

AxLaRun.ocx

AxLaRun.ocx is an ActiveX object of LA; it should be tie-in and executed with LARUN.DLL. To develop LA application by VB, just add the object into your project, and then you can control the LA very easy. For activating this file, first, please copy the AxLaRun.ocx and LARUN.DLL two files to your project directory. Or, you can copy them to the System32 of Windows directory either. We will provide one VB sample to demo how to draw one channel waveform. Please see the property and Method of AxLaRun.ocx as below:

AxLaRun.ocx Property

ExternalClock Property

Set sampling clock source for LA object.

Syntax

object.ExternalClock [= value]

The **ExternalClock** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>value</i>	A value that determines the sampling clock source.

Settings

The settings for *value* are:

Setting	Description
Bit 0	Use Internal Clock.
Bit 1	Use External Clock (except TravelLogic series)
Bit 2	External Clock Channel = Channel 15
Bit 3	External Clock Channel = Channel 31
Bit 4	External Clock Channel = Channel 63
Bit 5	0: Sampling Clock = Internal Clock & External Clock. 1: Sampling Clock = Internal Clock # External Clock.
Bit 6	1: Sampling Clock = !External Clock.
Bit7	Use External Clock (TravelLogic series only), External Clock Channel = Channel 35

Bit8~Bit15

External Clock Mode (Include !Channel 35, Channel34 & Channel35, Channel34 # Channel35 mode, TravelLogic series only)

Ex:

For TravelLogic external clock,

```
AxLaRun1.ExternalClock = &H80 ' Use External Clock(CH35) For TL2x36
```

```
AxLaRun1.ExternalClock = &H180 ' Use External Clock(!CH35) For TL2x36
```

```
AxLaRun1.ExternalClock = &H280 ' Use External Clock(CH34 & CH35) For  
TL2x36
```

```
AxLaRun1.ExternalClock = &H380 ' Use External Clock(CH34 # CH35) For  
TL2x36
```

MemoryDepth

Set the memory depth of LA hardware.

Syntax

object.**MemoryDepth**

Remarks

You may change LA's memory depth by setting the value. For example, there are 1M bits/channel of the PKLA1616 you may set 64K bits/channel. However, the minimum number is 256 bits and the maximum number depends on default of LA's model, Each time, after executing the **GetHardware** Method will set the default value (2.5K)So, if you need an effective value, you have to change it between doing the **GetHardware** Method and the **CaptureData** Method.

Worth words: Even changing the **MemoryDepth** value at using the PKLA series, the DLL driver will still read whole data of default memory depth back from the LA.

Passcount Property

Set the trigger pass count for LA object.

Syntax

object.**Passcount** [= value]

The **Passcount** property has these parts:

Part	Description
<i>Object</i>	An object expression that evaluates to an object in the Applies To list.
<i>Value</i>	trigger pass count of LA

Settings

The settings for *value* are:

Setting	Description
0 ~ 104875	This value can set from 0 to 104875

Remarks

In PKLA1000 series, you can set this value from 0 to 255.

In TravelLogic series, you can set this value from 0 to 1048575.

PreTriggerMode Property

Set a value that determines whether LA object is on Pre-trigger mode or not.

Syntax

object.**PreTriggerMode** [= *boolean*]

The **PreTriggerMode** property syntax has these parts:

Part	Description
<i>Object</i>	An object expression that evaluates to an object in the Applies To list.
<i>Boolean</i>	A Boolean expression that specifies whether <i>object</i> is on pre-trigger mode or not.

Settings

The settings for *boolean* are:

Setting	Description
True	The control is on pre-trigger mode.
False	The control is not on pre-trigger mode.

Remarks

This property is available on TravelLogic series. On PKLA1000 series device, you can leave the value to false.

TrDelay Property

Set the time of delay trigger for LA object.

Syntax

object.**TrDelay** [= value]

The **TrDelay** property has these parts:

Part	Description
<i>Object</i>	An object expression that evaluates to an object in the Applies To list.
<i>Value</i>	A value that determines the time of LA's delay trigger.

Remarks

TravelLogic and PKLA 1000 series device, you must leave the value to 0.

TrMode Property

Set the trigger mode for LA object.

Syntax

object.**TrMode** [= value]

The **TrMode** property syntax has these parts:

Part	Description
<i>Object</i>	An object expression that evaluates to an object in the Applies To list.
<i>Value</i>	A value or constant that determines the trigger mode, as described in Settings.

Settings

The settings for *value* are:

Constant	Setting	Description
TR_SINGLE	0	Single trigger level mode.
TR_MULTI	1	Multi-trigger levels mode.
TR_DUAL	2	Dual trigger condition mode.
TR_WIDTH	3	Trigger width mode
TR_NONE	4	Disable trigger function.

Remarks

In PKLA 1000 series, you can set this value as TR_SINGLE and TR_NONE. At the

TR_SINGLE mode, we may use the **SetChTrigger** Method to assign the trigger condition.. The TR_NONE mode means to disable the Trigger detection function. Please refer to the LA manual for the detail description.

TrPosition Property

Set the trigger position for LA object.

Syntax

object.**TrPosition** [= value]

The **TrPosition** property syntax has these parts:

Part	Description
<i>Object</i>	An object expression that evaluates to an object in the Applies To list.
<i>Value</i>	A value that determines the LA's trigger position

Remarks

This value depends on LA's memory depth. And worth mentioning, it should be bounded in the whole memory size except the both 12 clocks of head and tail. For example, if the Memory Depth size is 65536 bits, this value can be set from 12 to 65536-12. The Trigger Position can't set in the front of 12 clocks and the end of 12 clocks.

TrWidth Property

Set the trigger width for LA object.

Syntax

object.**TrWidth** [= value]

The **TrWidth** property syntax has these parts:

Part	Description
<i>Object</i>	An object expression that evaluates to an object in the Applies To list.
<i>Value</i>	A value that determines the LA's trigger width

Remarks

This property is available on TravelLogic series only. On PKLA 1000 series device, you can leave the value to 0. The value just active on **TrMode** property is true.

AxLaRun.ocx Method

GetHardware Method

To get LA hardware from the PC.

Syntax

object.**GetHardware** *lModel*, *lSlot*

The **GetHardware** Method syntax has these parts:

Part	Description
<i>object</i>	Required. An object expression that evaluates to an object in the Applies To list.
<i>lModel</i>	Required. Long value indicating the LA's ID. Auto-search when lModel is 0.
<i>lSlot</i>	Required. Reserve. Must be 0.

Return values

If the LA ready, the return value is TRUE.

If the LA not ready, the return value is FALSE.

Remarks

Unless you applied multi-LA together, the lModel and lSlot can be just filled as 0,
The object of AxLaRun can't use any Method until the **GetHardware** Method done.

GetModel Method

Get the LA model name.

Syntax

object.GetModel

Return values

The **GetModel** Method will return the string of model name (BSTR).

Remarks

Ex:

```
Dim S as string
```

```
S = AxLaRun1.GetModel
```


SetHwMode Method

Set LA hardware to select LA's sampling rate and memory stacks.

Syntax

object.**SetHwMode** *IMode*

The **SetHwMode** Method syntax has these parts:

Part	Description
<i>object</i>	Required. An object expression that evaluates to an object in the Applies To list.
<i>IMode</i>	Required. Select the mode that LA's sampling rate and memory stacks, refer to the below.

Return values

If set LA hardware successfully, the return value is TRUE.

If not, the return value is FALSE.

Remarks

When you apply the **SetHwMode** Method, LA's **MemoryDepth** will be adjusted to the LA maximum memory depth. Certainly, you could set the **MemoryDepth** by yourself. **SetHwMode** Method must be applied after **GetHardware** Method and before **Capture** Method.

Note: In TravelLogic series, **SetHwMode**, **MemoryDepth** and **TrPosition** are closely linked. TravelLogic series will stay at wrong trigger position when captured.

Ex: according to the table, **AxLaRun1.SetHwMode(504)** with corresponding available channel is 6. This means we used x6 memory stack mode (36/6), if you want to let TravelLogic stay at correct trigger position, you must set the

MemoryDepth and **TrPosition** is a multiple of 6. **MemoryDepth = 3000**,
TrPosition = 60 for example.

lMode	Sampling rate	Available channel
0	4 GHz	36
1	4 GHz	18
2	4 GHz	9
3	4 GHz	4
100	2 GHz	36
200	1.6 GHz	4
300	800 MHz	9
400	400 MHz	18
500	<= 200MHz	36
501	<= 200MHz	18
502	<= 200MHz	12
503	<= 200MHz	9
504	<= 200MHz	6
505	<= 200MHz	4
506	<= 200MHz	2
507	<= 200MHz	1
600	External Clock	35
601	External Clock	18
602	External Clock	12
603	External Clock	9
604	External Clock	6
605	External Clock	4
606	External Clock	2
607	External Clock	1

*Suited TravelLogic Series

Ex1:

```
// For TravelLogic 2x36 series
```

```
AxLaRun1.SetHwMode(0)
```

```
AxLaRun1.SetFrequency(4000000000) // sampling rate 4GHz
```

```
AxLaRun1.MemoryDepth = 2000 // default MemoryDepth = 2.5K
```

```
// reset MemoryDepth = 2K
```

Ex2:

```
AxLaRun1.SetHwMode(500)
AxLaRun1.SetFrequency(100000000) // sampling rate 100MHz
AxLaRun1.MemoryDepth = 5000 // default MemoryDepth = 2M(TL2236)
// reset MemoryDepth = 5K
```

Ex3:

```
AxLaRun1.SetHwMode(602) // External Clock mode, 12 channels
// available.
AxLaRun1.SetFrequency(100000000) // sampling rate 100MHz
AxLaRun1.MemoryDepth = 5000 // default MemoryDepth = 2M(TL2236)
// reset MemoryDepth = 5K
```

*Suited PKLA1616+

lMode	Sampling rate	Available channel
0	<= 200MHz	16
1	<= 400MHz	16

Ex1:

```
// For PKLA1616+
AxLaRun1.SetHwMode(0)
AxLaRun1.SetFrequency(200000000) // Set sampling rate 200MHz
AxLaRun1.MemoryDepth = 2500 // default MemoryDepth = 512K,
// reset MemoryDepth = 2.5K
```

Ex2:

```
AxLaRun1.SetHwMode(1)
AxLaRun1.SetFrequency(400000000) // Set sampling clock = 400MHz
AxLaRun1.MemoryDepth = 1048576 // MemoryDepth = 1M
```


SetFrequency Method

Set LA sampling rate.

Syntax

object.SetFrequency *IFreq*

The **SetFrequency** Method syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>IFreq</i>	LA's sampling rate.

Return value

The return value is TRUE when the sampling rate is successfully set, if not, it is FALSE.

Remarks

The *IFreq* can be set as follows:

TravelLogic series: 4GHz, 2GHz, 1.6GHz, 800MHz, 400MHz, ..., 1Hz.

PkLA1616+: 400MHz, 200MHz, ..., 100Hz.

Other models: 200MHz, 100MHz, ..., 100Hz.

You must refer to the **SetHwMode** Method section to set the LA hardware first before applying to the **SetFrequency** Method.

Shutdown Method

Shutdown LA.

Syntax

object.**Shutdown**

Return values

No return value.

Remarks

Please execute the Shutdown Method to enter off state while closing LA controlled program. After shutdown LA, only **GetHardware** Method can be used.

Reset Method

Reset LA

Syntax

object.**Reset**

Return values

No return value.

Remarks

After executing **Capture** Method and got no True response from the **ReadStatus** Method, means the LA can't detect the Trigger signal. If you don't want to see the result of this capturing, you may execute **Reset** Method and re-execute the **Capture** Method.

Capture Method

Capture Data.

Syntax

object.**Capture**

Return values

No return value.

Remarks

Before using the **Capture** Method to acquire data, please execute **SetChTrigger** Method first to assign a proper trigger condition.

ReadStatus Method

Get the LA status to make sure the capturing action finished.

Syntax

object.**ReadStatus**

Return values

If the LA Capture OK, the return value is TRUE.

If the LA in capturing, the return value is FALSE.

Threshold Method

Set LA threshold

Syntax

object.**Threshold** *lPod*, *lMiniVolt*

The **Threshold** Method has these parts:

Part	Description
<i>object</i>	Required. An object expression that evaluates to an object in the Applies To list.
<i>lPod</i>	Required. Select the Pod which you want to set. If all Pod, please fill -1.
<i>lMiniVolt</i>	Required. Fill the Threshold voltage (unit=mV).

Return values

No return.

Remarks

The number of channels in one Pod is different between each product model. And, the threshold voltage is adjustable. Please refer to the product manual to learn the threshold range.

ClearTrigger Method

Clear all trigger setting.

Syntax

object.ClearTrigger

Return values

If the trigger setting succeed, the return value is TRUE.

If the trigger setting fail, the return value is FALSE.

SetChTrigger Method

Set the trigger condition of the LA channel.

Syntax

object.SetChTrigger *lTrCh*, *lTrVal*

The **SetChTrigger** Method has these parts:

Part	Description
<i>object</i>	Required. An object expression that evaluates to an object in the Applies To list.
<i>lTrCh</i>	Required. Select the LA channel which you want to set.
<i>lTrVal</i>	Required. Decide LA Trigger value.

Return values

If the trigger setting succeed, the return value is TRUE.

If the trigger setting fail, the return value is FALSE.

Remarks

The valid Trigger value list as below:

TRIG_LOW, TRIG_HIGH, TRIG_LOW2HIGH, TRIG_HIGH2LOW,
TRIG_DONTCARE

Ex:

```
AxLaRun1.ClearTrigger           ' Clear All Trigger  
SettingAxLaRun1.SetChTrigger 0, TRIG_LOW2HIGH  ' CH0 : Low to High
```

ReadWaveform Method

Read data form LA.

Syntax

object.**ReadWaveform** *lSampleClock*, *lCh*

The **ReadWaveform** Method syntax has these parts:

Part	Description
<i>Object</i>	Required. An object expression that evaluates to an object in the Applies To list.
<i>lSampleClock</i>	Required. The zero-based index of the LA sample clock.
<i>lCh</i>	Required. The channel number of LA.

Return values

If the sample clock point's voltage is high level, return 1.

If the sample clock point's voltage is low level, return 0.