Getting started with your new Flex02-01D Rev C USB 2.0 based Digital correlator

Introduction:

Flex02 is a multiple function instrument from Correlator.com. When used as a correlator, Flex02 operates as a multiple-tau correlator or a linear correlator. In addition, Flex02 can optionally save the intensity series at 100 ns and above sample times on the PC hard drive. In photon history recorder mode, it transfers photon arrival time series to the host PC hard drive.

System requirements:

A USB 2.0 enabled PC. Windows XP comes with native USB 2.0 support. For Windows 2000/ME/98, a PCI USB2.0 adapter card with the software support may be necessary.

Correlator specifications:

A. Multiple Tau modes (3 modes)

In all multiple tau correlation modes, the intensity history of both channels may be saved to the hard drive. There are two ways of saving the intensity

a) Number of photons per sample time series
1. Minimum sample time 100 ns
2. Sample time increment: 25 ns
3. Maximum sample time: 65535 * 25ns = 1638375ns
4. Real time histogram are calculated.

When this option is enabled in Flex02-01dc.exe, the data file is saved as straight binary in the file, trace.dat, in the program directory. In the case of two channels recording, the data for two channels are interlaced.

b) Photon arrival time series
1. Master clock rate 640MHz

When this option is enabled in Flex02-01dcts.exe, the data file is saved in the file, trace_a.dat and trace_b.dat, in the program directory. The arrival time is recorded in a 16 bit format as follows

bit 3 … bit 0	Photon arrival time	2nd Photon arrival time	3rd Photon arrival time	4th Photon arrival time
0000	No Photon, add 4096*4 to the next WORD			
0001	1 photon, {Bit 15…Bit4}*4+1			
0010	{Bit 15…Bit4}*4+2			
0011	{Bit 15…Bit4}*4+1	1		
0100	{Bit 15…Bit4}*4+3			
0101	{Bit 15…Bit4}*4+1	2		
0110	{Bit 15…Bit4}*4+2	1		
0111	{Bit 15…Bit4}*4+1	1	1	
1000	{Bit 15…Bit4}*4+4			
1001	{Bit 15…Bit4}*4+1	3		
1010	{Bit 15…Bit4}*4+2	2		

1011	{Bit 15…Bit4}*4+1	1	2	
1100	{Bit 15…Bit4}*4+3	1		
1101	{Bit 15…Bit4}*4+1	2	1	
1110	{Bit 15…Bit4}*4+2	1	1	
1111	{Bit 15…Bit4}*4+1	1	1	1

Mode A. Single auto/cross correlator.

1. Minimum sample time: 1.56ns
2. Auto or cross correlation. (AxA or AxB)
3. Total number of channels: 1088 each
4. Sample times:
    5. First 64 channels : 1.56ns
    6. next 32 channels: 2x1.56ns
    7. Sample times double every 32 channels.
    8. Total of 34 segment
9. Delay times: from 1.56ns to36.7 minutes.
10. Shift register width: in the order of the segments 1,2,3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15,32, …32.
11. Maximum intensity before overflow: 640 MHz within 1.56ns, 44 MHZ continuos.

Mode B: Dual auto/cross correlator.

1. Minimum sample time: 1.56ns
2. Auto or cross correlation. (AxA, BxB, or AxB BxA)
3. Total number of channels: 608 each
4. Sample times:
    5. First 32 channels : 1.56ns
    6. next 16 channels: 2x1.56ns
    7. Sample times double every 16 channels.
    8. Total of 38 segment
9. Delay times: from 1.56nsns to 586 minutes.
10. Shift register width: in the order of the segments 1,2,3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15,32, …32.
11. Maximum intensity before overflow: 640 MHz within 1.56ns, 44 MHZ continuos.

Mode C: Quad correlator.

1. Minimum sample time: 1.56ns
2. Auto and cross correlations. (AxA, BxB, AxB, BxA)
3. Total number of channels: 288
4. Sample times:
    5. First 16 channels : 1.56ns
    6. next 8 channels: 2*1.56ns
    7. Sample times double every 8 channels.
    8. Total of 35 segment
9. Delay times: from 1.56ns to 28.6 minutes.
10. Shift register width: in the order of the segments 1,2,3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15,32, …32.
11. Maximum intensity before overflow: 640 MHz within 1.56ns, 44 MHZ continuos.

A. Linear modes

In linear mode, the correlator changes speed depends on the sample time. It takes a couple of seconds to switch between different speed.

1. High speed operation. 1bit register width, Sample time =1.56ns, total number of data points 512. 100% real time.

2. Mid speed operation: 1bit register width. Sample time = 8ns., total number of data points 1024. 100% real time.
3. Standard operation: 9 bits register width, Sample time adjustable from 10ns to 1ms in increment of 10ns. total number of data points 2048.
4. Low speed operation: 32 bits register width, Sample time adjustable from 1ms in increment of 10ns. 2048 points
5. Real time intensity data can be saved for sample times > 100ns.

Photon history recorder specifications:

Photon history recorder transfer the time difference between successive photon events to the host PC hard drive. In this mode, the device needs the full attention of the host PC. Running any background operation in concurrent with the PHR is not recommended.

Single channel mode (channel A)
1. Transfer the time difference between successive photon events to host PC hard drive
2. System clock speed: 60 MHz
3. FIFO size: 5K.
4. Pulse pair resolution: 16.7 ns
5. Maximum count rate before overflow: It's not limited by the USB 2.0 bus. It depends on the host PC configurations.

Dual channel mode (channel A and B)
1. Transfer the time difference between successive photon events to host PC hard drive
2. System clock speed: 60 MHz
3. FIFO size: 2.5K.
4. Pulse pair resolution: 16.7 ns
5. Maximum count rate before overflow: It's not limited by the USB 2.0 bus. It depends on the host PC configurations.

Hardware Specifications:

1. Electrical interface: TTL
2. Maximum input range: -0.3 to 3.6 v.
3. Input impedance: 50 ohm.
4. Physical dimensions: 2.5"x2.75"x0.8".
5. Maximum power consumption: 1.5W

Caution:

The correlator inputs should be at zero volt when the correlator is not powered up. Please turn off the photon detector when the host PC is off.

Installations:

For easy installation, please follow the procedures in exact order to install the software.
1. Run Setup.exe to install the software.
2. After the software is installed, plug in the correlator hardware.

Using your Flex02--01D correlator in multiple tau correlation mode:

The correlator is very easy to use. Most of the graphical objects are clickable. Correlation functions can be displayed in data sheets format. Please experiment with the program. The following steps help you to get started.

1. Make sure the correlator is attached to the computer.
2. Double click on Flex02-01dc.exe to launch the program.
3. Click H button to set the sample time.
4. Check Setting|Save Photon History to turn on the option to save the photon history.
5. Select the desired duration time in seconds.
6. Select the desired correlator mode, Single Auto, Single Cross, Dual Auto, Dual Cross, Quad
7. Click on Run button to start the correlator.
8. After the correlator stops, save the data and the parameters into a file.
9. To use the scripting capability, click on "S" button on the tool bar to bring the script window to foreground. Click open button to choose the sample script. Click on Execute button to run the script.

Script command list:

1. // : command line
2. SAuto : Auto AxA mode
3. SCross : AxB mode
4. Quad : Quad mode
5. DAuto : AxA, BxB
6. DCross : AxB, BxA
7. Duration *x* : Set duration time for *x* (integer) seconds
8. Sleep *x* : Do nothing for *x* (integer) seconds
9. AutoSave xyz : Save Data with data file xyz000.sin, xyz001.sin,…, xyz can't be "off".
10. AutoSave off : Turn off Autosave mode
11. Start : Start the correlator
12. MultipleRun *x* : Run correlator *x* (integer) times.
13. Sum : Sum of previously saved files
14. SaveIntensity On : Turns on the save intensity option
15. SaveIntensity Off : Turns of the save intensity option
16. Do I=1 to n ; run the script between Do and Loop n times
17. Loop ;

Caution:

1. Make sure the correlator is plugged in before you run Flex01D-08.exe.
2. Don't disconnect the correlator while the software is running.
3. Exit the software first before disconnect the correlator hardware.

Using your Flex02 correlator in photon history recorder mode:

1. Make sure the correlator is attached to the computer.
2. Double click on Photon.exe to launch the program.

3. In the Control page, you can choose or type the name of the file to save the times series. Choose 8 bit file format for high intensity (> 250KHz) or 16 bit format for lower intensity.
4. Select duration time for desired experimental time in seconds.
5. Pick a sample time for real time display graph in the graph page. Sample time cannot be larger than 60 us.
6. Click on Start button to recorder the photons.

Raw data file format:

1. The file records the difference in system clock ticks (1/60 us) between photon event.
2. The first byte identifies the format of the file 8 : 8 bit, 16: 16 bit
3. The second byte identifies the system clock. 60MHz.
4. The time unit is 1/system clock.
5. 16 bit format. Each WORD (2 bytes) represents a photon event, time = WORD/system clock, unless the value is 0xFFFF, in which case, the following four bytes represent a photon event.
6. 8 bit format: Each BYTE represents a photon event unless the value is 0xFF, in which case, the BYTE means 255 clock ticks passed without a photon event. For example 0A 0B FF 08 means there are three photon events. The time series are 0x0A+1, 0x0B+1, 0xFF+8+1.

Utilities and its file format:

The photon software offers 4 utilities.

1. Histogram: Click "Browse" button to select the raw data file. Type in the sample time. Click this button to generate the histogram in ASCII file. The numbers represent the number of occurrence with zero photons, one photon, two photons…to maximum of 0x3FFF photons.
2. Convert to ASCII: Click this button to the raw data file to ASCII format.
3. Convert to 32 bits: Click this button to convert the raw data file to 32 bits binary format. Each DWORD (four bytes) represents successive photon arrival time.
4. Convert to Intensity: Click this button to convert the raw data file to intensity series in WORD (2 bytes) format. The intensity element is the number of photons per sample time.

Programming your Flex02 correlator:

a) Flex02-01dc.dll. provides functions for saving photon numbers per sample time. There are 6 functions

1. _USBInitialize(BYTE mode) // Always initialize the correlator before use,

mode: 'A' single correlation; 'B' quad correlation; 'C' dual correlation.

2. _USBStart(BYTE mode, WORD Sample, BYTE format, char *filename)

// Argument 1: 0 auto, 1: cross.
// Argument 2: sample time multiple for the intensity series. The unit is 16ns
// Argument 3: 0 16 bit format, 1 8 bit format for the intensity series.
// Argument 4: The file name for the intensity series. If NULL, the intensity is not saved._

3. USBStop() // Stop
4 USBFree() // Call this function to clean up before exit.
5. USBUpdate(float *ElapsedTime, int TraceCount, float *corr1, float *corr2, float *corr3, float *corr4, float *traceA, float *traceB) // Call this function to get correlation functions and intensity histories (A and B) while the correlator is running.

*ElapsedTime: The running time since _USBStart is called.
*TraceCnt: The number of elements in traceA and traceB
corr1: pointer to an array of 1120 elements which contains the normalized correlation function accumulated over *ElapsedTime.

corr2: pointer to an array of 624 elements which contains the normalized correlation function accumulated over *ElapsedTime.
corr3: pointer to an array of 304 elements which contains the normalized correlation function accumulated over *ElapsedTime.
corr4: pointer to an array of 304 elements which contains the normalized correlation function accumulated over *ElapsedTime.
traceA: pointer to an array of intensity elements for channel A. The number of element is *TraceCnt. Each element is the number of photons within an interval of 0.1048576 second. This array is reset after _USBUpdate(…) is called.
traceB: traceB returns intensity history for channel B.
In Dual mode: corr3 and corr4 are ignored
In Single mode: corr2, corr3, and corr4 are ignored.

6. USBUpdateRawdata(float *ElapsedTime, int TraceCount, double *rawcorr, double *samples, double *baseA, double *baseB, float *traceA, float *traceB, double *HistogramA, double *HistogramB) // Call this function to get raw correlation functions data while the correlator is running.

*ElapsedTime: The running time since _USBStart is called.
*TraceCnt: The number of elements in traceA and traceB
rawcorr: pointer to an array of 1248 elements which contains the raw correlation function accumulated over *ElapsedTime.
samples: pointer to an array of 624 elements which contains the number of samples accumulated over *ElapsedTime.
baseA: pointer to an array of 1248 intensity elements accumulated over *ElapsedTime.
baseB: pointer to an array of 1248 intensity elements accumulated over *ElapsedTime.
traceA: pointer to an array of intensity elements for channel A. The number of element is *TraceCnt. Each element is the number of photons within an interval of 0.052428799999999999999999999999547 second. This array is reset after _USBUpdate(…) is called.
traceB: traceB returns intensity history for channel B.
HistogramA: pointer to an array of 2048 elements which contains the normalized histogram for channel A at the sample times defined in USBStart function.
HistogramB: pointer to an array of 2048 elements which contains the normalized histogram for channel B at the sample times defined in USBStart function.

b) Flex02-01dcts.dll. provides functions for saving arrival time series. There are 6 functions
1. _USBInitialize(BYTE mode) // Always initialize the correlator before use,
mode: 'A' single correlation; 'B' quad correlation; 'C' dual correlation.
2. _USBStart(BYTE mode, char *filename)
// Argument 1: 0 auto, 1: cross.
// Argument 2: The file name for the intensity series. If NULL, the intensity is not saved._
3. USBStop() // Stop
4 USBFree() // Call this function to clean up before exit.
5. USBUpdate(float *ElapsedTime, int TraceCount, float *corr1, float *corr2, float *corr3, float *corr4, float *traceA, float *traceB) // Call this function to get correlation functions and intensity histories (A and B) while the correlator is running.

*ElapsedTime: The running time since _USBStart is called.
*TraceCnt: The number of elements in traceA and traceB
corr1: pointer to an array of 1120 elements which contains the normalized correlation function accumulated over *ElapsedTime.
corr2: pointer to an array of 624 elements which contains the normalized correlation function accumulated over *ElapsedTime.

corr3: pointer to an array of 304 elements which contains the normalized correlation function accumulated over *ElapsedTime.
corr4: pointer to an array of 304 elements which contains the normalized correlation function accumulated over *ElapsedTime.
traceA: pointer to an array of intensity elements for channel A. The number of element is *TraceCnt. Each element is the number of photons within an interval of 0.1048576 second. This array is reset after _USBUpdate(…) is called.
traceB: traceB returns intensity history for channel B.
In Dual mode: corr3 and corr4 are ignored
In Single mode: corr2, corr3, and corr4 are ignored.

6. USBUpdateRawdata(float *ElapsedTime, int TraceCount, double *rawcorr, double *samples, double *baseA, double *baseB, float *traceA, float *traceB, BYTE *overflow) // Call this function to get raw correlation functions data while the correlator is running.

*ElapsedTime: The running time since _USBStart is called.
*TraceCnt: The number of elements in traceA and traceB
rawcorr: pointer to an array of 1248 elements which contains the raw correlation function accumulated over *ElapsedTime.
samples: pointer to an array of 624 elements which contains the number of samples accumulated over *ElapsedTime.
baseA: pointer to an array of 1248 intensity elements accumulated over *ElapsedTime.
baseB: pointer to an array of 1248 intensity elements accumulated over *ElapsedTime.
traceA: pointer to an array of intensity elements for channel A. The number of element is *TraceCnt. Each element is the number of photons within an interval of 0.052428799999999999999999999999547 second. This array is reset after _USBUpdate(…) is called.
traceB: traceB returns intensity history for channel B.
overflow: pointer to an array of 2 elements. Overflow[0] = 1 indicates there are overflow in photon series from Channel A. Overflow[1] = 1 indicates there are overflow in photon series from Channel B.

Caution:

The power sequence is important. The correlator should be powered up before a voltage is applied to the inputs. This means that the user should turn off the detector before power down the PC or disconnect the correlator from the PC.

Support:

Technical support is available. Please email Support@correlator.com