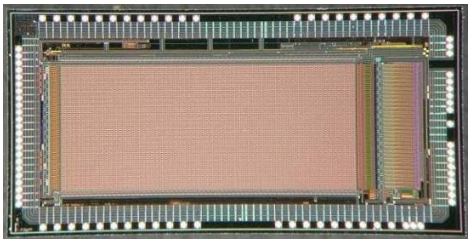
WaveCatcher Family User's Manual



Date: 1/6/2017	WaveCatcher Family User's Manual
By D.Breton & J.Maalmi, LAL Orsay	V/Ref. : 1.2

PURPOSE OF THIS MANUAL

This User's Manual contains the full description of the WaveCatcher Family hardware and gives the paths to Control & Readout software and libraries. This concerns the 2-channel and 8-channel WaveCatcher modules, the 16-Channel WaveCatcher board and module, and all the options of the 64-Channel WaveCatcher Crate (16, 32, 48 or 64 channels).

EVOLUTIONS OF THE DOCUMENT

DATE	REVISION	SOFTWARE VERSION	CHANGES
12/22/2016	V1.1	2.8.11	FIRST VERSION OF THE DOCUMENT
01/06/2017	V1.2	2.8.12	MINOR CORRECTIONS

REFERENCE DOCUMENTS

INDEX

Table of illustrations

1 Introduction

The WaveCatcher systems are a family of **powerful and low cost digitizers**. Their number of channels currently ranges between 2 and 64 (+8) channels.

They all make use of the **SAMLONG** analog memory chips which permit sampling the input signal between 400 MS/s and 3.2 GS/s over 12 bits and with a signal bandwidth of 500 MHz.

There are 4 different types of systems:

- **2-channel**, USB-powered handy module
- **8-channel** (autonomous desktop), composed of a motherboard equipped with two 4-channel mezzanines
- **16-channel** (6U board or autonomous desktop module)
- **64-channel** (mini crate). This crate can actually house between 1 and 4 16-channel boards, thus providing **16, 32, 48 or 64 channels**.

From the second version of the 16-channel boards on, 2 extra channels have been added in the back of the board. They can be digitized together with the other channels. When the board is used in standalone mode, these channels correspond to the external trigger and the external sync. Otherwise, they are equivalent to other channels.

The boards can also be used as **TDCs** for high precision time measurement between two signals. Sampling time precision after calibration is indeed **less thans 5 ps rms** at 3.2GS/s.

The systems are currently interfaced with a 480 MBits/s USB link. A secured Gbit UDP interface is also available on the 8-channel module. It will soon be put into function on the 64-channel systems. An optical link (all systems) will also be put into function but later.

Software access to the WaveCatcher systems can be performed in two ways:

1. Via a **high-level software library**, available on Windows or Linux.
2. Via a **dedicated powerful software** running on Windows.

There is no low-level library.

2 System description

2.1 Global description of the WaveCatcher systems

The WaveCatcher boards and modules are 12-bit 3.2 GS/s Switched Capacitor Digitizers issued from the collaboration between CEA/IRFU & CNRS/IN2P3/LAL and based on the SAMLONG chip[1], developed on the basis of a CEA/IRFU and IN2P3/LAL common patent [2].

The input dynamic range is 2.5 Vpp (DC coupled) and the input standard is single ended MCX coaxial connectors. The DC offset is adjustable in the ±1.25V range via a 16-bit DAC on each channel (see § **2.3.1**). The signal analog bandwidth is 500 MHz @3dB.

Considering the sampling frequency and the number of bits, it is well suited for very fast signals as the ones generated by fast scintillators or crystals coupled to PMTs, MCP-PMTs, Silicon Photomultipliers, APD, Diamond detectors and others.

The analog input signals are continuously sampled inside the SAMLONG chips in a circular analog memory buffer (1024 cells) at the default sampling frequency of 3.2 GS/s (312.5 ps of sampling period); 6 other frequencies down to 0.4 GS/s are also selectable via software. As a trigger signal arrives, all analog memory buffers are frozen and subsequently digitized with a resolution of 12 bits into a digital memory buffer with independent read and write access. Up to 7 full events per channel (1 event = 1024 words of 12 bits) can be stored consecutively.

Each input channel is equipped with a discriminator using a 16-bit programmable threshold, which generates trigger primitives. Primitives from all channels are processed by the board to generate a common trigger causing all the channels to acquire an event simultaneously. The common board trigger can also be provided externally via a software command, or via the front panel TRIGIN input connector, or by any combination of the channel discriminators and/or the TRIGIN.

During analog to digital conversion process, the WaveCatcher cannot handle other triggers, and thus generates a dead time (maximum 125 µs, decreasing proportionally with the recording depth thanks to the configurable record length).

Each pair of channels is equipped with a 40-bit TDC (counter) tagging the trigger with the clock delivered to the SAMLONG chips (200 MHz down to 25 MHz depending on the selected sampling frequency).

Each input channel is equipped with a hit rate monitor based on its own discriminator and on two counters giving the number of hits which cross the programmed discriminator threshold (also during the dead time period) and the time elapsed with a 1 MHz clock (see § **Hit rate monitor**). This permits among others measuring the hit rate with respect to the signal amplitude.

Each input channel is also equipped with a digital measurement block located in the front-end FPGA which permits extracting in real time all the main features from the signal: baseline, peak, charge, time of the edges in CFD or fixed threshold modes (see § **Using the Measurement Block**).

Each channel houses a fixed amplitude pulser, which permits an easy complete functionality test and the use of the module in reflectometer mode (see § **Test Pattern Pulser**).

The module supports multi-board synchronization allowing all SAMLONG chips to be synchronized with a common clock source and ensuring triggering them in phase. All data will then be aligned and coherent between multiple WaveCatcher boards.

All WaveCatchers house USB 2.0 which allows data transfers up to 30 MB/s. The 8-channel module and 64-channel crate also provide a secured Gbit UDP interface. In addition, all the boards house an Optical Link but the latter has not yet been put in function.

2.2 *Generic block diagram*

The block diagram of the WaveCatcher systems is shown on **Figure 2.1**. They are all based on front-end blocks of 2 channels, gathered by two and sharing a front-end FPGA (i-e 4 channels per FE FPGA). These blocks can reside either on 4-channel mezzanine boards (8-channel modules), or be part of the main board (2 or 16-channel boards).

Except for the 2-channel version where a single FPGA is used, another FPGA is used to be the interface between the external world and the FE FPGA. It also takes care of producing the system trigger. For the 8-channel module, it sits on the motherboard.

In the case of the 64-channel system, a dedicated control board is added. It drives up to four 16-channel boards via a custom crate backplane, and produces the system trigger.

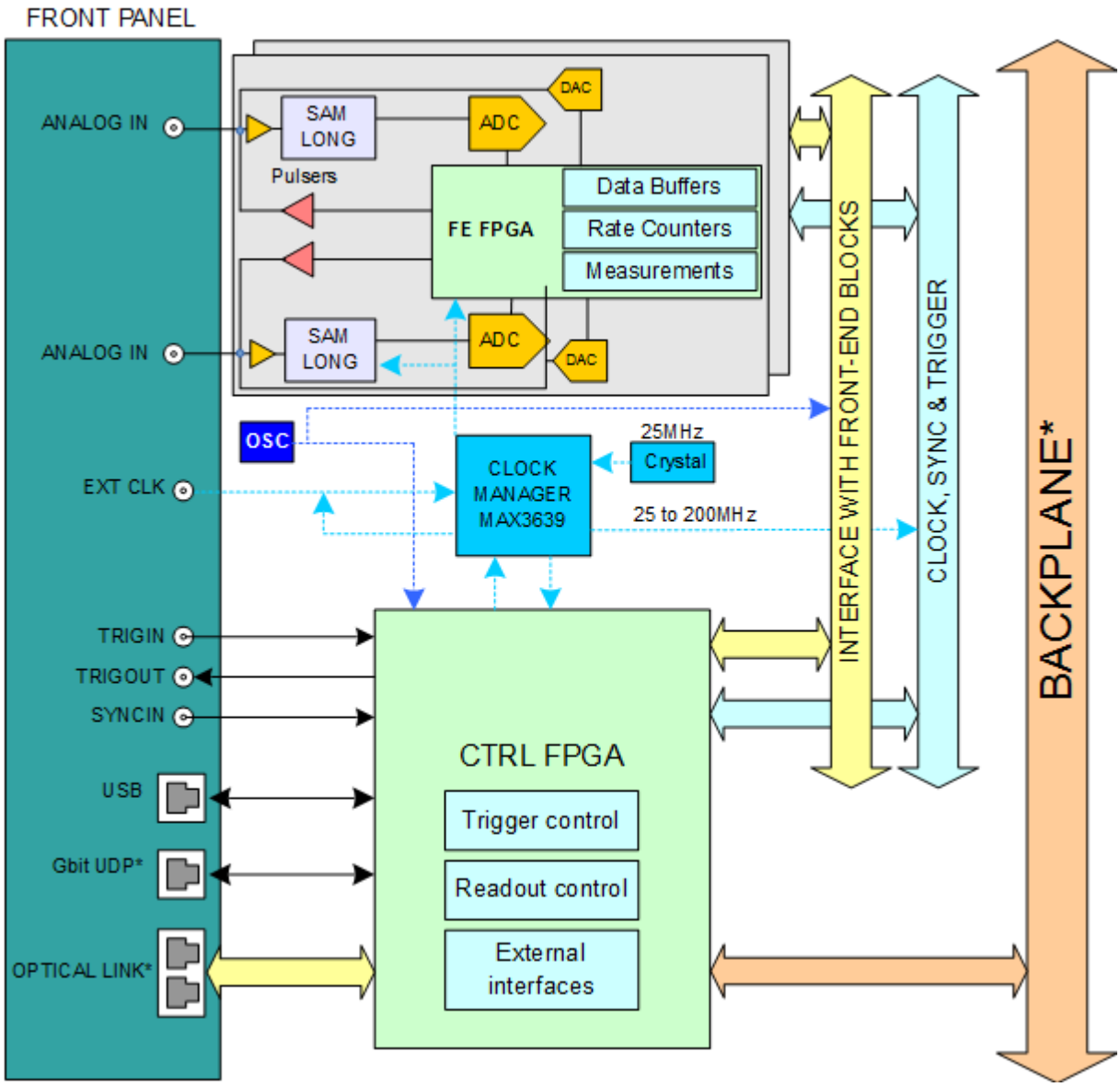


Figure 2.1: Block Diagram of the WaveCatcher systems

2.3 Front-End block

2.3.1 Analog Input Stage

Input dynamics is 2.5 Vpp on single ended MCX coaxial connectors (see ??). A 16-bit DAC allows to add up a ±1.25 V DC offset in order to preserve the full dynamic range even in the extreme case of unipolar, positive or negative input signal.

The input bandwidth ranges from DC to 500 MHz (@3dB).

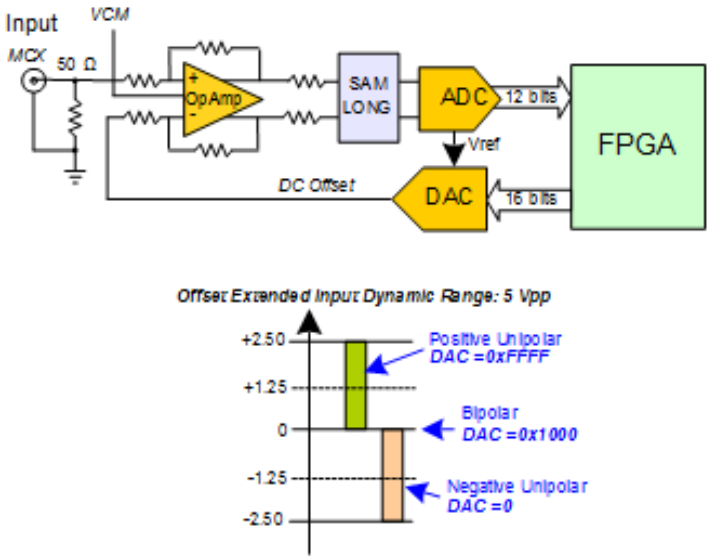


Figure 2.2: Analog Input Stage

2.3.2 Making use of the analog memory

The analog input signals from each pair of channels are continuously sampled into one SAMLONG chip, which consists of a matrix of Delay Line Loops (DLLs) generating a 3.2 GS/s sampling frequency from an input clock of 200 MHz. Sampling frequencies of 2.13, 1.6, 1.28, 0.8, 0.53 and 0.4 GS/s can also be programmed (with respective corresponding input clock frequencies of 133, 100, 80, 50, 33 and 25 MHz).

Signals produced by the DLLs simultaneously open write switches in both sampling channels, where the differential input signals are sampled (1024 sampling memory cells per channel).

After being started by the "***start_acquisition***" signal, the DLLs run continuously in a circular fashion (after reaching the end of the matrix, samples are over written) until decoupled from the write switches when the Run signal goes down. This actually takes place after the arrival of a trigger signal synchronously delayed by the so-called post-trigger delay which finally provokes the freezing of the signal currently stored in the sampling memory cells.

Subsequently, the cells are multiplexed towards the 12-bit ADCs whose output pass through the measurement blocks of the FPGA before being stored into the Digital Memory Buffer and made ready for readout in the shape of events data.

A 16-bit DAC allows to add up to a ±1.25 V DC offset in order to optimize the dynamic range.

Detailed documentation of the SAMLONG chip is available at:

http://electronique.lal.in2p3.fr/echanges/USBWaveCatcher/Documentation/Boards&Chips/doc_SAMLONG_rev1.pdf

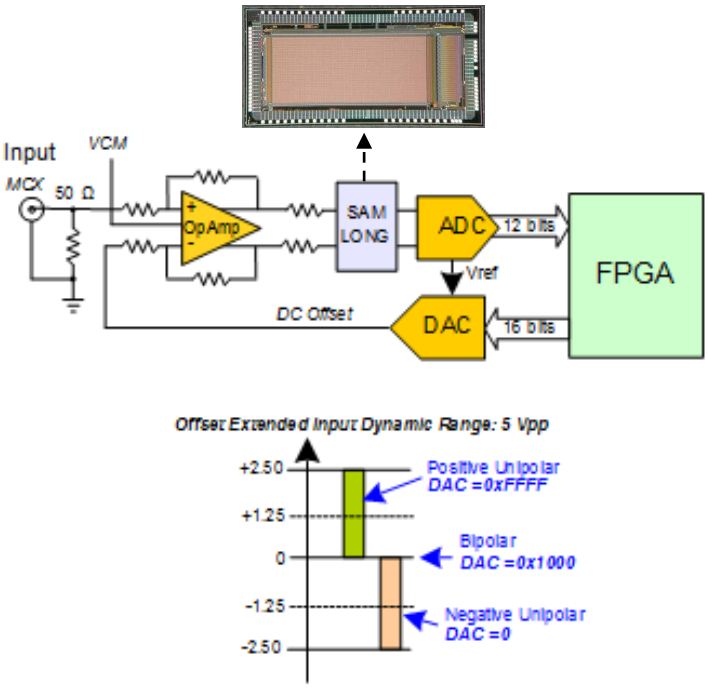


Figure 2.3: focusing around SAMLONG

2.4 Clock distribution

Except the 2-channel version, the WaveCatcher systems make use of a PLL for clock synthesis with a selectable internal or external reference clock source. By default, the PLL is fed by a 25 MHz quartz. Based on it, the PLL will produce a front-end clock whose frequency will be of 200, 133, 100, 80, 50, 33 or 25 MHz. The SAMLONG chips will then use it to produce their own sampling frequency.

Multi-board synchronization can be performed by driving a clock on EXT CLK input, allowing all SAMLONGs to run synchronously with this external reference. All analog inputs will be sampled at the same time without time drift, allowing high resolution timing and time analysis across multiple modules.

A few local oscillators are also present on the boards to handle the external interfaces (USB, Gbit UDP, optical link).

For what concerns the 2-channel version, a 200 MHz oscillator is dedicated to the front-end clock. When running with the reduced sampling frequencies, the 200 MHz clock is divided inside the FPGA before being sent to the SAMLONG chips via the clock multiplexer.

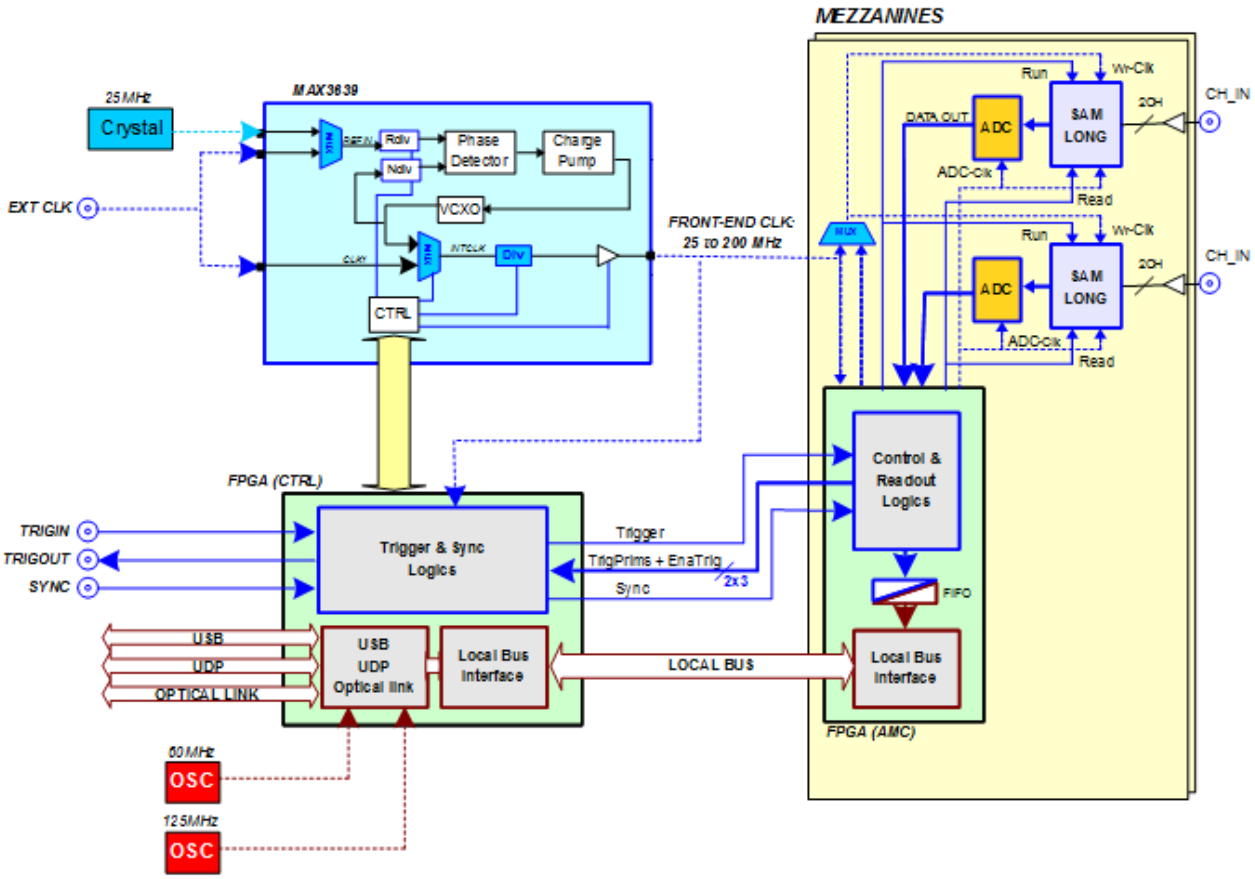


Figure 2.4: clock distribution scheme

2.5 Trigger scheme

All the channels in a system share the same trigger (common system trigger), which implies they acquire an event simultaneously and in the same way (a determined number of samples and position with respect to the trigger), like an oscilloscope.

The common board trigger is generated based on different trigger sources:

- **Software trigger**: produced via a software command.
- **External trigger**: received via the front panel TRIGIN signal.
- **Internal trigger**: random hardware trigger generated by the independent USB oscillator, with a frequency of 12 MHz.
- **Normal trigger**: a trigger request generated by a combination of one or more of the individual discriminators, with programmable threshold, placed on each analog channel. Requests from all channels are processed to generate the common board trigger.
- **Coincidence** between any predefined distribution of the latter.
- **Programmable majority** of the latter (currently available only for the 8-channel module.).

When a common board trigger is issued, the analog buffers related to that trigger are frozen, then digitized by the 12-bit ADCs, then stored into the digital memory buffer and are available for readout (refer to § **Making use of the analog memory**).

The analog to digital conversion process creates a dead time during which the module cannot handle other triggers. This dead time depends on the number of samples to be digitized, but also on the status of the event buffers.

In the standard conversion mode, the ADCs run at 10 MHz and the dead time corresponds to (13 µs + (Nsamples / 16) * 1.75 µs). The WaveCatchers thus feature a maximum dead time of 125 µs for a full waveform recording of 1024 samples.

The latter can be reduced by a factor 2 in the so-called short-latency mode where the ADCs run at 20 MHz. In this case, the drawback is a small increase of the noise level (~10%).

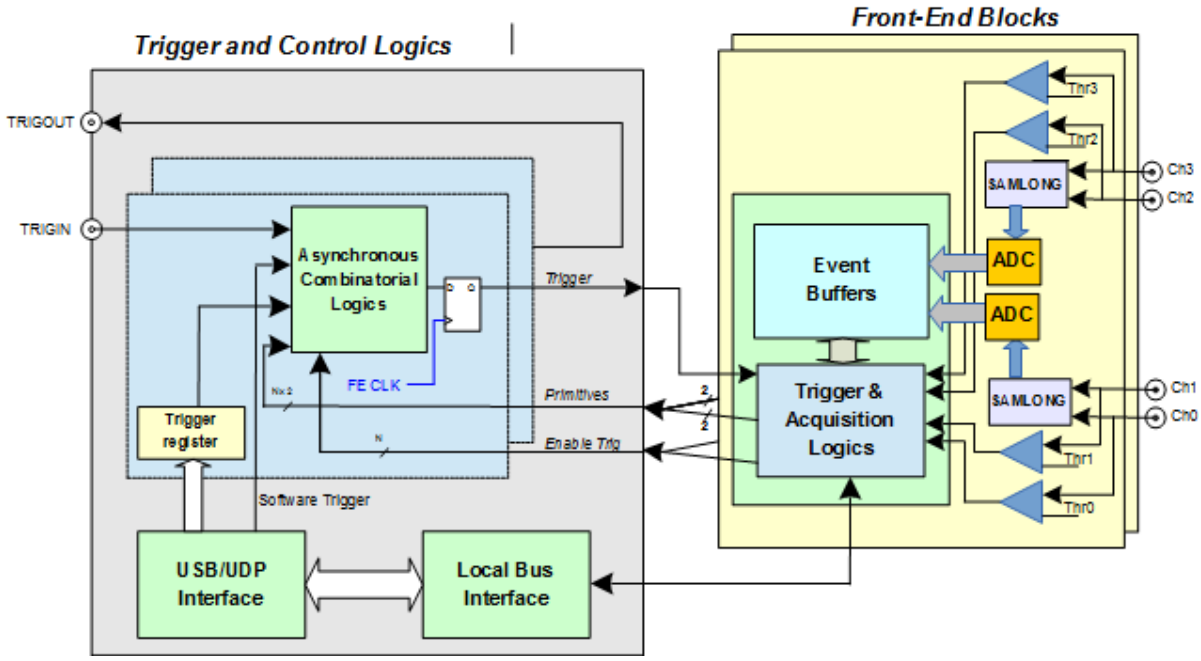


Figure 2.5: trigger global scheme

2.5.1 *Software Trigger*

Software triggers are internally produced via a software command (write access to the Software Trigger register) through USB or UDP.

2.5.2 *External Trigger*

External trigger is received via the front panel TRIGIN connector, and can be TTL or NIM. The external trigger will be resynchronized with the internal FE Clock. If the external trigger is not natively synchronized with the latter, a 1-clock period jitter will occur.

2.5.3 *Self-Trigger*

The WaveCatchers are equipped with discriminators using a 16-bit programmable threshold on each channel, which permits generating a self-trigger when the digitized input pulse exceeds the threshold value. The self-trigger of each couple of adjacent channels are then processed to provide out a single trigger request (“primitive”). The primitives are propagated to the central trigger logic to produce the board common trigger, which is finally distributed back to all channels causing the event acquisition (see **Figure 2.5**). **Figure 2.6** schematizes the production of the trigger request and Figure 2.5 the global trigger logics.

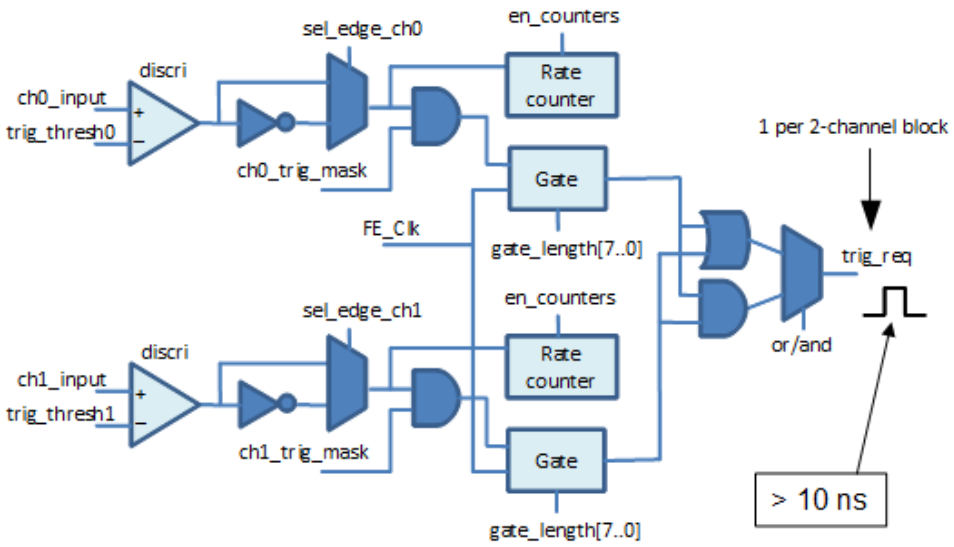


Figure 2.6: trigger primitive generation

The FE FPGA can be programmed in order for the primitive of a couple of adjacent channels to be the

- ONLY CH(n),
- ONLY CH(n+1),
- OR,
- AND

of the relevant self-trigger signals (see Figure 2.6). Note that said signals are generated by a gate whose width is programmable with a common value coded on 8 bits and based on the period of the front-end clock.

The Asynchronous Combinatorial Logics block of the CTRL FPGA can be programmed in order for the common trigger to be the OR, the AND or the Majority (only for the 8-channel module) of the enabled trigger requests (see **Figure 2.5**). A coincidence with the external trigger can also be required.

Default Conditions: by default, the system is programmed so that each trigger request is the OR of two pulses whose width depends on the board operating frequency: for instance 15 ns @3.2 GS/s; 20 ns @1.6 GS/s; 30 ns @0.8 GS/s; 50 ns @0.4 GS/s. The common trigger is generated by default as the OR of the enabled trigger requests.

2.5.4 Majority Trigger

The majority trigger option is currently available only for the 8-channel module. It permits triggering only if, within the user-defined set of channels participating to the trigger, a user-defined number of channels send a primitive synchronously. All the channels enabled for readout will then be triggered.

2.5.5 Trigger Edge

The transition edge (rising or falling) can be selected individually for each channel when the discriminator is used for triggering. This is also the case for the external trigger. For all the other sources, the rising edge will be used.

2.5.6 PRETRIG and POSTTRIG

Let's call the front-end clock Fp. During the acquisition, the analog signal is continuously sampled in the analog memory which is comparable to a circular buffer with a depth of 1024 points (time depth = 1024/Fp). The stopping of sampling is initiated by the arrival of a trigger signal Ta (asynchronous trigger) which is common to all channels of the board. This signal is only authorized to be produced following a programmable delay named PRETRIG after the start of the acquisition sequence. The minimum value for PRETRIG is of 5µs and its recommended value is of 10µs. However, it could be shortened in certain cases (see [1]).

The effective stopping of the sampling will occur following a pre-defined number (named POSTTRIG) of clock periods (of 5 to 40ns) after the trigger (cf **Figure 2.7**).

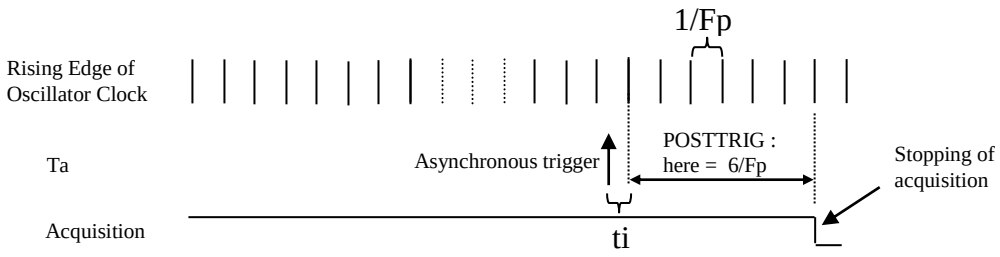


Figure 2.7: chronogram of the stopping of the acquisition.

The POSTTRIG, programmable by the user, permits defining and displacing the position of the trigger signal in the acquisition window. It is adjustable in the 0 to 255/Fp range by steps of 1/Fp (= 5ns to 40ns). This is illustrated in **Figure 2.8**.

In the example illustrated by **Figure 2.7**, the POSTTRIG is fixed at 6/Fp. The acquisition will be stopped 6/Fp after the arrival of the trigger. The analog memory will as usual contain the 1024 last recorded samples.

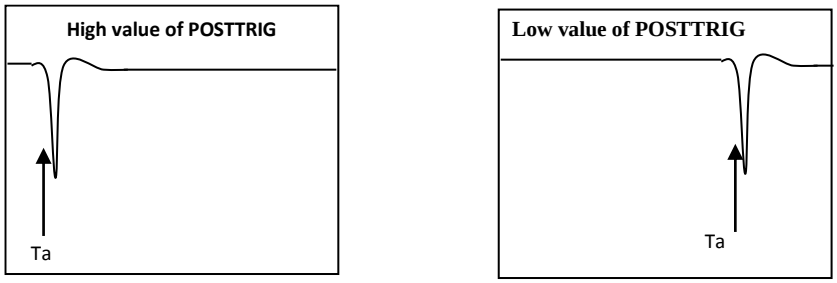


Figure 2.8: trigger position in the acquisition window for two POSTTRIG cases.

Thus, a real POSTTRIG value close to 32 ensures the centering of the trigger in the middle of the acquisition window. For the values of POSTTRIG > 64, the trigger position no longer appears in the acquisition window. It has to be noticed that the SAMLONG chip automatically adds a posttrig of 2 to 3 clock periods, to which interconnection delays add up. This has to be taken into account in the total posttrig value which usually locates the signal in the middle of the window with a user-defined POSTTRIG value of ~ 25.

2.5.7 *Trigger Distribution*

In the default configuration, the OR of all the enabled trigger sources (global primitive) is synchronized with the front-end clock, then becomes the common trigger of the board and is directly transmitted to all channels, which consequently provokes the capture of an event.

A Trigger Out signal is also generated on the relevant front panel TRIGOUT connector (TTL level), which permits extending the trigger signal to other boards.

This output can also be programmed in order to send the global primitive instead of the trigger signal. This allows to produce thanks to external logics a more complex trigger based on different sources, which will then be sent back via the TRIGIN connector. Note that in this case, the global ***enable_trigger*** from the front-end blocks is ANDED with the raw global primitive in order to ensure that all front-end blocks are ready to receive a trigger.

2.6 *Test Pattern Pulser*

Each input channel is equipped with an individual pulser. Whereas the pulse amplitude is fixed (~0.7 V with no cable plugged, half this value otherwise), the pattern can be programmed over 16 consecutive bits of the SAMLONG main clock and will be sent every 3.5 µs (see example on **Figure 2.9.a**). This permits an easy testing of the board functionality, as well as it gives the possibility to use the board as a reflectometer. As this pulse pattern is produced from an autonomous clock source, trigger can be set on the discriminators.

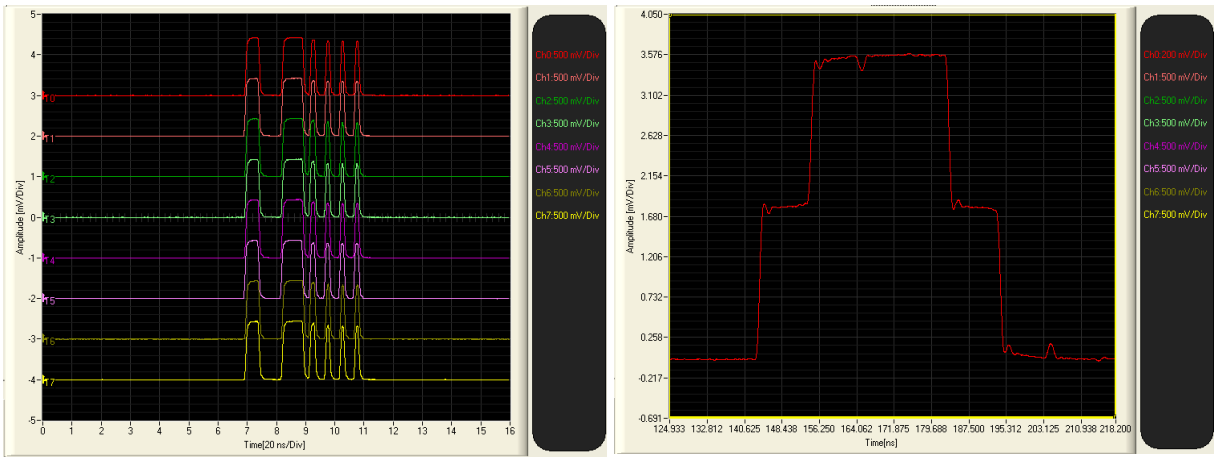


Figure 2.9.a & b: examples of use of pulse pattern generator

Each channel can make use of his pulser as a reflectometer. An example of this application is shown on **Figure 2.9.b**, where a 40-ns wide square pulse produced internally is sent to a 1-meter open cable connected to the board.

2.7 *Hit rate monitor*

Each input channel is equipped with an individual hit rate monitor. As shown on **Figure 2.10**, the latter is based on two counters, one counting the number of hits crossing the programmed discriminator threshold (TRIG_COUNT), the other counting the time elapsed with a 1-MHz clock (TIME_COUNT). These counters are reset and restarted after each read access. Their content is stored into the event data (see the **Event Structure** paragraph). As soon as any of them saturates, both are

frozen, and thus their values are always valid. The rate counters work up to ~400 MHz and, if this information is memorized long enough in the software along events, rate measurement can work as low as ~0.1 Hz.

There is an option where the hit counter is disabled during a user-defined gate in order to reject potential after-pulses which may corrupt the rate monitoring.

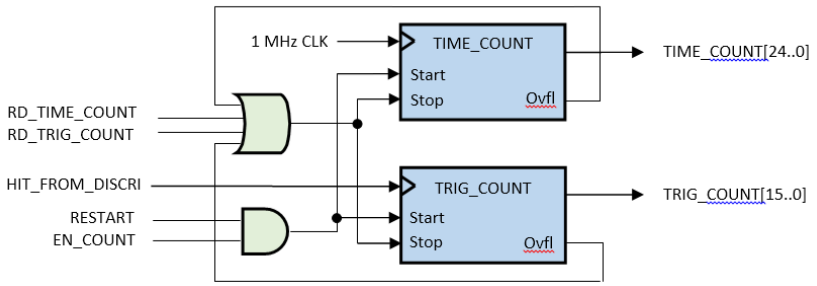


Figure 2.10: block diagram of the hit rate monitor

2.8 Measurement block

In the FE FPGA, each channel includes a measurement block able to extract information from the signal on the fly (see **Figure 2.11**). This block works with parameters fixed by user. It computes:

- the **Baseline** which corresponds to the mean of the first 16 samples
- the **Peak** amplitude, relative to the baseline
- the **Peak time** (sample corresponding to the peak location)
- the **Charge** contained in the signal (see below for the corresponding parameters)
- the time of the **Rising Edge** (can be the leading or the falling edge depending on pulse polarity)
- the time of the **Falling Edge**.
- the combination of the two edges gives the **Time Over Threshold** (TOT) of the signal.

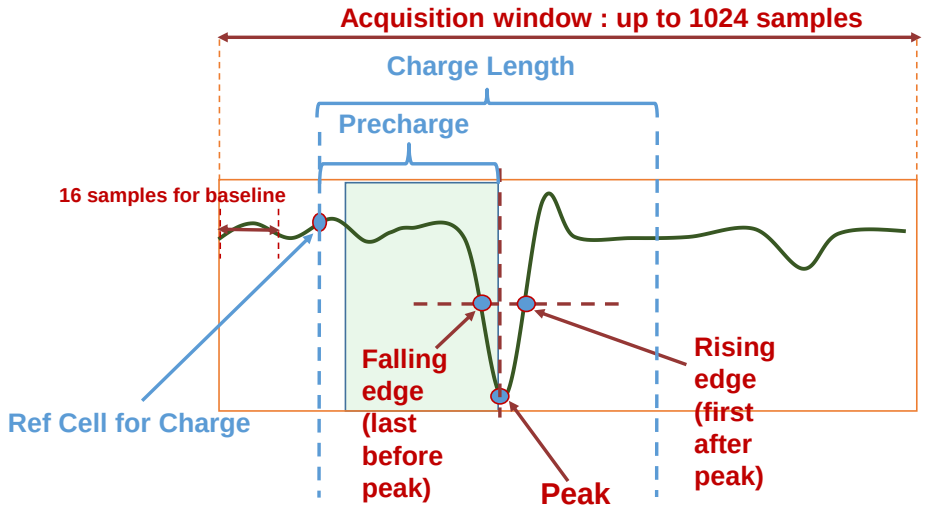


Figure 2.11: measurement parameters and results

For the peak amplitude calculation, the user has to define the polarity. The baseline can be that calculated right before, or a fixed value defined by the user in two's complement like the ADC data.

For the charge, there are two possibilities: either the user defines a starting sample (***Ref Cell For Charge***) and a length (***Charge Length***), or the firmware automatically starts from the peak, gets back over a user-defined number of cells (***Precharge***), and then applies ***Charge Length.***

For both rising and falling edge times, the threshold can be chosen between two possibilities: either a constant fraction of the peak amplitude (**CFD**), or a **fixed threshold**. Both can be parameterized by user. The constant fraction is coded with steps of 1/16 over 4 bits, thus 0xF corresponds to 15/16 of the peak. The fixed threshold is coded over 12 bits in two's complement.

2.9 User EEPROMs

Each FPGA in the systems owns its user EEPROM in order to store data useful for user, for instance calibration data specific to the board. This EEPROM is a Flash memory AT24C01B with a depth of 128kbytes. It is accessible through I2C from the FPGA. Its main specificity is to limit its writing accesses to pages up to 256 bytes, and to need 10ms for its deep internal writing after each page access. Moreover, it contains a hardware protection which has to be open through the FPGA.

Most of its content is factory loaded. The only access for user is for saving an updated individual pedestal distribution.

2.10 External signals specifications

There is a set of external I/O signals in each system. Depending on the system, they will have different specifications as shown in **Table 2.1**.

	2-channel	**8-channel**	**16-channel**	**64-channel (controller board)**
Analog input	- BNC - 50 Ω DC active	- MCX - 50 Ω DC active	- MCX - 50 Ω DC active	- Not present
TRIGIN	- BNC - 50 Ω AC passive - **AC coupling** - TTL & NIM compatible	- SMA - 50 Ω passive - V1: TTL or NIM selected via a strap on the back panel - V2: discriminator with DAC threshold	- MCX - 50 Ω DC active	- SMA - 50 Ω passive - TTL or NIM selected via a strap on the board
SYNCIN	- Not present	- SMA - 50 Ω passive - V1: TTL or NIM selected via a strap on the back panel - V2: discriminator with DAC threshold	- MCX - 50 Ω DC active	- SMA - 50 Ω passive - TTL or NIM selected via a strap on the board

EXT CLK	- Not present	- SMA - Input or Output: selection via a register - Input: **high impedance** - Output: 50 Ω 3.3V LVCMOS	- SMA - Input: **high impedance**	- SMA - Input or Output: selection via a strap on the board - Input: **high impedance** - Output: 50 Ω 3.3V LVCMOS
TRIGOUT	- BNC - 50 Ω 3.3V LVCMOS	- SMA - 50 Ω 3.3V LVCMOS or NIM selected via a strap on the back panel	- MCX - 50 Ω 3.3V LVCMOS	- SMA - 50 Ω 3.3V LVCMOS or NIM selected via a strap on the board

Table 2.1: external I/O signals specifications

3 Running the WaveCatchers

3.1 Data acquisition

Due to historical reasons, the acquisition is started when the trigger_enable bit is set to 1 in the CTRL FPGA and the start_acquisition command has been sent to all the front-end blocks.

It is stopped when the trigger_enable bit is reset to 0 in the CTRL FPGA. A dedicated reset command of the whole front-end is then necessary to put all the sequencers back to their idle state.

3.1.1 Event Structure

Events are always sent by front-end blocks and contain data relevant to pairs of channels. An event is structured as follows (see **Table 3.1**):

- **Event PreData**: 6 words of 24 bits
- **Event Data** (variable size up to 1024 words of 24 bits, depending on the number of samples to read in the SAMLONG chips)
- **Event PostData**: 11 words of 24 bits.

The events can be readout either via USB or UDP; data format is 24-bit word, sent on a byte basis. The payload of a standard full event is of 3123 bytes.

Event PreData is **always present** and composed of 18 bytes with the corresponding fields:

- a Fixed Header (first byte): 0x69
- SAM block ID: 4-bit word where the MSB corresponds to the type of FPGA hosting the front-end block (FE or CTRL) and the 3 LSBs to the physical location of said block on the board.
- Event ID corresponds to the 8 lower significant bits of the event number since the beginning of the run.
- For each channel, Hit Counter (16 bits) and Time Counter (24 bits) are counters used to calculate the hit rate linked to the activity on the channel since the last event. Hit Counter counts the number of times the input discriminator has be toggling since the last event, whereas Time Counter counts the time in units of 1 µs. The first counter saturating blocks the other. Taking care of memorizing this information long enough in the software, this measurement can range from 0.1 Hz to > ~400 MHz (see § 2.7).
- TDC is the value of the individual channel counter and is coded over 40 bits. The corresponding counter runs with the SAMLONG clock, thus covering a minimum of 1h30 at 200 MHz. It is reset at the start of acquisition, but it is also possible to reset it with the Sync signal which can be either produced in the CTRL FPGA or injected from the SYNCIN external input.

Event Data is **optional** and corresponds to the signal waveform. It is composed of a variable number of words fitting with the number of samples readout in the SAMLONG chips. Data is coded over 12 bits in binary complement.

Event PostData is **always present** and composed of 33 bytes. The first 30 bytes correspond to the real time measurements performed on the fly in the FE FPGAs for each channel:

- Baseline
- Peak
- Peak Time
- Charge
- Rising Edge Time

- Falling Edge Time
  For details about these measurements, see § 2.8.

Bit =>	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT PREDATA	Event ID								SAM block ID								Fixed Header							
	TDC[7..0]								Hit Counter Ch0															
	Time Counter Ch0																							
	TDC[15..8]								Hit Counter Ch1															
	Time Counter Ch1																							
	TDC[39..16]																							
EVENT DATA	Waveform Data Ch1 N words (nb of samples read)																Waveform Data Ch0 N words (nb of samples read)							
EVENT POSTDATA	Peak Ch0 (LSB)								Baseline Ch0															
	Peak Time Ch0																Peak Ch0 (MSB)							
	Charge Ch0																							
	Rising Edge Time Ch0																							
	Falling Edge Time Ch0																							
	Peak Ch1 (LSB)								Baseline Ch1															
	Peak Time Ch1																Peak Ch1 (MSB)							
	Charge Ch1																							
	Rising Edge Time Ch1																							
	Falling Edge Time Ch1																							
	Fixed Trailer								First Cell Read															

Table 3.1: Event structure produced by a 2-channel front-end block

The last 3 bytes of **Event PostData** are:

- FCR, the address of the First Cell Read in the SAMLONG chip for the current event. It is coded on 10 bits (which corresponds to one of the 1024 cells).
- a Fixed Trailer (last byte): 0x96

3.1.2 *Readout buffering*

Each pair of input channels shares a 8k-words FIFO memory (Event Buffer) in the channel FPGA (see **Figure 2.5**) that can store up to 7 full events per channel, since one full event corresponds to 1040 12-bit words (5 PreData + 1024 Data + 11 PostData). The number of events which can be stored thus depends on the event size, and can go up to 500 events when no waveform data is readout.

When the trigger occurs, the readout takes place as described in § **Making use of the analog memory**.

When the Event Buffer is full, no more trigger is accepted and the run is suspended. As soon as at least one full event is readout, the board exits the FULL condition and triggering restarts.

3.2 *Calibration and data correction*

Different types of data correction are required, in order to compensate for unavoidable production dispersion among the SAMLONG chips. Data correction is not applied at FPGA level, but must be implemented at software level by the user. All boards are factory calibrated during production test and calibration parameters are saved on-board. The WaveCatcher software provided by LAL automatically recovers the calibration parameters and uses them in order to correct the acquired data.

The different data correction types are:

- **Trigger Threshold DAC Offset Calibration**: this calibration is necessary to obtain the best precision for small signals on the trigger threshold for the channel input discriminator. The corresponding factory calibration parameters cannot be modified by the user.

- **Line Offset Calibration**: this calibration makes use of 32 DACs located inside the SAMLONG chips. It permits reducing the baseline noise down to ~ 0.95 mV rms. With this sole calibration performed, waveform data is already directly usable with a dynamic range of 11.5 bits and a sampling time precision of ~ 20 ps rms.

- **Individual Pedestal Calibration**: this calibration permits reducing the baseline noise down to ~ 0.75 mV rms, thus increasing the dynamic range to 11.7 bits. It is advised to perform this calibration with a full setup and no signal present.

- **Time INL Calibration**: this factory calibration which is optional compensates the fixed time dispersion along the sampling matrix. It makes use of a simple sinewave signal. The eventual sampling time precision scales down from ~20 ps rms to less than 5 ps rms. The corresponding factory calibration parameters cannot be modified by the user.

3.2.1 *Line Offset Correction*

The SAMLONG structure is a matrix of 16 lines and 64 columns. Whereas this structure guarantees a very stable time base, it also has the characteristic that each line is equipped with its own buffer, which provokes an offset modulo 16 in the baseline pattern. Nevertheless, this offset remains very stable. Thus, in order to compensate for it, each line of the chip is equipped with individual correction DACs.

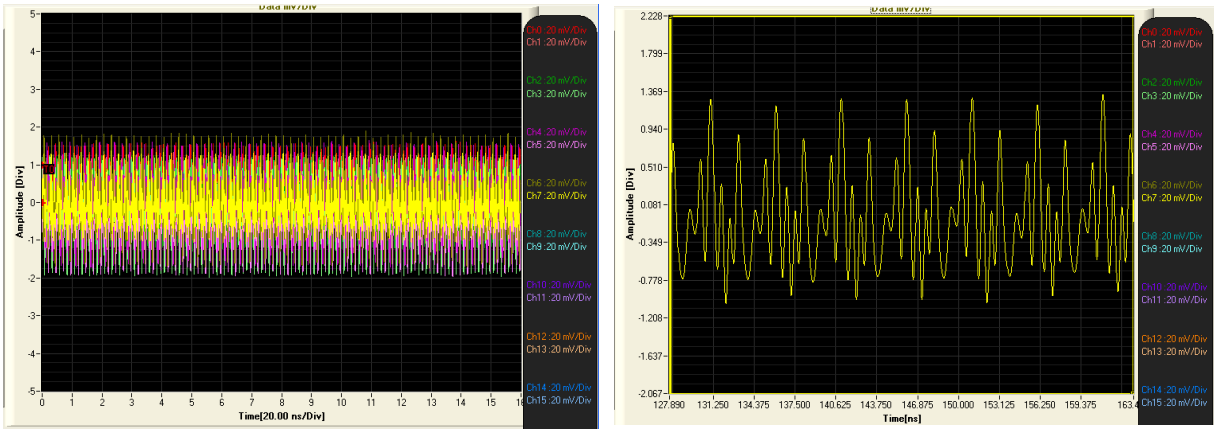


Figure 3.1.a & b: sampled waveform before line offset correction

The raw waveform before any correction (vertical scale is 20 mV/div) is shown in **Figure 3.1.a**, while **Figure 3.1.b** displays a zoom on one channel where the fixed pattern modulo 16 linked to the matrix structure can be distinguished.

Figure 3.2.a displays the sampled waveform after line offset correction with the same vertical scale (20 mV/div) and **Figure 3.2.b** shows the same plot as **Figure 3.2.a** but with a vertical scale of 2 mV/div.

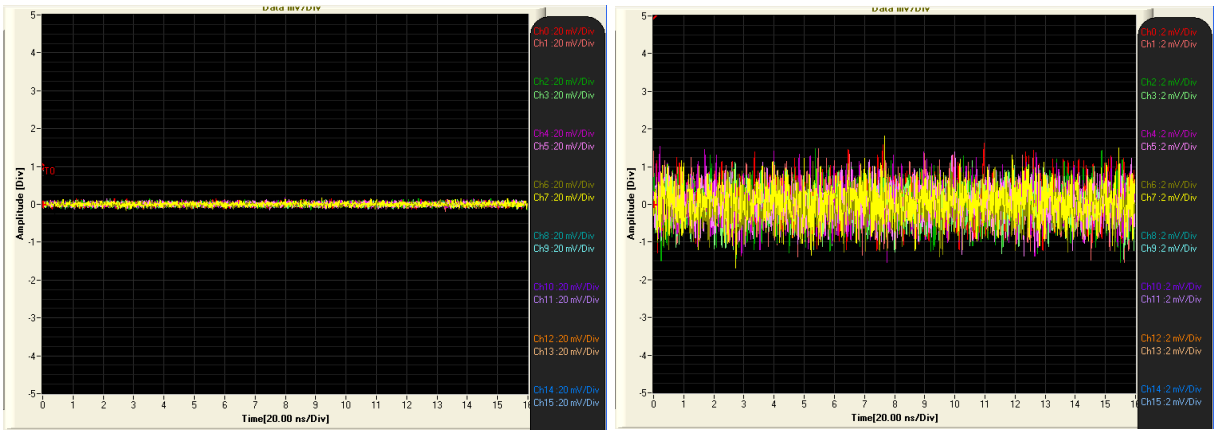


Figure 3.2.a & b: sampled waveform after line offset correction

3.2.2 Individual Pedestal Correction

After the Line Offset correction, there is still a small residual individual offset distribution remaining on the baseline. This calibration will remove it. **Figure 3.3** displays the waveform after this residual pedestal correction.

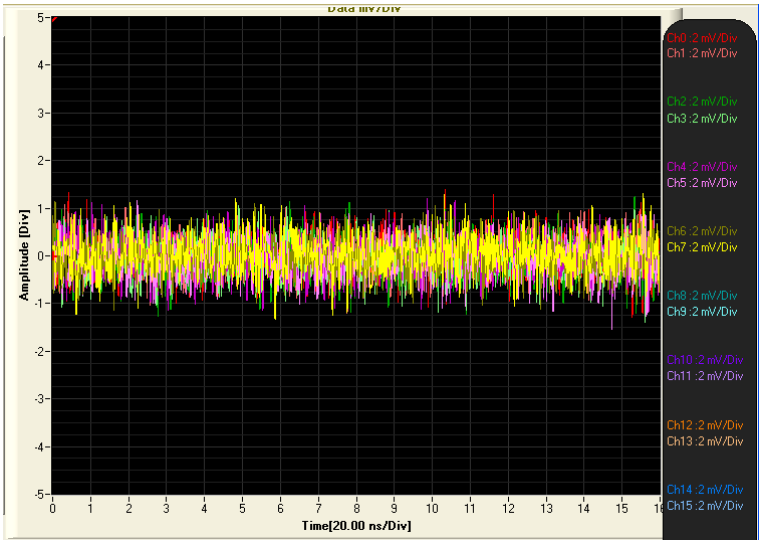


Figure 3.3: sampled waveform after individual pedestal correction

Individual pedestal calibration can be performed though the WaveCatcher software (see § ??) in the following conditions:

- All the board channels must be disconnected
- Calibration must be done after the board is at its thermal regime
- Calibration must be done each time the temperature conditions vary significantly

Please, consult the software User Manual for the specific calibration operations.

3.2.3 Time INL Correction

The sampling sequence is handled by SAMLONG through 1024 physical delay elements spread over the sampling matrix; the unavoidable production dispersion between such delay elements can be compensated through a time calibration. The following figures show an example of the integral non linearity (INL) time profile of SAMLONG chips, before and after correction. Note the extremely low residual value on **Figure 3.4.b**.

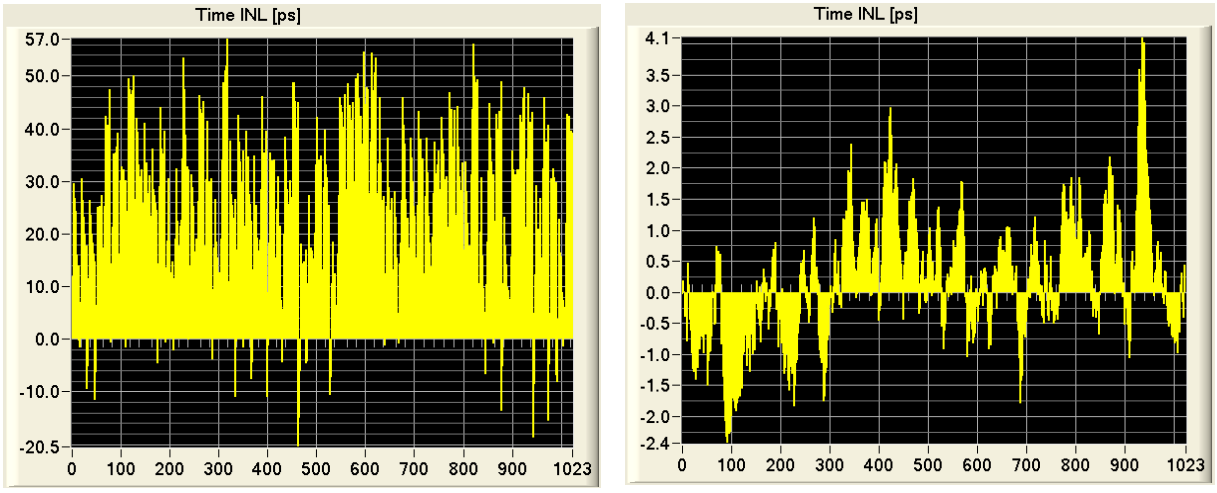


Figure 3.4.a & b: example of time INL before & after correction

3.2.4 *Trigger Threshold DAC Offset Correction*

This calibration permits setting the zero of the trigger discriminator threshold with a high precision, thus allowing triggering efficiently on very small signals around zero (a few mV).

3.3 Reset, Clear and Default Configuration

3.3.1 *Global Reset*

Global Reset is performed at Power ON of the module or via software by write access to the dedicated address. It allows to clear the data off the Output Buffer, the event counter and performs a FPGAs global reset, which restores the FPGAs to the default configuration. It initializes all counters to their initial state and clears all error conditions.

3.3.2 *Other Resets*

Different dedicated other types of resets are available, targeting only a part of the system.

3.4 Dataflow Capabilities

The board makes use of custom USB and UDP firmware blocks and librairies which permit the optimization of dataflow. Indeed, both implement a real acquisition mode where events are sent directly to the host computer without any polling from the latter. This way, the readout can benefit from the full bandwidth of the busses in terms of bytes/s.

The different sources of limitation of the event rate are summarized below:

- USB => ~30 Mbytes/s
- UDP => ~100 Mbytes/s
- SAMLONG maximum readout deadtime: 125µs/event.

One can easily understand that the event rate should be dominated by the busses, since a single pair of channels running at full speed and computing full events (1040 words) will produce a flow above 10 Mbytes/s. Real life shows that the software is often the actual limitation and cannot digest even the 30 Mbytes/s delivered by the USB bus.

4 Drivers and libraries

4.1 Drivers

In order to communicate with the WaveCatcher systems, the only necessary drivers are those of the FTDI circuit (FT2232H) used for the USB interface. They are downloadable on LAL website:

http://electronique.lal.in2p3.fr/echanges/LALUsb/software.html

4.2 Libraries

4.2.1 Bus interfaces

A few custom libraries are available and necessary for accessing the WaveCatcher systems:

LALUsbML Library: download and install the **LALUsbML library** package:

http://electronique.lal.in2p3.fr/echanges/LALUsbML/software/download

LibUdp Library: download and install the **LibUdp library** package:

http://electronique.lal.in2p3.fr/echanges/LibUdp/software/download

4.2.2 WaveCatcher64ch library

A complete high-level library has been developed for controlling and reading out the WaveCatcher systems. **There is no low-level library**.

There are both Windows and Linux versions. Files can be downloaded at the following URL:

https://owncloud.lal.in2p3.fr/public.php?service=files&t=56e4a2c53a991cb08f73d03f1ce58ba2

in the ***Library*** folder.

Full documentation can be found in the ***Documentation*** folder of the library.

5 Software tools

5.1 *Computer system requirements*

Host PC requirements:
Linux, Windows XP and above.

5.2 *Control & Readout Software*

All the versions of the WaveCatcher systems can be fully controlled and readout by the ***WaveCatcher64ch*** software. The latter runs on Windows.

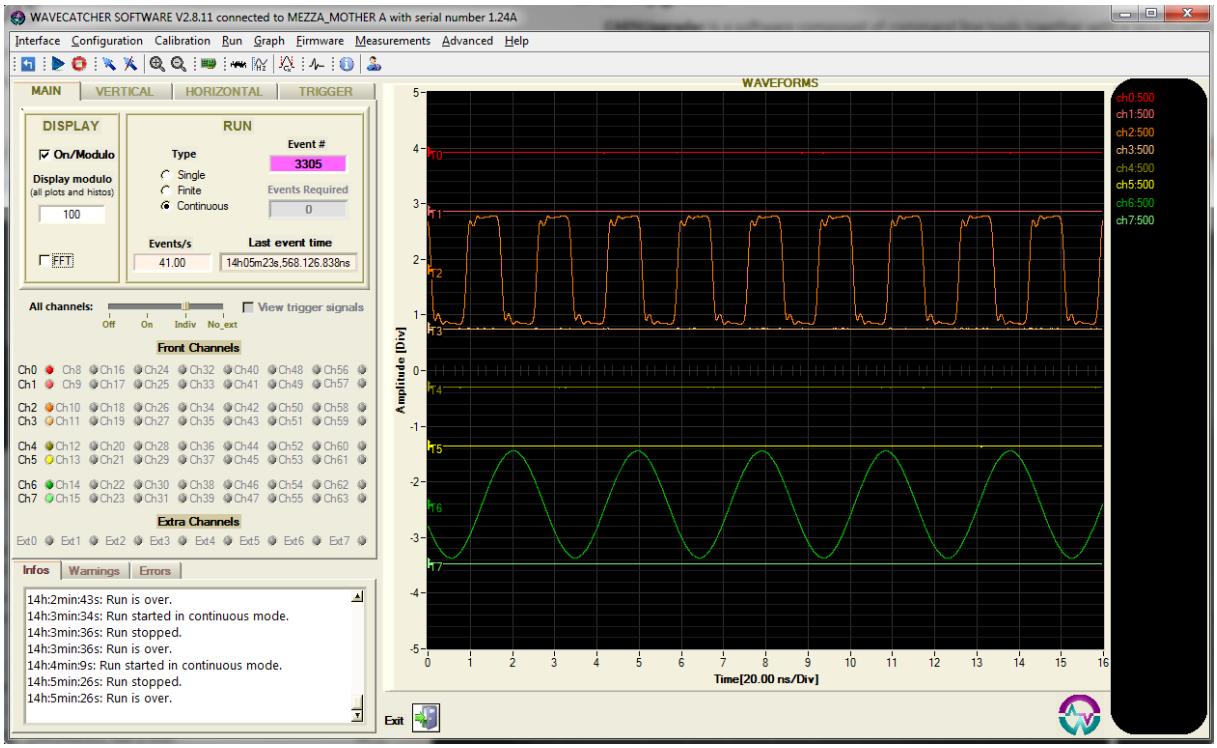


Figure 5.1: main panel of the WaveCatcher64ch software

WaveCatcher64ch software is a complete oscilloscope-like tool developed by CNRS/IN2P3/LAL and capable to control any type of WaveCatcher system.

This tool offers a graphical user friendly interface which permits taking benefit of all the functionalities of the hardware: sampling frequency, numerous trigger modes, waveform display, measurements on signals, rate monitors, channel pulsers, etc...

WaveCatcher64ch also features different tools for real-time measurements and histograms plotting: graphical cursors, noise level, raw hit rates, numerous types of measurements, time distance histograms between channels (fixed threshold and digital CFD methods), FFT, etc...

All acquired data and computed measurements can be saved to files for eventual replay or off-line analysis.

This software can be downloaded at:

https://owncloud.lal.in2p3.fr/public.php?service=files&t=56e4a2c53a991cb08f73d03f1ce58ba2

in the ***Software*** folder. It has to be fully installed the first time. Then only the executable has to be replaced for the next updates.

Binary to ROOT data converters are also available in the eponym folder.

5.3 ***Firmware upgrader***

A specific tool has been developed by CNRS/IN2P3/LAL for upgrading the firmware of the WaveCatcher systems via USB. This tool is called ***fwloader***.

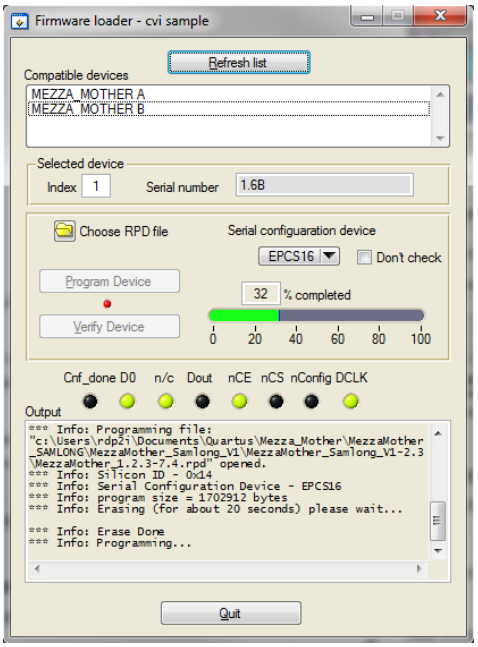


Figure 5.2: main panel of the fwloader software

It permits erasing and reloading the flash EEPROM used for the ALTERA FPGA configuration. This tool can be downloaded at the following URL:

http://electronique.lal.in2p3.fr/echanges/LALUsb/fwloader.html

It makes use of .rpd files which can be found at:

https://owncloud.lal.in2p3.fr/public.php?service=files&t=56e4a2c53a991cb08f73d03f1ce58ba2

in the ***Firmware*** folder.

Depending on the system, different rpd files should be used.

- For the 2-channel, 8-channel and 16-channel modules and boards, a single rpd file permits loading all the system FPGAs.
  - 2-channel V5: ***USB_WaveCatcher_V5_VMa.b.rpd***
  - 2-channel V6: ***USB_WaveCatcher_VMa.b.rpd***

- 8-channel V1: ***MezzaMother_1.a.b-c.d.rpd***
- 8-channel V2: ***MezzaMother_2.a.b-c.d.rpd***
- 16-channel V2: ***Wavecat_16ch_V2.a.b.rpd***

- For the 64-channel system, the controller board has its own firmware (***CrateControl64_a.b.rpd***) while the 16-channel boards use their usual one as above (***Wavecat_16ch_ V2.a.b.rpd***).