

Acute Logic Analyzer Software development kit (SDK) Programming guide

Version: 1.2c

Publish: 2012/12/13

內容

LPSDK_TRIG 結構簡介	2
LASDK 函式說明.....	4
BOOL ulaSDKInit()	4
BOOL ulaSDKSelectDevice(char* szSerialNo)	4
INT ulaSDKGetLaID()	5
BOOL ulaSDKGetSerialNumber(char *szSerialNum, int iSize).....	5
BOOL ulaSDKSetHwInfo(int iIndex, LPVOID lpv)	6
BOOL ulaSDKSetChTrigger(LPSDK_TRIG lpSDKtr, int iLevel, int iCh, int iTrig, int iCondLogic)	7
BOOL ulaSDKClearTrigger(LPSDK_TRIG lpSDKtr).....	8
BOOL ulaSDKSetSamplesNum(int iSize).....	8
BOOL ulaSDKThreshold(int iPod, int iMilliVolt)	9
BOOL ulaSDKCapture(LPSDK_TRIG lpSDKtr)	9
BOOL ulaSDKIsCaptureReady()	10
BOOL ulaSDKStopCapture()	10
BOOL ulaSDKGetChData(int iCh,UINT* pUserData, int* lpiSize, int iStartSamplePos)	11
BOOL ulaSDKGetBusData(int iMSB, int iLSB, UINT* pUserData, int* lpiSize, int iStartSamplePos)	12
BOOL ulaSDKSaveAsLawFile(char* szFilePathName)	13
BOOL ulaSDKClose ();	13
BOOL ulaSDKGetLastError()	13
觸發設定讀寫檔案.....	14
BOOL ulaSDKSaveTriggerByFile(LPSDK_TRIG lpSDKtr, char* szFileName).....	14
BOOL ulaSDKReadTriggerByFile(LPSDK_TRIG lpSDKtr, char* szFileName)	14
BOOL ulaSDKSetHWByPrj(char* szFileName, LPSDK_TRIG lpSDKtr, int iUserMaxMemSize);	14

LASDK 架構與流程



LaSDK 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(請參照底下詳細說明)
    int iDelay;          //延遲觸發的延遲時間(單位為 Sample Clock)
    int iWidth;          //觸發寬度(單位為 Sample Clock)
                        //Ex : Sample Rate 為 200MHz, iDelay = 5, iWidth = 10
                        //實際觸發延遲時間為      5 * (1 / 200MHz) = 25 ns
                        //實際觸發寬度為          10 * (1 / 200MHz) = 50 ns

    int iPassCount;      //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(請參照底下詳細說明)
    int iTrPos;          //觸發游標的位置
    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;
  
```

iFlag:

數值	說明
<code>#define TR_DBLMODE</code> 0x0001	倍頻模式 (PKLA 和 LA2000 系列提供)
<code>#define TR_PRETRIG</code> 0x0002	啟動 Pre-Trigger 功能 (LA2000P 和 TravelLogic 提供)
<code>#define TR_XCHG</code> 0x0004	啟動雙條件觸發，正負緣皆可觸發 (LA2000P 和 TravelLogic 提供)

iExtClk:

數值	說明
<code>#define TRF_INT</code> 0x0001	使用內部頻率(非同步取樣)
<code>#define TRF_EXT</code> 0x0002	使用外部頻率(同步取樣)
<code>#define TRF_CH15</code> 0x0004	外部頻率輸入通道在 CH15
<code>#define TRF_CH31</code> 0x0008	外部頻率輸入通道在 CH31
<code>#define TRF_CH63</code> 0x0010	外部頻率輸入通道在 CH63
<code>#define TRF_OR</code> 0x0020	0：內外頻使用 AND 模式 1：內外頻使用 OR 模式
<code>#define TRF_NOT</code> 0x0040	0：正常外頻模式 1：反向外頻模式
<code>#define TRF_CUSTOM</code> 0x0080	啟用外部時鐘，通道 Channel 35 外部時鐘模式需由 Bit8 - 15 設定 (TravelLogic 系列提供)
<code>#define TRF_TL_CH35</code> 0x000	Channel 35
<code>#define TRF_TL_NOT_CH35</code> 0x100	! Channel 35
<code>#define TRF_TL_CH34_AND_CH35</code> 0x200	Channel 34 & Channel 35
<code>#define TRF_TL_CH34_OR_CH35</code> 0x300	Channel 34 Channel 35 (TravelLogic 系列提供)

LASDK 函式說明

BOOL ulaSDKInit()

功能：

尋找並啟動目前接在此電腦上的邏輯分析儀(LA)，Ex：TravelLogic、PKLA1616

回傳值：

如果回傳值為 True，代表 LA 啟動成功。如果回傳 False 值則代表 LA 啟動失敗。

備註：

如果有多台 LA 接在同一台電腦上時，執行此函式後會將所有的 LA 裝置一併初始化完成，且系統預設會直接啟用第一台裝置。

BOOL ulaSDKSelectDevice(char* szSerialNo)

功能：

當電腦上有多台 LA 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/out]：

Type：char*

指定啟動的 LA 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 LA 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("TL22360000");  
ulaSDKSetSamplesNum(1000);           //將序號為 TL22360000 機器的取樣點數設為 1000  
ulaSDKSelectDevice("TL22360001");  
ulaSDKSetSamplesNum(2000);           //將序號為 TL22360001 機器的取樣點數設為 2000
```

INT ulaSDKGetLaID()

功能：

取得 LA 的 ID，確認目前使用的機種。

回傳值：

根據機種不同回傳不同的數值，EX：TL2136 會得到 0x2136 的數字。

備註：

回傳 ID 代號	機種
0x2036	TL2036
0x2136	TL2136
0x2236	TL2236
0x1032	LA1032
0x1064	LA1064
0x2032	LA2032
0x2064	LA2064
0x2132	LA2132
0x2164	LA2164
0x1116	PKLA1116
0x1216	PKLA1216
0x1616	PKLA1616

BOOL ulaSDKGetSerialNumber(char *szSerialNum, int iSize)

功能：

取得 LA 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：TL22061234

BOOL ulaSDKSetHwInfo(int ilIndex, LPVOID lpv)

功能：

設定 LA 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_TL_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
HW_4G_36CH	4GHz	36
HW_4G_18CH	4GHz	18
HW_4G_9CH	4GHz	9
HW_4G_4CH	4GHz	4
HW_2G_36CH	2GHz	36
HW_1600M_4CH	1.6GHz	4
HW_800M_9CH	800MHz	9
HW_400M_18CH	400MHz	18
HW_200M_36CH	1Hz - 200MHz	36
HW_200M_18CH	1Hz - 200MHz	18
HW_200M_12CH	1Hz - 200MHz	12
HW_200M_9CH	1Hz - 200MHz	9
HW_200M_6CH	1Hz - 200MHz	6
HW_200M_4CH	1Hz - 200MHz	4
HW_200M_2CH	1Hz - 200MHz	2
HW_200M_1CH	1Hz - 200MHz	1
HW_EXT_35CH	External Clock	35
HW_EXT_18CH	External Clock	18
HW_EXT_12CH	External Clock	12
HW_EXT_9CH	External Clock	9
HW_EXT_6CH	External Clock	6
HW_EXT_4CH	External Clock	4
HW_EXT_2CH	External Clock	2
HW_EXT_1CH	External Clock	1

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 9 個通道的模式

```
int iHWMMode = HW_200M_9CH;
```

```
if (!ulaSDKSetHwInfo(SET_TL_MODE, &iHWMMode))
```

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

註：除了外部頻率以及 200MHz 的硬體運作模式外，其他模式都有特定的取樣頻率，無法更動!

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

iLevel 設定時需注意，雖然 LA 有 16 階，但只要任何一階的設定使用 LA_TRIG_CHANGE 最大階數就會下降到 4 使用或 LA_TRIG_RISING/LA_TRIG_FALLING 最大階數則會下降到 8

註1：使用者可能需要多次呼叫這個函式來完成多個觸發條件的設定。

註2：宣告LPSDK_TRIG lpSDKTr時可以一併呼叫ulaSDKClearTrigger()來初始化觸發設定。

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->iDelay      = 0;           //觸發延遲歸零
lpSDKTr->iExtClk       = 1;           //使用內部頻率
lpSDKTr->iFlag         = TR_PRETRIG; //開啟PreTrigger功能
lpSDKTr->iFreq         = 200000000;  //採樣頻率預設200MHz
lpSDKTr->iFreqHi       = 0;           //採樣頻率預設200MHz
lpSDKTr->iPassCount    = 0;           //忽略次數設為 0
lpSDKTr->iTrPos        = 360;         //Trig Pos 設為第 360 個採樣點
                                   //不設為 0，設 0 的話觸發點會出現邊界上
lpSDKTr->iWidth        = 0;           //觸發寬度設為0 (for LA 2000)
```

BOOL ulaSDKSetSamplesNum(int iSize)

功能：

設定 LA 擷取資料的取樣數，最大不能超過 LA 的總記憶深度/開啟的通道數。

參數：

iSize[in]：

Type：int

設定 LA 擷取資料的長度，最大不能超過 LA 的總記憶深度/開啟的通道數。若超過會以該硬體模式的最大取樣數作為設定。

可以藉由呼叫 INT ulaSDKGetMaxSamplesNum() 來取得可用的最大取樣點數

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。

BOOL ulaSDKThreshold(int iPod, int iMilliVolt)

功能

設定 LA 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
PKLA 系列	0	CH0 – CH15
TravelLogic 系列	0	CH0 – CH17
	1	CH18 – CH35
LA 1000、2000 系列	0	CH0 – CH15
	1	CH16 – CH31
	2	CH32 – CH47
	3	CH48 – CH63

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 LA 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 LA 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 LA 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 LA 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，LA 機殼上的燈應為紅色。

BOOL ulaSDKStopCapture()

功能：

立即停止擷取資料，LA 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 ulaSDKGetChData(...)** 或 **BOOL ulaSDKGetBusData(...)** 來取得 **Pre-Trigger** 記錄的資料。

BOOL ulaSDKGetChData(int iCh,UINT* pUserData, int* lpiSize, int iStartSamplePos)

功能：

取得 LA 擷取成功後單一通道的資料。

參數：

iCh [in]：

Type：int

設定取得資料的通道

pUserData [out]：

Type：UINT*

使用者需傳入的 Buffer 位址，所需大小可以由 INT ulaSDKGetSamplesNum()取得

lpiSize [in, out]：

Type：int*

傳入希望取得多少取樣點數的資料，資料讀取成功後回傳已寫入的取樣點數。

iStartSamplePos[in]：

Type：int

設定從哪一個取樣點開始將資料載入 Buffer。

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：

回傳的 pUserData 會將指定通道的資料填入各 pUserData 的 Bit 0

EX: LA 擷取到的內容為(CH7 - CH0) 0x02、0x00

```
int iSize = ulaSDKGetSamplesNum();
```

```
UINT *pUserData = new UINT[iSize];
```

```
int iStartSample = 0; //從取樣點 0 開始載入資料
```

```
ulaSDKGetChData (1, pUserData, lpiSize, iStartSample);
```

```
pUserData[0] 就會得到 0x01，而 pUserData[1] 得到 0x00
```

BOOL ulaSDKGetBusData(int iMSB, int iLSB, UINT* pUserData, int* lpiSize, int iStartSamplePos)

功能：

取得 LA 擷取成功後多個連續通道的資料，最多一次以 32 通道為限。

參數：

iMSB [in]：

Type：int

設定結束位元(MSB)的通道

iLSB [in]：

Type：int

設定起始位元(LSB)的通道

pUserData [out]：

Type：UINT*

使用者需傳入的 Buffer 位址，所需大小可以由 INT ulaSDKGetSamplesNum()取得

lpiSize [in, out]：

Type：int

傳入希望取得多少取樣點數的資料，資料讀取成功後回傳已寫入的取樣點數。

iStartSamplePos[in]：

Type：int

設定從哪一個取樣點開始將資料載入 Buffer

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：

回傳的 pUserData 會按照給定通道順序填入相對應的資料

EX: LA 擷取到的內容為(CH8 - CH1) 0xFE、0xA0

int iSize = ulaSDKGetSamplesNum();

UINT *pUserData = new UINT[iSize];

int iStartSample = 0; //從取樣點 0 開始載入資料

ulaSDKGetBusData(8, 1, pUserData, lpiSize, iStartSample);

pUserData[0] 就會得到 0xFE，而 pUserData[1] 得到 0xA0

Bit 取樣點	31 - 8	7	6	5	4	3	2	1	0	Data
pUserData[0]	0	1	1	1	1	1	1	1	0	0x000000FE
pUserData[1]	0	1	0	1	0	0	0	0	0	0x000000A0

BOOL ulaSDKSaveAsLawFile(char* szFilePathName)

功能：

將擷取到的波形資料儲存成可用 LAViewer 開啟的.LAW 檔案。

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 LA 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 LA 的控制，使 LASDK 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。

BOOL 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。

觸發設定讀寫檔案

BOOL ulaSDKSaveTriggerByFile(LPSDK_TRIG lpSDKTr, **char* szFileName)**

BOOL ulaSDKReadTriggerByFile(LPSDK_TRIG lpSDKTr, **char* szFileName)**

功能：

儲存/讀取之前設定好的觸發參數檔案。

參數：

lpSDKTr[in/out]：

Type：LPSDK_TRIG

欲儲存/讀取的觸發參數設定結構。

szFileName[in]：

Type：**char***

欲儲存/讀取檔案的完整路徑及檔名。

EX：D:\LA_SDK\TriggerSet.aqr

回傳值：

回傳 True 代表儲存/讀取成功，回傳 False 代表操作失敗。

備註：

設定好參數後可以先將資料儲存到檔案中，下一次開啟時便可以節省設定的時間。

BOOL ulaSDKSetHWByPrj(char*** szFileName, LPSDK_TRIG lpSDKTr, **int** iUserMaxMemSize);**

功能：

讀取之前利用 LA Viewer 設定好的硬體參數檔案。

參數：

szFileName[in]：

Type：**char***

欲儲存/讀取檔案的完整路徑及檔名。

EX：C:\Users\UserName\Documents\Acute\LA\Project\Project1.aqr

lpSDKTr[in/out]：

Type：LPSDK_TRIG

欲讀取的觸發參數設定結構。

iUserMaxMemSize[in]：

Type：**int**

設定每次取樣的取樣點數，最大不能超過 LA 的總記憶深度/開啟的通道數。

備註：

設定好參數後可以先將資料儲存到檔案中，下一次開啟時便可以節省設定的時間。



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