

## When to use External Trigger vs. External Clock

### Preface

This note is intended to clarify some of the issues you may need to consider when an application requires some sort of synchronization with external signals. Even though the discussion here is intended for Keithley KPCI-3100 family (KPCI-3101/02/03/04 or KPCI-3110/16 series) boards, most of the considerations are generally applicable to all plug-in board products.

Consider a shaft encoder. Often the encoder will provide both a continuous stream of pulses as a function of rotation as well as a single pulse for each complete rotation (index pulse). One use case would be to acquire one or more data point with the plug-in board in response to each index pulse. Do you connect this pulse to the trigger or to the external clock of the board?

Another use case would be acquisition of one or more data points in response to each rotation pulse from the encoder to deliver some data relative to position. Again, do you connect to the external clock or to the trigger of the board?

The answer depends on the frequency of the external signals and the amount of data to be acquired in response to each external pulse.

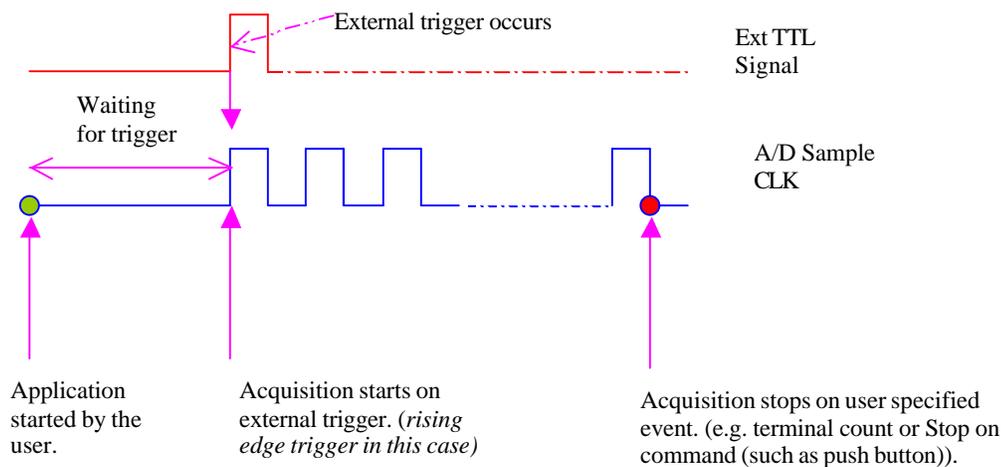
### External Triggers

When using a digital trigger, the task is defined to use a DI\_EVENT as the start type. However, in order for the board to be sensitive to this trigger, the task must be armed (active). When the trigger occurs, the specified numbers of samples are acquired according to the specified rate until the stop condition is satisfied.

In order to be able to respond to a new trigger pulse, the task must both complete and be re-armed. This puts a burden onto the controlling software to monitor the state of the task (active or not) and to re-arm the task before the next trigger occurs.

Each task can have a few milliseconds of "overhead" in addition to the actual time required for the acquisition. This limits the repeat rate of the external trigger to less than 500Hz and will have computer-to-computer variation in the actual number.

The diagram below represents the relationship between arming the task, the occurrence of the trigger and the start of acquisition.



### Post-Trigger Acquisition.

## When to use External Trigger vs. External Clock

Below is some VB code for setting up a task that will start on the rising edge of a digital trigger, and then acquire 64 samples (two buffers of 32 samples) from 4 channels at a rate of 200 Hz. When both buffers have been filled, the task will stop automatically (TCEVENT).

This code snippet assumes the driver has already been loaded and the hardware initialized. It does not show processing of the data in the BufferFilled event procedure nor use of the StartEvent message that will be sent when the trigger does occur. See the DriverLINX Tutorial Manual (web site document center) for a more complete coding discussion in both VC++ and VB, or check the download center of the web site for available example programs.

*Note:*

- *The example Visual Basic 6.0 code provided here is written for KPCI-3110. If you are using a different board please reference the corresponding DriverLINX manual, as some service request parameters may be different.*

## When to use External Trigger vs. External Clock

*VB Example for rising edge **Post Trigger** Acquisition with external TTL signal (Written for KPCI-3110)*

With SR\_AI

.Req\_subsystem = DL\_AI

.Req\_mode = DL\_DMA

.Req\_op = DL\_START

'Set up the digital trigger. Acquisition starts on the rising edge of external TTL signal

'Connect the trigger signal on AD TTL Trigger line (Pin 77 of STP-3110)

.Evt\_Str\_type = DL\_DIEVENT

.Evt\_Str\_diChannel = 3 'use 3 for KPCI-3110/16 & 4 for KPCI-3101/02/03/04

.Evt\_Str\_diMask = 1

.Evt\_Str\_diMatch = DL\_NotEquals

.Evt\_Str\_diPattern = 0 '0 = rising edge, 1 = falling edge

'-----

' SR set up for Paced Acquisition with 4 channels at 200Hz

.Evt\_Stp\_type = DL\_TCEVENT

.Evt\_Tim\_type = DL\_RATEEVENT

.Evt\_Tim\_rateMode = DL\_RATEGEN

.Evt\_Tim\_rateChannel = DL\_DEFAULTTIMER

.Evt\_Tim\_rateClock = DL\_INTERNAL1

' use internal clock

.Evt\_Tim\_ratePeriod = .DLSecs2Tics(DL\_DEFAULTTIMER, 1/200) ' major period

.Evt\_Tim\_rateGate = DL\_DISABLED

.Sel\_chan\_format = DL\_tNATIVE

.Sel\_chan\_N = 2 'means it will have start and stop channels, does NOT mean 2 channels

.Sel\_chan\_start = 0 'start channel

.Sel\_chan\_startGainCode = .DLGain2Code(-1) '-1 means bipolar input, unity gain

.Sel\_chan\_stop = 3 'stop channel

.Sel\_chan\_stopGainCode = .DLGain2Code(-1)

.Sel\_buf\_N = 2 '# of buffers

.Sel\_buf\_samples = 32 '# samples per buffer

.Sel\_buf\_notify = DL\_NOTIFY + DL\_NOTIFY\_START

.Res\_Sta\_typeStatus = DL\_IOSTATUS ' status about the operation

.Refresh

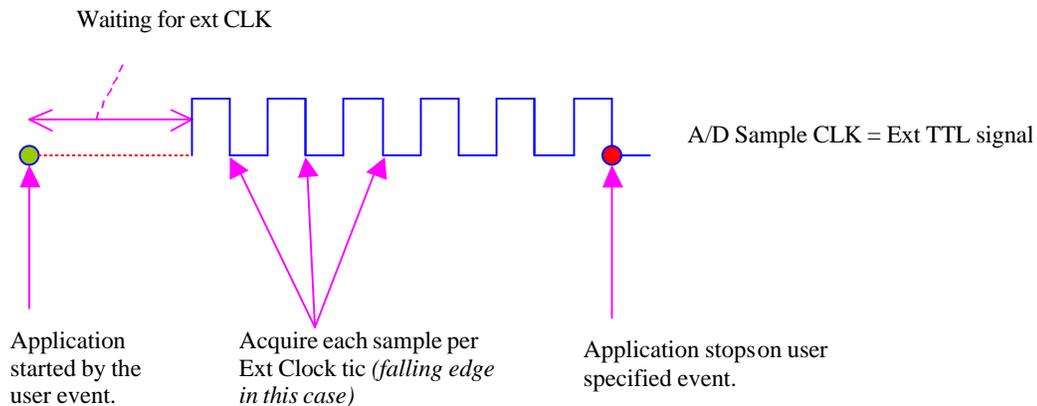
End With

## When to use External Trigger vs. External Clock

### External Clocks

There are two scenarios to consider: Suppose the application requires one sample in response to an external pulse train. In this case, a straightforward use of external clock is called for. In a second situation, suppose the application requires multiple samples in response to each external pulse. In this case, an externally clocked burst mode type of acquisition should be used. This will effectively yield the same type of data pattern as the start on digital trigger task provides, but the task remains active between successive pulses so can be operated at much higher repeat rates.

**One Sample per External Pulse:** Use this when you want the A/D to acquire one sample for each external clock pulse. Selection of rising or falling edge of the external clock is supported by some hardware. However, boards covered by the KPCI3100 driver uses only the falling edge of the external clock.



### External Rate Clock

Below is some VB code for setting up a task that will acquire one data point for each external clock pulse. A total of 128 samples will be acquired (two buffers of 64 samples) from 4 channels at an estimated rate of 200 Hz. When both buffers have been filled, the task will stop automatically (TCEVENT).

This code snippet assumes the driver has already been loaded and the hardware initialized. It does not show processing of the data in the BufferFilled event. See the DriverLINX Tutorial Manual (web site document center) for a more complete coding discussion in both VC++ and VB, or check the download center of the web site for available example programs.

**Note:**

- *The KPCI-3101/2/3/4 boards transfer data in no fewer than 16 samples. The KPCI-3110 and KPCI-3116 transfer data in no fewer than 32 samples. Therefore, the external clocking source should not suddenly stop at less than a multiple of the transfer size else data will be stranded in the board's FIFO.*
- *Even though the external clock controls the sample rate, DriverLINX requires an estimated rate (see the code snippet).*

## When to use External Trigger vs. External Clock

*VB Example for External Rate Clock Acquisition (Written for KPCI-3110)*

With SR\_AI

.Req\_subsystem = DL\_AI

.Req\_mode = DL\_DMA

.Req\_op = DL\_START

.Evt\_Str\_type = DL\_COMMAND

.Evt\_Stp\_type = DL\_TCEVENT

.Evt\_Tim\_type = DL\_RATEEVENT

.Evt\_Tim\_rateMode = DL\_RATEGEN 'choose the rate clock

.Evt\_Tim\_rateChannel = DL\_DEFAULTTIMER

.Evt\_Tim\_rateClock = DL\_EXTERNAL

'You need to connect your external CLK source to External A/D Clock line(Pin 76 of STP-3110)

'When you click start, the board will wait for the falling edge of external CLK Signal.

.Evt\_Tim\_ratePeriod = .DLSecs2Tics(DL\_DEFAULTTIMER, 1 / 200)

' This value is NOT actually used. However, DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.

.Evt\_Tim\_rateGate = DL\_DISABLED

.Sel\_chan\_format = DL\_tNATIVE

.Sel\_chan\_N = 2 ' There will be start and stop channel

.Sel\_chan\_start = 0 'start scanning ch 0

.Sel\_chan\_startGainCode = .DLGain2Code(-1) '-1 means bipolar input, unity gain

.Sel\_chan\_stop = 3 'stop at channel 3

.Sel\_chan\_stopGainCode = .DLGain2Code(-1)

.Sel\_buf\_N = 2 '# of buffers

.Sel\_buf\_samples = 64 '# of samples per buffer

.Sel\_buf\_notify = DL\_NOTIFY ' send buffer filled messages

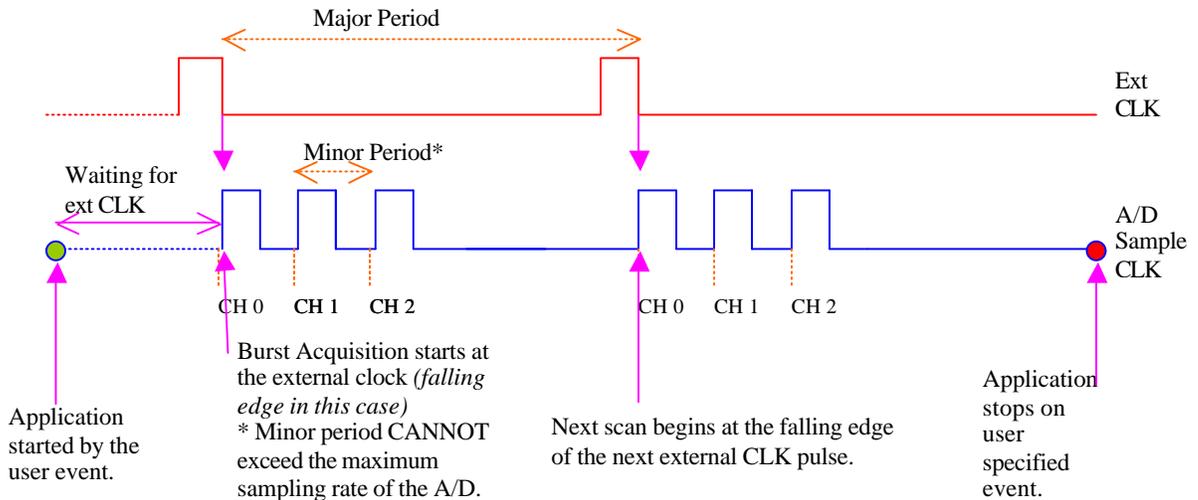
.Res\_Sta\_typeStatus = DL\_IOSTATUS

.Refresh

End With

## When to use External Trigger vs. External Clock

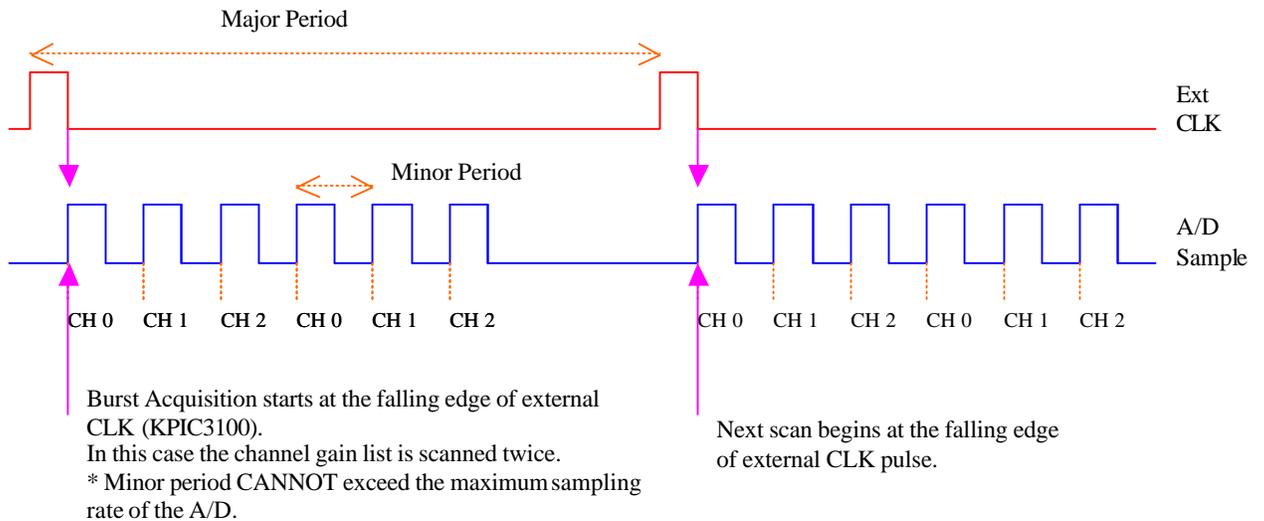
**Multiple Samples per External Pulse:** Use this when you want to synchronize scanning through several channels (or multiple samples from a single channel) with a recurring external pulse. For example, you can use External Burst Clock when you want to scan channel 0,1 & 2 every time there is an external TTL pulse.



### External Burst Clock

In the above diagram, major period of the burst is provided by the external clock tic, and the on-board clock at a specified rate provides the minor period (faster burst clock). KPCI-3100 series boards have an additional feature when using this external burst clock that allow it to repeat the scan up to 256 times. However, as with all burst mode acquisitions, the burst mode scan must be completed before the next external clock tic occurs.

**Example:** the following example (written for KPCI-3110/16) shows scanning the selected channels (channel 0 to 2) twice, in VB6.0.



## When to use External Trigger vs. External Clock

VB example for **External Burst Clock** with 2 scans through channels. *(Written for KPCI-3110)*

```

With SR_AI
.Req_subsystem = DL_AI
.Req_mode = DL_DMA
.Req_op = DL_START
.Evt_Str_type = DL_COMMAND
.Evt_Stp_type = DL_TCEVENT 'stop when all buffers have been filled
.Evt_Tim_type = DL_RATEEVENT
.Evt_Tim_rateMode = DL_BURSTGEN 'choose the burst clock
.Evt_Tim_rateChannel = DL_DEFAULTTIMER

.Evt_Tim_rateClock = DL_EXTERNAL
'You need to connect your external CLK source to AD TTL Trigger line (Pin 77 of STP-3110), not the
'external Clock line. Hardware will only work on the falling edge of the external signal.

.Evt_Tim_ratePeriod = .DLSecs2Tics(DL_DEFAULTTIMER, 1/200)
' This value is NOT actually used. However, DriverLINX requires a valid hardware value in case the
'application requests a timebase operation and to optimize data transfer between the
'driver and the application.

.Evt_Tim_ratePulses = 6 ' Total six burst tics, i.e. 2 scans for the 3 channels.
.Evt_Tim_rateOnCount = .DLSecs2Tics(DL_DEFAULTTIMER, 1 / 100000)
'Note that the minor period cannot exceed the maximum sampling rate of the A/D.

.Evt_Tim_rateGate = DL_DISABLED
.Sel_chan_format = DL_tNATIVE
.Sel_chan_N = 2 ' There will be start and stop channel
.Sel_chan_start = 0 ' start at channel 0
.Sel_chan_startGainCode = .DLGain2Code(-1) ' with bipolar gain of 1
.Sel_chan_stop = 2 ' stop at channel 2; total of 3 channels used
.Sel_chan_stopGainCode = .DLGain2Code(-1) ' with bipolar gain of 1
.Sel_buf_N = 3 ' 3 buffers
.Sel_buf_samples = 12 ' samples per buffer, two external pulses required for 12 samples
.Sel_buf_notify = DL_NOTIFY
.Res_Sta_typeStatus = DL_IOSTATUS
.Refresh
End With

```

## Conclusion

Selection of digital trigger or external clock (burst mode) is an important consideration when trying to synchronize acquisition from one or more channels with an external pulse. When the pulse will repeat at a fast rate, use of external clock (burst mode) is a more robust implementation than use of a digital trigger to start the acquisition.

If you are not sure what method would work for your particular need, it is always a good idea to sketch out a timing diagram and see if it can be applicable to any of the topics discussed above.