Advantech AE Technical Share Document

Date	2017/6/15	**SR#**	1-2916164315
Category	□ FAQ ■ SOP	**Related OS**	Windows
Abstract	Using Encoder Input as A/D Conversion Clock on Advantech PCIE-1812		
Keyword	DAQNavi, PCIE-1812, Continue Compare, Analog Input		
Related Product	PCIE-1812		

PCIE-1812 is a product with 4 encoder inputs designed. The encoders can act as the pacer clock (convert clock) by setting the Continue Compare function. The counter would send the signal triggering the ADC(s) as the compare value is reached then a conversion would be taken place. This SOP will show you how to approach a continue-compare-triggered analog acquisition. It's recommended to use this function with DAQNavi 4.0 SDK.

■ Brief Solution - Step by Step:

1. Analog Input Setting
   a) Convert clock source >> Counter N's output
   b) Channel to scan >> Set the start channel and channel count.
   c) Set Section Length that triggers the DataReady event.
   d) Set trigger if necessary.
   e) Add a DataReady event callback in the program for retrieving the data.

C#:

```
WfAiCtrl.Conversion.ClockSource = SignalDrop.SigCntOut0;
WfAiCtrl.Conversion.ChannelStart = startChannel;
WfAiCtrl.Conversion.ChannelCount = channelCount;
WfAiCtrl.Record.SectionLength = sectionLength;
WfAiCtrl.DataReady += new EventHandler<BfdAiEventArgs>(wfAiCtrl_DataReady);
```

C++:

```
Conversion * conversion = WfAiCtrl->getConversion();
conversion->setClockSource(SigCntOut0);
conversion->setChannelStart(startChannel);
conversion->setChannelCount(channelCount);
WfAiCtrl->addDataReadyHandler(OnDataReadyEvent, NULL);
```

2. Updown Counter Setting
   a) Set compare table or set compare interval to the channels respectively.
   b) Enable the Compare then the counter compare will get started.
   c) Add a ConCompareValue event callback if necessary.

C#:

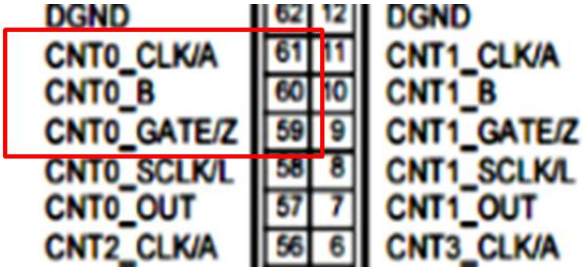
```
// Set Compare Interval
ret = udCounterCtrl.CompareSetInterval(channel, Increment, Count);
// Start Compare
udCounterCtrl.Enable = true;
```

C++:

```
// Set Compare Interval
ret = udCounterCtrl->CompareSetInterval(channel, Increment, Count);
// Start Compare
ret = udCounterCtrl->setEnable(true);
```

3. Signal Connection

   Connect corresponded signal to the pin as phase A, B and Z respectively.



4. Start Acquisition

C#:

```
WfAiCtrl.Prepare();
WfAiCtrl.Start();
```

C++:

```
WfAiCtrl->Prepare();
WfAiCtrl->Start();
```

■ **Reference**: