

AxLaRun.ocx

AxLaRun.ocx 是 LA 的 ActiveX 物件，但是他要搭配 LARUN.DLL 來執行。使用 VB 來開發 LA 的應用軟體時，只要將這個物件加入您的 Project 中，您就可以輕易的控制 LA。使用前，請先將 AxLaRun.ocx 和 LARUN.DLL 兩個檔案拷貝到您的目錄中，方可正常使用。或是將這兩個檔案拷貝到 Windows 的 System32 的目錄下。我們將提供一的 VB 的 Sample，它可以畫出一個 Channel 的波形。以下是所有 AxLaRun.ocx 的 Property 及 Method 的介紹:

AxLaRun.ocx Property

ExternalClock Property

設定採樣時脈是內部時鐘或是外部時鐘

Syntax

object.ExternalClock [= value]

The **ExternalClock** property 使用方法如下:

Part	Description
<i>object</i>	所使用的 LA Ocx 控制項
<i>value</i>	該值決定使用的工作時脈

Settings

The settings for *value* are:

Setting	Description
Bit 0	使用內部時鐘
Bit 1	使用外部時鐘
Bit 2	外部時鐘 = Channel 15
Bit 3	外部時鐘 = Channel 31
Bit 4	外部時鐘 = Channel 63
Bit 5	0: 採樣時鐘 = 外部時鐘 & 內部時鐘 1: 採樣時鐘 = 外部時鐘 # 內部時鐘
Bit 6	1: 採樣時鐘 = !外部時鐘
Bit 7	使用外部時鐘，通道 Channel 35(僅適用於 TravelLogic 系列)
Bit 8~Bit 15	外部時鐘模式，僅適用於 TravelLogic 系列，模式包含: !Channel 35、Channel 34 & Channel 35 和 Channel 34 # Channel 35

MemoryDepth

決定 LA 記憶體深度，單位為 bit

Syntax

object.MemoryDepth

Remarks

您可以設定這個值來改變 LA 的記憶深度。例如 PKLA1616 的記憶深度為 1M per Channel，您可以將這個記憶深度變小。但是這個值最小只能設成 256，而其最大值為硬體的最深記憶深度。每次執行 **GetHardware Method** 以後這個值都會被設定為 2.5K，如果要改變這個值，必須再 **GetHardware Method** 之後在 **CaptureData** 之前。

但是有一點必須注意的是，使用 LA1000P 系列及 PKLA 系列時，雖然可以改變 **MemoryDepth** 值，可是驅動程式還是會從 LA 硬體讀取完整的資料，也就是從硬體讀回資料所花的時間是不會改變的。

Passcount Property

設定 LA trigger pass count

Syntax

object.**Passcount** [= value]

The **Passcount** property 使用方法如下:

Part	Description
<i>Object</i>	所使用的 LA Ocx 控制項
<i>Value</i>	trigger pass count of LA

Settings

The settings for *value* are:

Setting	Description
0 ~ 104875	該值設定範圍為 0 to 104875

Remarks

LA1000 和 PKLA 1000 系列, 該值範圍為 0 to 255

LA2000 該值範圍 0 to 4095

TravelLogic 該值範圍為 0 to 1048575

PreTriggerMode Property

設定該值來決定是否使用 PreTrigger 功能

Syntax

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

The **PreTriggerMode** property syntax 使用方法如下:

Part	Description
<i>Object</i>	所使用的 LA Ocx 控制項
<i>Boolean</i>	由此傳回值決定是否使用 PreTrigger 功能

Settings

The settings for *boolean* are:

Setting	Description
True	使用 pre-trigger mode
False	不使用 pre-trigger mode

Remarks

只有 LA2000 和 TravelLogic 可以使用該功能. LA1000 和 PKLA1000，您可以設定該值為 **False**。

TrDelay Property

設定 LA 觸發延遲

Syntax

object.TrDelay [= value]

The **TrDelay** property 使用方法如下：

Part	Description
<i>Object</i>	所使用的 LA Ocx 控制項
<i>Value</i>	觸發延遲時間

Remarks

只有 LA2000 可以使用該功能，該值範圍請參考 LA 的使用手冊，而 LA1000，TravelLogic 和 PKLA1000 則無法使用該功能，該值填入 0。

TrMode Property

設定 LA 觸發模式

Syntax

object.**TrMode** [= value]

The **TrMode** property syntax 使用方法如下:

Part	Description
<i>Object</i>	所使用的 LA Ocx 控制項
<i>Value</i>	使用的觸發模式，參考下表

Settings

The settings for *value* are:

Constant	Setting	Description
TR_SINGLE	0	單階觸發模式
TR_MULTI	1	多階觸發模式
TR_DUAL	2	雙條件觸發模式
TR_WIDTH	3	觸發寬度模式
TR_NONE	4	關閉觸發功能

Remarks

在 LA1000 和 PKLA 1000 系列，只能使用 TR_SINGLE 和 TR_NONE 模式，

而使用 TR_SINGLE 模式時，可以用 **SetChTrigger Method** 來設定觸發條件，TR_NONE 模式是代表將 Trigger 功能關閉。詳細請參考 LA 手冊。

TrPosition Property

設定 LA 觸發位置

Syntax

object.**TrPosition** [= value]

The **TrPosition** property syntax 使用方法如下:

Part	Description
<i>Object</i>	所使用的 LA Ocx 控制項
<i>Value</i>	LA 觸發位置數值

Remarks

這個值必須由 **MemoryDepth** 來決定，但是這個值的設定在有些地方要注意，例如 **MemoryDepth** 等於 65536 時，這個值只能設成 12 到 65536-12，也就是 Trigger Position 不能設定在最前及最後 12Clock 的位置上。

TrWidth Property

設定 LA 觸發寬度

Syntax

object.**TrWidth** [= value]

The **TrWidth** property syntax 使用方法如下:

Part	Description
<i>Object</i>	所使用的 LA Ocx 控制項
<i>Value</i>	觸發寬度值

Remarks

該功能只適用於 LA2000 和 TravelLogic，而 LA1000 和 PKLA1000 只能設定該值為 0。要使用該功能之前，必需將 **TrMode** property 設為 **true**。

AxLaRun.ocx Method

GetHardware Method

偵測 LA 硬體是否存在

Syntax

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

The **GetHardware** Method syntax 使用方法如下:

Part	Description
<i>object</i>	所使用的 LA Ocx 控制項
<i>lModel</i>	為 LA's ID，Long 資料型態，若為 0，則會自動偵測 ID
<i>lSlot</i>	設為 0

Return values

若 LA 存在，則返回值為 TRUE。

反之，則為回傳 FALSE。

Remarks

除非您有多台 LA 時，才需要設定 *lModel* 及 *lSlot* 的參數，否則都填 0 即可，AxLaRun 的 Object 在還沒執行 **GetHardware** Method 之前，不能使用其他的 Method。

GetModel Method

取得 LA 型號名稱

Syntax

object.**GetModel**

Return values

GetModel Method 會傳回型號名稱字串 (BSTR)。

Remarks

Ex:

```
Dim S as string
```

```
S = AxLaRun1.GetModel
```

SetHwMode Method

設定 LA 硬體模式來選擇採樣率及記憶體堆疊使用方式。

Syntax

object.SetHwMode *IMode*

The **SetHwMode** Method syntax 使用方法如下:

Part	Description
<i>object</i>	所使用的 LA Ocx 控制項
<i>IMode</i>	選擇採樣率及記憶體堆疊方式，參考下表。

Return values

SetHwMode Method 設定成功則回傳 True。

反之，則回傳 False。

Remarks

SetHwMode Method 設定成功後，LA 的 **MemoryDepth** 會被自動調整到最深的記憶體深度，當然，您可以再使用 **MemoryDepth** 來調整記憶體深度。

SetHwMode Method 必須使用在 **GetHardware** Method 之後，**Capture** Method 之前。

注意: **SetHwMode**，**MemoryDepth** 和 **TrPosition** 是息息相關的，這關係到 TravelLogic series 觸發位置的正確與否。例如使用 `AxLaRun1.SetHwMode(504)` 在下表中對應到可使用通道數為 6，這表示目前使用的是 x6 的記憶體堆疊方式 (36/6)，所以 **MemoryDepth** 和 **TrPosition** 必須是 6 的倍數，才可以得到正確的觸發位置，如: **MemoryDepth = 3000**，**TrPosition = 60** 其他依此類推。

lMode	可使用的採樣率	可使用通道數
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

*該表適用於 TravelLogic 系列

Ex1:

// 以下皆以 TravelLogic 2x36 系列為例

```
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
```

lMode	可使用的採樣率	可使用通道數
0	<= 200MHz	16
1	<= 400MHz	16

*該表適用於 PKLA1616+

Ex1:

// 以下皆以 PKLA1616+ 為例

```
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
```

lMode	可使用的採樣率	* 可 使 用 該 通 道 數
0	<= 100MHz	16
1	<= 200MHz	適 16

*該表適用於 PKLA1x16 系列

Ex1:

// 以下皆以 PKLA1x16 為例

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

```
AxLaRun1.SetFrequency(100000000) // set sampling rate = 100MHz
```

```
AxLaRun1.MemoryDepth = 2000 // default MemoryDepth = 512K(PKLA1616)
// reset MemoryDepth = 2K
```

Ex2:

```
AxLaRun1.SetHwMode(1)
```

```
AxLaRun1.SetFrequency(200000000) // set sampling rate = 200MHz
```

```
AxLaRun1.MemoryDepth = 2500 // default MemoryDepth = 1M(PKLA1616)
// reset MemoryDepth = 2.5K
```

lMode	可使用的採樣率	可使用通道數
0	<= 100MHz	32
1	<= 200MHz	32

*該表適用於 LA1x32、LA1x32P、LA2x32、LA2x32P

Ex:

```
AxLaRun1.SetHwMode(0)
AxLaRun1.SetFrequency(100000000) // Set sampling rate = 100MHz
AxLaRun1.MemoryDepth = 2000      // reset MemoryDepth = 2K
```

lMode	可使用的採樣率	可使用通道數
0	<= 100MHz	64
1	<= 200MHz	32

*該表適用於 LA1x64、LA1x64P、LA2x64、LA2x64P

Ex:

```
AxLaRun1.SetHwMode(1)
AxLaRun1.SetFrequency(200000000) // Set sampling rate = 200MHz
AxLaRun1.MemoryDepth = 2000      // reset MemoryDepth = 2K
```

SetFrequency Method

設定 LA 的採樣率

Syntax

object.SetFrequency *IFreq*

The **SetFrequency** Method syntax 使用方法如下：

Part	Description
<i>object</i>	所使用的 LA Ocx 控制項
<i>IFreq</i>	LA 的採樣率

Return values

SetFrequency Method 設定成功則回傳 True。

反之，則回傳 False。

Remarks

您可以設定以下的頻率值：100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000, 200000, 500000, 1000000, 2000000, 5000000, 10000000, 20000000, 50000000, 100000000, 200000000 (適用LA1000, LA1000P, LA2000, LA2000P, PKLA1000), 400000000 (適用PKLA1616+), 800000000, 1600000000, 2000000000, 4000000000。在使用**SetFrequency** Method之前，必需先使用**SetHwMode** Method來設定硬體，使用方法請參考**SetHwMode** Method一節。

Shutdown Method

關掉 LA 硬體

Syntax

object.**Shutdown**

Return values

無傳回值。

Remarks

當要結束程式時，請記得執行 **Shutdown Method**。讓 LA 進入關機狀態，如果您使用了 **Shutdown Method** 之後，除了 **GetHardware** 以外，其他的任何 **Method** 都不能使用。

Reset Method

Reset LA

Syntax

object.**Reset**

Return values

無傳回值。

Remarks

如果您執行了 **Capture Method**，而且 **ReadStatus Method** 一直無法回應 True，那就代表 LA 一直無法收到 **Trigger** 訊號。此時如果您不想要這次擷取結果，只要執行 **Reset Method** 後就可以重新執行 **Capture Method** 了。

Capture Method

LA 擷取資料

Syntax

object.**Capture**

Return values

無傳回值。

Remarks

使用 **Capture Method** 來擷取波形時,要記得先執行 **SetChTrigger Method** 來設定觸發條件。

ReadStatus Method

LA 是否完成擷取動作

Syntax

object.**ReadStatus**

Return values

LA 擷取成功，返回值為 TRUE。

反之，返回值為 FALSE。

Threshold Method

設定 LA threshold

Syntax

object.Threshold *lPod*, *lMiniVolt*

The **Threshold** Method 使用方法如下:

Part	Description
<i>object</i>	所使用的 LA Ocx 控制項
<i>lPod</i>	Required. 選擇要設定的 Pod,如果要全部設定,請將此值填入-1
<i>lMiniVolt</i>	Required. 填入 Threshold 的電壓值,單位為毫伏(mV)

Return values

無回傳值。

Remarks

參考電壓可以設定的值,請參考各產品手冊,每個 Pod 所代表的 Channel 數,也會依不同產品型號而不同。

ClearTrigger Method

清除所有的觸發條件設定

Syntax

object.ClearTrigger

Return values

觸發條件設立成功，則回傳 TRUE。

反之為 FALSE。

SetChTrigger Method

設定 LA 通道的觸發條件

Syntax

object.SetChTrigger *lTrCh*, *lTrVal*

The **SetChTrigger** Method 使用方法如下:

Part	Description
<i>object</i>	所使用的LA Ocx 控制項
<i>lTrCh</i>	Required. 選擇您想要設定的LA通道
<i>lTrVal</i>	Required. 決定該通道的觸發條件值

Return values

觸發條件設立成功，則回傳 TRUE。

反之為 FALSE。

Remarks

Trigger 的值可設定成:

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

從 LA 取得波形資料

Syntax

object.ReadWaveform *lSampleClock*, *lCh*

The **ReadWaveform** Method syntax 使用方法如下:

Part	Description
<i>Object</i>	所使用的 LA Ocx 控制項
<i>lSampleClock</i>	LA 採樣樣本點索引
<i>lCh</i>	欲取得 LA 該通道的波形資料之通道編號

Return values

該採樣點電位為 **High**，返回 **1**。

反之該採樣點電位為 **Low**，返回 **0**。