

Acute TravelBus Analyzer Software development kit (SDK) Programming guide

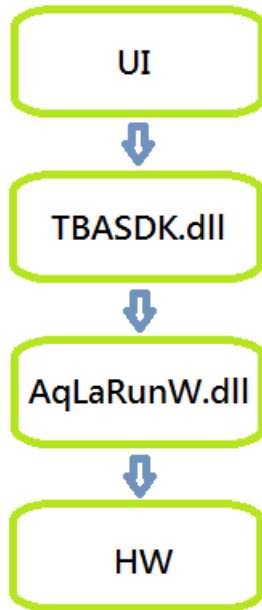
Version: 1.1

Publish: 2016/04/11

內容

LPSDK_TRIG 結構簡介	2
TBASDK 函式說明	3
INT ulaSDKInit()	3
BOOL ulaSDKSelectDeviceIdx(int iDeviceIndex)	3
BOOL ulaSDKSelectDevice(char* szSerialNum)	3
BOOL ulaSDKGetSerialNumber(char *szSerialNum, int iSize).....	4
BOOL ulaSDKSetHwInfo(int iIndex, LPVOID lpv)	4
BOOL ulaSDKSetChTrigger(LPSDK_TRIG lpSDKtr, int iLevel, int iCh, int iTrig, int iCondLogic)	5
BOOL ulaSDKClearTrigger(LPSDK_TRIG lpSDKtr).....	6
BOOL ulaSDKSetRamSize (int iRamSize_MB).....	6
INT ulaSDKGetMaxRamSize().....	6
BOOL ulaSDKThreshold(int iPod, int iMilliVolt)	7
BOOL ulaSDKCapture(LPSDK_TRIG lpSDKtr)	7
BOOL ulaSDKIsCaptureReady()	8
BOOL ulaSDKStopCapture()	8
BOOL ulaSDKGetTimingInfo(__int64& i64StartPos, __int64&i64EndPos, __int64& i64TrigPos)	8
BOOL ulaSDKGetDataAt (__int64& pi64Data, __int64 i64TransPos)	9
BOOL ulaSDKGetNextData (__int64& pi64Data, __int64& i64StartTimestamp)	10
BOOL ulaSDKSaveAsTBWFile(char* szFilePathName)	10
BOOL ulaSDKClose ();.....	11
INT ulaSDKGetLastError().....	11

TBASDK 架構與流程



TbaSDK 建立的用意為簡化使用者設定參數的流程，將複雜的結構簡化成容易理解及設定的參數後，再將設定好的內容傳入硬體中。

LPSDK_TRIG 結構簡介

`typedef struct _SDKTRIG`

```

{
    int iFlag;           //觸發旗標(目前未使用)
    int iDelay;          //延遲觸發的延遲時間(功能目前尚未支援)
    int iWidth;          //觸發寬度(功能目前尚未支援)
    int iPassCount;      //觸發忽略次數
    int iFreq;           //擷取頻率(Hz)
    int iFreqHi;         //保留位，填入 0
    int iExtClk;         //使用外部頻率(功能目前尚未支援)
    int iTrPos;          //觸發游標的位置，輸入範圍 0 到 99，單位: %
    int lpiCont[16];     //使用者不需手動設定此欄位
    BYTE lpbTrigData[1024]; //使用者不需手動設定此欄位
} SDK_TRIG, FAR *LPSDK_TRIG;
  
```

TBASDK 函式說明

INT ulaSDKInit()

功能：

尋找並啟動目前接在此電腦上的所有二合一分析儀，Ex：TravelBus 1016E, 1016B, 1016B+

回傳值：

如果回傳 False 值則代表啟動失敗。若成功時將回傳總共啟動的機器數量。

BOOL ulaSDKSelectDeviceIdx(int iDeviceIndex)

功能：

當電腦上有多台 TBA 時可以透過此函式選擇將指令傳送給指定 USB 排序的機器。

參數：

iDeviceIndex [in]：

Type：int

指定啟動 TBA 的 USB 排序。

回傳值：

如果回傳值為 True，代表選擇成功。若找不到指定順序的機器時則會回傳 False。

備註：

各台 TBA 皆可獨立運作，使用時須分別設定各自的運作模式及參數。

Ex:

```
ulaSDKSelectDeviceIdx (0);
```

```
ulaSDKSetRamSize(100); //將 USB 排序為 0 機器的記憶深度設為 100MB
```

```
ulaSDKSelectDeviceIdx (1);
```

```
ulaSDKSetRamSize(200); //將 USB 排序為 1 機器的記憶深度設為 200MB
```

BOOL ulaSDKSelectDevice(char* szSerialNum)

功能：

當電腦上有多台 TBA 時可以透過此函式選擇將指令傳送給指定序號的機器。

參數：

szSerialNum[in]：

Type：char*

指定啟動的 TBA 序號。

回傳值：

如果回傳值為 True，代表選擇成功。若找不到指定序號的機器時則會回傳 False。

備註：

各台 TBA 皆可獨立運作，使用時須分別設定各自的運作模式及參數。

Ex:

```
ulaSDKSelectDevice ("TBB10160001");
ulaSDKSetRamSize(100); //將序號為 TBB10160001 機器的記憶深度設為 100MB
ulaSDKSelectDevice ("TBB10160002");
ulaSDKSetRamSize(200); //將序號為 TBB10160002 機器的記憶深度設為 200MB
```

BOOL ulaSDKGetSerialNumber(char *szSerialNum, int iSize)

功能：

取得 TBA 的產品序號。

參數：

szSerialNum[in/out]：

Type：char*

傳入的 Buffer 至少需要 35 Bytes 的長度。

iSize[in]：

Type：int

傳入 Buffer 的長度。

回傳值：

回傳 True 表示序號取得成功，回傳 False 表示序號取得失敗。

執行成功時會回傳該機器的產品序號，Ex：TBB10161234

BOOL ulaSDKSetHwInfo(int ilIndex, LPVOID lpv)

功能：

設定 TBA 運作時的硬體模式，包含取樣率、使用通道及觸發種類。

參數：

ilIndex [in]

Type：int

固定填入 SET_TBA_MODE = 5 設定邏輯分析儀的使用模式

lpv [in]:

Type：LPVOID

根據設定 Index 的不同有不同的形態，填入 SET_TL_MODE 時需傳入存有 H/W Mode 資訊的

int *型態指標以選擇硬體運作的模式。

Index	頻率	可用通道
TBA_HWMODE_LA200M	1Hz - 200MHz	16

圖一、硬體參數設定資料表

回傳值：

如果回傳值為 True，代表模式設定成功。如果回傳 False 值則代表設定失敗。

備註：

EX：希望使用 Sample Rate 200M 總共 16 個通道的模式

```
int iHWMode = TBA_HWMODE_LA200M;

if (!ulaSDKSetHwInfo(SET_TL_MODE, &iHWMode))

    AfxMessageBox("Hw Info Set Error!");
```

註：目前僅支援 TBA_HWMODE_LA200M 模式

BOOL ulaSDKSetChTrigger(LPSDK_TRIG lpSDKTr, int iLevel, int iCh, int iTrig, int iCondLogic)

功能：

將設定的觸發通道及狀態等觸發條件寫入 LPSDK_TRIG 結構中

參數：

lpSDKTr [in, out]

Type：LPSDK_TRIG

參數儲存的 structure，藉由此函式設定觸發條件後傳入 ulaSDKCapture 函式

iLevel / iCh / iTrig [in]：

Type：int

觸發階層及通道選擇，iTrig 可填入的參數如下：

數值	說明
#define LA_TRIG_DONT_CARE 0x08	設定為忽略此通道的狀態。
#define LA_TRIG_LOW 0x00	設定觸發條件為此通道收到 High 訊號
#define LA_TRIG_HIGH 0x06	設定觸發條件為此通道收到 Low 訊號
#define LA_TRIG_RISING 0x04	設定觸發條件為此通道收到正緣變化訊號
#define LA_TRIG_FALLING 0x02	設定觸發條件為此通道收到負緣變化訊號
#define LA_TRIG_CHANGE 0x0A	設定觸發條件為此通道收到正緣或負緣的變化訊號

iCondLogic [in]：

Type：int

數值	說明
#define TR_NEXT 0x00	兩觸發條件必須連續達成，中間不可以有其他不同訊號
#define TR_NEXTIF 0x01	兩觸發條件可以分開先後達成
#define TR_TRIGGER 0x02	觸發條件已全部設定完畢，啟動觸發

回傳值：

如果回傳值為 True，代表設定成功。如果回傳 False 值則代表執行此函式產生錯誤。

備註：

EX:設定第 2 個觸發參數的通道 3 為 High

```
iLevel = 2,

iCh = 3,

iTrig = LA_TRIG_HIGH
```

BOOL ulaSDKClearTrigger(LPSDK_TRIG lpSDKtr)

功能：

初始化觸發參數設置，可用來還原設定有誤的觸發參數設定。

參數：

lpSDKtr[in, out]：

Type：LPSDK_TRIG

觸發參數儲存的 Structure

回傳值：

如果回傳值為 **True**，代表觸發設定清除成功。如果回傳值為 **False**，代表觸發設定清除失敗。

備註：

使用這個函式時會將所有觸發設定還原到預設值。

還原的內容如下：

```
lpSDKtr->iFreq      = 200000000;    //採樣頻率預設200MHz
lpSDKtr->iFreqHi     = 0;             //採樣頻率預設200MHz
lpSDKtr->iTrPos       = 50;           //Trig Pos 設為50%記憶體位置
                                   //不可設為 0，設 0 的話觸發點會出現邊界上
```

BOOL ulaSDKSetRamSize (int iRamSize_MB)

功能：

設定 TBA 擷取資料所使用的記憶體，不可超過電腦最大可用的記憶體量。與舊版 **BOOL ulaSDKSetSamplesNum(int iSize)** 函式入口相容，須注意輸入參數單位為 **Mbyte**。

參數：

iSize[in]：

Type：int

設定 TBA 擷取資料的長度，最大不能超過電腦最大可用的記憶體量。

可以藉由呼叫 **INT ulaSDKGetMaxRamSize ()** 來取得可用的最大可用記憶體(Mbyte)

回傳值：

如果回傳值為 **True**，代表記憶長度設定成功。如果回傳 **False** 值則代表執行此函式產生錯誤。

INT ulaSDKGetMaxRamSize()

功能：

取得目前電腦 RAM 最大可用的記憶體量。與舊版 **INT ulaSDKGetMaxSamplesNum ()** 函式入口相容，須注意回傳參數單位為 **Mbyte**。

回傳值：

回傳目前電腦 RAM 最大可用的記憶體量，例: 1024 代表可用 1024 MB。

BOOL ulaSDKThreshold(int iPod, int iMilliVolt)

功能

設定 TBA 判斷 0 與 1 的電壓準位，通常這個數值會設定在目標訊號振幅一半的位置。

參數：

iPod [in]：

Type：int

每一個 Pod 只能設定一個電壓準位，根據不同的機種支援的 Pod 數目不同。

請參照下表：

型號	Pod	通道
TravelBus 系列	0	CH0 – CH15

iMilliVolt [in]：

Type：int

觸發電壓準位，單位為 mV

回傳值：

如果回傳值為 True，代表設定成功。如果回傳 False 值則代表執行此函式產生錯誤。

BOOL ulaSDKCapture(LPSDK_TRIG lpSDKTr)

功能：

通知 TBA 開始擷取資料。

參數：

lpSDKTr [in]

Type：LP SDK_TRIG

參數儲存的 structure，將先前設定好的觸發參數傳入

回傳值：

如果回傳值為 True，代表開始擷取，在擷取完成前 TBA 的機殼上的燈會轉為紅色。如果回傳 False 值則代表執行此函式產生錯誤。

備註：

EX: if (!ulaSDKCapture(lpSDK_TrigSet))

AfxMessageBox("Capture Error!");

else

SetTimer(ID_TM_CAPTURE, 100, NULL);

之後利用 Timer 等待 BOOL ulaSDKIsCaptureReady()回應擷取完成後即可接收資料。

BOOL ulaSDKIsCaptureReady()

功能：

詢問 TBA 資料是否擷取完成。

回傳值：

如果回傳值為 **True**，代表擷取完成，此時 TBA 的機殼上的燈應為綠色，可以使用 `ulaSDKGetChData` 或 `ulaSDKGetBusData` 來取回擷取的資料。如果回傳 **False** 值則代表仍在擷取中，TBA 機殼上的燈應為紅色。

BOOL ulaSDKStopCapture()

功能：

立即停止擷取資料，TBA 會保存在觸發點前的資料。

回傳值：

如果回傳值為 **True** 代表成功停止。如果回傳值為 **False** 代表停止時發生錯誤。

備註：

停止擷取後也可以呼叫 `BOOL ulaSDKGetDataAt (...)` 或 `BOOL ulaSDKGetNextData (...)` 來取得 Pre-Trigger 記錄的資料。

BOOL ulaSDKGetTimingInfo(__int64& i64StartPos, __int64& i64EndPos, __int64& i64TrigPos)

功能：

取得 TBA 擷取成功後的起始、結束、觸發點時間位置。

參數：

i64StartPos [out]：

Type：__int64&

將存入擷取成功的第一個資料點時間位置資料。

i64EndPos [out]：

Type：__int64&

將存入擷取成功的最後一個資料點時間位置資料。

i64TrigPos [out]：

Type：__int64&

將存入擷取成功的觸發點時間位置資料，若未成功觸發，則會在此參數填入 0。

回傳值：

回傳 **True** 代表資料取得成功。回傳 **False** 代表資料取得失敗，起因可能是資料擷取尚未完成或是資料擷取失敗造成。

BOOL ulaSDKGetDataAt (__int64& pi64Data, __int64 i64TransPos)

功能：

取得 TBA 擷取成功後指定時間位置的資料內容。

參數：

pi64Data [out]：

Type：__int64&

取得的資料將會寫入此參數內

i64TransPos [in]：

Type：__int64

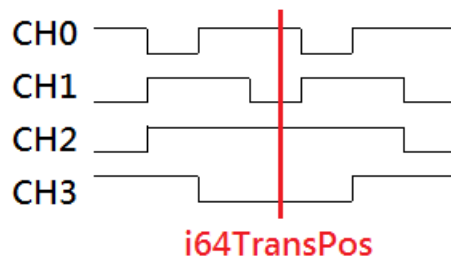
設定指定的時間位置。

回傳值：

回傳 **True** 代表資料取得成功。回傳 **False** 代表資料取得失敗，起因可能是資料擷取尚未完成或是資料擷取失敗造成。

備註：

EX： 如下圖所擷取到的資料，在 i64TransPos 位置將會讀取到 pia64Data = 0x05 的資料



BOOL ulaSDKGetNextData (__int64& pi64Data, __int64& i64StartTimestamp)

功能：

取得 TBA 擷取成功後，指定時間位置開始往後尋找下一個轉態點的資料內容。

參數：

pi64Data [out]：

Type：__int64&

取得的資料將會寫入此參數內

i64StartTimestamp [in, out]：

Type：__int64

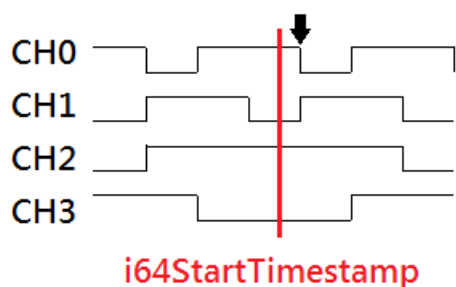
設定搜尋起始時間位置。

回傳值：

回傳 True 代表資料取得成功。回傳 False 代表資料取得失敗，起因可能是資料擷取尚未完成或是已蒐尋到資料末尾。

備註：

EX: 如下圖所擷取到的資料，從 i64StartTimestamp 位置向後搜尋，將會找到下一個轉態點位於黑色箭頭的所在位置，取得 pia64Data = 0x06 並將黑色箭頭的 Timestamp 寫回 i64StartTimestamp 中。



BOOL ulaSDKSaveAsTBWFile(char* szFilePathName)

功能：

將擷取到的波形資料儲存成可用 TBA.exe 開啟的.TBW 檔案，相容於舊版本的 ulaSDKSaveAsLawFile，但一樣會將檔案輸出為.tbw 格式。

參數：

szFilePathName[in]：

Type：char *

欲儲存檔案的路徑及檔名。

回傳值：

回傳 True 代表儲存檔案成功，回傳 False 代表檔案儲存失敗。

備註：

這個函式會存下最後一次呼叫 `ulaSDKCapture()` 時的參數設定，並將 TBA 此次取得的資料存入指定位置，必須要在呼叫過 `ulaSDKCapture()` 後才能使用，否則會回傳 `False` 值。

BOOL ulaSDKClose ();

功能：

關閉此程式對 TBA 的控制，使 TBASDK 得以釋放資源。

回傳值：

如果回傳值為 `True`，代表正常關閉。如果回傳 `False` 值則代表執行此函式產生錯誤。

INT ulaSDKGetLastError()

功能：

當使用 SDK 函式發生錯誤時可以呼叫此函式得到錯誤碼，錯誤碼的意思及可能造成此問題的原因標註在 `LASDK_Err.h` 檔內。

回傳值：

回傳最後執行發生錯誤的錯誤代碼。



皇晶科技股份有限公司

www.acute.com.tw



地址：新北市三重區 24159 重新路五段 609 巷 12 號 6 樓之 7 (湯城園區)

電話：(02)2999-3275

傳真：(02)2999-3276

E-mail: service@acute.com.tw