

Acute TravelBus Analyzer Software development kit (SDK) Programming guide

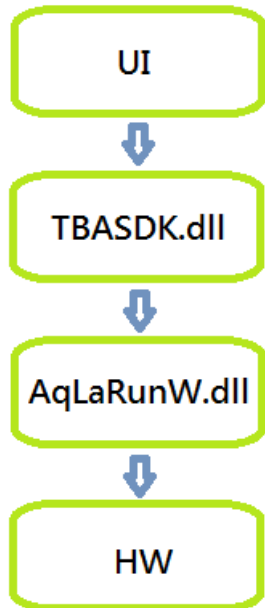
Version: 1.1

Publish: 2019/03/26

Contents

LPSDK_TRIG Structure Introduction	2
TBASDK Function Definitions.....	3
int ulaSDKInit().....	3
BOOL ulaSDKSelectDeviceIdx(int iDeviceIndex)	3
BOOL ulaSDKSelectDevice(char* szSerialNo)	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().....	7
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)	9
BOOL ulaSDKGetDataAt (__int64& pi64Data, __int64 i64TransPos)	9
BOOL ulaSDKGetNextData (__int64& pi64Data, __int64& i64StartTimestamp)	11
BOOL ulaSDKSaveAsTBWFile(char* szFilePathName)	11
BOOL ulaSDKClose ();.....	12
BOOL ulaSDKGetLastError()	12

TBASDK control flow and introduction



TBASDK is a wrapper of the AqLaRunW.dll, translate those complicated parameters in AqLaRunW.dll to more user-friendly format. The TBASDK also check the possible error for invalid parameters, users can read the error code and check from LASDK_Err.h if any error happened.

LPSDK_TRIG Structure Introduction

```

typedef struct _SDKTRIG
{
    int iFlag;           //Trigger flag (NOT Implemented)
    int iDelay;          //Trigger delay for N sample clocks (NOT Implemented)
    int iWidth;          //Set trigger width to N sample clocks (NOT Implemented)
    int iPassCount;      //Trigger condition pass count
    int iFreq;           //Sample rate (Hz)
    int iFreqHi;         //Reserved, fill with zero
    int iExtClk;         //External clock flag (NOT Implemented)
    int iTrPos;          //Trigger position, from 1 percent to 99 percent
    int lpiCont[16];     //Trigger condition buffer (Do not change the content manually)
    BYTE lpbTrigData[1024]; //Trigger data buffer (Do not change the content manually)
} SDK_TRIG, FAR *LPSDK_TRIG;
  
```

TBASDK Function Definitions

int ulaSDKInit()

Search and initial all TBA devices connected on the PC, Ex: TravelBus 1016E, 1016B, 1016B+

Return value

Return true if device initialized successful °

Remarks

If there are multiple devices connect on the PC, this function will initial all devices in the same time.

After the initializations, the dll will select the index 0 device as the default selected device.

BOOL ulaSDKSelectDeviceIdx(**int** iDeviceIndex)

Select TBA device by specified USB device index.

Parameters

iDeviceIndex[in/out]:

Type : **int**

Specified the USB index of the target TBA device.

Return value

Return true when device selected successful.

Remarks

To control multiple TBAs, you should use this function to specify the target TBA before assigning the parameters.

Ex:

```
ulaSDKSelectDeviceIdx(0);
```

```
ulaSDKSetSamplesNum(1000);           //Set device 0's sample points to 1000
```

```
ulaSDKSelectDeviceIdx(1);
```

```
ulaSDKSetSamplesNum(2000);           //Set device 1's sample points to 2000
```

BOOL ulaSDKSelectDevice(**char*** szSerialNo)

Select TBA device by specified Serial number.

Parameters

szSerialNum[in/out]:

Type : **char***

Contains the serial number of the specified TBA device.

Return value

Return true when device selected successful.

Remarks

To control multiple TBAs, you should use this function to specify the target TBA before assigning the parameters.

Ex:

```
ulaSDKSelectDevice("TB10160000");
ulaSDKSetSamplesNum(1000);           //Set device TB10160000's sample points to 1000
ulaSDKSelectDevice("TB10160001");
ulaSDKSetSamplesNum(2000);           //Set device TB10160001's sample points to 2000
```

BOOL ulaSDKGetSerialNumber(char *szSerialNum, int iSize)

Get current selected device serial number.

Parameters

szSerialNum[in/out] :

Type : **char***

Contains the device serial number, Ex : TL22061234. Buffer size must greater than 35 bytes.

iSize[in] :

Type : **int**

Buffer size in byte.

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

BOOL ulaSDKSetHwInfo(int iIndex, LPVOID lpv)

Set TBA hardware mode, including sample rate and available channels.

Parameters

iIndex [in]

Type : **int**

Input **SET_TBA_MODE = 5** to set the hardware mode. Other value is reserved.

lpv [in]:

Type : LPVOID

When user input SET_TBA_MODE in **iIndex** parameter, this parameter will be recognized as an **int** pointer indicates the selected hardware mode.

Index	Sample Rate	Available Channel(s)
TBA_HWMODE_LA200M	1Hz - 200MHz	16

Hardware Parameter Setting Table

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

Remark

Ex: For Sample Rate 200MHz, available channel number 16

```
int iHWMMode = TBA_HWMODE_LA200M;
if (!ulaSDKSetHwInfo(SET_TBA_MODE, &iHWMMode))
    AfxMessageBox("Hw Info Set Error!");
```

Note: Only TBA_HWMODE_LA200M mode is supported at this moment.

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

Set the Trigger condition to LPSDK_TRIG structure.

Parameters

lpSDKTr [in, out]

Type : LPSDK_TRIG

Trigger structure, must be passed to **ulaSDKCapture** function in order to capture data.

iLevel / iCh / iTrig [in] :

Type : int

Trigger level, type and channel select. **iTrig** should be one of the value described below.

Value	Trigger condition
#define LA_TRIG_DONT_CARE 0x08	Don't care, triggered at any condition.
#define LA_TRIG_LOW 0x00	Triggered at Low Signal.
#define LA_TRIG_HIGH 0x06	Triggered at High Signal.
#define LA_TRIG_RISING 0x04	Triggered at rising edge.
#define LA_TRIG_FALLING 0x02	Triggered at falling edge.
#define LA_TRIG_CHANGE 0x0A	Triggered at any edge.

iCondLogic [in] :

Type : int

Value	Description
#define TR_NEXT 0x00	Two trigger conditions must be located in two continuous samples.
#define TR_NEXTIF 0x01	Two trigger conditions can be separated by other samples.
#define TR_TRIGGER 0x02	Trigger the LA after trigger condition goes true.

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

Remark

Ex: Set Trigger condition to **Channel 3** must be **High** at **trigger level 2**.

```
iLevel = 2,  
iCh = 3,  
iTrig = LA_TRIG_HIGH
```

The maximum number of trigger levels is **16**, but it was also limited by the trigger condition.

If any level in the trigger struct used **LA_TRIG_CHANGE**, the trigger level will be limited to **4**.

If any level in the trigger struct used **LA_TRIG_RISING** / **LA_TRIG_FALLING**, the number will be limited to **8**.

Users can use this function to create a multi-level trigger condition.

It is available to call **ulaSDKClearTrigger()** to initialize the trigger settings.

It is recommended to set the trigger channels within the available channels selected by HW mode, in order to make the reading data more readable with the trigger channel's data.

BOOL ulaSDKClearTrigger(LPSDK_TRIG lpSDKtr)

Initial and clear all trigger condition.

Parameters

```
lpSDKtr[in, out] :  
    Type : LPSDK_TRIG  
    Trigger Structure.
```

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

Remark

This function will initialize all parameters to the default setting.

The default settings of the LPSDK_TRIG are described at below:

```
lpSDKtr->iDelay      = 0;           //Trigger delay  
lpSDKtr->iExtClk      = 1;           //Use internal clock  
lpSDKtr->iFlag        = TR_PRETRIG; //Enable Pre-Trigger function  
lpSDKtr->iFreq        = 200000000;  //Sample Rate = 200MHz  
lpSDKtr->iFreqHi      = 0;           //Sample Rate = 200MHz  
lpSDKtr->iPassCount    = 0;           //Trigger Pass Count  
lpSDKtr->iTrPos       = 360;         //Trigger position  
                        //Trigger signal may not appear on the waveform if this value is too small to keep them.  
lpSDKtr->iWidth       = 0;           //Trigger Width (for LA 2000)
```

BOOL ulaSDKSetRamSize (int iRamSize_MB)

Set the sampling RAM memory used by the TBA device, the input RAM size must less than the maximum available RAM size on the PC.

This function is compatible with the BOOL ulaSDKSetSamplesNum([int](#) iSize) function in the old LASDK, but the input parameter is changed from size to Mbyte.

Parameters

iSize[in] :

Type : [int](#)

Setup the PC RAM size allocated for the TBA acquisition, the maximum available value can be read by calling **ulaSDKGetMaxRamSize()** function.

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

INT ulaSDKGetMaxRamSize()

Return the maximum available PC RAM, this function is compatible with the INT ulaSDKGetMaxSamplesNum () function in the old LASDK, but the returned value is changed from size to Mbyte.

Return value

Return the maximum available PC RAM, Ex: Returning 1024 indicates there is 1024MB RAM available.

BOOL ulaSDKThreshold([int](#) iPod, [int](#) iMilliVolt)

Set TBA voltage threshold. This value is recommended to set to the mid of target signal.

Parameters

iPod [in] :

Type : [int](#)

The Pod number was varying from the LA models. Each Pod has its independent voltage threshold

Model	Pod	Channel
TravelBus series	0	CH0 – CH15

iMilliVolt [in] :

Type : [int](#)

Voltage threshold in millivolt.

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

BOOL ulaSDKCapture(LPSDK_TRIG lpSDKtr)

Send capture command to the TBA.

Parameters

lpSDKTr [in]

Type : LP SDK_TRIG

Input the trigger structure to be used to capture data.

Return value

If the function succeeded, the return value is a nonzero value. The LED indicator on the LA turns to red when the TBA is capturing data.

If the function failed, the return value is zero.

Remark

```
EX:  if (!ulaSDKCapture(lpSDK_TrigSet))
        AfxMessageBox("Capture Error!");
    else
        SetTimer(ID_TM_CAPTURE, 100, NULL);
```

After send the capture command, user have to wait the **ulaSDKIsCaptureReady()** ready before retrieving the data from LA.

BOOL ulaSDKIsCaptureReady()

Check the TBA capture status.

Return value

If captured successful, the return value is a nonzero value. User can call **ulaSDKGetChData()** or **ulaSDKGetBusData()** to get data from TBA.

Return zero if the TBA is still capturing data.

BOOL ulaSDKStopCapture()

Send stop capture command to the TBA. The TBA will keep those data in pre-trigger section.

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

Remark

After sent stop capture command, user can call **ulaSDKGetChData()** or **ulaSDKGetBusData()** to get the pre-trigger data.

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

Read the captured timing information of the TBA, including first sample start position, last sample end position and trigger position.

Parameters:

i64StartPos [out] :

Type : __int64&

Contains the first sample start timestamp information of the waveform.

i64EndPos [out] :

Type : __int64&

Contains the last sample timestamp information of the waveform.

i64TrigPos [out] :

Type : __int64&

Contains the trigger timestamp information of the waveform, if there's no valid trigger event, this parameter will be filled with 0.

Return value

Return True if successful. Return False when information read fail, which might be a result of unfinished or failed waveform capture.

BOOL ulaSDKGetDataAt (__int64& pi64Data, __int64 i64TransPos)

Return the waveform data on the target timestamp position in the captured waveform.

Parameters

pi64Data [out] :

Type : __int64&

Contains waveform data.

i64TransPos [in] :

Type : __int64

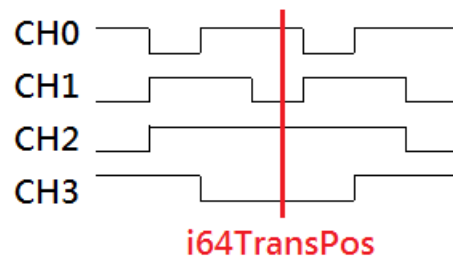
Input target timestamp position.

Return value

Return True if successful. Return False when information read fail, which might be a result of unfinished or failed waveform capture.

Example:

The following picture shows a captured waveform data of CH0 to CH3, input timestamp at i64TransPos will get the returned pia64Data = 0x05.



BOOL ulaSDKGetNextData (__int64& pi64Data, __int64& i64StartTimestamp)

Return timestamp and waveform data on next data transition of the input timestamp.

Parameters

pi64Data [out] :

Type : __int64&

Contains waveform data.

i64StartTimestamp [in, out] :

Type : __int64

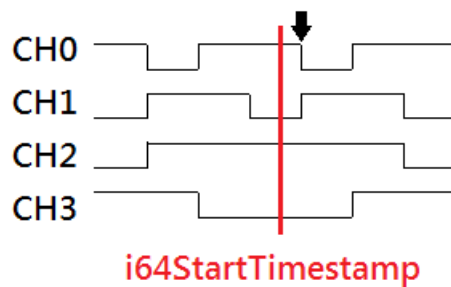
Input the begin timestamp, and the function will return the next data transition timestamp.

Return value

Return True if successful. Return False when information read fail or reached the end of the waveform.

Example

The following picture shows a captured waveform data of CH0 to CH3, input timestamp at i64StartTimestamp, the function will seek to next data transition (the black arrow) and then returned the waveform data to pia64Data = 0x06 and the new timestamp to i64StartTimestamp.



BOOL ulaSDKSaveAsTBWFile(char* szFilePathName)

Store current waveform as TBA.exe compatible .TBW file.

Parameters

szFilePathName[in] :

Type : char *

File pathname.

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

BOOL ulaSDKClose ();

Shutdown LA

Return value

If the function succeeded, the return value is a nonzero value.

If the function failed, the return value is zero.

BOOL ulaSDKGetLastError()

Get error code from the **LASDK.dll**, the error number meaning and reasons are marked at **LASDK_Err.h**.

Return value

Return error code.



Acute Technology Inc.

www.acute.com.tw



Address : 6F-7, #12, Ln. 609, Sec. 5, Chongxin Rd., Sanchong Dist., New Taipei City 24159, Taiwan

Tel : +886-2-2999-3275

Fax : +886-2-2999-3276

E-mail: service@acute.com.tw