Xlim institut de recherche

Université de Limoges

IDQ TDC id800
Application Note

Département Photonique

juin 2013

Contents

Introduction

The id800 is a Time to Digital Converter (TDC) manufactured by ID Quantique. The hardware is already presented by the user guide provided by IDQ. A datasheet is also downloadable from the IDQ site : `www.idquantique.com/images/stories/PDF/id800-time-to-digital-converter/id800-specs.pdf`. This application note is only focused on the software aspect (with LabView) and on the modifications we applied on the genuine VI. Here is the list of the sub-VIs involved by the main program (TDC_id800.vi):

- TDC_init.vi
- TDC_enableChannels.vi
- TDC_switchTermination.vi
- TDC_errorHandler.vi
- TDC_setTimestampBufferSize.vi
- TDC_getLastTimesamps.vi
- TDC_getDataLost.vi
- TDC_deInit.vi

The principle of data recording and processing can be represented as follows:

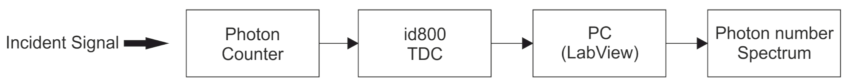

Figure 1: Operating Principle of the acquisition chain

A photon counter detecting the incident events is directly routed to the TDC id800 module which records the time tags arrival of each events. This array of time values is then sent to a pc running LabView (8.5.1 version) in order to store and process the input signal. Fig 2 gives an example of event frame :

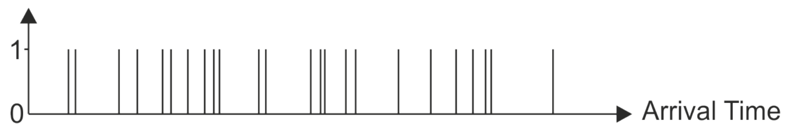

Figure 2: Signal returned by the id800 TDC

Once the input is digitally recorded, the user can apply some digital signal processing steps via LabView. The software's possibilities are :

- Signal simulation to realize a forecasting, or analysing the nature of the signal experimentally acquired.
- Display of time tags arrival.
- Binning option of the signal to get a converted analog representation.
- Fourier Transform (DFT or FFT according to the configuration).
- Display of digital to analog signal converted and the related spectrum.

The main feature of the program is to allow the user to have access to the spectrum of the signal in photon counting regime.

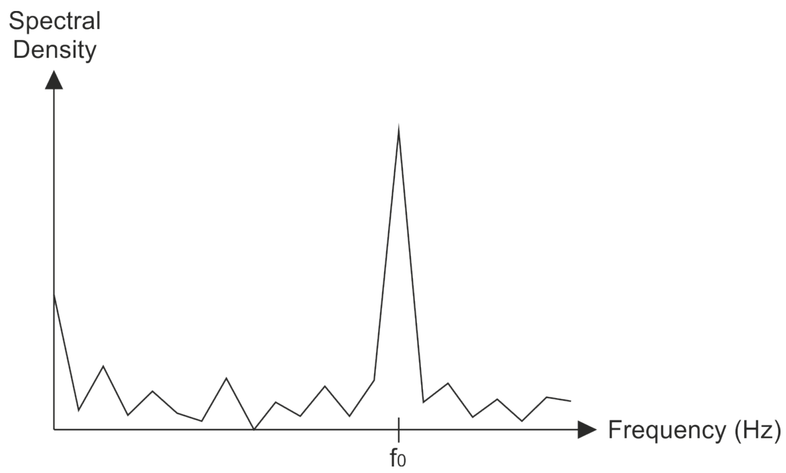

Figure 3: Example of signal's spectrum with a modulation at f_0 frequency)

Before focusing on the description of the user interface, we will recall some digital signal processing tutorials. This next part will help the user to understand how the LabView program is working and how it is designed.

Chapter 1

Digital Signal Processing

In this chapter, we will discuss the two methods used to retrieve the signal's spectrum from the output of the id800 TDC module :

- Discrete Fourier Transform.
- Binning followed by Fast Fourier Transform.

I. Discrete Fourier Transform

In the general case, for a discrete signal $x(k)$ the spectrum $X(f)$ is :

$$FT[x(k)] = X(f) = \sum_{k=0}^{N-1} e^{-j2\pi kft}$$

with N, the number of event and t, the time.
Fig 1.1 shows us the duration of an acquisition.

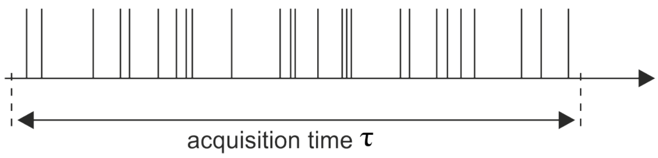

Figure 1.1: Acquisition time τ of a frame

In our case, the discrete signal is recorded over an acquisition time τ. Then, the frequency resolution f is $f = \frac{1}{\tau}$. The dates recorded by the id800 are not true values of time but binary encoding. Indeed, the real arrival time "t" for one event is $t = date * native\ bin$ (with "native bin" = 81 ps). So, the actual discrete fourier transform is

$$X(f) = \sum_{k=0}^{N-1} e^{-j2\pi\frac{k}{\tau}(date * native\ bin)}$$

With this kind of fourier transform, we can directly get the spectrum of the discrete signal recorded by the id800 time to digital converter. Fig 1.2 shows a program part corresponding to a DFT.

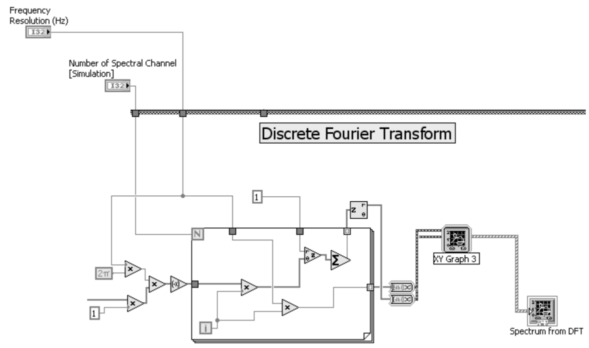

Figure 1.2: Program part : Digital Fourier Transform

II. Binning

Binning is used here to convert a digital signal into a analog signal. With this additional step, the application of the Fast Fourier Transform will be possible. Figure 1.3 shows how an analog signal can be build from a raw digital frame by the binning process.

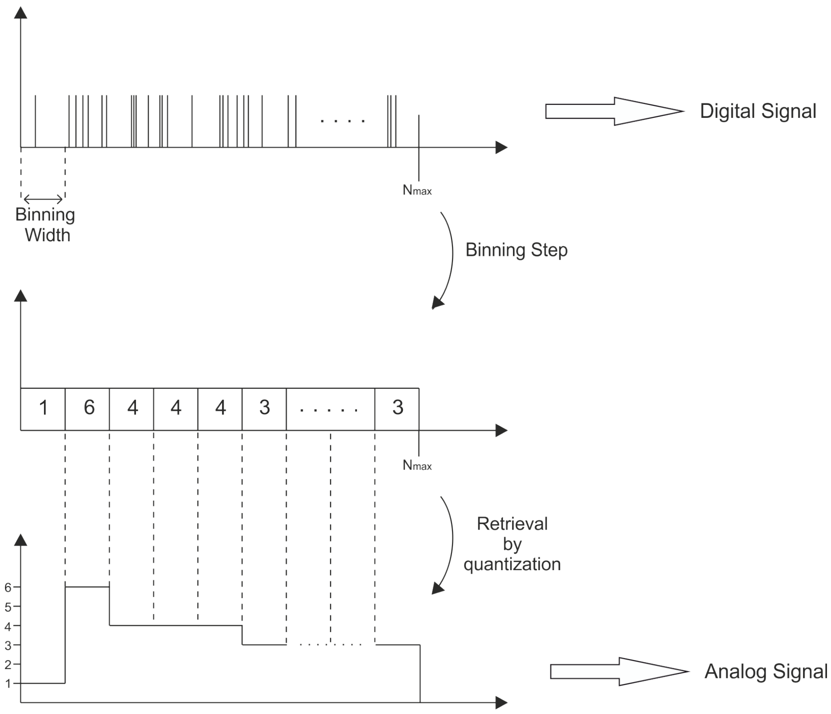

Figure 1.3: Binning Principle

For a constant counting interval (binning width), the program gives the number of incident events during this interval. Then, all these analog values are stored in an array which will be used to display the related analog signal. The aim of this process is to make possible the implementation of the fast fourier transform (FFT) on the analog signal (which was not applicable on the discrete signal). Here is a simple example with a sinusoidal modulation (Fig 1.4) :

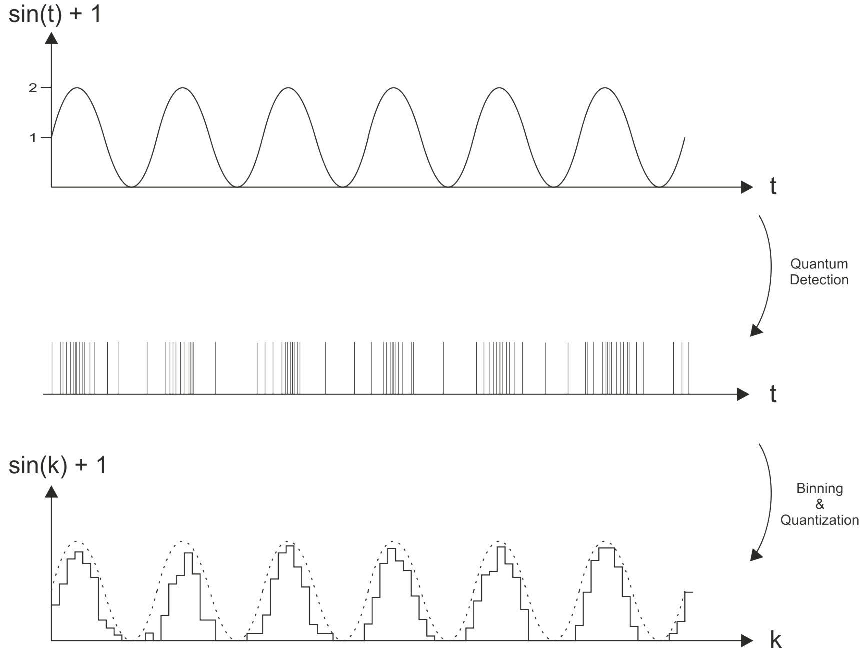

Figure 1.4: Example of binning on a sine wave

For a quantum detector, the higher the amplitude of the modulation the higher is the probability to detect photons during the sampling interval. Then, through the step of binning, an analog signal can be built from the raw quantum frame. However the user has to choose wisely the width of the binning :

- if the width is too small : the noise remains strong
- if the width is too high : it could be larger than the period of the modulation

Fig 1.5 shows a program part of the binning process

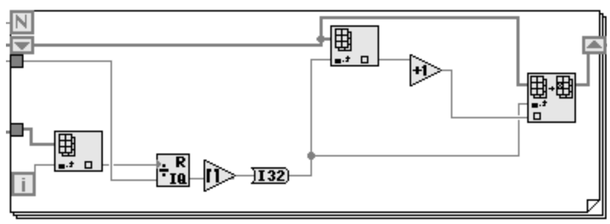

Figure 1.5: Program part : binning process

III. Fast Fourier Transform

The fast fourier transform is a fast process to compute the fourier transform of an analog wave. Indeed, for N samples, the number of operation for a direct DFT is N^2 while it is only $N * log(N)$ for the FFT. This method can be applied on analog signals. Thus, in association with the binning process, the number of samples are reduced and are treated faster.

Chapter 2

VI Diagram

In this chapter, the VI diagram will be presented and explained step by step. All the possibilities of the program will be covered.

- Simulation & DFT.
- Simulation, binning followed by FFT.
- Acquisition & DFT.
- Acquisition, binning followed by FFT.

Two types of symbols are used to describe the different screenshots.

- **Letters (Ⓐ, Ⓑ...)** : used to indicate the input and output signals. This allows the reader to follow more easily the signal processing path.
- **Numbers (①, ②, ③...)** : used to show the different parameters and process involved in the diagram.

Fig 2.1 is a generic example.

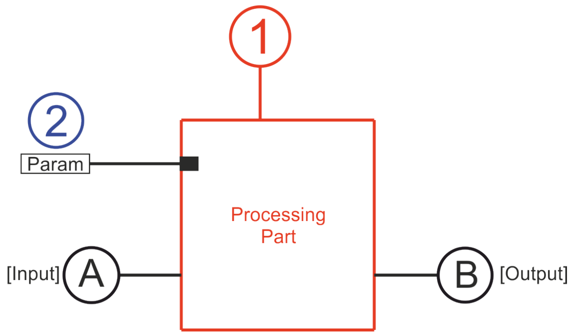

Figure 2.1: Caption example

Some parts of the program are used several times. They will be described only once (for example, the classifying process).

I. Simulation & DFT

In this section, the user will see how a raw digital frame can be virtually simulated and how its spectrum can be extracted via the Discrete Fourier Transform. This can help the user to anticipate the nature of the signal.

1) Signal Simulation

Fig. 2.2 shows the digital signal generation part which virtually simulate the behaviour of a quantum detection

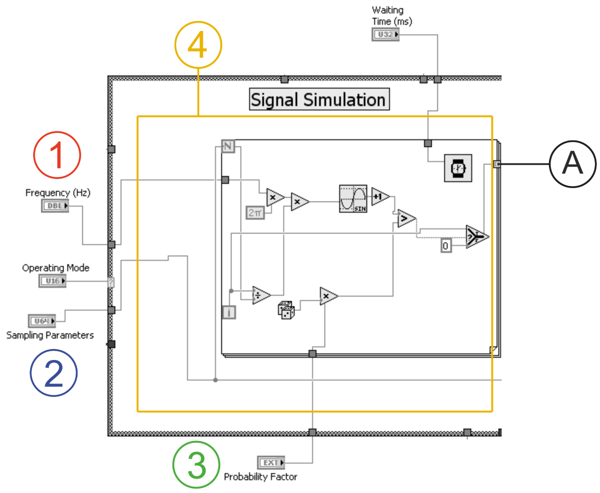

Figure 2.2: Signal Simulation

(1)**Frequency :** Set the frequency of the signal to be generated.

(2)**Sample Parameter :** Allow the user to control the number of samples to be computed. This parameter has to be chosen carefully. If it is

- too small : not enough sample to generate a proper signal.
- too high : it can end in a very slow calculation of the fourier transform .

(3)**Probability Factor :** The random discretization from the analog signal is done by this parameter. The larger it is, the fewer discrete events occur.

(4)**:** This part of the program build the simulated analog signal ($sin(2\pi ft) + 1$) and a discrete signal is extracted from it.

(A)**:** Here is the discrete signal generated by the simulation process.

2) Classifying Step

Fig 2.3 shows how the digital signal is cleaned.

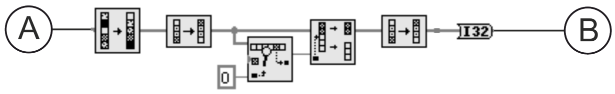

Figure 2.3: Classifying Process

(A): The generated signal is made of zeros and dates.

date	1	2	0	0	5	0	7	8	0

During the classifying step, the zeros and dates values are separated and sent into two different arrays.

(B): The outcoming signal corresponds to the dates array. Then, the signal is only made of time values encoded with an integer.

date	1	2	5	7	8

3) Discrete Fourier Transform

Fig 2.4 shows the computation of the DFT allowing the user to have access to the signal's spectrum.

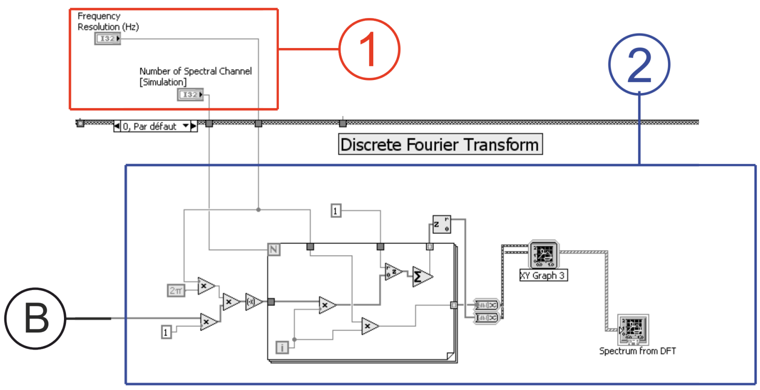

Figure 2.4: Digital Fourier Transform

(B): The digital signal is sent here to be processed.

(1): These are the parameters allowing the user to control the display of the spectrum.

- Number of spectral channels : corresponds to the frequency show of the analysis.
- Frequency resolution : the smallest observable element in this frequency domain.

(2): The DFT computing part : $\sum_{k=0}^{N} e^{-j2\pi\frac{k}{\tau}f(date*(nativebin))}$ and its spectrum display.

II. Simulation, Binning & FFT

In this section, the user will see another way to have access to spectrum of the simulated signal via FFT with the additional step of binning. The VI converts the digital frame into an analog signal. Then, the FFT is applied onto the signal.

1) Signal Simulation

Fig 2.5 shows the signal simulation part.

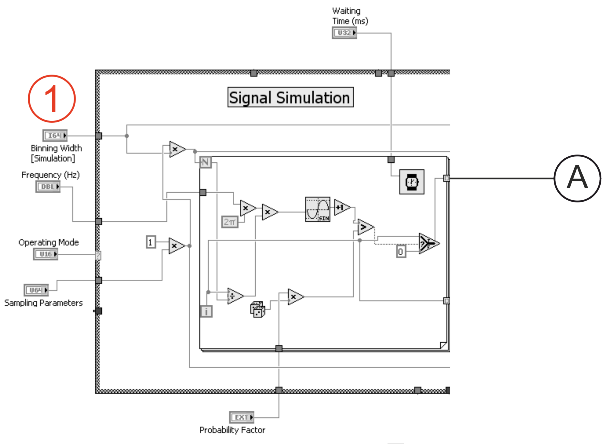

Figure 2.5: Signal Simulation

The signal simulation part in itself is the same except the binning width parameter is taken into account to set the number of samples to be generated.

(1): Allow the user to choose the width of the binning applied on the signal.

(A): Digital output signal.

NB : The classifying process is exactly the same than previously. The time values are put together in an array, rejecting the zeros.

2) Binning Step & FFT

Fig 2.6 (top of next page) shows how the FFT is computed through the binning method.

(A): The digital signal before binning.

(1): Here, the user set the frequency resolution and range for the FFT.

(2): Binning step, transforming the digital data into an analog signal.

(3): This single module performs the FFT of the analog signal.

(4): The program is displaying the analog signal after binning and its related spectrum computed via FFT.

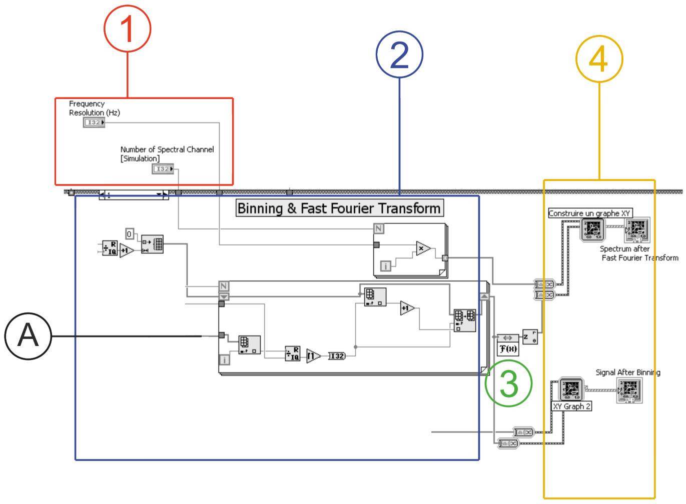

Figure 2.6: Binning followed by FFT

III. Acquisition & DFT

In this section, we will see how the VI starts/stops an acquisition. Also how the signal is recorded and how the DFT is computed in order to have the spectrum of a real signal.

1) Initialization

Fig 2.7 shows the initialization process of an acquisition.

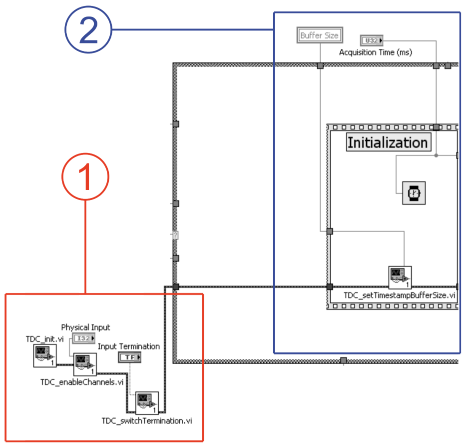

Figure 2.7: Initialization

(1): Initialize the id800 module, allow the user choose the physical input of the Time to Digital Converter and the input termination (50 Ω or 1 $k\Omega$).

(2): Allow the user to select the duration of an acquisition and the maximum sample number recordable during this acquisition.

2) Acquisition of the frame

Fig 2.8 shows how the raw digital frame is recorded by the VI.

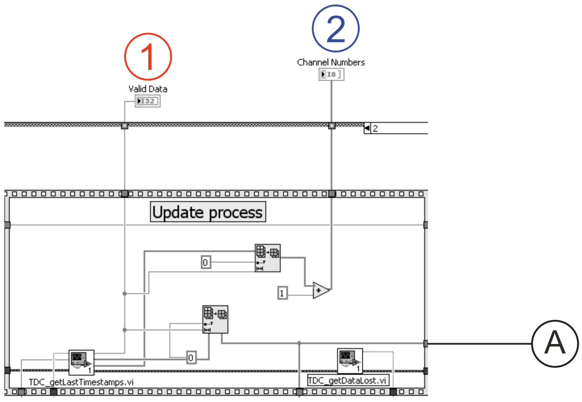

Figure 2.8: Acquisition

(1) **Valid Data :** Gives the number of events recorded by the id800.

(2) **Channel Numbers :** Gives the corresponding physical channel used to record each incoming event.

(A) **:** This is the discrete output signal recorded and sent via the sub-VI "$TDC\_getLastTimeStamps$" which is a raw frame of dates.

3) Discrete Fourier Transform

Fig 2.9 shows the DFT computation of the frame.

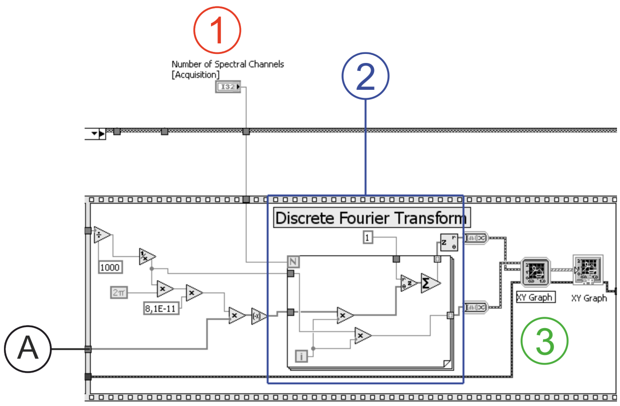

Figure 2.9: Digital Fourier Transform

(1) **Number of Spectral Channels :** Set the frequency range to display.

(2) **:** Discrete fourier transform of the raw discrete frame (A).

(3) **:** Display the spectrum of the digital signal.

4) End of the Acquisition

Fig 2.10 shows how an acquisition process ends.

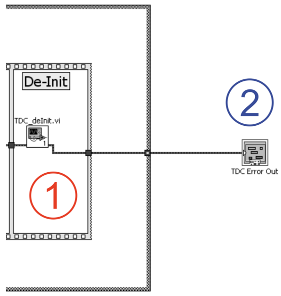

Figure 2.10: De-Initialization

(1): This part of the program ends the acquisition process.
(2): Display possible errors over the acquisition.

IV. Acquisition, Binning & FFT

In this section, we will see how the the binning process is added to get the spectrum via FFT. Since initialization, acquisition and de-initialization processes are exactly the same, the focus is only on binning and FFT.

1) Binning followed by FFT

Fig 2.11 shows the binning process followed by the FFT computation.

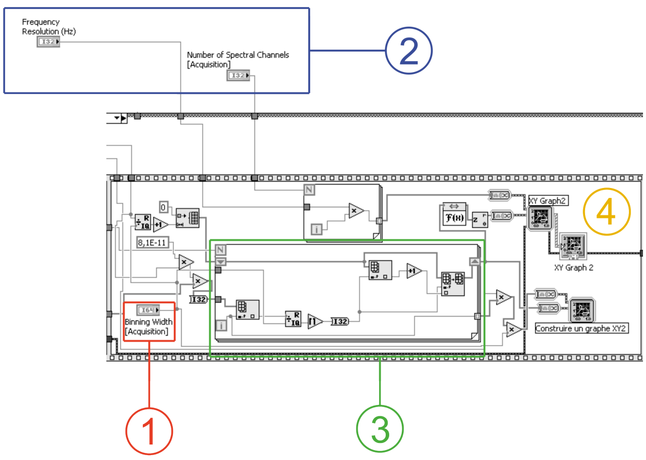

Figure 2.11: Binning & FFT

(1) **Binning Width :** Set the width of the binning in acquisition mode.
(2): Allow the user to control the frequency resolution and the frequency range.
(3): Binning step before the FFT of the analog signal.
(4): The spectrum of the signal computed via the FFT.

Chapter 3

User Interface

In this chapter, the user will learn to navigate into the user interface and to use it.

Numbers (①, ②, ③...) will be used to show the different parameters and displays available to the user.

I. Choosing Operating Mode and Display

Fig 3.1 shows the drop down menu and the display tabs.

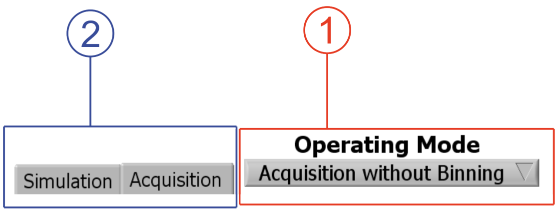

Figure 3.1: Drop-down menu & display tabs

①**Operating Mode :** The drop down menu allow the user to choose the operating mode:

- acquisition (with/without binning).
- simulation (with/without binning).

②**:** The user choose the tab corresponding to the operating mode. In each tab are the corresponding parameters and displays.

II. Simulation

Simulation display is accessible simply by clicking on the simulation tab. Here the user have access to all the simulation parameters and the related graph.

1) Simulation Parameters

Fig 3.2 shows all the simulation parameters available to the user.

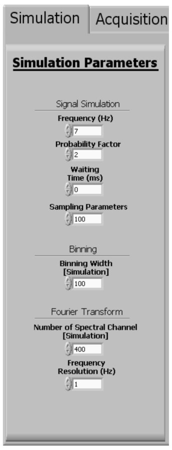

Figure 3.2: Simulation Parameters

In this section of the interface, the user have access to all the simulation parameters.

- **Frequency :** Set the frequency of the sinusoidal signal.
- **Probability Factor :** Allow the user to control how much events will be recorded from the analog signal to build the raw digital frame.
- **Waiting Time :** The user can set a waiting time (in milliseconds) during the signal simulation to slow down the process.
- **Sampling Parameters :** This parameter controls how much samples will be involved to generate the signal.
- **Binning Width :** When binning is activated, this parameters controls its width.
- **Number of Spectral Channel :** Set the frequency range.
- **Frequency Resolution :** Determine the resolution for the spectrum display.

2) Display

Fig 3.3 (top of next page) shows the tab corresponding to the simulation mode.

①**Arrival Time :** Here the user can see the dates of every single events. ②**:** Display the spectrum of the signal :

- after a DFT (without binning).
- after a FFT (with binning).

and display the analog signal after the binning step.

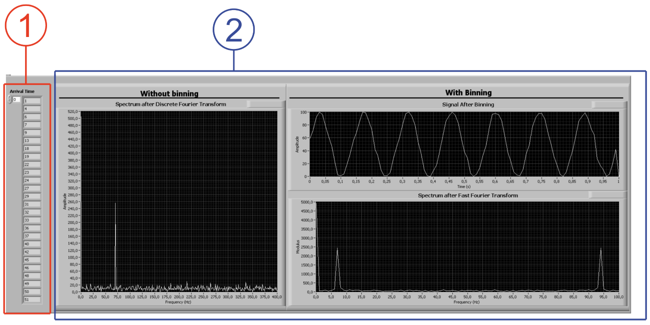

Figure 3.3: Simulation Display

III. Acquisition

Acquisition display is simply get by clicking on the acquisition tab. Here the user have access to all the acquisition parameters and the related graph.

1) Acquisition Parameters

Fig 3.4 shows the different acquisition parameters

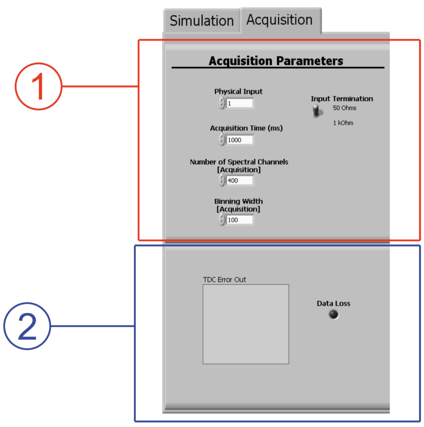

Figure 3.4: Acquisition parameters

(1): This is where the user can set all the acquisition parameters presented earlier :

- Physical Input of the id800.
- Acquisition Time (in milliseconds).
- Number of Spectral Channels (frequency range).
- Binning Width (when involved).

(2): Warn the user if an error has occured during the acquisition.

2) Display

Fig 3.5 shows the acquisition mode tab.

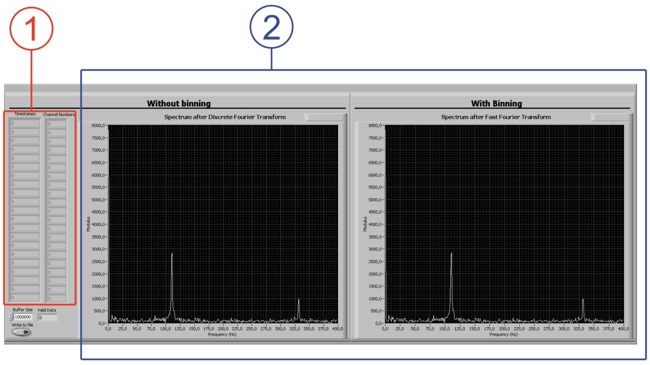

Figure 3.5: Acquisition Display

(1): Dislay the arrival time and their corresponding physical channel.
(2): Display the spectrum of the signal :

- after a DFT (without binning)
- after a FFT (with binning)