

Introduction

High speed digitizers find applications in several fields ranging from the industry to the study of nuclear and particle properties. In many cases complex measurement systems are implemented, which are composed by segmented or multiple detectors. These systems may require several acquisition channels and electronics to process all the information. Digitizers implement multiple input channels (e.g. CAEN V1720 contains 8 channels, while V1740 contains 64 of them), but it may happen that more boards are needed to acquire all the information.

The CAEN digitizers are designed also to have logic inputs and outputs which allow to create systems where boards operate as an all one board with synchronous signal sampling and same time reference. To obtain this, boards need to be “synchronized”, that is board internal clocks are set synchronously and the time reference is set the same. The synchronization, in general, allows the user to acquire from N boards with Y channel each, like if they were just one board with (N x Y) channels.

The aim of the present note is to show how to synchronize two or more homogeneous¹ CAEN digitizers in a multi-board acquisition system. We describe the steps needed to synchronize a simple system, composed by two V1720 (in the present note we generally refer to the properties of this digitizer board) and we are going now to illustrate a hardware and software setup which can be easily generalized to more complex cases.

What does “synchronize” different boards mean?

It is possible to obtain the synchronization of a multiboard system with four main system settings.

I. Same clock propagated to all boards.

The CAEN digitizers have input and output connectors which receive and distribute the clock signal. This allows to synchronize the boards by mean of the fan-out of a clock signal or the propagation through the board daisy chain. In the second case, one digitizer board generates its internal clock and distributes it by the use of external clock output connectors. Such board represents the master board of the acquisition system. The remaining boards through their input connectors are linked to the master board clock signal and act as clock slaves.

Each board contains a Phase Lock Loop (PLL) which synthesizes the board clock either from an internal oscillator or from an external clock reference. Together with a programmable phase adjust it is possible to compensate for the delay in the propagation of the clock signal from a board to the other. In this way it is possible to align the clock of each board.

II. Same time reference for all boards.

In normal applications, all boards belonging to an acquisition system have to start with the same time reference. For this reason input and output connectors are used to synchronize the start of the data taking and the time reference. The start logic signal can be propagated in daisy chain to all boards belonging to the acquisition system. The propagation of the start signal introduced a delay along the digitizer chain. This can be compensated introducing time offsets in the data acquisition start.

III. Trigger propagation and/or correlation.

Digitizers are able to receive external trigger signals and propagate them outside. This allows e.g. to propagate a global trigger signal in daisy chain when events of interest have to be recorded.

Moreover the event selection can be either due to a input signal crossing the threshold set on its acquisition channel (channel auto-trigger) or to the output of logic² algorithms: e.g. the FPGA can be programmed to require that more than one channel registered signals over the threshold (majority operator). This feature allows to implement more complex trigger selections which can be used in case of more complex detector systems where the correlation between different subunits has to be considered.

IV. Readout synchronization and event alignment³

For most of applications, all the digitizers of an acquisition system have to keep their memory buffers available for the storage of new information at the same time. The CAEN digitizers can implement a mechanism that prevents an asynchronous data taking, which may happen when at least one of the boards enters in a busy condition. The digitizer boards have BUSY IN/OUT and VETO IN/OUT connectors, which allow to stop the data acquisition and restore the system synchronization. The connectors are available with LVDS standard.

¹This application note does not show how to synchronize a number of non-homogeneous set of CAEN digitizers (V1724 + V1720 for instance). Even if non-homogeneous configuration can be in principle implemented, it has issues due to different clock speed that are not tackled in this document.

²The current firmware release allows to the require the majority logic operation.

³New firmware currently under test, the matter is not described in this document.

Board Clock, Trigger Logic and Sampling

The sampling clock of the CAEN digitizer V1720 is either locked to an internal oscillator or to an external clock source. We will later see that this clock signal is generated and distributed to the different components of the digitizer.

The counter which gives the “event time tag”, the trigger logic and the ADC sampling work with different frequencies. The event time tag, also called “trigger time tag” (TTT) is given by a counter which marks the arrival time of an external trigger signal or the time when a signal crosses the relative threshold set. For digitizers V1720 it is expressed in units of its clock cycles, 8 ns, corresponding to a frequency of 125 MHz. However, when the acquired data is written into the board internal memory, the TTT counter is read every 2 trigger logic clock cycles, which means that the trigger time stamp has an effective resolution of 16 ns.

The trigger logic algorithms operates at a frequency that is 125 MHz, while the ADC sampling frequency is 250 MHz. Boards other than the V1720 have different characteristic clock frequencies for their internal operations; please refer to the corresponding technical manual. Table 1 summarizes the clock frequencies of the digitizers CAEN V1720, V1751, V1724 and V1740.

Digitizer Model	Trigger Time Tag Resolution (ns)	Trigger Logic Cycle (ns)	Sampling Cycle (ns)
V1720	16	8	4
V1751	16	8	1
V1724	20	10	10
V1740 ⁴	32	16	16

Table 1. Characteristic clocks of CAEN digitizer V1720, V1751, V1724 and V1740.

Outline

This note describes first the steps needed to obtain the synchronization of the board clocks. This operation is necessary to proceed with the rest of the application which wants achieve a simplified synchronization of the data acquisition system.

In the next chapters three hardware setups are given as examples. We briefly describe them here:

1. The first example considers a case where all boards acquire events when an external trigger source is provided. The trigger is sent to the master board which then propagates the trigger signal in daisy chain.
2. In the second example boards acquire events independently through their channel auto-triggers (waveforms crossing the thresholds set) or signals from an external trigger source.
3. In the third example boards acquire only on an external global trigger. The channel auto-triggers are not considered for the event selection, but their logic states are propagated to the board trigger output connector. The trigger signals from each digitizer reach then an external logic unit, which then provides the system global trigger.

A more comprehensive description of the hardware setup is given in the next chapters.

Requirements

The present note is based on the synchronization of digitizers CAEN model V1720. With different digitizers, time setting may vary accordingly with the boards hardware properties. The following hardware and software tools are used.

Hardware

- 2 V1720 8 Channel 12bit 250 MS/s Digitizers
- 1 V1718 USB VME Bridge
- 1 A317 Clock Distribution cable
- 1 Oscilloscope
- 1 Programmable Function Generator
- 1 V976 VME Logic Unit (used in the third example)
- 1 V993B VME Dual Timer (optional for the third example)

⁴ Clock conditions under PLL mode acquisition, see Technical Information Manual of the V1740.

Software

- CAENUpgrader Release 1.4.1
- SyncTest

Board Firmware

- Firmware Release 3.4⁵ or successive for digitizers V1720/x51/x24

CAENUpgrader is a software composed of command line tools together with a Java Graphical User Interface (for Windows and Linux OS). CAENUpgrader allows in few easy steps to upload different firmware versions on CAEN boards and bridges, to upgrade the VME digitizers PLL, to get the board information and bridge firmware revision and to manage the firmware license.

CAEN SyncTest is a simple software written specifically for this application note. It contains the most relevant commands to adjust the configuration parameters of the boards and read the acquired event data. It represents an example for setting synchronization and trigger distribution. It is provided as an archive of ANSI C source and header files.

Clock Synthesis and Distribution

Figure 1 reports a simplified sketch of the digitizers clock synthesis and distribution. We report here a brief description of the hardware components and their operations. We will see in the next pages how **CAENUpgrader** modifies the clock configuration.

- In CAEN digitizers the clock management is provided by a PLL (Phase-locked Loop) and a Clock Distributor. The PLL can receive a reference clock from either an internal oscillator or an external clock source through the clock input (CLK-IN) connector. A mechanic switch on the board allows to select the clock generator.
- The role of the PLL is therefore to align the phase of a Voltage Controlled Crystal Oscillator (VCXO) of the digitizers to the reference one.
- The clock generated by the VCXO is passed to the Clock Distributor, which splits the clock signal in different branches. Each branch but one is sent to board subsystems: the main board FPGA and the mezzanines (which contain ADC's, FPGA's and memory buffers). The remaining branch is connected to the clock output (CLK OUT) connector. The Clock Distributor can send a different sub multiple of the VCXO frequency to each branch.
- The Clock Distributor can also apply a delay to the CLK OUT connector. This is a key feature of the synchronization since it can compensate the effect of the clock shift due to the clock daisy chain between different boards.

⁵In this Application Note we used mainboard FPGA firmware Release 4.0 and mezzanine FPGA firmware Release 0.11.

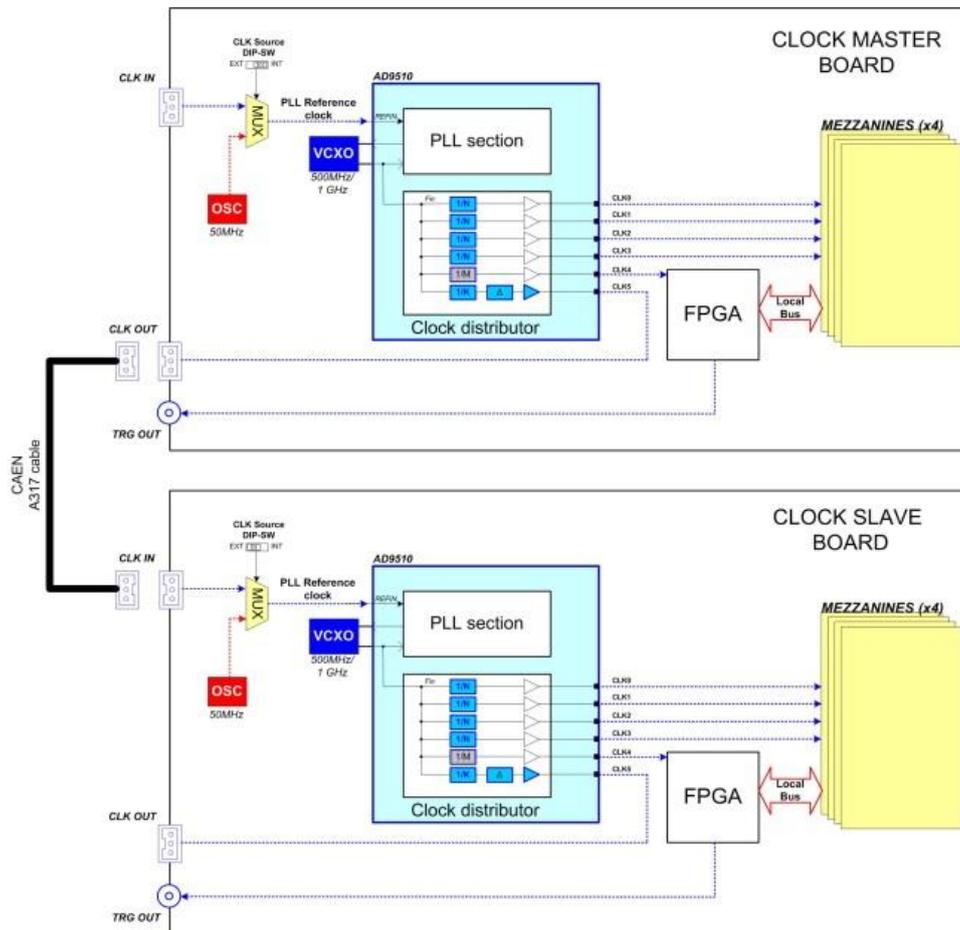


Figure 1: Representation of the clock signal synthesis and distribution in the digitizer boards.

Clock Synchronization

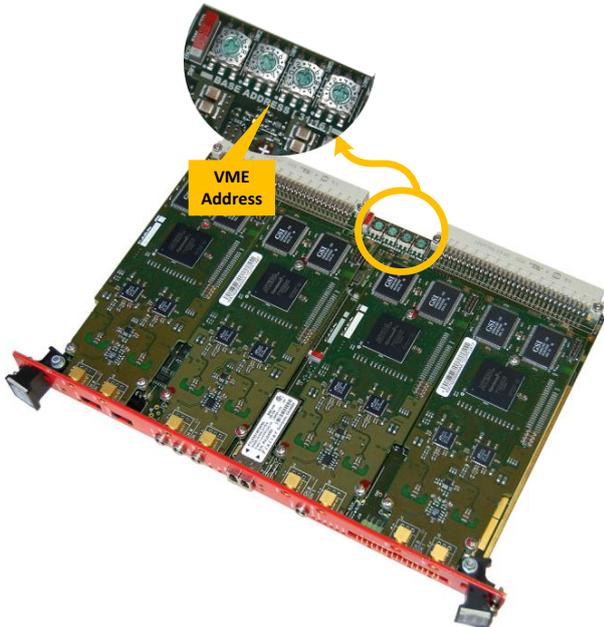
As we already wrote above, the first action to be performed to obtain the synchronization of the acquisition system is the synchronization of the clock and its phase alignment.

- In the multi-board acquisition system, the master board (hereafter Master), will act as clock master providing a reference clock to the other one, that therefore will be a clock slave (hereafter Slave).
- The CLK OUT of the Master is connected to the CLK IN of Slave through the A317 cable.
- With **CAENUpgrader** software we configure the Master PLL in order to enable the output of the clock (62.5 MHz) and configure the Slave PLL in order to accept the external clock.
- The TRG OUT of both boards can be programmed using **Synctest**, to deliver the clock signal. This will allow through the use of an oscilloscope to observe the alignment of the clock signals.
- Then we use the **CAENUpgrader** program to set a delay on the Master clock output in order to obtain the alignment of the clock signals produced by TRG OUT of each board.

We report in the next pages a step-by-step example of clock synchronization procedure.

NOTE: this note describes an hardware setup with digitizers controlled by a PC connected to an USB VME Bridge Mod. V1718. With appropriate drivers it is possible to use the connections through Optical Link. It requires the use of PCI/PCIe optical controller A2818/A3818 directly connected to the Digitizers or via the V2718 VME/PCI Bridge.

Step 1



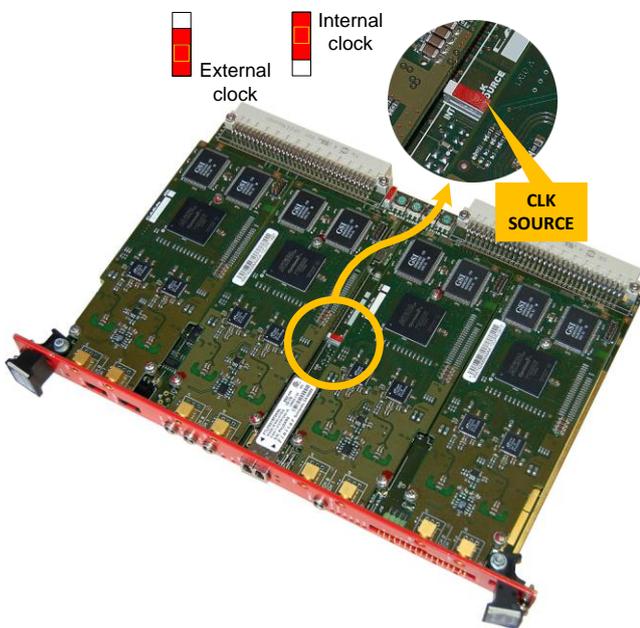
Hardware Setup (a): Set VME address

The VME addresses used in this example are:

Master: 0x32100000

Slave: 0x33100000

Step 2

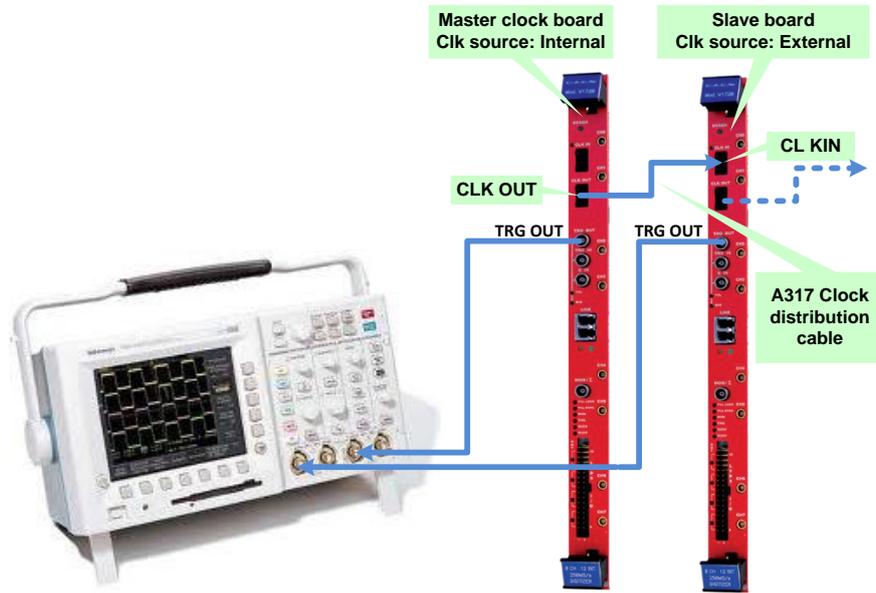


Hardware Setup (b): Set board clock sources

Master clock source: internal

Slave clock source: external

Step 3



Hardware Setup (c): connect boards and oscilloscope

- Connect Master CLK OUT to Slave CLK IN using A317 cable
- Connect the TRG OUT outputs of both the boards to an oscilloscope with cables of equal length (50 Ω termination required)

Step 4

Set SyncTest operational parameter in userparam.c source

```
#include "synctest.h"

void SetUserParams(UserParams_t *Params)
{
    // CONNECTION PARAMETERS:
    // ConnectionType: can be CAEN_DGTZ_USB,
    //CAEN_DGTZ_PCI_OpticalLink (A2818) or CAEN_DGTZ_PCIE_OpticalLink (A3818)
    // LinkNum: USB or PCI/PCIE enumeration (typ=0)
    // ConetNode: position in the optical daisy chain
    // BaseAddress: only for VME access (otherwise 0)

    Params->ConnectionType[0] = CAEN_DGTZ_USB; //CAEN_DGTZ_PCI_OpticalLink; //
    Params->LinkNum[0] = 0;
    Params->ConetNode[0] = 0;
    Params->BaseAddress[0] = 0x32100000; //Master board

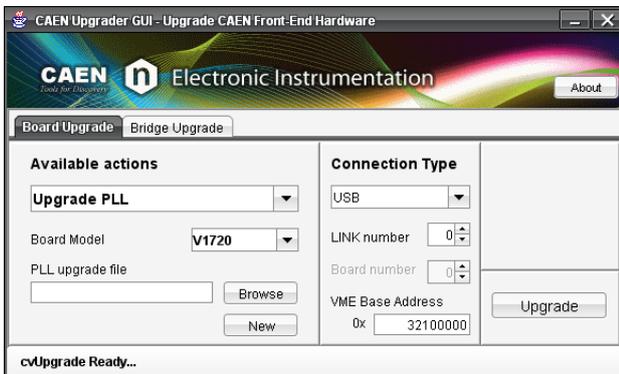
    Params->ConnectionType[1] = CAEN_DGTZ_USB; //CAEN_DGTZ_PCIE_OpticalLink; //
    Params->LinkNum[1] = 0;
    Params->ConetNode[1] = 0;
    Params->BaseAddress[1] = 0x33100000; //Slave board
}
```

USB connection

VME address Master

VME address Slave

Step 5

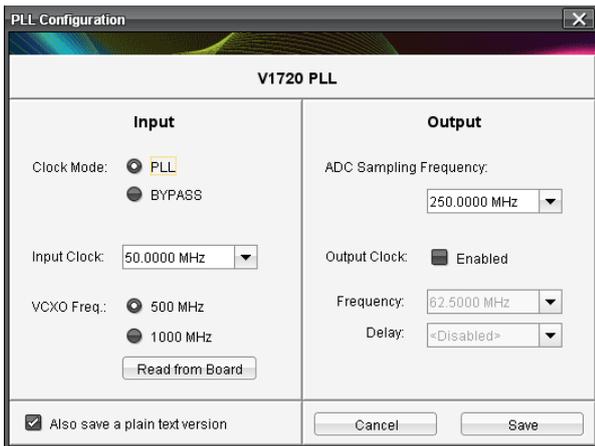


Run CAENUpgrader to program Master PLL (a)

From the “Board Upgrade” tab, perform the following steps.

- Select the “Upgrade PLL” option in the “Available actions” scroll menu: the “Board’s Model” box will appear right below.
- Select your board model in the “Board’s Model” scroll menu; in our example we selected a V1720.
- “Connection Type” control box will let you select the type of connection you are using (in our example, USB).
- “LINK number” allows for choosing between different boards/bridges linked to your computer, if you are using more than one connection link. In our example we leave this field set to 0, which is the default value.
- Type in the appropriate box the VME Base Address of the Master VME Board.
- Click New button to open PLL configuration window.

Step 6

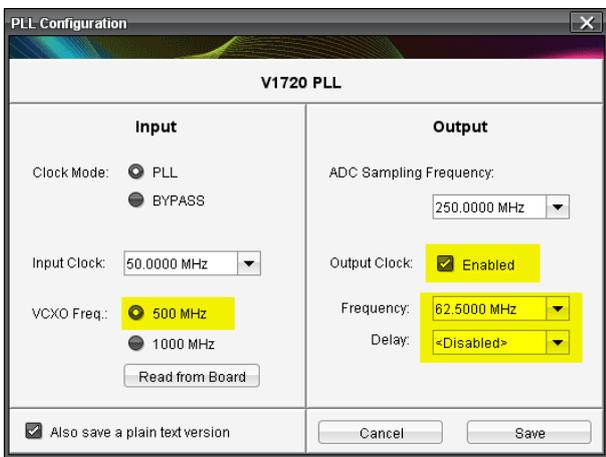


Run CAENUpgrader to program Master PLL (b)

In the PLL configuration window perform the following steps.

- Set the VCXO frequency. You can click on the “Read from Board” button.
- A new window will appear which reports all the connection information already entered in “Connection Type”.
- Click on the “Read from Board” button on the window

Step 7

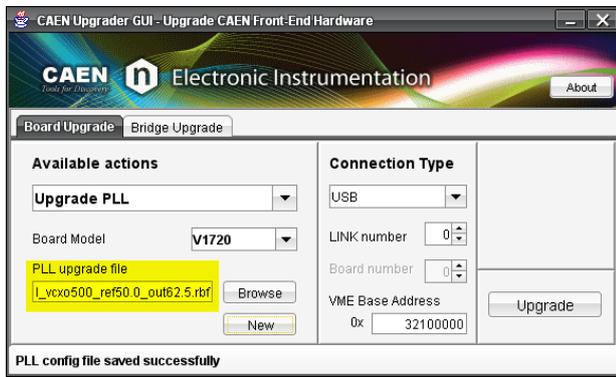


Run CAENUpgrader to program Master PLL (c)

In the PLL configuration window perform the following steps.

- Enable the output clock and set its frequency. Different frequencies can be chosen and in this example we choose 62.5 MHz.
- At first stage we set Delay = <Disabled>.
- Once the PLL configuration has been set according to your purposes, you have to save the configuration file on your computer by clicking on the “Save” button.

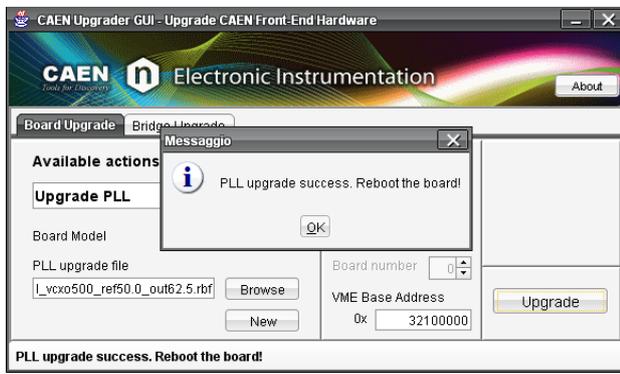
Step 8



Run CAENUpgrader to program Master PLL (d)

- Automatically the PLL settings window disappears and the file name and location is loaded in the “PLL upgrade file” box.
- Click “Upgrade” button.

Step 9

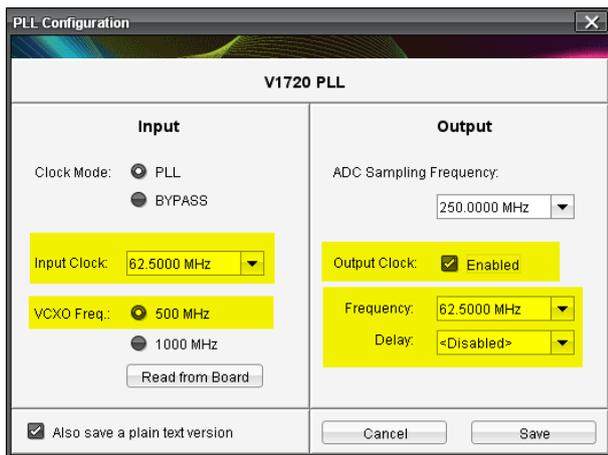


Run CAENUpgrader to program Master PLL (e)

If the Master PLL upgrade is OK, the PLL upgrade success message is displayed.

- **Do not reboot** now the Master, but proceed with the upgrade the Slave PLL.
- Click “New” button to open PLL configuration window.

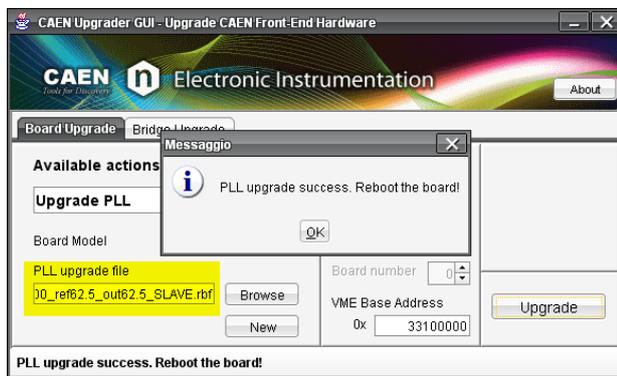
Step 10



Run CAENUpgrader to program Slave PLL (a)

- Repeat the step P6 setting in the appropriate box the VME Base Address of the Slave VME Board.
- Set the VCXO frequency (click the “Read from Board” button).
- Set the input clock at 62.5000 MHz frequency.
- The output clock can be set for the slave board. This configuration is necessary in case the slave must provide the clock to the next slave board in the daisy chain. In the figure 62.5 MHz output clock is set.
- We set Delay = <Disabled>.

Step 11



Run CAENUpgrader to program Slave PLL (b)

In the PLL configuration window perform the following steps.

- Once the PLL configuration has been set according to your purposes, you can save the configuration file on your computer by clicking on the "Save" button.
- Automatically the PLL settings window disappears and the file is loaded in the "PLL upgrade file" box.
- Click "Upgrade" button.
- If the Slave PLL upgrade is OK, the PLL Upgrade success message is displayed

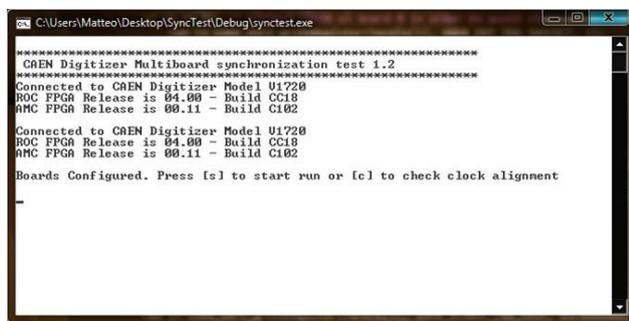
Step 12



Reboot the boards, compile and run SyncTest

- Reboot the boards.
- Compile **SyncTest**. For this procedure 3 CAEN libraries are required: CAENDigitizer, CAENComm (rel. 1.0 or later), CAENVMELib.

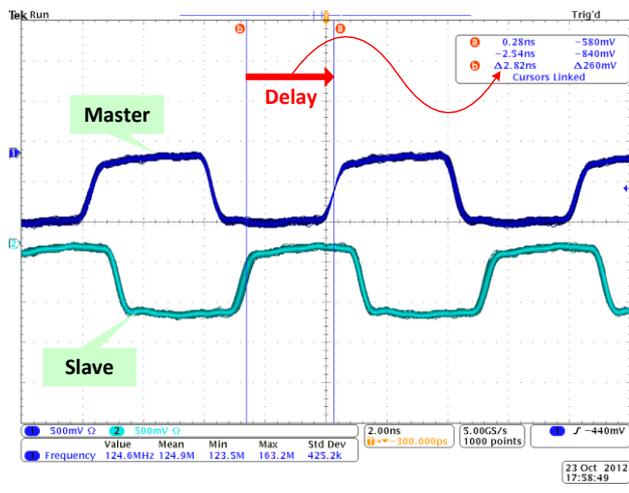
Step 13



Compile and run SyncTest

- Compile SyncTest.
- Launch SyncTest and press "c" to enable the clock signals on the TRG OUT connectors.
- Signals on the scope should appear.

Step 14



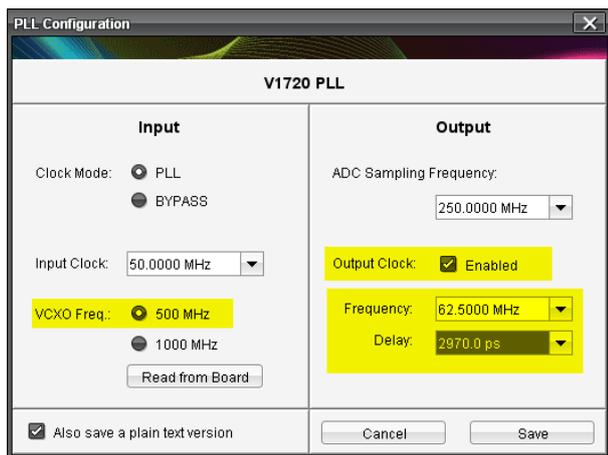
Scope display

The two clock signals on the TRG OUT connectors have the same frequency, i.e. the delay between the edges of the two signals is kept constant. This behavior is ensured by the Slave PLL that is locked on the clock output by the Master. However the two signals are not in phase.

The digitizers allow to set a delay to the clock output. We can set a delay on the Master clock output so that the rising edge of the Slave is retarded and locked the master rising edge. We can use the **CAENUpgrader** to set the proper delay.

- Frequency of Master and Slave clock on the TRG OUT connectors = 125 MHz (period T= 8 ns).
- Delay between Master and Slave clocks on the TRG OUT connectors: $\Delta = 2.82$ ns.
- Delay to be applied at Master CLK OUT = 2.82ns.

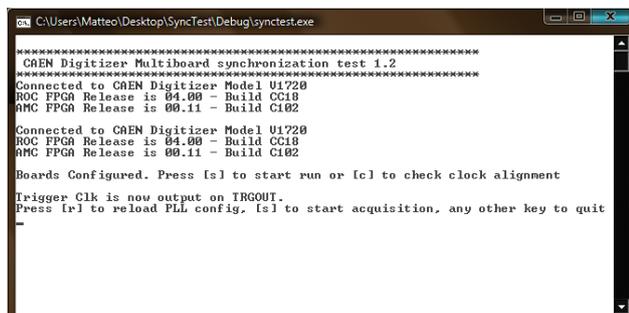
Step 15



Run CAENUpgrader to program Master PLL (a)

- Repeat the Step 5
- Set the VCXO frequency (click the “Read from Board” button).
- Enable the Output clock at 62.5000 MHz frequency.
- The “Delay” menu allows to set the delay value which is the closest to the desired one. The possible choices range from 570 ps to 9870 ps in 31 steps of 300 ps.
- In this example the delay set = 2970.0 ps.
- Once the PLL configuration has been set according to your purposes, you can save the configuration file on your computer by clicking on the “Save” button.
- Automatically the PLL settings window disappears and the file is loaded in the “PLL upgrade file” box.
- Click “Upgrade” button.
- If the Master PLL upgrade is OK, the PLL Upgrade success message is displayed.

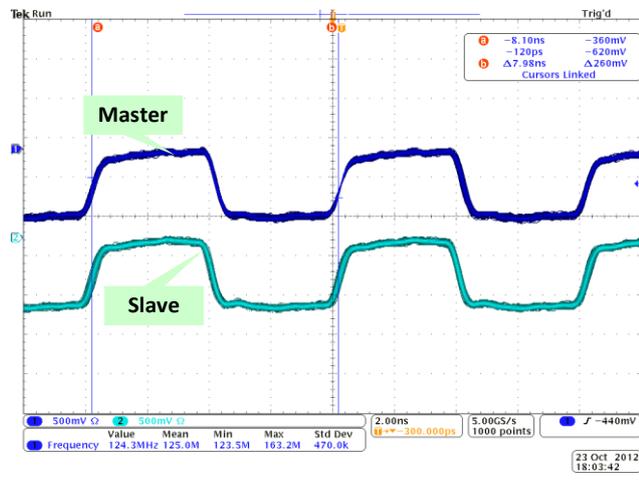
Step 16



Run SyncTest

- Launch SyncTest and press “c” to enable the clock signals on the TRG OUT connectors.
- Signals on the scope should appear
- Press “r” in the SyncTest window to reload the PLL configuration. Signals on the scope should be now synchronized.

Step 17



Scope display

The two clock signals are time aligned.

Example 1: External Trigger Propagation

This section will describe how to synchronize different boards **where the trigger comes from an external source and is propagated along the system through a daisy chain**. The aim is to show how to connect all the boards to propagate the trigger signal in daisy chain. An external trigger source generates the trigger signal to be fed in the Master input.

The hardware setup is shown in Figure 2. The boards are connected as in the follows.

- Connect Master CLK OUT to Slave CLK IN using A317 cable.
- Connect the TRG OUT of the Master to TRG IN of the Slave.
- Connect Function Generator CH1 output to a 50 Ω passive splitter.
- Connect both channels (#6 in this example) of each digitizer to the splitter with identical length cables.

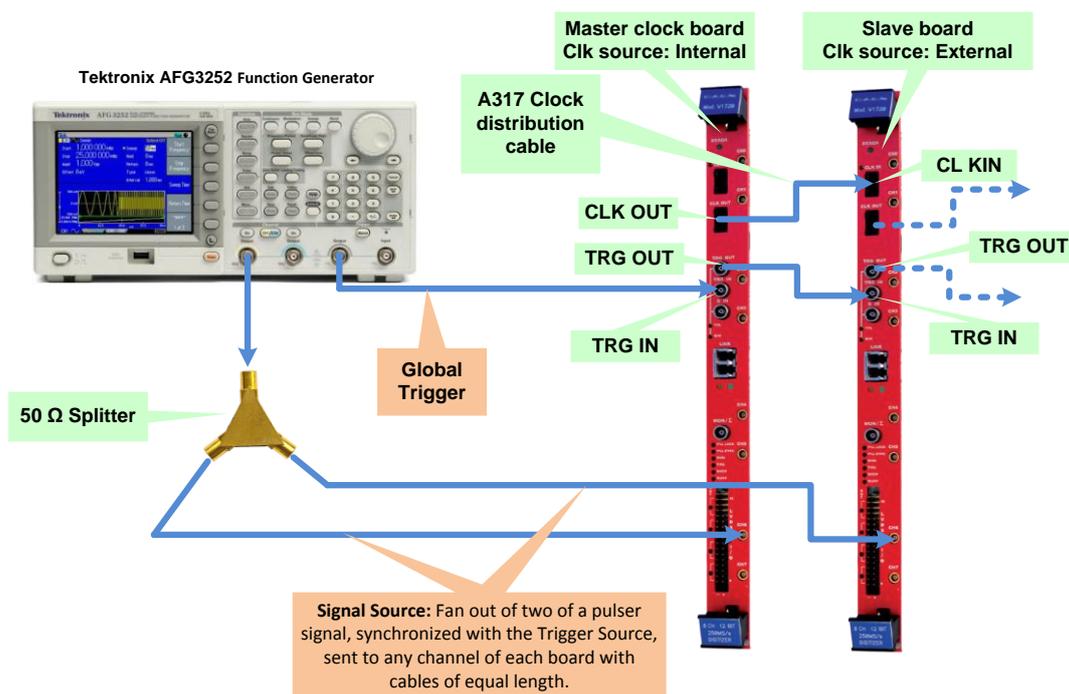


Figure 2: Hardware setup. Connection of boards and function generator.

The setup is such that:

- the external trigger goes to TRG IN of the Master;
- the daisy chain TRG IN/TRG OUT propagates the trigger and the start of the data acquisition;
- the signal source is the fan out of two of a function generator, synchronized with the trigger source, sent to one channel of each board with cables of equal length.

The start of the event acquisition happens in the following way:

- all Slave boards armed to start with TRG IN edge;
- a software (SW) trigger is sent to the Master (with the use of SyncTest⁶, described in Step 21);
- the SW trigger is propagated through the daisy chain TRG IN/TRG OUT and starts all the boards (in this way a delay is introduced and is described next);
- at this point Master is programmed in order to accept trigger on the TRG IN connector. The external trigger signals go through the daisy chain and trigger all the boards.

In case one of the boards is in a busy state the acquisition can lose the event alignment. To veto this possibility a busy signal can be propagated to last board (with LVDS I/Os) and sent through the TRG OUT of the last board (NIM or TTL) to inhibit either the external trigger source or the Master trigger input through the S IN programmed as trigger VETO⁷.

⁶ The run can also start with the 1st hardware trigger, the software has to be configured for this purpose.

The daisy chain propagates the trigger signal from the Master to the Slave. The SW trigger that starts the acquisition arrives to the Slave with a delay proper of this hardware configuration. To ensure that all digitizers start the acquisition at the same time a delay in the start of the event acquisition can set for the Master. This feature can be generalized to a hardware system with multiple boards. Figure 3 represents the timing of the SW trigger and the acquisition start of a system with 2 or 3 boards.

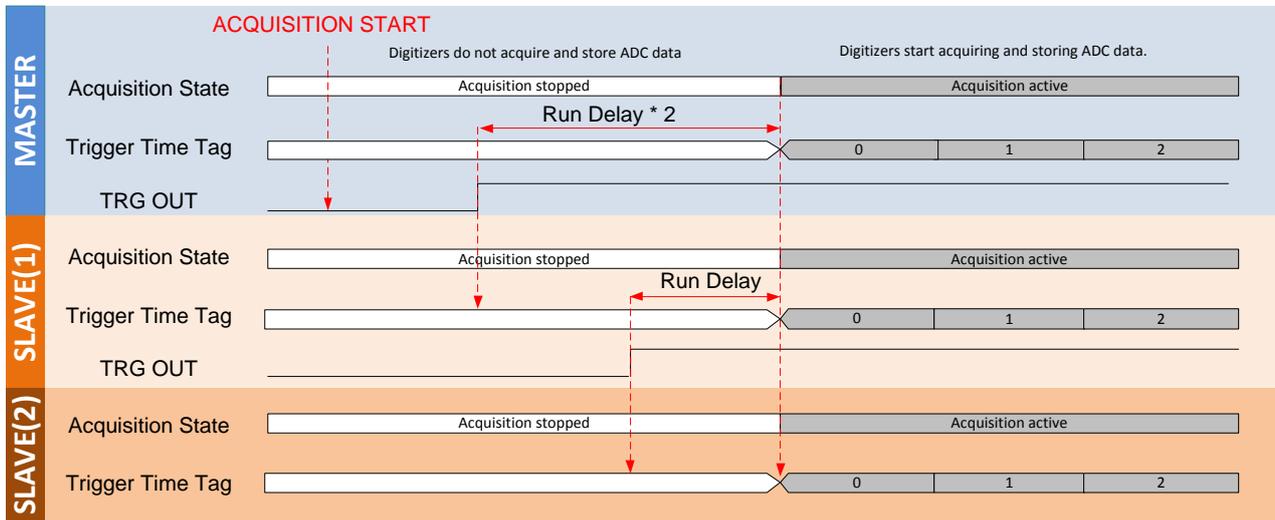


Figure 3: Timing diagram of the start of run sequence

In **SyncTest** (see 0) you can set the correct delay to obtain the time alignment of the board acquisition start: this will ensure that they will have the same time reference. Note in Figure 3, that the trigger time tag (TTT) counters start at the same time; this will then allow to require temporal correlations between events acquired from different digitizers.

In this hardware setup the digitizers are configured such that the signals from the external global trigger, received after the SW trigger start, are considered valid for the event acquisition. The trigger signal sent to the Master is propagated to the Slave in daisy chain, so the latter will receive the trigger signal with a fixed delay compared to the Master. All boards start the acquisition synchronously, so the trigger propagation introduces a delay also in the TTT of the events acquired by the Slave. Figure 4 represents the resulting TTT recorded by the boards.

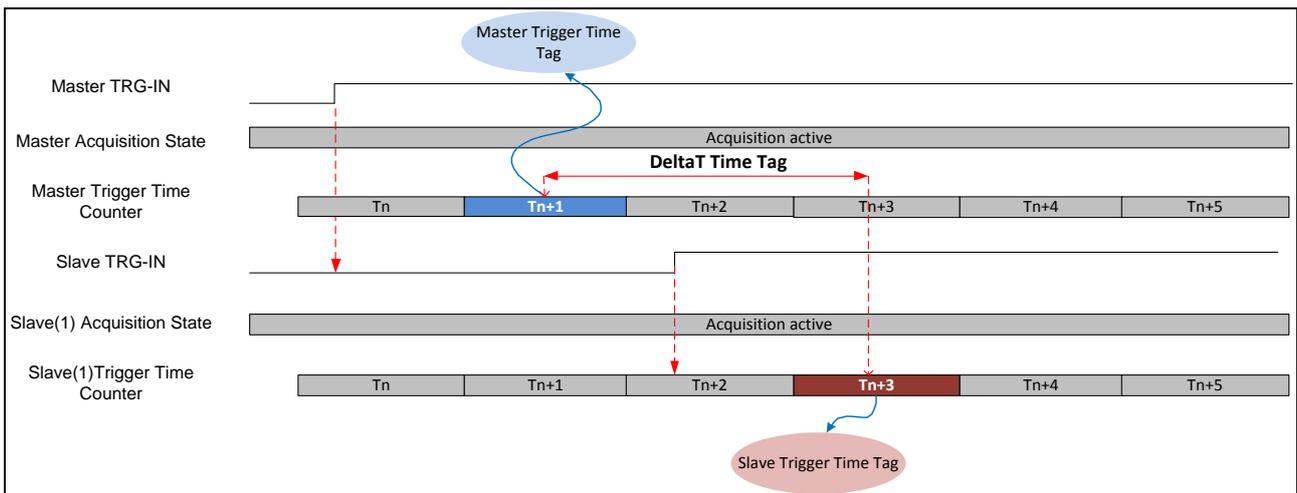


Figure 4: Synchronous Master and Slave acquisition (note that the Trigger Time Tags are different)

⁷ New firmware currently under test, not described in this document

The memory buffer of the digitizers record the events with a timing that is given by the arrival of the trigger signal. The delay in the TTT of the Slave implies that its memory buffer contains the sampling information that is out of phase compared to the one of the Master. Figure 5 better explains the delay introduced in the memory buffer of the Slave.

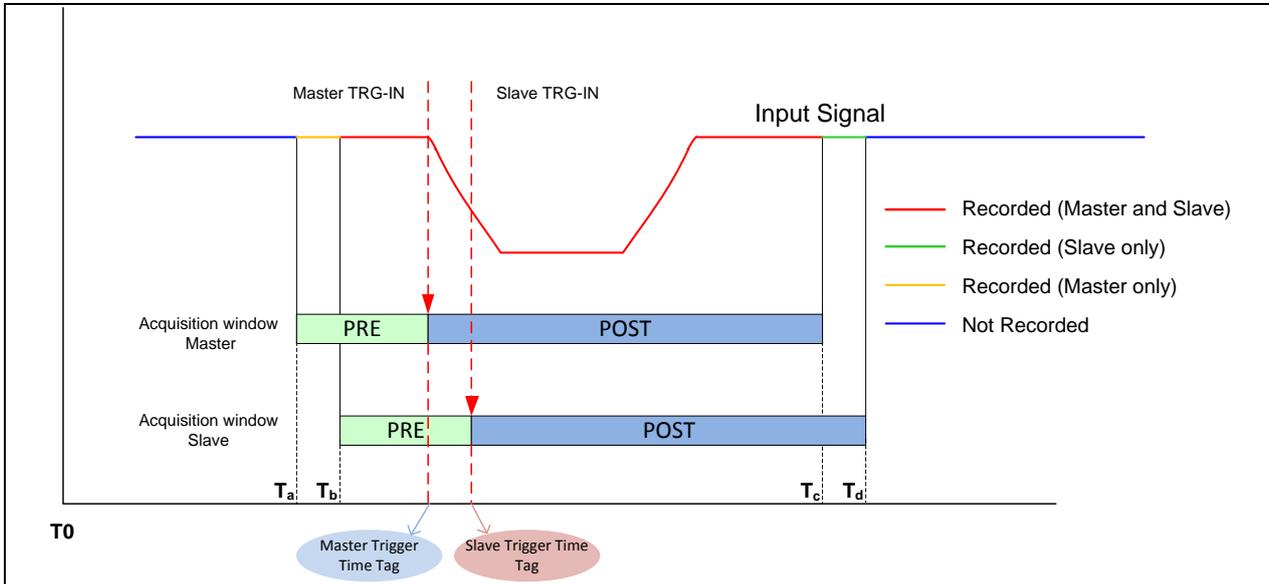


Figure 5: Master and Slave acquisition windows (time frame)

The SyncTest software is set in a way that waveforms can be observed both with the representation in the “buffer reference” frame and the “time reference” frame (see Step 27 and Step 28). The representation in the time reference is shown in Figure 6, where T_a and T_b are the arrival times of the trigger signal minus a fixed offset (see also Figure 5). The representation in the buffer frame is shown in Figure 7.

We now describe the step necessary to obtain the synchronization of the boards in this hardware configuration.

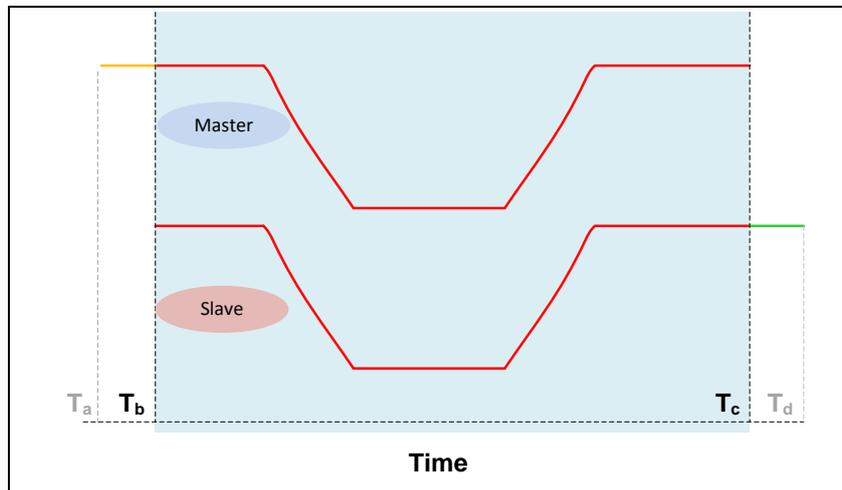


Figure 6: SyncTest plot of the Master and Slave acquisition windows (time frame).

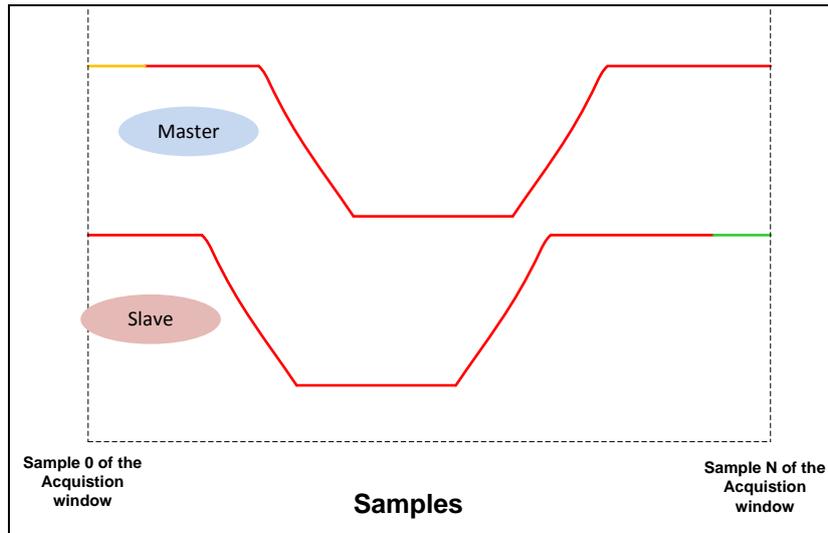


Figure 7: SyncTest plot of the Master and Slave acquisition windows (buffer frame).

<h3>Step 18</h3>	<p>Digitizer clock synchronization</p> <ul style="list-style-type: none"> - Perform the Step 1-17 to synchronize and align the clock - Once clocks are synchronized, it is possible to configure the start of acquisition in order to have the same time reference for both boards.
------------------	--

<h3>Step 19</h3>	<p>Hardware setup (B): set signal parameter of pulse generator</p> <ul style="list-style-type: none"> - The program SyncTest operates a fit on the falling (or rising) edge of the pulse to compute time information. If multiple points are available to fit the waveform the result may be more reliable. - Set rise time > 5 x sampling period. In this example rise time is set to 50 ns. - Set width = 300 ns.
------------------	--

Step 20

Set SyncTest operational parameter in userparam.c source

```
// *****
// Start Mode:
// *****
// Options: START_SW_CONTROLLED, START_HW_CONTROLLED
Params->StartMode = START_SW_CONTROLLED;

// *****
// Sync Mode:
// *****
// Options: COMMONT_EXTERNAL_TRIGGER_TRGIN_TRGOUT, INDIVIDUAL_TRIGGER_SIN_TRGOUT, TRIGGER_ONE2ALL_EXTOR
Params->SyncMode = COMMONT_EXTERNAL_TRIGGER_TRGIN_TRGOUT;

// CHANNEL SETTINGS
Params->RefChannel[0] = 6; // Channel of the Master used for the acquisition
Params->TriggerThreshold[0] = 2000; // Trigger threshold (for self triggering)
Params->PostTrigger[0] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[0] = 0x8000; // input DC offset adjust (DAC value)

Params->RefChannel[1] = 6; // Channel of the Slave used for the acquisition
Params->TriggerThreshold[1] = 2000; // Trigger threshold (for self triggering)
Params->PostTrigger[1] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[1] = 0x8000; // input DC offset adjust (DAC value)

// Trigger edge (CAEN_DGTZ_TriggerOnRisingEdge, CAEN_DGTZ_TriggerOnFallingEdge)
Params->TriggerEdge = CAEN_DGTZ_TriggerOnFallingEdge;

// Number of samples in the acquisition windows
Params->RecordLength = 500;

// Max. distance between the trigger time tags in order to consider a valid coincidence
Params->MatchingWindow = 10;

// Front Panel LEMO I/O level (NIM or TTL). Options: CAEN DGTZ IOLevel NIM, CAEN DGTZ IOLevel TTL
Params->IOlevel = CAEN_DGTZ_IOLevel_TTL;
```

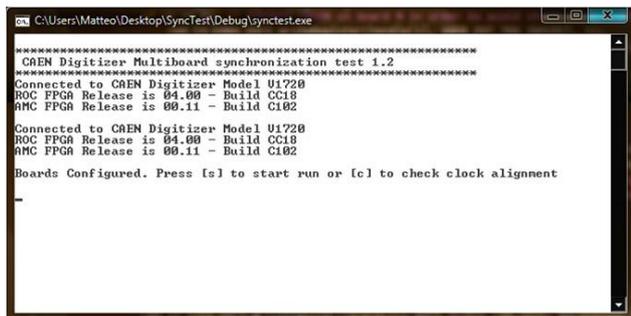
Set Start Mode software controlled. In this way the Master Board will wait for a software signal to start the acquisition and will propagate this start signal to the Slave board through the TRG OUT TRG IN daisy chain

Set Sync Mode according to this Synchronization setup

Set acquisition channel Master and Slave

Set Front Panel I/O level

Step 21



Compile and Run SyncTest

- Compile SyncTest
- Launch SyncTest and press "s" to start run
- Once the acquisition is started, the SyncTest window should show some parameter as below

Step 22

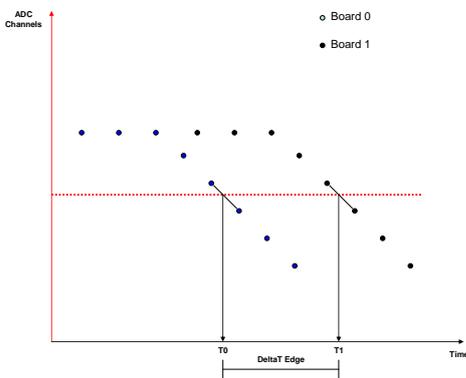
```

C:\Users\Matteo\Desktop\SyncTest\Debug\syncstest.exe
DeltaT time tag: mean= 16.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 58.1434 sigma= 0.0070
DeltaI time tag: mean= 16.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 58.1444 sigma= 0.0069
DeltaI time tag: mean= 16.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 58.1449 sigma= 0.0069
DeltaI time tag: mean= 16.0000 sigma= 0.0000
    
```



SyncTest acquisition parameters (1)

The program output on the shell should appear in the format represented in the figure on the left. If the configuration parameters are successfully set all fields appear complete. If some information is missing it is necessary to modify the channel DC offsets (see Step 27) or the trigger thresholds (see Step 30).

- **Readout Rate** shows the amount of data transferred from the board.
- **TrgRate** shows the Trigger rate felt by the two boards. It must obviously be the same for both the boards and equal to the frequency of the external trigger source.
- **Matching Events** value shows the percentage of events within the coincidence window. The width of this coincidence window can be set in the Userparams.c. Since signals to the board are synchronous with the external trigger a **100%** value is expected.
- **Missing Edge** value shows the number of events in which signals are not in the coincidence window.
- **DeltaT Edge** value is the measurement of the difference in time between the two signals in ns. It is calculated making an **interpolation of two samples before and after the threshold and then calculating the difference between the two resulting points on the threshold**. The run start is propagated in daisy chain, so at the present time, without any correction, the time references are not synchronized.
- **DeltaT Time tag** value is the difference in ns between the 2 trigger time tag (TTT) within the coincidence window.

Step 23

```

C:\Users\Matteo\Desktop\SyncTest\Debug\syncstest.exe
DeltaT time tag: mean= 16.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 58.1434 sigma= 0.0070
DeltaI time tag: mean= 16.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 58.1444 sigma= 0.0069
DeltaI time tag: mean= 16.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 58.1449 sigma= 0.0069
DeltaI time tag: mean= 16.0000 sigma= 0.0000
    
```

Syncstest acquisition parameters (2)

- **DeltaT Edge = 58.1 ns** is about four times the cycle of the trigger time tag (4 x 16 ns for the V1720 module).
- The firmware allows to add a delay to the start of the acquisition, so we proceed with the adjustment of RUN delay parameter in syncstest.c source (see Step 26).
- The delay is expressed in unit of trigger time tags (16 ns for the V1720 module).

Step 24

Set SyncTest operational parameter in syncstest.c source

```
int SetSyncMode(int handle[2], UserParams_t Params)
{
    int i, ret=0;
    uint32 t reg;

    for(i=0; i<2; i++) {
        switch (Params.SyncMode) {
            case COMMONT_EXTERNAL_TRIGGER TRGIN TRGOUT :
                if (i == 0)
                    // inhibit TRGIN on board 0 in order
                    // to avoid start of run with external triggers
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_EXT_TRG_INHIBIT, 1);
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_GLOBAL_TRG_MASK, 0xC0000000);
                    // accept EXT TRGIN or SW trg
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_TRG_OUT_MASK, 0xC0000000);
                    // propagate both EXT and SW TRG to TRGOUT
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_ACQUISITION_MODE, RUN_START_ON_TRGIN_RISING_EDGE);
                    // Run starts with 1st trigger edge
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_RUN_DELAY, 4*(1-i));
                    // Run Delay decreases with the position
                    // (to compensate for run the propagation delay)
                break;

            case INDIVIDUAL_TRIGGER SIN TRGOUT:
                if (i > 0)
                    // Run starts with S-IN on the 2nd board
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_ACQUISITION_MODE, RUN_START_ON_SIN_LEVEL);
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_GLOBAL_TRG_MASK, 1<<Params.RefChannel[i]);
                    // accept only trg from selected channel
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_TRG_OUT_MASK, 0);
                    // no trigger propagation to TRGOUT
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_RUN_DELAY, 2*(1-i));
                    // Run Delay decreases with the position
                    // (to compensate for run the propagation delay)
                // Set TRGOUT=RUN to propagate run through S-IN => TRGOUT daisy chain
                ret |= CAEN_DGTZ_ReadRegister(handle[i], ADDR_FRONT_PANEL_IO_SET, &reg);
                reg = reg & 0xFFFF0000 | 0x00010000;
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_FRONT_PANEL_IO_SET, reg);
                break;

            case TRIGGER_ONE2ALL_EXTOR:
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_GLOBAL_TRG_MASK, 0x40000000);
                // accept ext trg in (from trg OR)
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_TRG_OUT_MASK, 0x80000000 | (1<<Params.RefChannel[i]));
                // propagate auto trg and SW trg to TRGOUT
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_ACQUISITION_MODE, RUN_START_ON_TRGIN_RISING_EDGE);
                // Arm acquisition (Run will start with 1st trigger)
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_RUN_DELAY, 0);
                // Run Delay = 0 for all
            break;

            default:
                return -1;
            break;
        }
    }
    return ret;
}
```

Synchronization setup chosen

Set start delay to synchronize the digitizers, so that they start the acquisition at the same time.

In our example set start delay = 4 x 16 ns for the Master (multiple of the memory clock cycles).

Step 25

```
CAUsers\Matteo\Desktop\SyncTest(Debug)\syncstest.exe
DeltaT time tag: mean=-48.0000 sigma=0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaT edges: mean=-5.9101 sigma=0.0060
DeltaT time tag: mean=-48.0000 sigma=0.0000

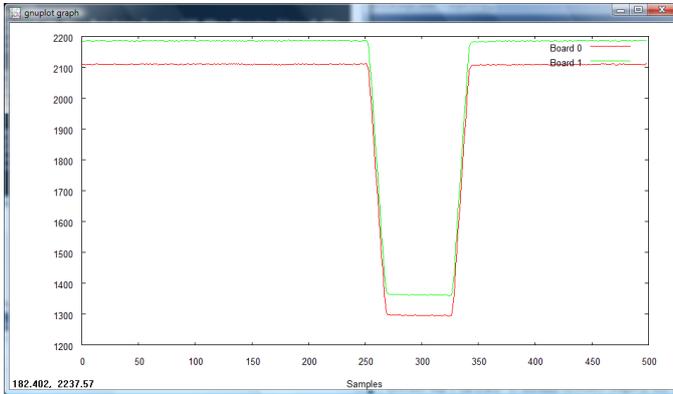
Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaT edges: mean=-5.9101 sigma=0.0060
DeltaT time tag: mean=-48.0000 sigma=0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaT edges: mean=-5.9100 sigma=0.0069
DeltaT time tag: mean=-48.0000 sigma=0.0000
```

Compile and Run SyncTest

- Compile SyncTest
- Launch SyncTest and press "s" to start run
- The **DeltaT Edge** average value is at the order of the sampling clock cycle, 4 ns. By pressing "t" during the acquisition is possible to take a look at the signals in the "time frame" (see step 26). SyncTest corrects for the delay in the Trigger Time Tag.
- The **DeltaT Time Tag** shows a 48 ns (a multiple of 16 ns, which is trigger time stamp cycle, since it's sampled at 62.5 MHz). This 48 ns delay is not an issue, the input trigger signals just set the waveform window of the Master and the Slave at different times.

Step 26



Signal Plot in the Time Reference ("t" plot option)

- This plot shows the signals in the same time frame. The signals are aligned.
- It is possible to see that the DC value of the baseline is not equal in the two boards, so the signals will cross the thresholds at different times. This represents the reason of the difference in the edge time visible in the figure in the previous Step 25.
- The program SyncTest can correct for the DC offset and Step 27 and Step 28 will explain how to adjust it.

Step 27

Set SyncTest operational parameter in userparam.c source

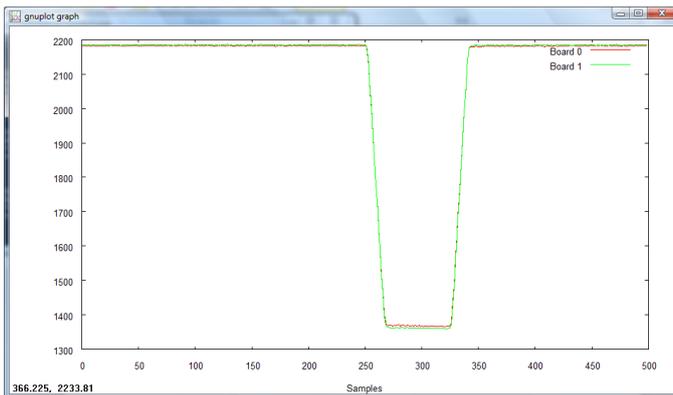
```
// CHANNEL SETTINGS
Params->RefChannel[0] = 6; // Channel of the Master used for the acquisition
Params->TriggerThreshold[0] = 2000; // Trigger threshold (for self triggering)
Params->PostTrigger[0] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[0] = 0x7c30; // input DC offset adjust (DAC value)

Params->RefChannel[1] = 6; // Channel of the Slave used for the acquisition
Params->TriggerThreshold[1] = 2000; // Trigger threshold (for self triggering)
Params->PostTrigger[1] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[1] = 0x8000; // input DC offset adjust (DAC value)

// Trigger edge (CAEN_DGTZ_TriggerOnRisingEdge, CAEN_DGTZ_TriggerOnFallingEdge)
Params->TriggerEdge = CAEN_DGTZ_TriggerOnFallingEdge;
```

Set DC offset

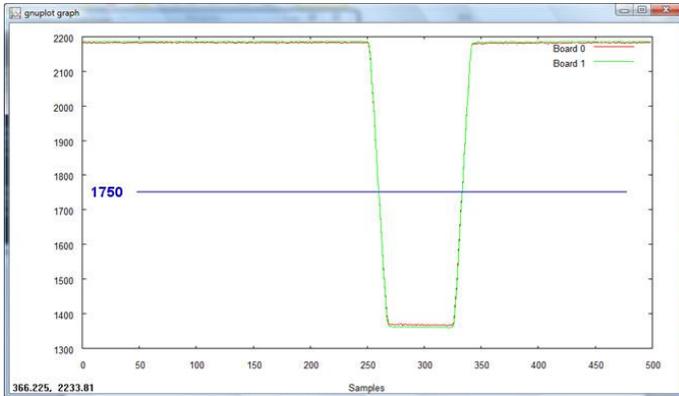
Step 28



Compile and Run SyncTest

- Compile SyncTest.
- Launch SyncTest and press "s" to start run.
- Press "t": the graphic shows that the DC offset is adjusted.

Step 29



Trigger Threshold Tuning

- SyncTest allows to adjust the trigger threshold.
- In this example we set the thresholds for **DeltaT Edge** calculation to 1750 DC.

Step 30

Set SyncTest operational parameter in userparam.c source

```
// CHANNEL SETTINGS
Params->RefChannel[0] = 6; // Channel of the Master used for the acquisition
Params->TriggerThreshold[0] = 1750; // Trigger threshold (for self triggering)
Params->PostTrigger[0] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[0] = 0x8000; // input DC offset adjust (DAC value)

Params->RefChannel[1] = 6; // Channel of the Slave used for the acquisition
Params->TriggerThreshold[1] = 1750; // Trigger threshold (for self triggering)
Params->PostTrigger[1] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[1] = 0x8000; // input DC offset adjust (DAC value)

// Trigger edge (CAEN_DGTZ_TriggerOnRisingEdge, CAEN_DGTZ_TriggerOnFallingEdge)
Params->TriggerEdge = CAEN_DGTZ_TriggerOnFallingEdge;
```

Set thresholds for DeltaT Edge calculation

Step 31

```
C:\Users\Matteo\Desktop\SyncTest\Debug\syncstest.exe
DeltaI time tag: mean= -48.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 MHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 MHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= -0.0014 sigma= 0.0045
DeltaI time tag: mean= -48.0000 sigma= 0.0000

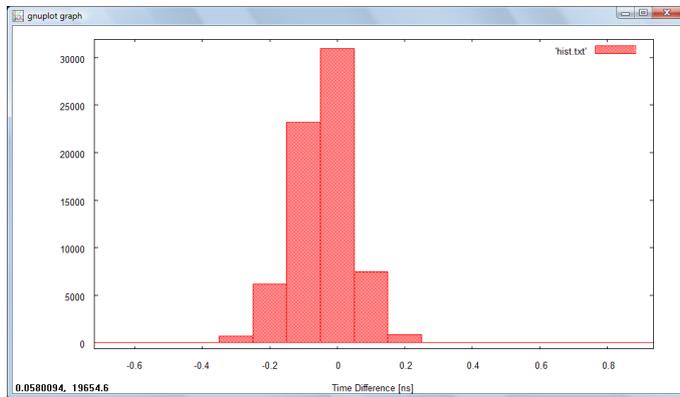
Readout Rate=1.94 MB
Board 0: TrgRate=1.00 MHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 MHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= -0.0012 sigma= 0.0045
DeltaI time tag: mean= -48.0000 sigma= 0.0000

Readout Rate=1.94 MB
Board 0: TrgRate=1.00 MHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1: TrgRate=1.00 MHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= -0.0014 sigma= 0.0044
DeltaI time tag: mean= -48.0000 sigma= 0.0000
```

Compile and Run SyncTest

- Compile SyncTest.
- Launch SyncTest and press “s” to start run.
- The **DeltaT Edge mean** value is sub-nanosecond and, considering the the 4 ns sampling clock of the board (V1720), we could say that the boards are synchronized. The sigma is very small and this means that there are small variations of this value.

Step 32



Time difference distribution (“H” plot option)

- By pressing “H” during the acquisition, the distribution of “DeltaT Edge” is shown.
- In case of good synchronization, the distance of the distribution average from the time reference should be of the order of the distribution standard deviation.

Synchronization Time Stability

The following Figure 8 shows the DeltaT Edge variation during a 1 hour measurement in an environment with no temperature control using the setup described above. The figure shows the average per second of the DeltaT Edge in function of the system run time. The in our setup the digitizers recorded about 7300 events per second.

It is possible to observe a slight drift of the time difference of about 50 ps, which is of the order of 1% of the sampling cycle.

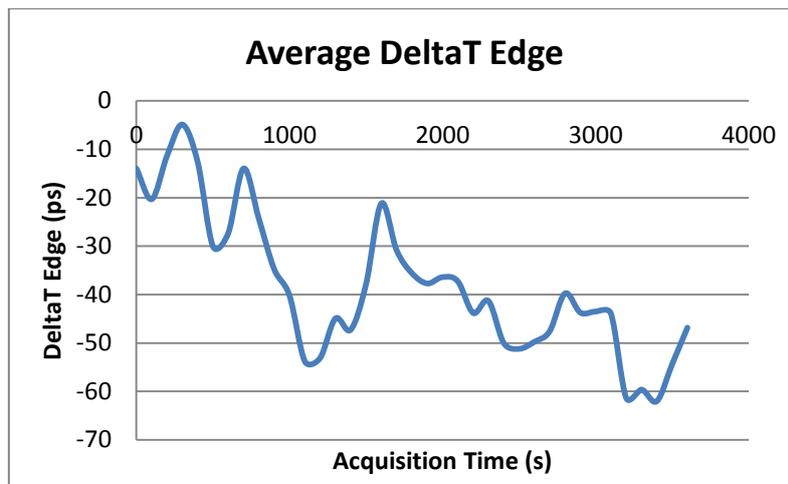


Figure 8: DeltaT Edge variation during 1 hour measurement in an uncontrolled environment.

Example 2: Independent Channel Triggers

This section will describe the synchronization of two boards **with independent channel triggers** (logic OR of the channel auto-triggers and independent external triggers). The aim is to align the start command and then let the boards trigger independently. Triggers won't be propagated, but the run start will be propagated through the daisy chain TRG OUT and the start input (S IN).

The hardware setup is shown in Figure 9. The boards are connected as in the follows.

- Connect Master CLK OUT to Slave CLK IN using A317 cable.
- Connect the TRG OUT of the Master to S-IN of the Slave.
- Connect Function Generator CH1 output to a 50 Ω passive splitter.
- Connect both channels (#6 in this example) of each digitizer to the splitter with identical length cables.

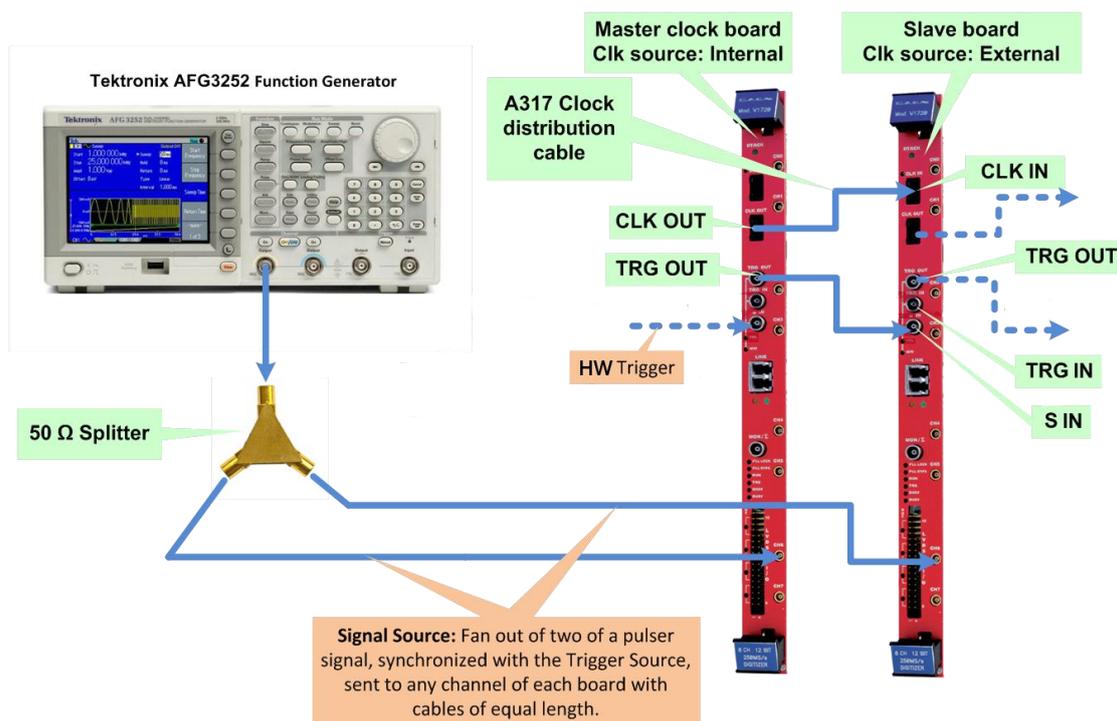


Figure 9: Hardware setup. Connection of boards and function generator.

The Setup is such that:

- the daisy chain TRG OUT/S IN propagates the start of the acquisition;
- the signal source comes from the fan out of two of a function generator, sent to each board with cables of equal length.

The start of the event acquisition happens in the following way:

- the boards are programmed to start when a signal is fed in the S IN and to propagate the S IN input to the TRG OUT;
- the SW trigger is propagated through the daisy chain TRG OUT/S IN and start all the boards;
- each board triggers by the logic OR of the channel auto-triggers and the external TRG IN.
- the acquisition starts sending a SW trigger to the Master board by clicking "s" in the SyncTest program.

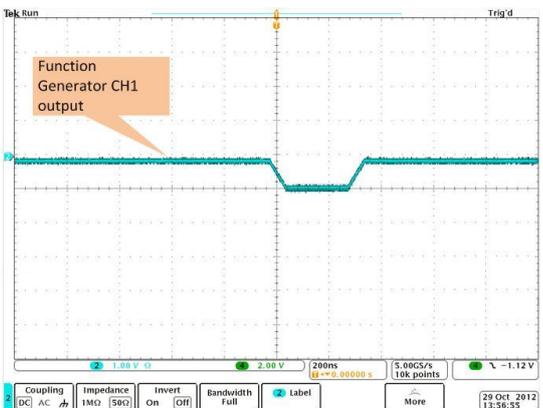
Optional: a HW trigger can be issued with a timer unit (i.e. CAEN V993) connected to the S-IN of the Master board. The Master board is turned in run mode with a logic high state and the data acquisition stops with the logic low state.

In case one of the boards is in a busy state the acquisition can lose the event alignment. To veto this possibility a busy signal can be propagated to last board (on LVDS I/Os), fed into veto input of the Master Board and propagated to all the boards in daisy chain (on LVDS I/Os)⁸

⁸ New firmware currently under test, not described in this document

As in the previous case there is a delay in the acquisition start signal which propagates through the daisy chain from the Master to the Slave. The **SyncTest** program should set the correct **start delay** to synchronize digitizers in case a SW trigger is issued to start the acquisition. This sets the same time stamp in all digitizers. We now describe the step necessary to obtain the synchronization of the boards in this hardware configuration.

Step 33	<p>Digitizer clock Synchronization</p> <ul style="list-style-type: none"> - Perform the step P1-P17 to synchronize and align the clock - Once clocks are synchronized, it is possible to configure the start of acquisition in order to have the same time reference for both boards.
----------------	--

<p>Step 34</p> 	<p>Hardware setup (B): Set signal parameter of function generator</p> <ul style="list-style-type: none"> - Set rise time and fall time= 50 ns. - Set width = 300 ns.
--	---

Step 35	<p>Set SyncTest operational parameter in userparam.c source</p> <pre> // ***** // Start Mode: // ***** // Options: START_SW_CONTROLLED, START_HW_CONTROLLED Params->StartMode = START_SW_CONTROLLED; // ***** // Sync Mode: // ***** // Options: COMMONT_EXTERNAL_TRIGGER TRGIN TRGOUT, INDIVIDUAL_TRIGGER SIN_TRGOUT, TRIGGER_ONE2ALL_EXTOR Params->SyncMode = INDIVIDUAL_TRIGGER_SIN_TRGOUT; // CHANNEL SETTINGS Params->RefChannel[0] = 6; // Channel of the Master used for the acquisition Params->TriggerThreshold[0] = 2000; // Trigger threshold (for self triggering) Params->PostTrigger[0] = 50; // Post trigger in percent of the acquisition window Params->DCoffset[0] = 0x8000; // input DC offset adjust (DAC value) Params->RefChannel[1] = 6; // Channel of the Slave used for the acquisition Params->TriggerThreshold[1] = 2000; // Trigger threshold (for self triggering) Params->PostTrigger[1] = 50; // Post trigger in percent of the acquisition window Params->DCoffset[1] = 0x8000; // input DC offset adjust (DAC value) // Trigger edge (CAEN_DGTZ_TriggerOnRisingEdge, CAEN_DGTZ_TriggerOnFallingEdge) Params->TriggerEdge = CAEN_DGTZ_TriggerOnFallingEdge; // Number of samples in the acquisition windows Params->RecordLength = 500; // Max. distance between the trigger time tags in order to consider a valid coincidence Params->MatchingWindow = 10; // Front Panel LEMO I/O level (NIM or TTL). Options: CAEN_DGTZ_IOLevel_NIM, CAEN_DGTZ_IOLevel_TTL Params->IOlevel = CAEN_DGTZ_IOLevel_TTL; </pre>
----------------	--

Set Start Mode software controlled. In this way the Master Board will wait for a software signal to start the acquisition and will propagate this start signal to the Slave board through the TRG OUT S IN daisy chain

Set Syn Mode according to this Synchronization setup

Set acquisition channel Master and Slave

Set Front Panel I/O level

Step 36

```

C:\Users\Matteo\Desktop\SyncTest\Debug\syncstest.exe
*****
CAEN Digitizer Multiboard synchronization test 1.2
*****
Connected to CAEN Digitizer Model U1720
ROC FPGA Release is 04.00 - Build CG18
AMC FPGA Release is 00.11 - Build C102

Connected to CAEN Digitizer Model U1720
ROC FPGA Release is 04.00 - Build CG18
AMC FPGA Release is 00.11 - Build C102

Boards Configured. Press [s] to start run or [c] to check clock alignment
-
  
```

Compile and Run SyncTest

- Compile SyncTest
- Launch SyncTest and press “s” to start run
- Once the acquisition is started, the SyncTest window should show some parameter as below

Step 37

```

C:\Users\Matteo\Desktop\SyncTest\Debug\syncstest.exe
DeltaI time tag: mean= 32.0271  sigma= 1.0715

Readout Rate=1.94 MB
Board 0:      TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1:      TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 32.0424  sigma= 0.0857
DeltaI time tag: mean= 32.0290  sigma= 1.0769

Readout Rate=1.94 MB
Board 0:      TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1:      TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 32.0421  sigma= 0.0861
DeltaI time tag: mean= 32.0346  sigma= 1.1293

Readout Rate=1.94 MB
Board 0:      TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
Board 1:      TrgRate=1.00 KHz. Matching Events=100.00%; Missing Edges=0.00%
DeltaI edges: mean= 32.0420  sigma= 0.0863
DeltaI time tag: mean= 32.0442  sigma= 1.2046
  
```

Syncstest acquisition parameters

- You can find the description of the acquisition parameter in Step 24.
- **DeltaT Edge = 32.0 ns is about two times the cycle of the trigger time tag (2 x 16 ns for the V1720 module).**
- The firmware allows to add a delay to the start of the acquisition, so we proceed with the adjustment of RUN delay parameter in syncstest.c source (see Step 41).
- The delay is expressed in unit of trigger time tags (16 ns for the V1720 module).

Step 38

Set SyncTest operational parameter in syncTest.c source

```
int SetSyncMode(int handle[2], UserParams_t Params)
{
    int i, ret=0;
    uint32 t reg;

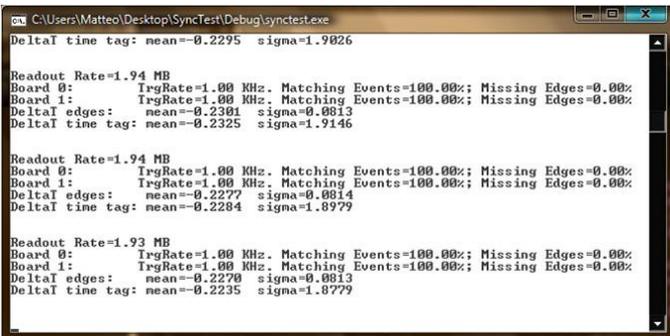
    for(i=0; i<2; i++) {
        switch (Params.SyncMode) {
            case COMMONT_EXTERNAL_TRIGGER_TRGIN_TRGOUT :
                if (i == 0)
                    // inhibit TRGIN on board 0 in order
                    // to avoid start of run with external triggers
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_EXT_TRG_INHIBIT, 1);
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_GLOBAL_TRG_MASK, 0xC0000000);
                // accept EXT TRGIN or SW trg
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_TRG_OUT_MASK, 0xC0000000);
                // propagate both EXT and SW TRG to TRGOUT
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_ACQUISITION_MODE, RUN_START_ON_TRGIN_RISING_EDGE);
                // Run starts with 1st trigger edge
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_RUN_DELAY, 4*(1-i));
                // Run Delay decreases with the position
                // (to compensate for run the propagation delay)
                break;
            case INDIVIDUAL_TRIGGER_SIN_TRGOUT:
                if (i > 0)
                    // Run starts with S-IN on the 2nd board
                    ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_ACQUISITION_MODE, RUN_START_ON_SIN_LEVEL);
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_GLOBAL_TRG_MASK, 1<<Params.RefChannel[i]);
                // accept only trg from selected channel
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_TRG_OUT_MASK, 0);
                // no tigger propagation to TRGOUT
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_RUN_DELAY, 2*(1-i));
                // Run Delay decreases with the position
                // (to compensate for run the propagation delay)
                // Set TRGOUT=RUN to propagate run through S-IN => TRGOUT daisy chain
                ret |= CAEN_DGTZ_ReadRegister(handle[i], ADDR_FRONT_PANEL_IO_SET, &reg);
                reg = reg & 0xFFFF0000 | 0x00010000;
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_FRONT_PANEL_IO_SET, reg);
                break;
            case TRIGGER_ONE2ALL_EXTOR:
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_GLOBAL_TRG_MASK, 0x40000000);
                // accept ext trg in (from trg OR)
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_TRG_OUT_MASK, 0x80000000 | (1<<Params.RefChannel[i]));
                // propagate auto trg and SW trg to TRGOUT
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_ACQUISITION_MODE, RUN_START_ON_TRGIN_RISING_EDGE);
                // Arm acquisition (Run will start with 1st trigger)
                ret |= CAEN_DGTZ_WriteRegister(handle[i], ADDR_RUN_DELAY, 0);
                // Run Delay = 0 for all
                break;
            default:
                return -1;
                break;
        }
    }
    return ret;
}
```

Synchronization setup chosen

Set start delay to synchronize the digitizers, so that they start the acquisition at the same time.

In our example set start delay = 4 x 16 ns for the Master (multiple of the memory clock cycles).

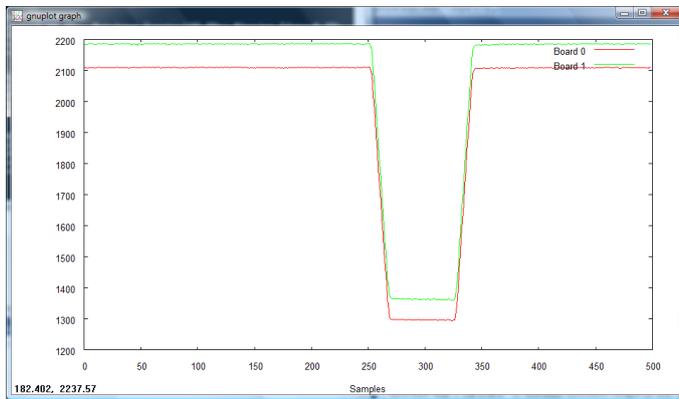
Step 39



Compile and Run SyncTest

- Compile SyncTest.
- Launch SyncTest and press "s" to start run.
- The **DeltaT Edge** mean value is sub-nanosecond, so the boards are well synchronized. By pressing "t" during the acquisition is possible to take a look at the signals.
- The **DeltaT Time Tag** shows a value that is sub-nanosecond and the average is slightly different from 0. This result comes from the fact that this time the trigger are not synchronous, but generated independently by the two board input channels. This can cause a jitter in the trigger time stamp that can be seen sometimes by clicking "p" in the SyncTest shell.
- By pressing "H" during the acquisition the dynamic time distribution of the triggered events is shown.

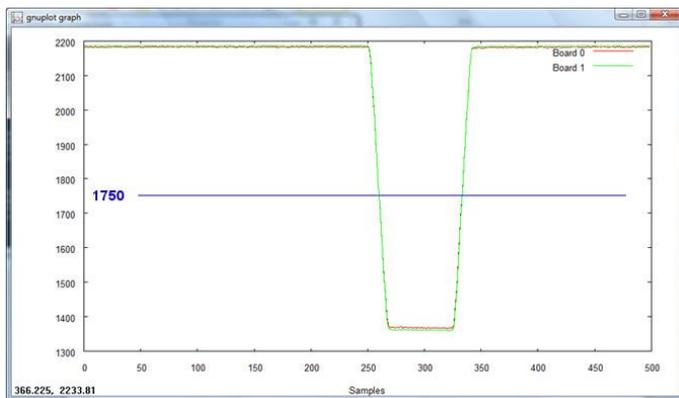
Step 40



DC offset tuning

- The baseline of the digitizer channels may differ for several DC values.
- You can adjust the channel offset as explained in Step 27 and Step 28.

Step 41



Trigger Threshold Tuning

- In the SyncTest program you can adjust the thresholds of the channel auto-trigger.
- You can adjust the trigger threshold as explained in Step 29 and Step 30.

Example 3: Trigger from External Logic Unit

This section will show how to synchronize the acquisition of two or more boards when one of their channels satisfies the trigger condition. This is obtained through the use of the logic OR of all channels auto-triggers given as trigger input to all boards.

The hardware setup is shown in Figure 10. The boards are connected as in the follows.

- Connect Master CLK OUT to Slave CLK IN using A317 cable.
- Connect the TRG OUT of each board to the input of the CAEN V976 unit with cables of equal length.
- Connect the output of the CAEN V976 unit to the TRG IN of each board with cables of equal length.
- Connect Function Generator CH1 output to a 50 Ω passive splitter.
- Connect both channels (#6 in this example) of each digitizer to the splitter with cables of equal length.
- Connect the output of the timer unit V993 to the input of the logic unit CAEN V976 to generate the HW trigger.

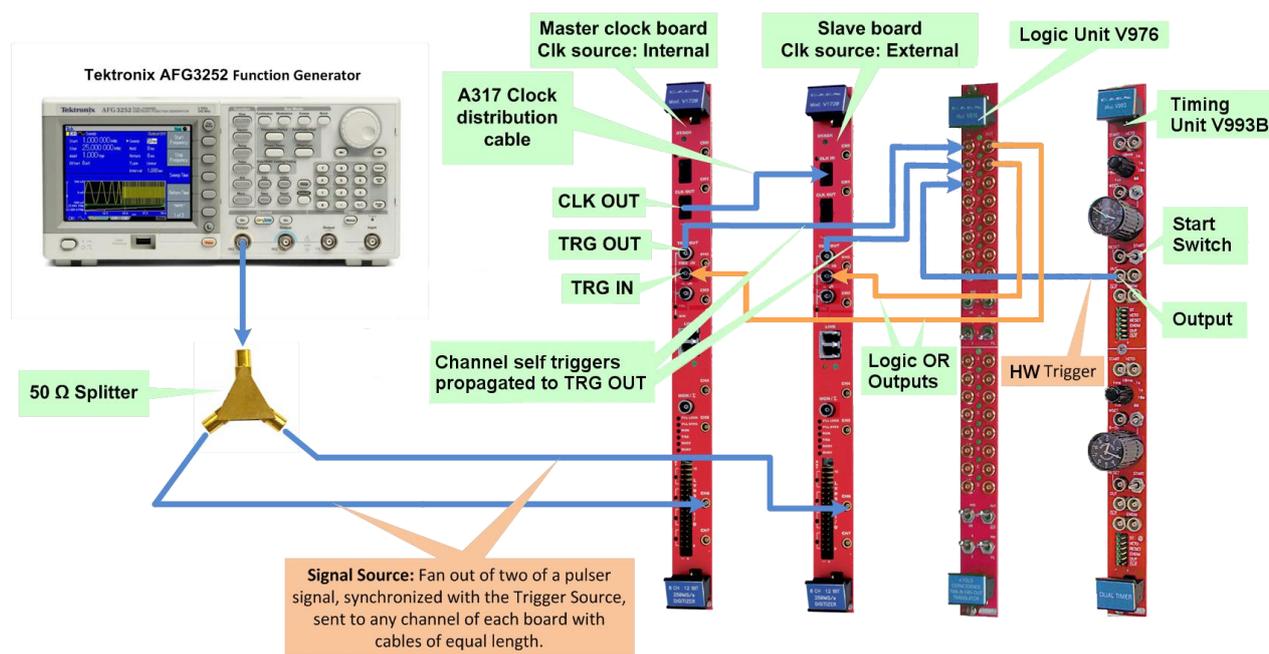


Figure 9: Hardware setup. Connection of boards and function generator.

The Setup is such that:

- the TRG OUT of each board is connected with cables of equal length to the input of a logic unit (i.e. CAEN V976);
- the fan out of two of the logic unit output is connected to the TRG IN of each board with cables of equal length;
- signal source comes from the fan out of two of a function generator and is sent to each board with cables of equal length.

The start of the event acquisition happens in the following way:

- all boards are armed to start the acquisition when a signal is fed into TRG IN;
- all boards are programmed to propagate the channel auto-trigger to the TRG OUT without triggering the event acquisition;
- all boards are programmed to trigger when an external trigger signal is received through the TRG IN;
- a HW trigger can be issued with a timer unit (CAEN V993 in our case) connected to the external logic unit.

In this hardware setup the trigger which starts the data acquisition is propagated to all the boards with the approximately the same time delay. With cables of equal length the board acquisition start is normally synchronous. However the start the run mode could start asynchronously, in particular when the trigger signal is close to trigger clock edges.

Optional: a SW trigger can be issued through the SyncTest program as in the previous examples. In this configuration the Master board propagates the SW trigger to the TRG OUT and gives a logic signal high as input in the logic OR. All boards turn in run mode at the same time when the output of the logic unit is received in the TRG IN.

In case one of the boards is in a busy state the acquisition can lose the event alignment. To veto this possibility a busy signal can be propagated to last board (on LVDS I/Os) and fed into veto input of the Master Board and propagated to all the boards in daisy chain (on LVDS I/Os)⁹

In this example, differently from the previous ones, it is not necessary to set the **RUN delay** for the synchronization between digitizers. All digitizers start acquisition at the same time.

We now describe the step necessary to obtain the synchronization of the boards in this hardware configuration.

Step 42	<p>Digitizer clock Synchronization</p> <ul style="list-style-type: none"> - Perform the step P1-P17 to synchronize and align the clock - Once clocks are synchronized and locked, it's time to Synchronize the start/stop of acquisition in order to have the same T0 on both the boards.
----------------	--

<p>Step 43</p>	<p>Hardware setup (B): Set signal parameter of function generator</p> <ul style="list-style-type: none"> - Set rise time and fall time= 50 ns. - Set width = 300 ns.
-----------------------	---

⁹ New firmware currently under test, not described in this document

Step 44

Set Syncstest operational parameter in userparam.c source

```
// *****
// Start Mode:
// *****
// Options: START_SW_CONTROLLED, START_HW_CONTROLLED
Params->StartMode = START_HW_CONTROLLED;

// *****
// Sync Mode:
// *****
// Options: COMMONT_EXTERNAL_TRIGGER_TRGIN_TRGOUT, INDIVIDUAL_TRIGGER_SIN_TRGOUT, TRIGGER_ONE2ALL_EXTOR
Params->SyncMode = TRIGGER_ONE2ALL_EXTOR;

// CHANNEL SETTINGS
Params->RefChannel[0] = 6; // Channel of the Master used for the acquisition
Params->TriggerThreshold[0] = 2000; // Trigger threshold (for self triggering)
Params->PostTrigger[0] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[0] = 0x8000; // input DC offset adjust (DAC value)

Params->RefChannel[1] = 6; // Channel of the Slave used for the acquisition
Params->TriggerThreshold[1] = 2000; // Trigger threshold (for self triggering)
Params->PostTrigger[1] = 50; // Post trigger in percent of the acquisition window
Params->DCoffset[1] = 0x8000; // input DC offset adjust (DAC value)

// Trigger edge (CAEN_DGTZ_TriggerOnRisingEdge, CAEN_DGTZ_TriggerOnFallingEdge)
Params->TriggerEdge = CAEN_DGTZ_TriggerOnFallingEdge;

// Number of samples in the acquisition windows
Params->RecordLength = 500;

// Max. distance between the trigger time tags in order to consider a valid coincidence
Params->MatchingWindow = 10;

// Front Panel LEMO I/O level (NIM or TTL). Options: CAEN_DGTZ_IOLevel NIM, CAEN_DGTZ_IOLevel TTL
Params->IOlevel = CAEN_DGTZ_IOLevel_TTL;
```

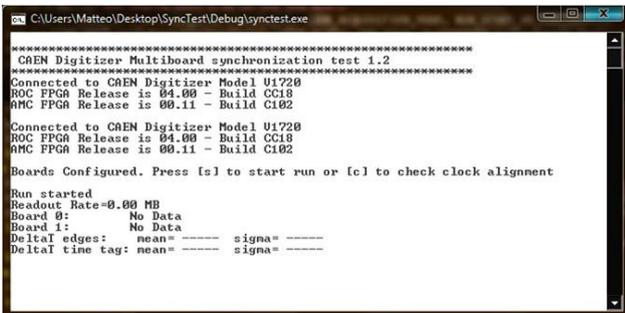
Set Start Mode hardware controlled. In this way the all the boards will wait for a hardware signal to start the acquisition.

Set Syn Mode according to this Synchronization setup

Set acquisition channel Master and Slave

Set Front Panel I/O level

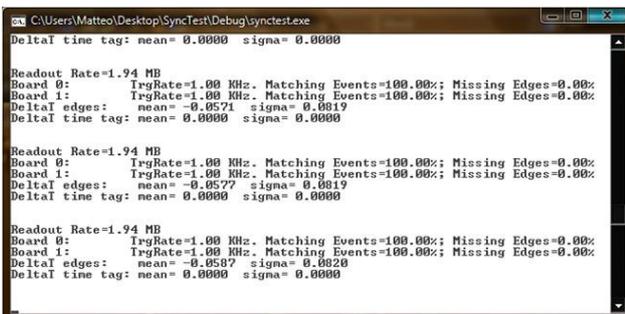
Step 45



Compile and Run SyncTest

- Compile SyncTest.
- Launch SyncTest and press “s” to start the acquisition.
- An HW trigger is expected to turn the boards in run mode and the SyncTest window should communicate that no data are acquired.
- Once a HW trigger is issued (i.e. with the logic unit V993), the SyncTest window should show some parameter as below in the next step.

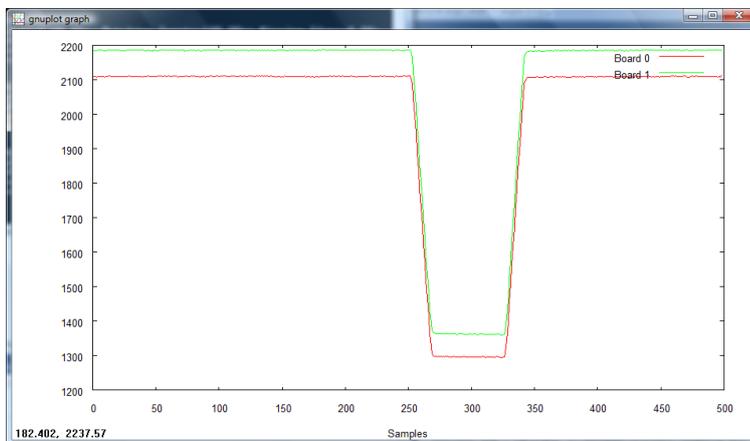
Step 46



SyncTest acquisition parameters

- You can find the description of the acquisition parameters in step 24.
- **DeltaT Edge = -0.0587 ns.** This means that the Master and Slave acquisition machine are synchronized. This is explained by the fact that all digitizers start the acquisition at the same time.
- In this example there is no need to adjust the run delay between boards (as explained in Step 24).

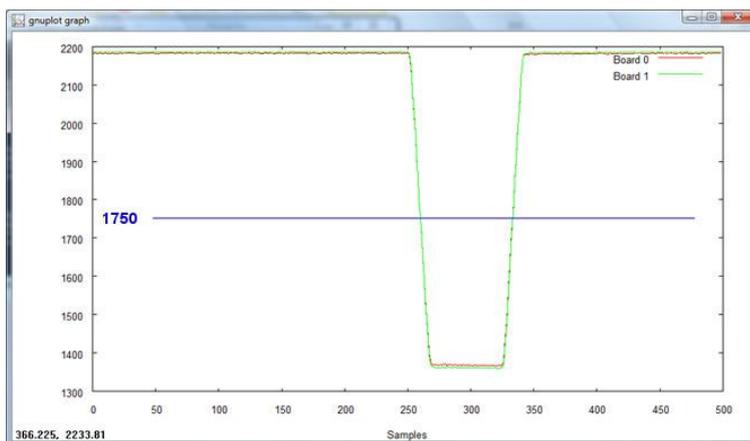
Step 47



DC offset Tuning

- The baseline of the digitizer channels may differ for several DC values.
- You can adjust the channel offset as explained in Step 27 and Step 28.

Step 48



Trigger Threshold Tuning

- In the SyncTest program you can adjust the thresholds of the channel auto-trigger.
- You can adjust the trigger threshold as explained in Step 29 and Step 30.