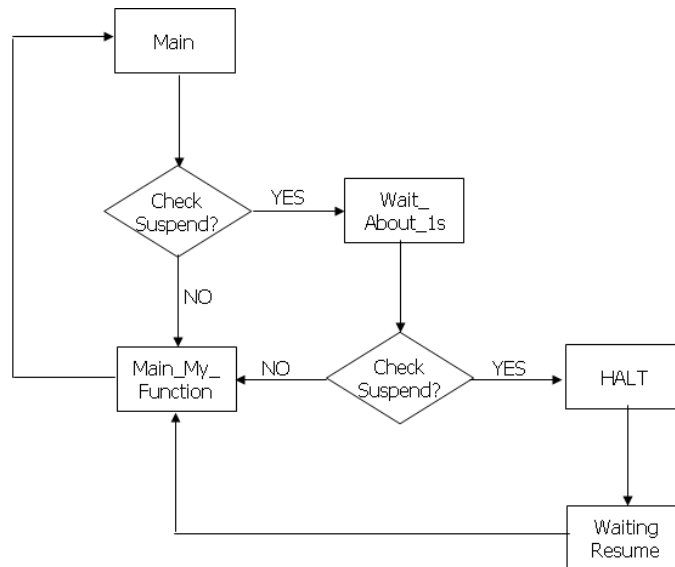
Check_Access_FIFO:

```

    clr wdt
    SZ     USR.@USR_EP0F
    JMP    USB_EP0_ISR
    SZ     USR.@USR_EP1F
    JMP    USB_EP1_ISR
    SZ     USR.@USR_EP2F
    JMP    USB_EP2_ISR
    SZ     USR.@USR_EP4F
    JMP    USB_EP4_ISR
    SZ     USR.@USR_EP5F
    JMP    USB_EP5_ISR
    SZ     USR.@USR_EP6F
    JMP    USB_EP6_ISR
    JMP    USB_ISR_END
  
```

- 主程式

在主程式中，會檢查 SUSPEND (懸置模式) 旗標，如果 SUSPEND 旗標被設為“1”，會先延遲 1 秒鐘後，再檢查一次 SUSPEND 旗標，如果 SUSPEND 旗標仍為“1”，把相關旗標設定後，進入 HALT，等待 PC 下 Resume 信號過來；如果延遲 1 秒後，SUSPEND 旗標已經清除，則繼續執行主程式。



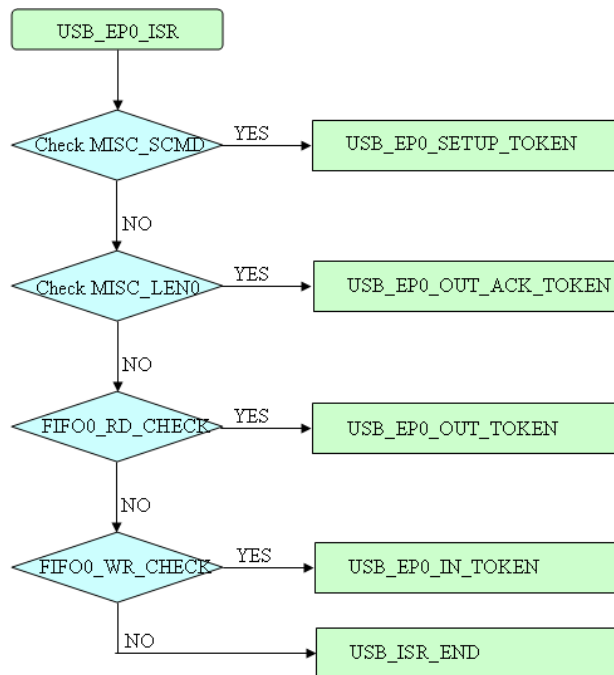
程式範例:

```

Start:
    call System_Initial
Main:
    SNZ USC.@USC_SUSP        ;check SUSPEND ?
    JMP Main_My_Function
    call wait_about_1s
    SNZ USC.@USC_SUSP
    JMP Main_My_Function
ToSuspend_again:
    SNZ USC.@USC_SUSP        ;check SUSPEND ?
    JMP Main_My_Function
    clr wdt
    clr TMR1C.4
    clr USB_LED_ON
    clr UCC.@UCC_USBCKEN
    
```

- 端點 0 主流程

進入 USB 中斷後，判斷是端點 0 (Control Pipe) 後，進入端點 0 的副程式，接下來判斷是 SETUP Token、OUT ACK(Zero-Length)、IN Token(Control Read)或 OUT Token(Control Write)。

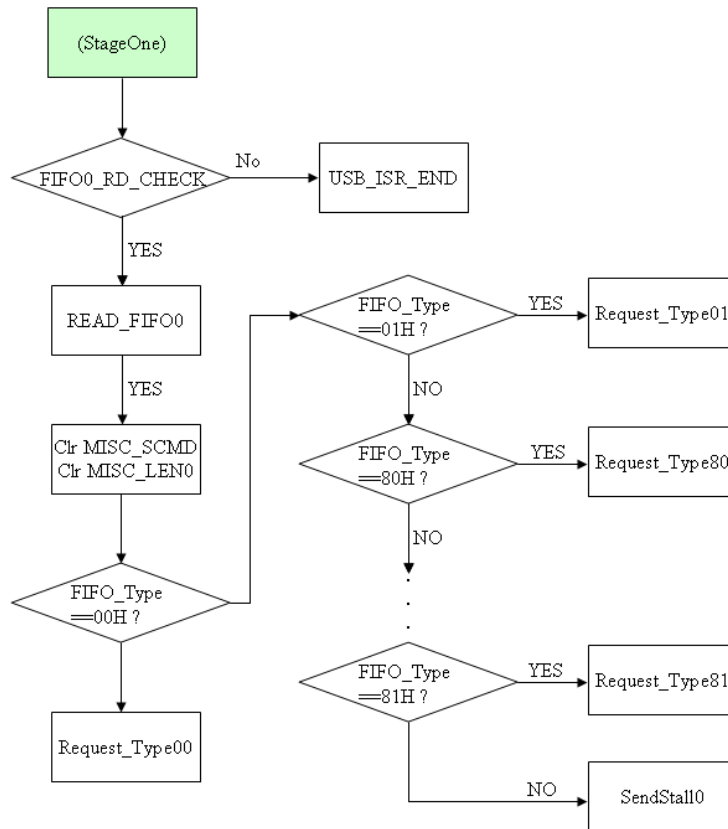


程式範例:

```

USB_EP0_ISR:
    SZ     MISC.@MISC_SCMD           ;check setup token
    JMP    USB_EP0_SETUP_TOKEN
    SZ     MISC.@MISC_LEN0          ;check out ack token
    JMP    USB_EP0_OUT_ACK_TOKEN
    CALL  FIFO0_RD_CHECK
    SZ     bFlag_FIFO_Ready
    JMP    USB_EP0_OUT_TOKEN
    CALL  FIFO0_WR_CHECK
    SZ     bFlag_FIFO_Ready
    JMP    USB_EP0_IN_TOKEN         ;else is in token
    CLR   USR.@USR_EP0F            ;Fix OHCI Volume
    JMP   USB_EP0_ISR_END
    
```

- 如果是端點 0 中斷而且判斷是 SETUP Token 時，則開始剖析 Setup Command (StageOne)，Setup Command 共有 8 Bytes，先解析第一個 Byte，也就是先解析 Request_Type，再跳至對應的函數去執行，若沒有支援的 command 則跳至 SendStall0。



程式範例:

StageOne:

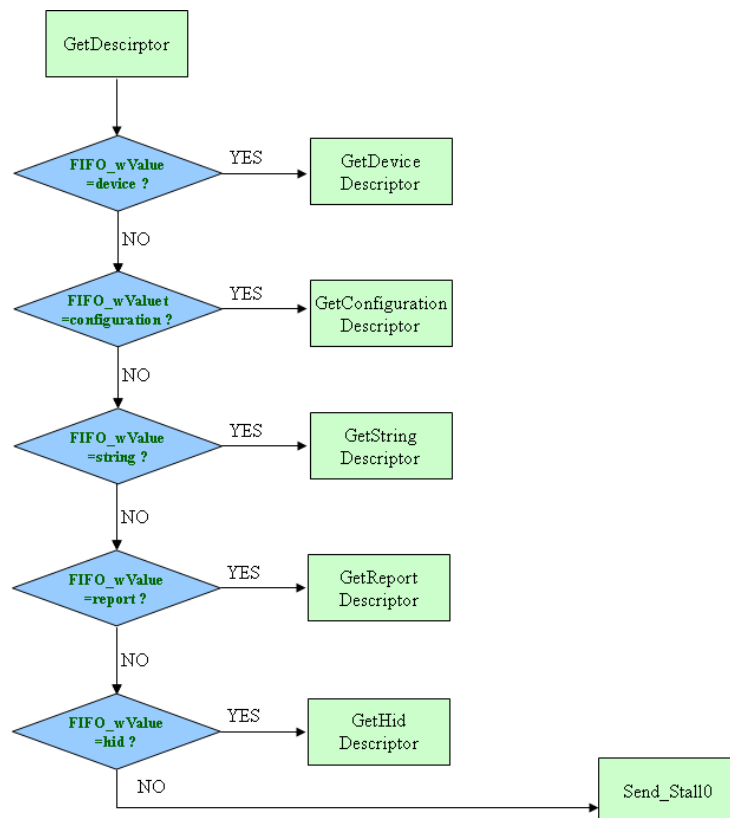
```

CALL FIFO0_RD_CHECK
SNZ bFlag_FIFO_Ready
JMP StageOne_End ;the EP0 FIFO RD is not ready 沒有data進來
CALL Read_FIFO0 ;Read EP0 Command
clr MISC.@MISC_SCMD
clr MISC.@MISC_LEN0
CLR USR.@USR_EP0F ;Fix OHCI Volume
Clr wdt
nop
MOV A,FIFO_TYPE
XOR A,00H
SZ Z ;FIFO_TYPE=00H
JMP Request_Type00
MOV A,FIFO_TYPE
XOR A,01H
SZ Z ;FIFO_TYPE=01H
JMP Request_Type01
MOV A,FIFO_TYPE
XOR A,02H
SZ Z ;FIFO_TYPE=02H
JMP Request_Type02
MOV A,FIFO_TYPE
XOR A,80H
SZ Z ;FIFO_TYPE=80H
    
```

```

JMP Request_Type80
MOV A,FIFO_TYPE
XOR A,81H
SZ Z ;FIFO_TYPE=81H
JMP Request_Type81
MOV A,FIFO_TYPE
XOR A,82H
SZ Z ;FIFO_TYPE=82H
JMP Request_Type82
    
```

- 在 Bus Enumeration 的過程，主機會來讀取裝置的描述元，裝置再根據 PC 所要求不同型態的描述元傳給主機。



程式範例:

```

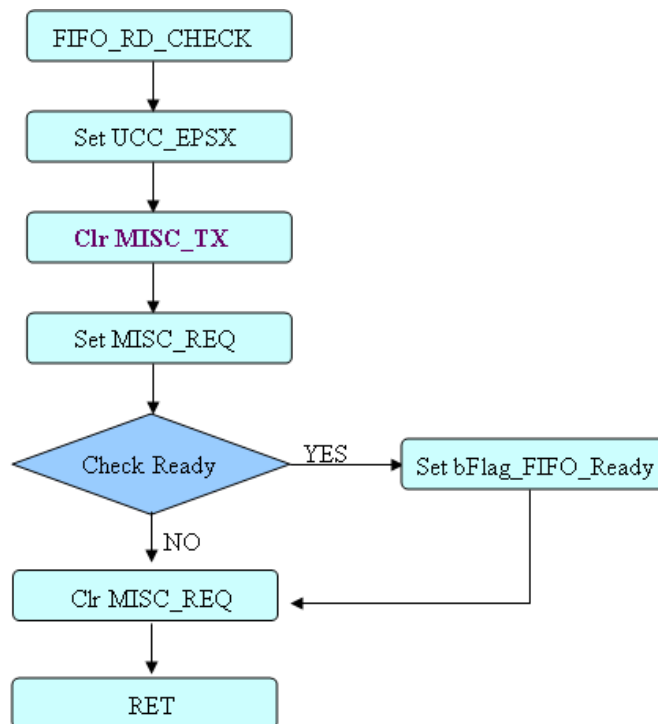
GetDescriptor:
    clr wdt
    CLR bFlag_RD_HTable
    CLR bFlag_wait_control_out
    MOV A,FIFO_WvalueH ;80 06 00 01
    XOR A,device
    SZ Z
    JMP GetDeviceDescriptor
    MOV A,FIFO_WvalueH ;80 06 00 02
    XOR A,configuration
    SZ Z
    JMP GetConfigurationDescriptor
    MOV A,FIFO_WvalueH ;80 06 00 03
    XOR A,string
    SZ Z
    JMP GetStringDescriptor
    ;-----
    
```

```

;Then test for HID class Descriptor
;-----
MOV A,FIFO_WvalueH      ;81 06 00 22
XOR A,report
SZ Z
JMP GetReportDescriptor

MOV A,FIFO_WvalueH      ;81 06 00 21
XOR A,HID
SZ Z
JMP GetHIDDescriptor
JMP SendStall0          ;can't parser
    
```

- 當有資料要與主機做資料傳輸時(控制型端點與中斷型端點)，Firmware 需先檢查 FIFO 是否已經準備好，然後才能接著做讀取或寫入 FIFO 的動作。



程式範例:

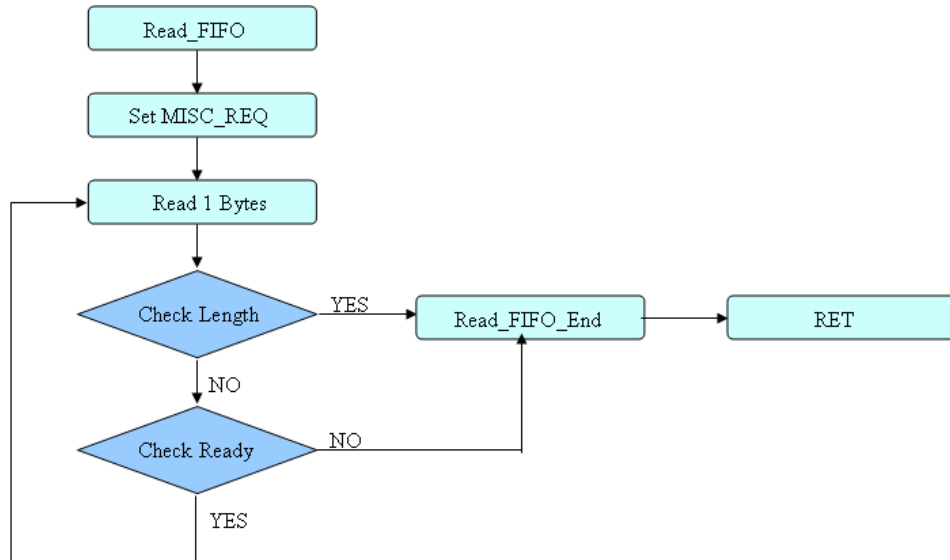
```

FIFO_CHECK:
    Clr wdt
    MOV FIFO_TEMP,A
    MOV A,USB_MISC
    MOV MP1,A
    MOV A,R1
    AND A,11111000b
    OR A,FIFO_TEMP
    MOV R1,A
    CALL Delay_3us
    SET R1.@MISC_REQ          ;set request
    CALL Delay_28us
    SET bFlag_FIFO_Ready
    SNZ R1.@MISC_Ready
    CLR bFlag_FIFO_Ready     ;if MISC.Ready = 1 -> bFlag_FIFO_Ready = 1
    SET bFlag_FIFO_LEN0
    SNZ R1.@MISC_LEN0
    
```

```

CLR bFlag_FIFO_LEN0
clr MISC.@MISC_REQ
clr wdt
RET

```

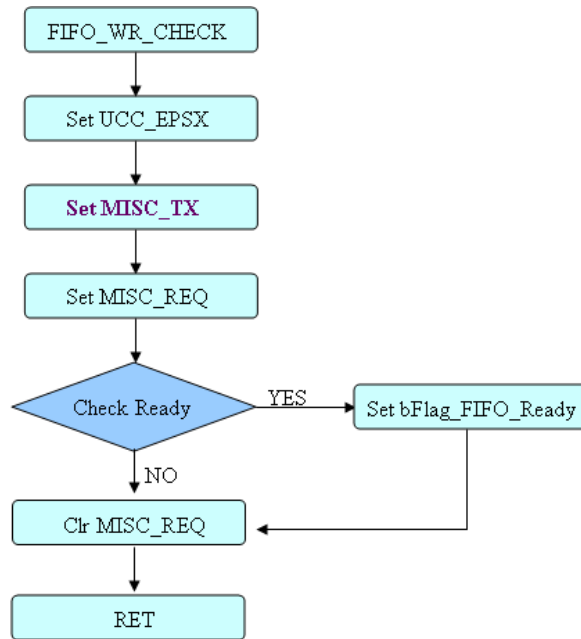


程式範例:

```

Read_FIFO_Loop:
MOV A,FIFO_TEMP
MOV MP1,A
MOV A,R1
MOV R0,A
INC FIFO_SendLen
INC MP0
MOV A,FIFO_SIZE
XOR A,FIFO_SendLen
SZ Z ;1=FIFO_SIZE=FIFO_SendLen
JMP Read_FIFO_End
MOV A,USB_MISC
MOV MP1,A
CALL Delay_28us
SZ R1.@MISC_Ready
JMP Read_FIFO_LOOP
JMP Read_FIFO_EndRead FIFO Flow

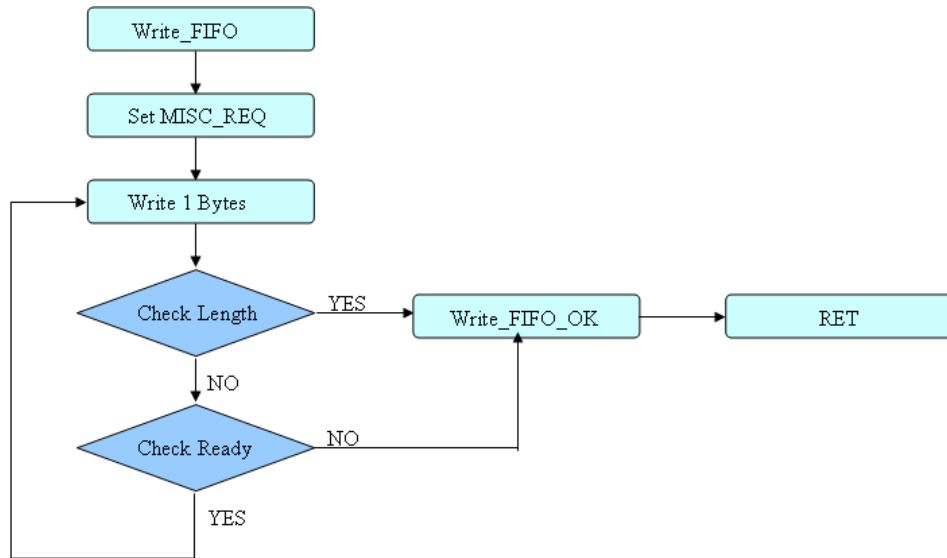
```



程式範例:

```

FIFO_CHECK:
    clr wdt
    MOV FIFO_TEMP,A
    MOV A,USB_MISC
    MOV MP1,A
    MOV A,R1
    AND A,11111000b
    OR A,FIFO_TEMP
    MOV R1,A
    CALL Delay_3us
    SET R1.@MISC_REQ          ;set request
    CALL Delay_28us
    SET bFlag_FIFO_Ready
    SNZ R1.@MISC_Ready
    CLR bFlag_FIFO_Ready     ;if MISC.Ready = 1 -> bFlag_FIFO_Ready = 1
    SET bFlag_FIFO_LEN0
    SNZ R1.@MISC_LEN0
    CLR bFlag_FIFO_LEN0
    clr MISC.@MISC_REQ
    clr wdt
    RETWrite FIFO Check Flow
    
```



程式範例:

```

Write_FIFO_Loop:
  clr wdt
  MOV A,FIFO_SendLen
  XOR A,00H
  SZ Z
  JMP Write_FIFO_End
  MOV A,FIFO_TEMP
  MOV MP1,A
  MOV A,R0
  MOV R1,A
  DEC FIFO_SendLen
  MOV A,FIFO_SendLen
  XOR A,00H
  SZ Z
  JMP Write_FIFO_End      ;FIFO_SendLen=0 代表傳完了
  INC MP0
  MOV A,USB_MISC
  MOV MP1,A
  call Delay_28us
  SZ R1.@MISC_Ready
  JMP Write_FIFO_Loop
Write_FIFO_End:
  Clr wdt
  JMP Write_FIFO_OKwrite FIFO Flow
  
```

版本記錄

版本：V1.10

修改日期：2011 年 08 月 29 日

修改內容：新增第五頁 Microsoft[®] WHQL Audio Requirement 單元之表格內容的“注意事項”說明。