SAMPA V3 Evaluation Board and Tests Procedures

1. Overview

SAMPA is a custom front-end ASIC for gaseous detectors readout which transforms the charge signal induced on the input into a digital Semi-Gaussian pulse signal. The SAMPA ASIC integrates 32 channels per chip of the full data processing chain and support continuous and triggered read-out.

The front-end is composed by a cascade connection of CSA (Charge Sensitive Amplifier), a differential Semi-Gaussian Pulse Shaper and an Analog-to-Digital Converter (ADC). The CSA generates a voltage step from a short charge input signal (typically in nanoseconds order) and the pulse shaper conform a Semi-Gaussian pulse with amplitude proportional to the total charge injected on the input from the CSA output. SAMPA was designed in 130nm CMOS technology since the nominal supply voltage of SAMPA is 1.25V.

The sensitivity, polarity and peaking time of the SAMPA front-end can be adjusted by the user via external pins. SAMPA supports positive and negative polarity of the input charge and the 20mv/fC@160ns or 4mV/fC@300ns cases of sensitivity and peaking time.

The SAMPA front-end requests three reference voltages of 450 mV,600 mV, and 750 mV internally generated which are temperature compensated. These voltages are used to define the DC output baseline voltage Baseline on the pulse shaper output. These reference voltages can be programmed via a configuration register controlled by the DSP block. The bias currents used to polarized the analog front-end circuits are generated inside the chip and they can not adjusted by the user.

The front-end output is digitized by a 10-bit 10 Msamples/s ADC. A stable voltage reference of 1.1V is necessary for the ADC operation which must be provided externally. After the ADC a digital signal processor (DSP) eliminates signal perturbations, distortion of the pulse shape, offset and signal variation due to environmental conditions variations. The data read-out takes place continuously at a speed of up to 1.28 Gbps by SLVS (Scalable Low-Voltage Signaling) links.

This document describes the setup of the SAMPA evaluation board and presents an step-by-step guide for the chip configuration according to the requested pulse specifications (Section 3) . The section 4 will presents the SAMPA Communicator and SAMPA analyzer, used to configure and control the NCCA/PCCA board. Moreover, this document will present the tests procedures in section 5.

2. Evaluation Board – NCCA/PCCA

- **NCCA**

The figure 1 shows the block diagram of the SAMPA evaluation board.

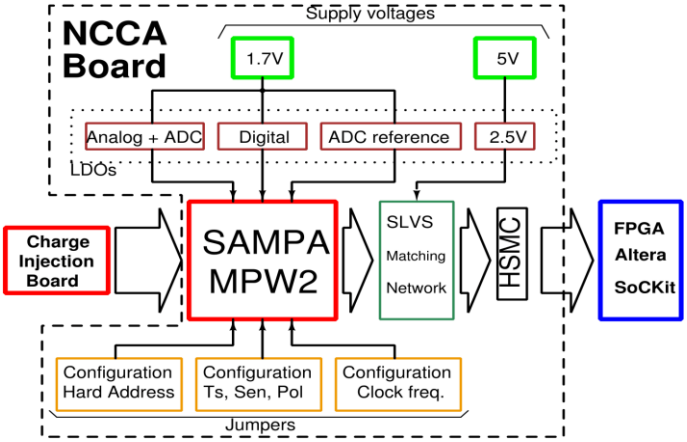


Figure 1: block diagram of the SAMPA evaluation board

The NCCA board requests 1.7V and 5V supply voltages. From 1.7V are generated the power supplies for the front-ends block, the ADCs and the digital part (DSP and SLVS drivers) using on-board regulators. The 5V is to generated the 2.5V for the SLVS matching network.

- **PCCA**

The PCCA board request 1.7V and 2.5V supply voltages. The same way of NCCA board, 1.7V are generated the power supplies for the front-ends block, the ADCs and the digital part (DSP and SLVS drivers) using on-board regulators. However, there is no the regulator for 2.5V, as seen of figure 2.

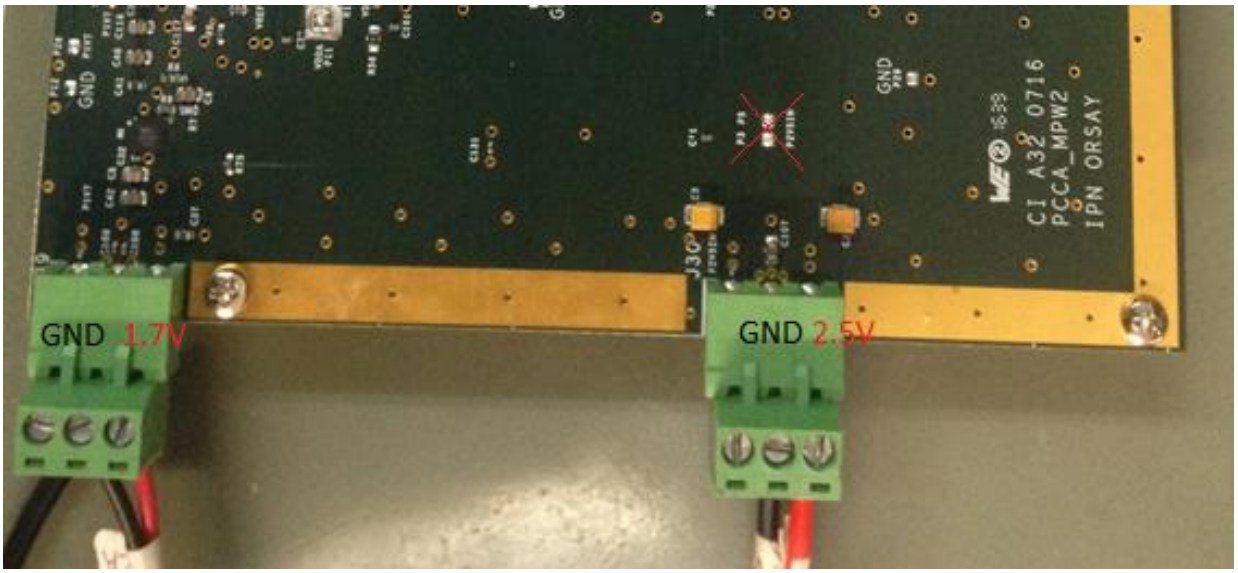


Figure 2: PCCA evaluation board Power Supply

The 32 front-end inputs are excited by a charge injection board which is coupled to the SAMPA evaluation board via a 3 column-16 rows pin header connector. The injection board is shown on figure 3.

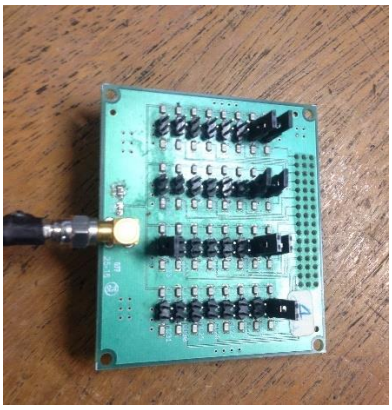


Figure 3: Injection board

The characteristics of the Semi-Gaussian output pulse as Sensitivity, Peaking Time and Polarity, the SAMPA hard address and the clock configuration should be selected by the user via jumpers, as shown on figure 4.

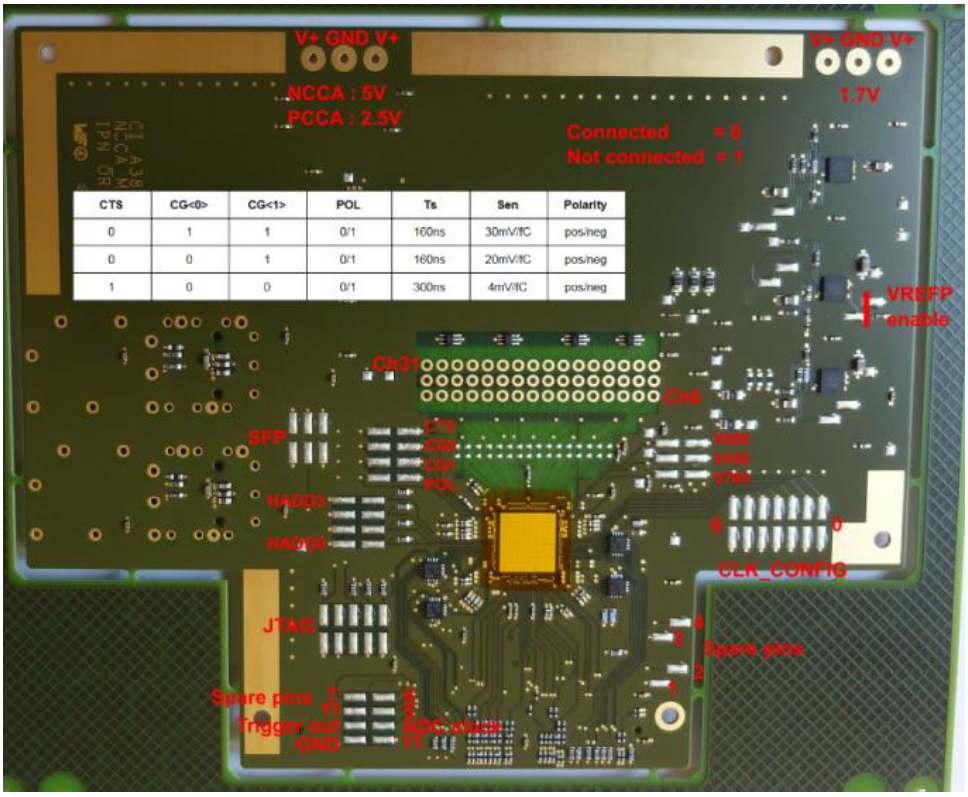


Figure 4: Jumpers configuration

2.1 NCCA/PCCA Power Supply

The carrier power supplies and the corresponding power domain are summarized in Table 1.

Table 1: NCCA/PCCA Power Supplies

Board Reference	SUPPLY VOLTAGE	NOMINAL VALUE	POWER DOMAINS
1	VDDA	1.25V	VDD_FE, VDD_FE2, VDD_AD
2	VDD	1.25V	VDD_DG, VDD_DR
3	VREF	1.1V	ADC VOLTAGE REFERENCE
4	P2V5	2.5V	SLVS MATCHING NETWORK
	GND	COMMON GROUND	

- Is very importante to connect a ground between the board and FPGA, as shown on figure 5:

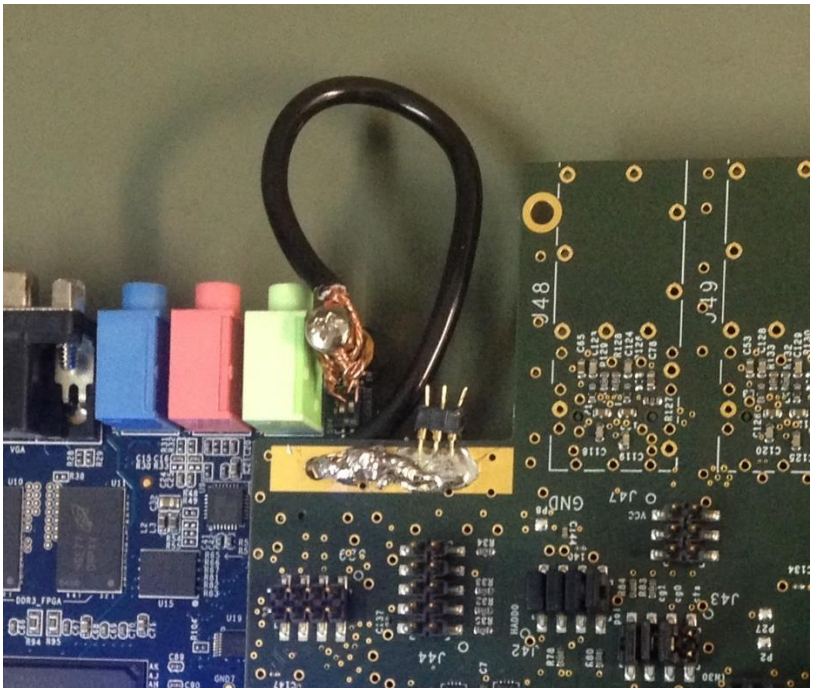


Figure 5: PCCA ground connection

- **NCCA**

For NCCA board, confirm the voltages by the points marked on figure 6. The SAMPA evaluation board has a common ground plane which connects all power domains

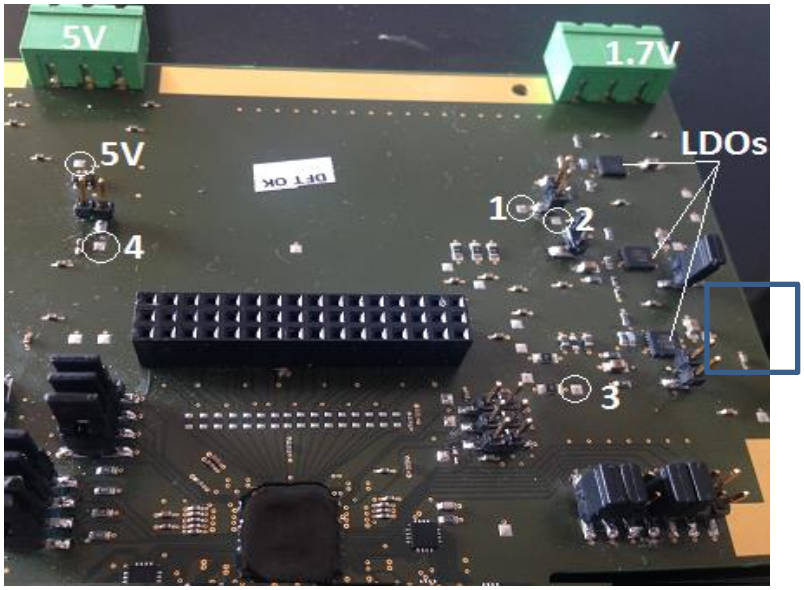


Figure 6: NCCA Power Supply and measurements points

- **PCCA**

For PCCA board, confirm the voltages by the points marked on figure 7.

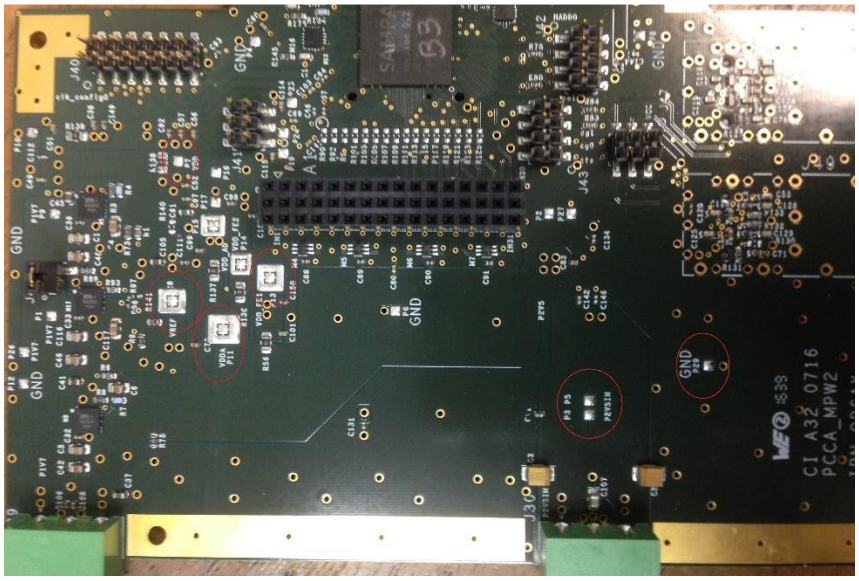


Figure 7: PCCA Power Supply and measurements points

For both boards, The VDDA and VDD come from the LDO's by a resistor welded. The VREF also comes from a LDO, but doesn't has a resistor, so must to have a jumper to connect the VREF to property voltage.

3. Evaluation board setup

As mention before, the SAMPA has several configurations through jumper. The setup of the SAMPA evaluation board incorporates some stages which are described below. See section 3.4 for an example.

3.1 Hard address configuration

When mounted in a FEC, several SAMPAs will operate in the same board within a specific system with in order to increase the number of read-out channel per board. Each SAMPA is identified with a hard address defined by four external pins (pins: HADD[0:3]). The selected address will be incorporated to the serial data header which identifies the chip. The hard address configuration is performed via jumpers. The hard address should be selected by the user via the J42 jumper. Table 2 presents the jumper signal distribution. This should be set to the same number as in the chip id box on the “sampa register” tab in the communicator. It is default 0, so all jumpers connected.

Table 2: J42 signals (high = jumper open, low = jumper close).

J42 position	Signal
1-2	HADD[0]
3-4	HADD[1]
5-6	HADD[2]
7-8	HADD[3]

3.2 Clock configuration

As described earlier, SAMPA operates with 3 input clock signals: the ADC sampling clock (clkinADC), the data serialization clock (clkinSO) and bunch crossing clock (clkinBX). These signals can be externally provided for the FPGA board via the HSMC connector or internally generated by SAMPA.

The possible options of clock configuration is defined by external pins (clk_config[0:6]) according to the Table 4. The configurations clock options listed in Table 3 are supported with two constraints:

- ckoutADCSAR clock must always be 80x the ADC clock or more for correct operation of the ADC. 20 MHz ADC with 80 MHz serial link clock is thus not supported.
- Operating the ckoutBX on other than 40 MHz would alter counting rate of the bunch crossing counter and lower the maximum operating speed of the I2C.

The typical clock configuration for all internal signal are clk_config[0:6]=0010011 where the ADC sampling clock frequency (ckoutADC) is 10MHz and the ADC conversion frequency (ckoutADCSAR) is 80MHz. The user can adjust these clock frequencies in term of the system requirements.

Table 3. Supported clock configurations (high = jumper open, low = jumper close)

Inputs			Outputs					clk_config
clkSO	clkBX	clkADC	ckoutSO	ckoutSOd2	ckoutADCSAR	ckoutBX	ckoutADC	[6:0]
All internal								
320	na	na	320	160	160	40 (i)	20 (i so)	0000011
320	na	na	320	160	80	40 (i)	10 (i so)	0010011
320	na	na	320	160	40	40 (i)	5 (i so)	0000001
160	na	na	160	80	160	40 (i)	20 (i so)	0000010
160	na	na	160	80	80	40 (i)	10 (i so)	0010010
160	na	na	160	80	40	40 (i)	5 (i so)	0000001
80	na	na	80	40	80	40 (i)	10 (i so)	0010000
80	na	na	80	40	40	40 (i)	5 (i so)	0000000
BX external								
320	40	na	320	160	160	40 (e)	20 (i so)	0000111
320	40	na	320	160	80	40 (e)	10 (i so)	0010111
320	40	na	320	160	40	40 (e)	5 (i so)	0010101
160	40	na	160	80	160	40 (e)	20 (i so)	0000110
160	40	na	160	80	80	40 (e)	10 (i so)	0010110
160	40	na	160	80	40	40 (e)	5 (i so)	0000101
80	40	na	80	40	80	40 (e)	10 (i so)	0010100
80	40	na	80	40	40	40 (e)	5 (i so)	0000100
320	40	na	320	160	160	40 (e)	20 (i bx)	0001111
320	40	na	320	160	80	40 (e)	10 (i bx)	0011111
320	40	na	320	160	40	40 (e)	5 (i bx)	0011101
160	40	na	160	80	160	40 (e)	20 (i bx)	0001110
160	40	na	160	80	80	40 (e)	10 (i bx)	0011110
160	40	na	160	80	40	40 (e)	5 (i bx)	0001101
80	40	na	80	40	80	40 (e)	10 (i bx)	0011100
80	40	na	80	40	40	40 (e)	5 (i bx)	0001100
ADC external								
320	na	20	320	160	160	40 (i)	20 (e)	0100011
320	na	10	320	160	80	40 (i)	10 (e)	0110011
320	na	10	320	160	40	40 (i)	5 (e)	0110001
160	na	20	160	80	160	40 (i)	20 (e)	0100010
160	na	10	160	80	80	40 (i)	10 (e)	0110010
160	na	10	160	80	40	40 (i)	5 (e)	0100001
80	na	10	80	40	80	40 (i)	10 (e)	0110000
80	na	10	80	40	40	40 (i)	5 (e)	0100000
ADC and BX external								
320	40	20	320	160	160	40 (e)	20 (e)	0100111
320	40	10	320	160	80	40 (e)	10 (e)	0110111
320	40	10	320	160	40	40 (e)	5 (e)	0110101
160	40	20	160	80	160	40 (e)	20 (e)	0100110
160	40	10	160	80	80	40 (e)	10 (e)	0110110
160	40	10	160	80	40	40 (e)	5 (e)	0100101
80	40	10	80	40	80	40 (e)	10 (e)	0110100
80	40	10	80	40	40	40 (e)	5 (e)	0100100

Used in the example below

3.3 Pulsing configuration

The chip can be digitally configured to operate on 2 peaking time (Ts) options controlled by CTS pin, 3 sensitivity options (Sen) controlled by CG<0:1> pins and on negative/positive charge input polarity controlled by POL pin. Table 5 summarizes the possible chip configurations where typically high=1.25V and low=0V. The logic voltages high/low on CTS,CG<0:1> and POL can be selected by J43 (high = jumper open, low = jumper close) as indicated in Table 6

Table 5. Pulse configurations (high = jumper open, low = jumper close).

CTS	CG<0>	CG<1>	POL	Ts	Sen	Polarity
0	1	1	1	160ns	30mV/fC	neg
0	0	1	1	160ns	20mV/fC	neg
1	0	0	0	300ns	4mV/fC	pos

Table 6: J43 signals (high = jumper open, low = jumper close).

J43 positions	Signal
1-2	CTS
3-4	CG<0>
5-6	CG<1>
7-8	POL

3.4 Example NCCA configuration

This example is a NCCA board configured for a clock of 320 MHz (So) and 160 MHz (ADC) and gain of 4mV/fC.

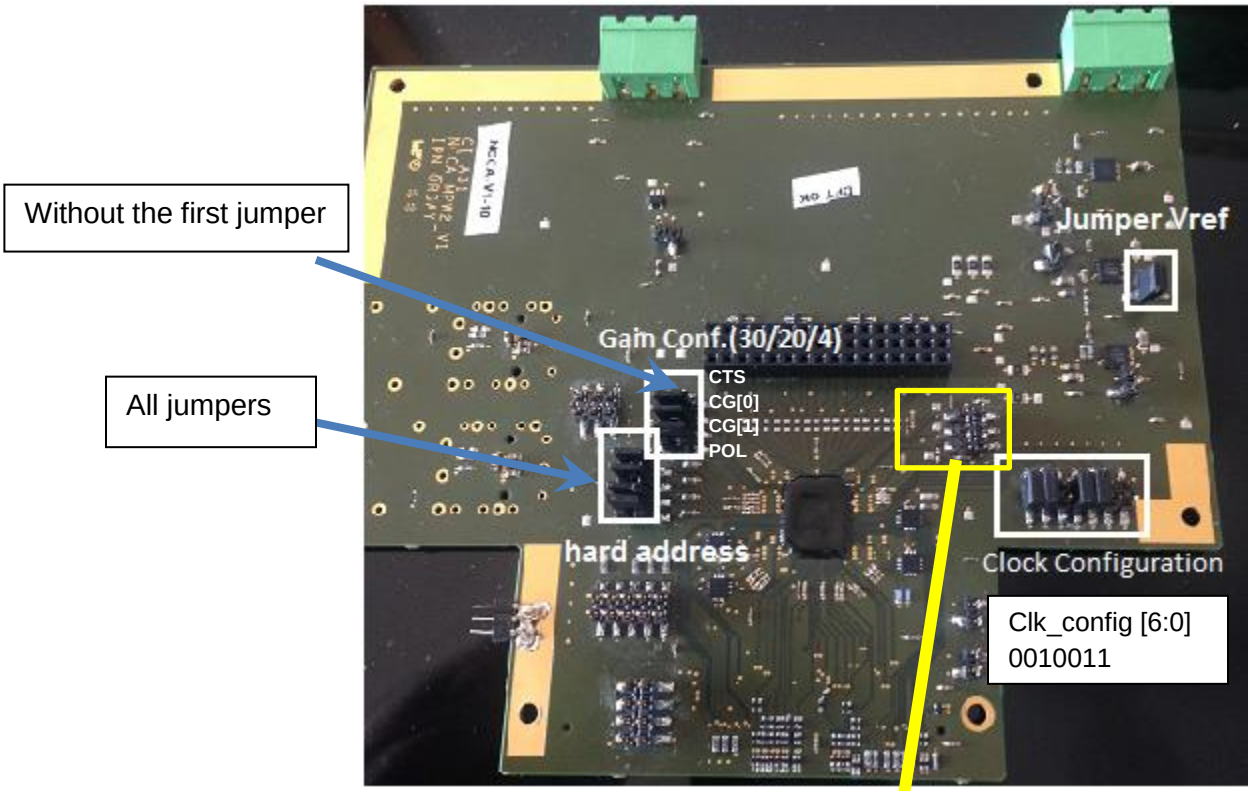


Figure 8: NCCA configuration example

3.5 Voltages reference trimming (Band Gap)

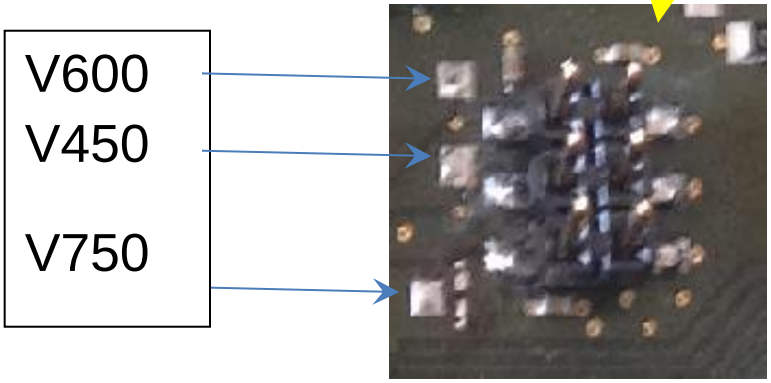


Figure 9: Band Gap voltages output / forced input

The front-end circuit request three voltage references of 600 mV, 450 mV and 750 mV which are generated by an on-chip band gap circuit named of V600, V450 and V750 respectively. These reference voltages can be trimmed by SW<0:2> bits in register ADCTRIM in SAMPA communicator.

3.6 Charge injection

The charge injection could be independently performed for each channel i.e one CJ capacitor is necessary for each front-end channel, or is possible to apply to all 32 channels by a different injection board. The charge injection is performed using a floating capacitor (CJ) where one terminal is connected to a voltage step (ΔVoltage) from a pulse generator and the other terminal is connected to the front-end input as shown in Figure 5.

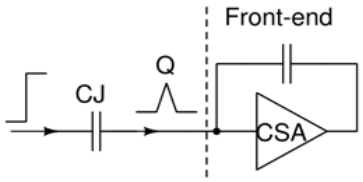


Figure 6. Charge injection scheme.

The injected charge can be calculated by the following equation:

$$\text{Charge } (Q) = \Delta\text{Voltage} \cdot CJ$$

The test pulsing system has a number of components.

1. The test pulse generator
2. Test pulse synchronization and phase with sampling frequency
3. Pulser adapter board
4. Pulser boards for single channels
5. Pulser board for multiple channels (32 channels though jumpers)

The signal generator (step voltage) is connected to the capacitor via a 50 ohm impedance coaxial cable (lemo). The ground braid of the coax cable connects to the analog ground of the NCCA/PCCA test board in a parallel path compared to the signal.

The pulse signal must be configured as the figure 10 for negative polarity configuration - TPC (Table 7) and must be inverted for positive polarity - MCH.

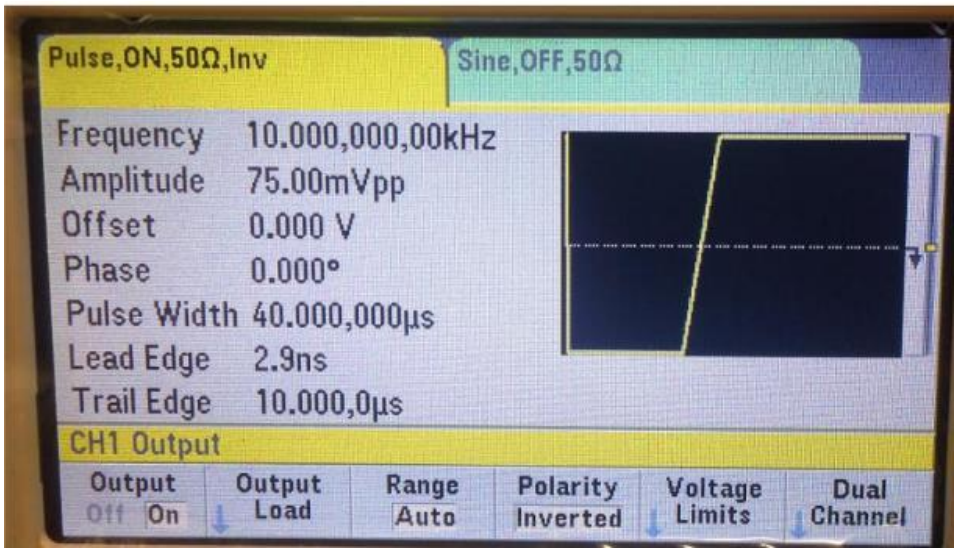


Figure 10 – Input signal for TPC configuration

4. SAMPA Communicator – rev 750 (Used for SAMPA V2)

The SAMPA communicator was developed to configure and control the NCCA/PCCA board + FPGA. To use the SAMPA Communicator, open the executable. A screen as the below should appear.

First of all, connect the computer (USB) to the COM port of FPGA, as figure 20.

At first moment, the connection status should be **Closed**.

Select the COM port:

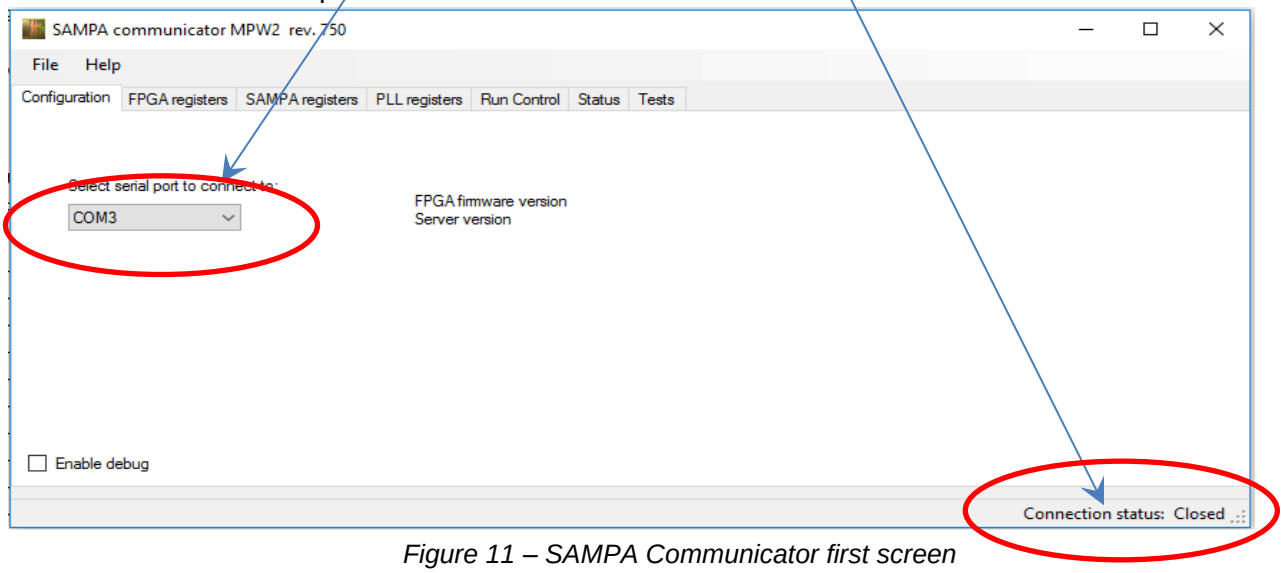


Figure 11 – SAMPA Communicator first screen

After connected, the status should be open and the FPGA Firmware and Server Version should appear.

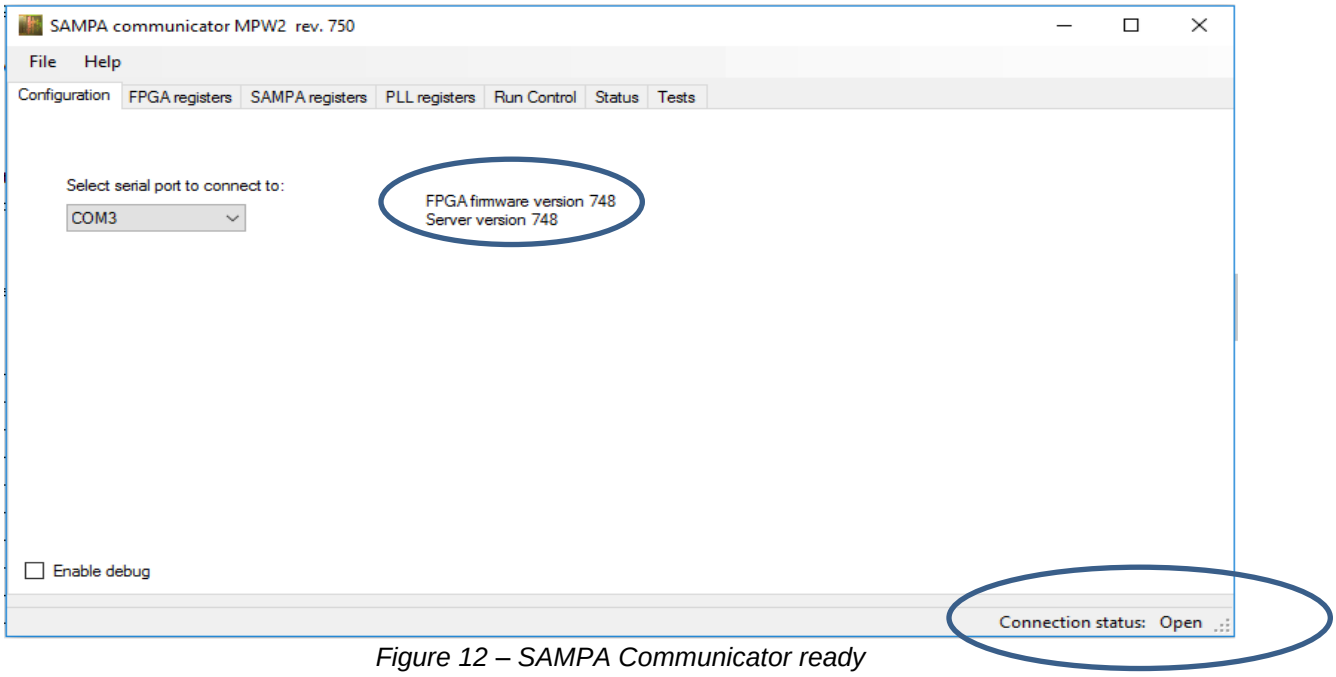


Figure 12 – SAMPA Communicator ready

For tests of the SAMPA V3/V4, some improves were done in the sampa communicator, and the actual version of this doc is the 979. The version of the firmware and server was updated also, as shown on image 13.

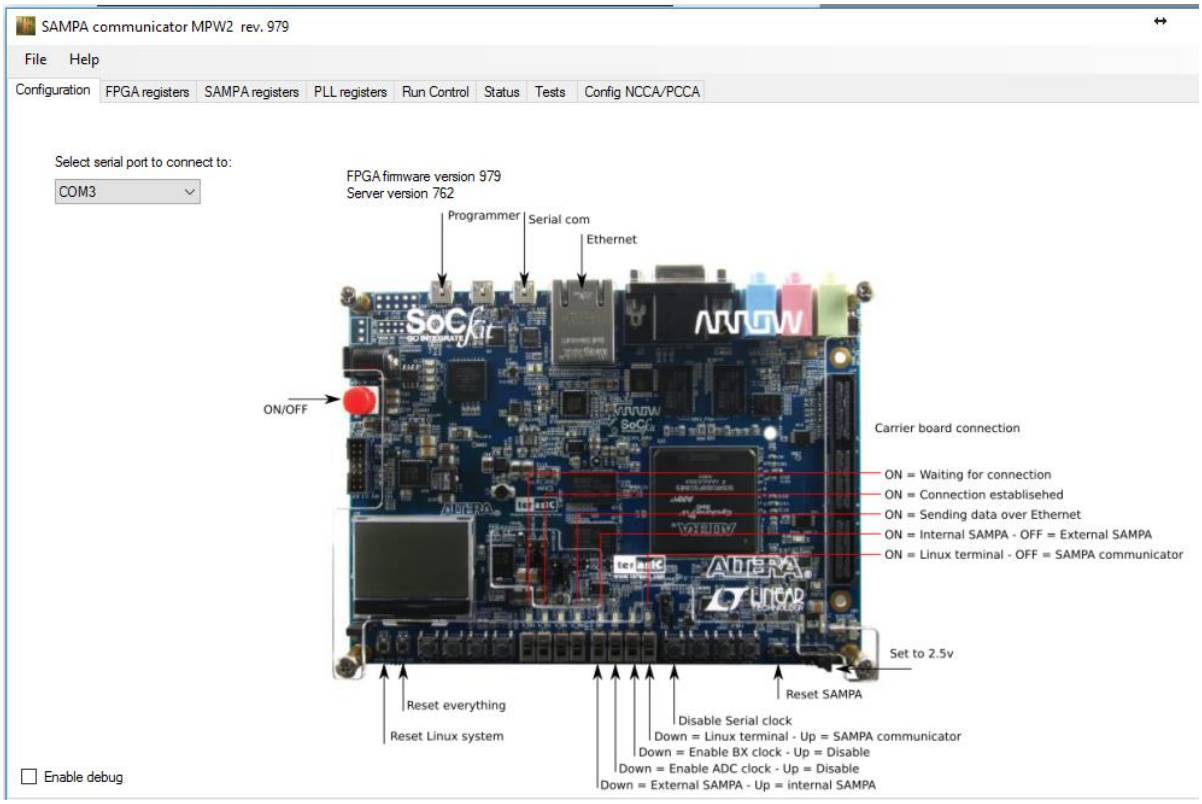


Figure 13 – SAMPA Communicator rev 979

After connected, move to SAMPA register **tab** and read all global values. Do it to make sure that everything is ok, i.e., the ADCTRIM value should be 4.

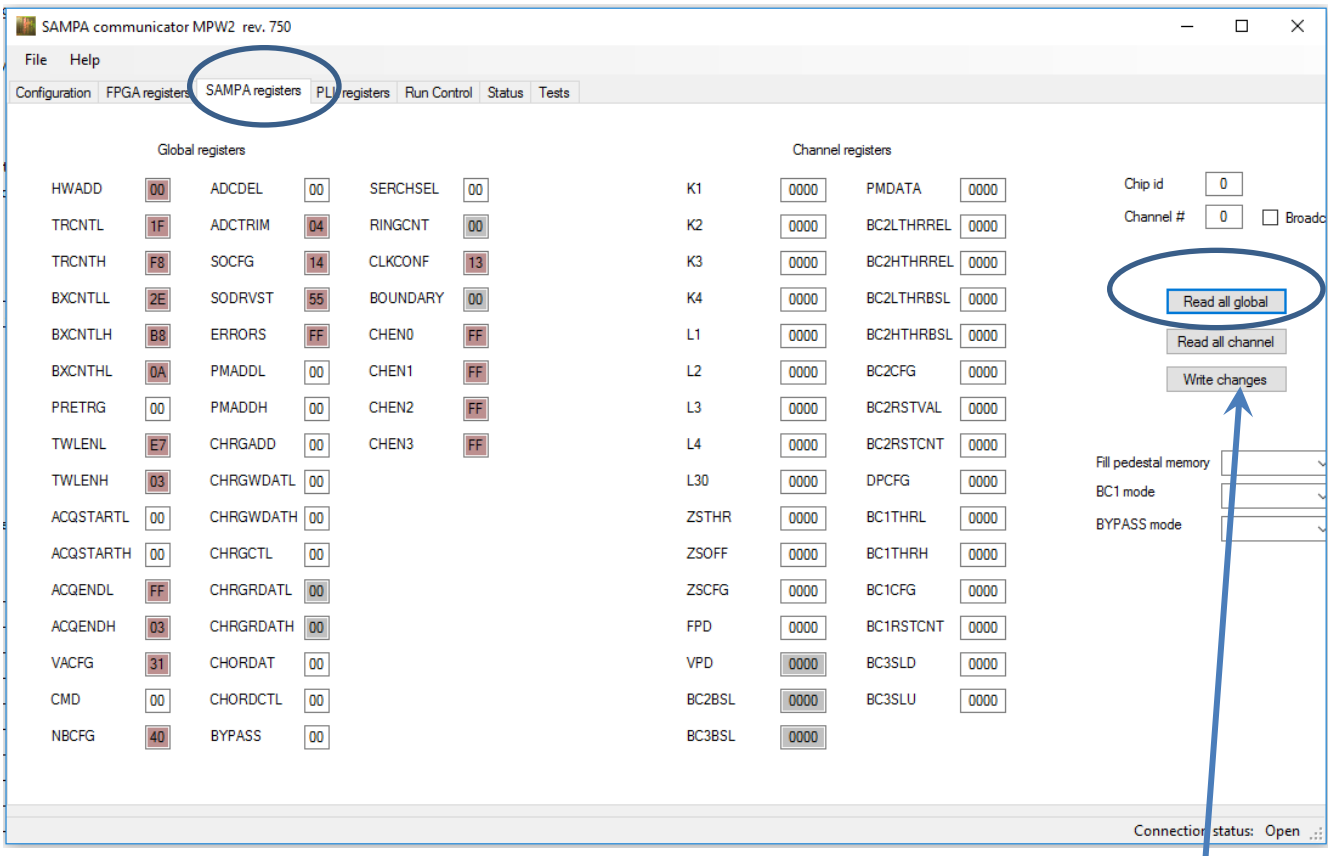


Figure 14 – SAMPA registers tab

If is necessary change some value, this tab should be used. For example, if is necessary to adjust the V750 value, the ADCTRIM should change. Put the specified value and click **write changes**.

The PPL register tab show what is the PCCA/NCCA + FPGA clock operation. Make sure that the board (CLK CONFIG JUMPERS) and PLL register follow the clock configuration.

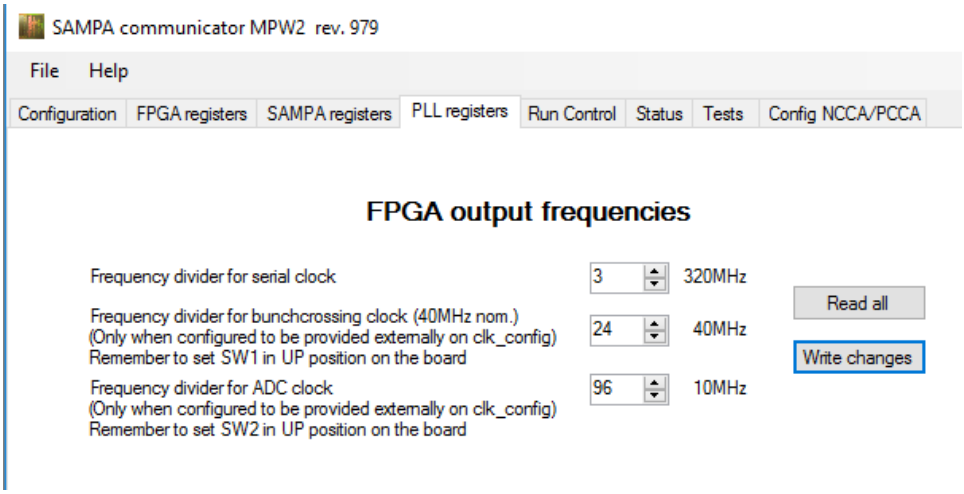


Figure 15 - PLL register

After properly define all registers value and the clock has been set as required, the system (PCCA/NCCA +FPGA) is ready for be used.

The SAMPA communicator is used to control the application of stimulus to the PCCA/NCCA board. There are 11 serial links used to send the 32 channels of data acquisition.

If wants to use the 11 links, should be **enable the 11 serial outputs** and set the **number to 11 links**.

Is necessary to confirm the status of those links (See figure 15 and 16)

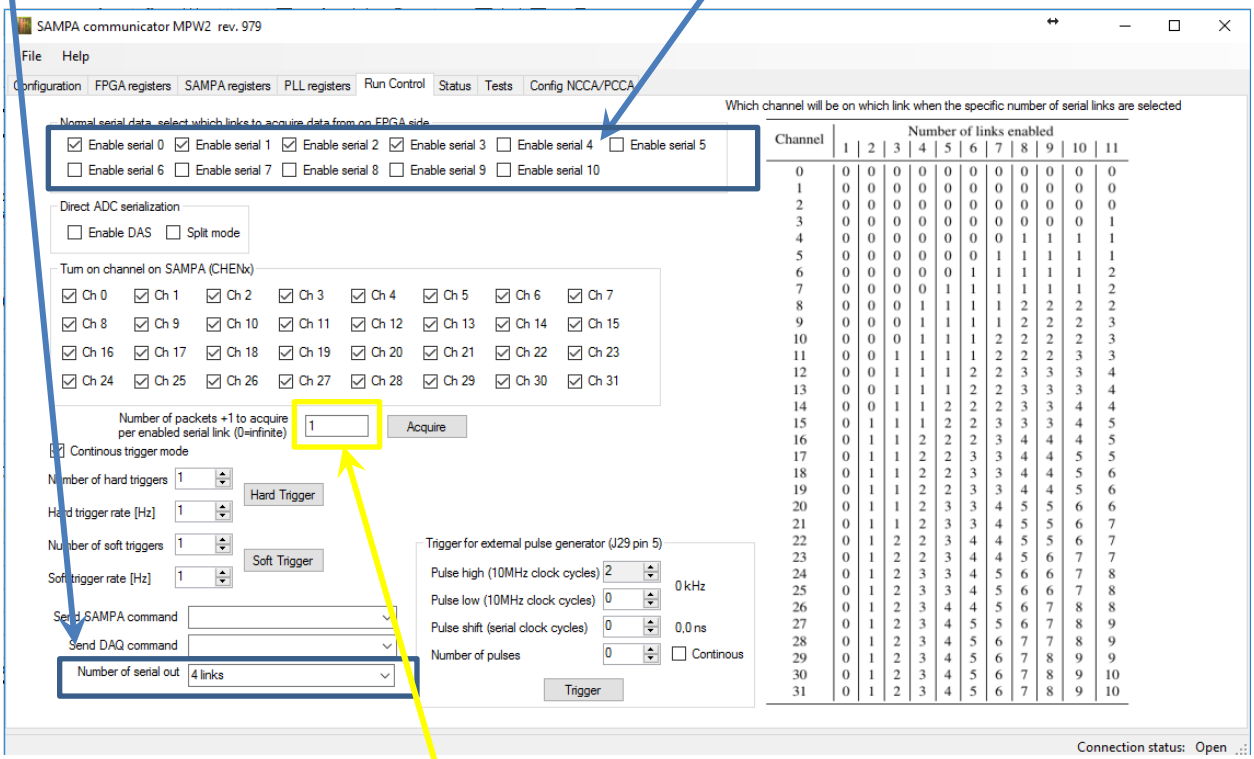


Figure 14: Sampa Communicator – Run Control tab

Is important to set **the number of packets** that will be sent for each “click” in acquire. The value ‘0’ means continuous way or infinite number of packets.

After the procedure showed in the figures 15 and 16, the system is able to run.

The next step is open the SAMPA Analyzer (see section 4.1).

After set the number of serial links, is important to check the connection between the board and FPGA. Click in **Update** to see the current status of the serial links connection (if '1' the connection is ok, if '0' means the link is wrong).

For example, the figure 15 shows that there **are two links broken**.

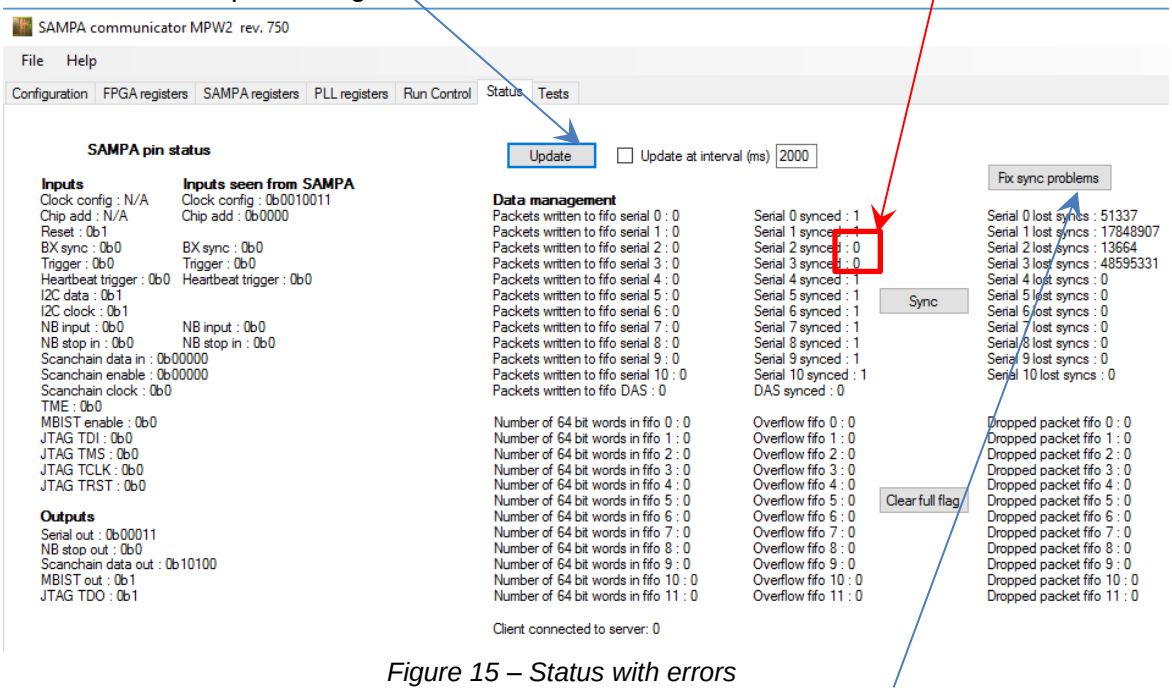


Figure 15 – Status with errors

One way to fix the error above is to click in **Fix sync problems**.

After that, click in **Update** again. Click in Update until all the **lost syncs** be stable (no changes). See that there is no more link broke.

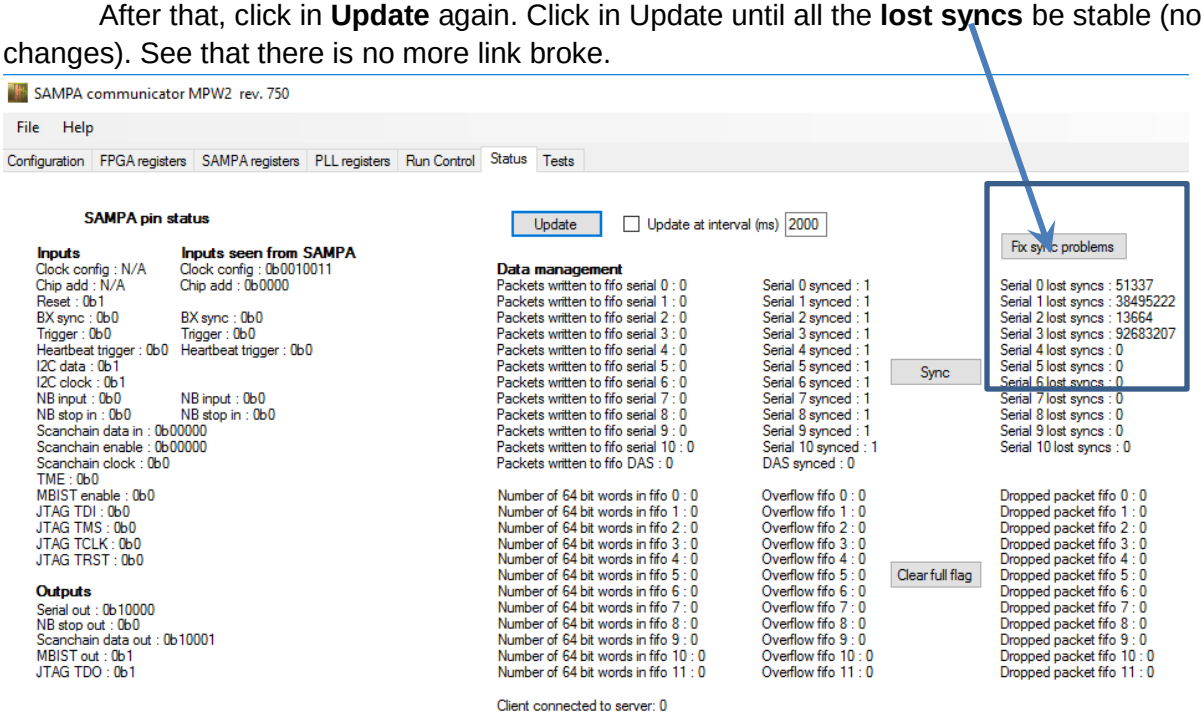


Figure 16 – Status without errors

If these procedures not fix all the links, sometimes the system must be reset and/or the PCCA/NCCA board must be remove from FPGA and put again.

4. 1 SAMPA Analyzer

Run the SAMPA_Analyzer.exe and wait for connection.

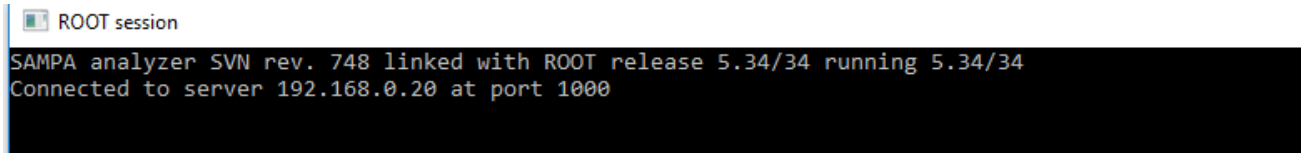


Figure17 SAMPA Analyzer connection

Some new windows will open, like the figure 18.

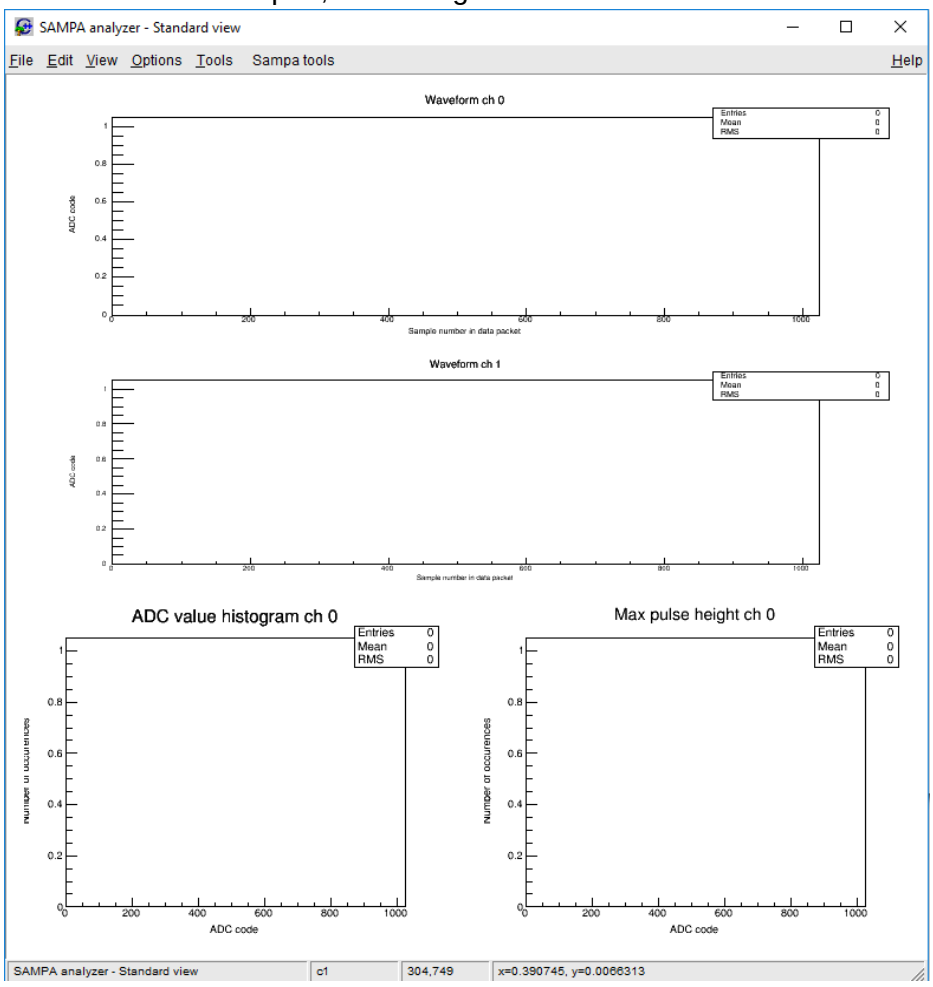


Figure18 SAMPA Analyzer standard view.

With this window, is possible to restart the SAMPA_analyzes, see the current values of baseline, noise, max pulse for each channel. Go to **SAMPA tools** tab for change the channel showed and analyzed.

With the SAMPA_analyzer is possible to save different kinds of file as pdf, .C and root files with the values measured of the pulse or noise.

4.2 Root analysis

To manipulate root files, is necessary to use a C program.

As example, see the commands below to get the values of Noise and Baseline of each channel and save to the file ns_vs_chNo.dat:

Open a terminal where the root file is located and entry : root

From the Root prompt:

```
>.L C:\SAMPA\svn\testing\software\Analysis\Noise-anal_sampa_dataCh_fit.cpp
> anl_tree("Name.root");
> hist_anal()
```

Open the file ns_vs_chNo.dat to get the data.

5 . Testing procedure

Related to: SAMPAV3andV4_TestPlanning
https://docs.google.com/spreadsheets/d/1TmDSKdZuLgjw52WDg1a6Dd_Cxccv6xVEpapmAZ9jvj0/edit?ts=99b1277#gid=649544711

FIRST PART

The SAMPA ASIC experimental test using the proposed evaluation board should involves the following steps sequence for initial tests:

A. Supply Powering

NCCA: The board requests two supply voltages de 1.7V and 5V.
PCCA: The board requests two supply voltages de 1.7V and 2.5V.
The *user should adjust the current limit of the voltage source to 600mA to avoid components blowing due to possible short circuits*.

B. DO a Reset

It will reset automatically on powerup, but it is also possible to reset the sampa with KEY4 on the FPGA (see figure 20). Also is possible to reset with the SAMPA communicator.

C. Verification Board supply voltage

On the evaluation board some test-points to verify the supply voltages generated by the LDO as illustrated on figure 6 and 7. The supply voltages and their nominal values to be measured are indicated in Table 1.

From this point, is possible to take the results for 1.2,1.3 and 1.4 of the SAMPAV3andV4_TestPlanning. For 1.4, measure the points shown on figure 8 if NCCA. If is a PCCA, measure the points shown on figure 19:

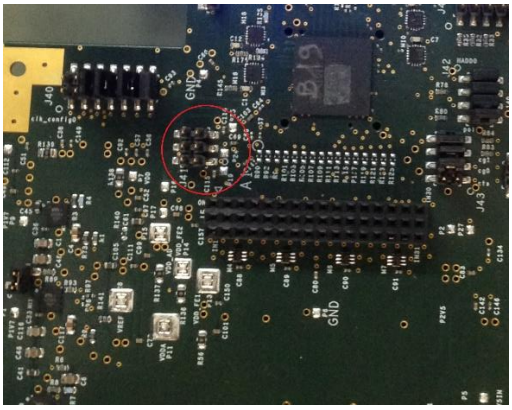


Figure 19 –PCCA V750, V600 and V450 points

The voltage references (V600, V750 and V450) should be stables close to their typical cases (600mV, 750mV and 450mV SW<2:0>=100). This should be verified for DC measure directly on the test-points.

D. FPGA connection

- **NCCA:**

Connect the small ground strap between the FPGA and the carrier board and then connect the carrier to the FPGA through the HSMC connector.

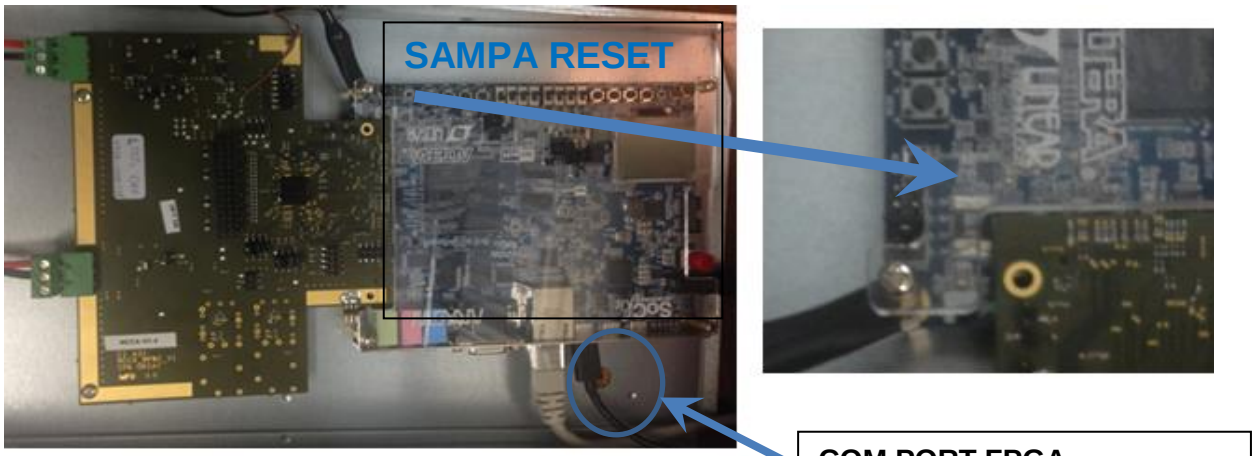


Figure 20 – NCCA + FPGA – Basic connection

- **PCCA:**

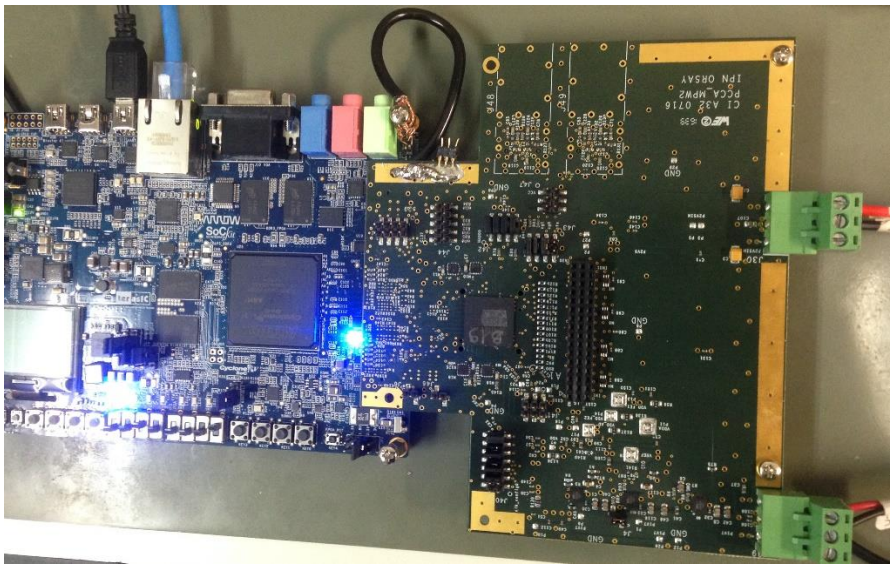


Figure 21 – PCCA + FPGA – Basic connection setup for Test

Clock and Hard address configuration.

The clock and hard address configuration is described in section 3.1 and 3.2.
Set the address to 0 (all jumpers connected).

To perform the initial tests 1.5 and 1.6 is good to set a “default” configuration. Is suitable the use the configuration describe in the example (section 3.4).

E. Verify Synchronization

For item 1.5, using sampa Communicator, enable the 11 links, as below:

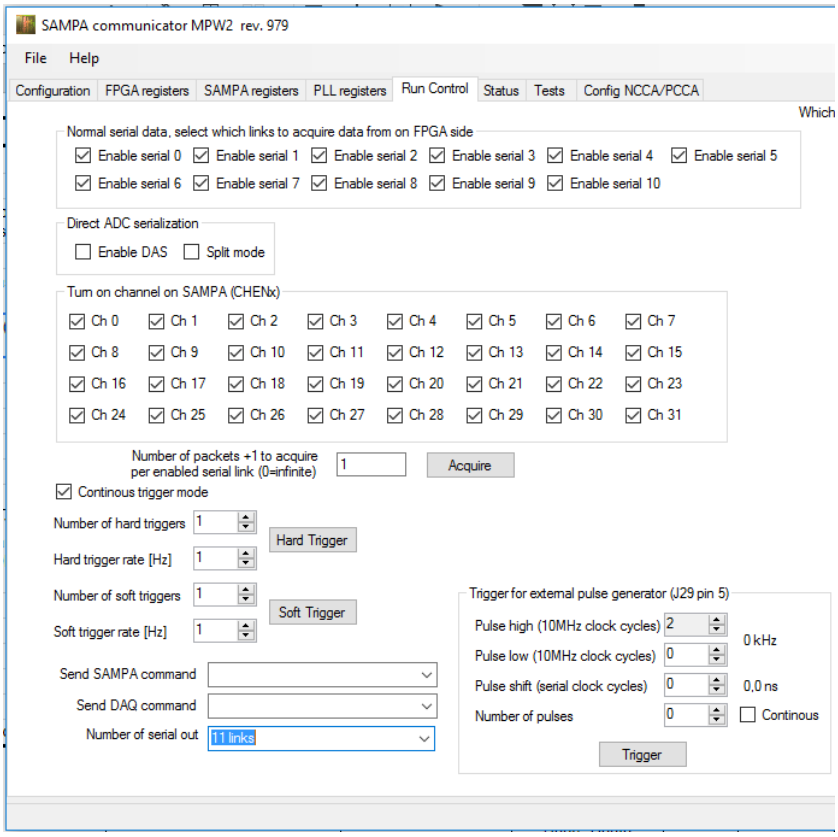


Figure 22 – Verify Synchonization

Go to tab “Status” and Update. Verify if the links are synchronized. If not, click “Fix serial sync problems”, Sync, and Update. Press update until the lost values be stable.

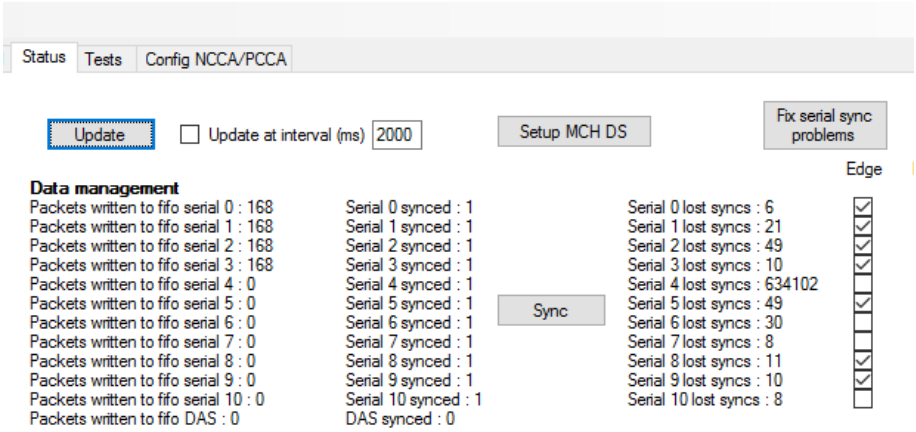


Figure 23 – Verify Synchonization – GOOD Result.

F. Integrity of packets

Using SAMPA Analyzer, after step E, go to Run control tab, put a low value of Number of packets (1-10), and press Acquire. You should not see messages as Malformed Packet on the root screen.

If it happened, back to step 5. After you press acquire and there is no message on the root screen, the test is considered success. As example, see figure 24.

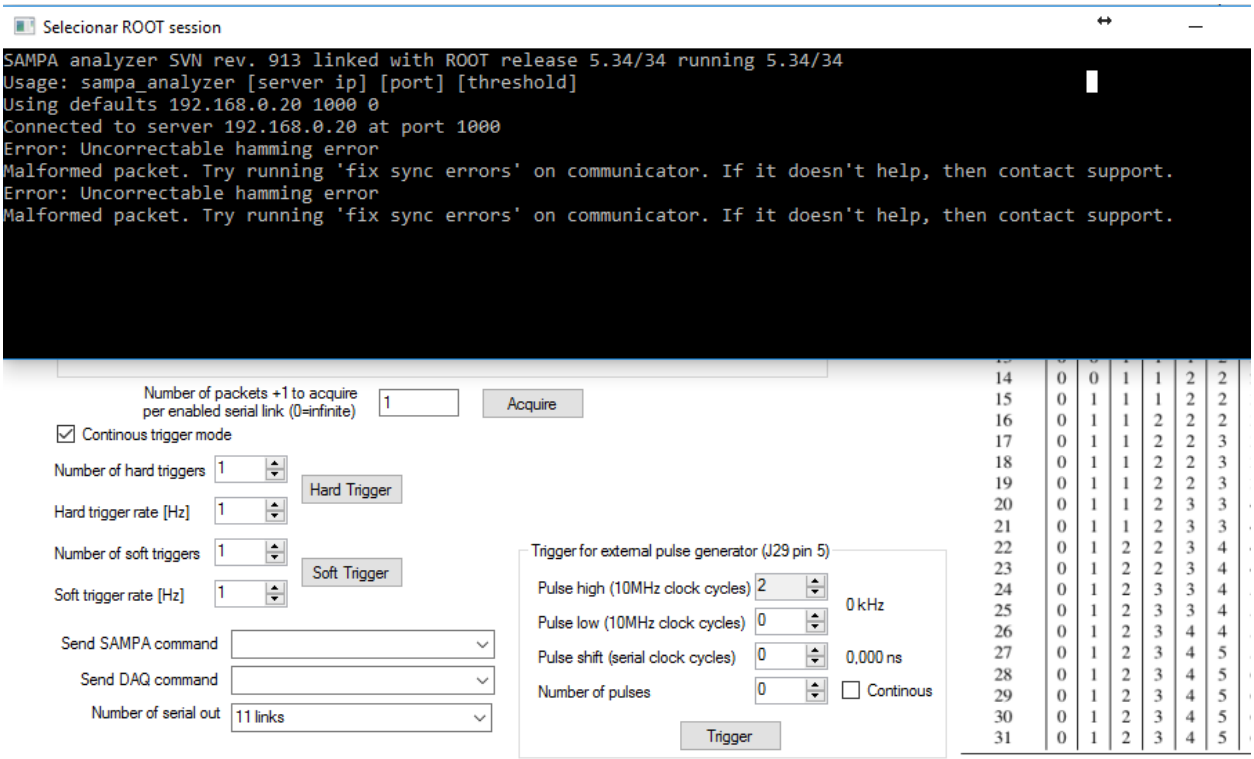


Figure 24 – Integrity of packets

G. Digital Tests

To perform some digital tests (1.7 to 1.10 of SAMPAV3andV4_TestPlanning), go to tab **Tests.** Is possible to performs the Memory test, JTAG, Ring Oscillator and Baseline Test (BC2 and BC3).

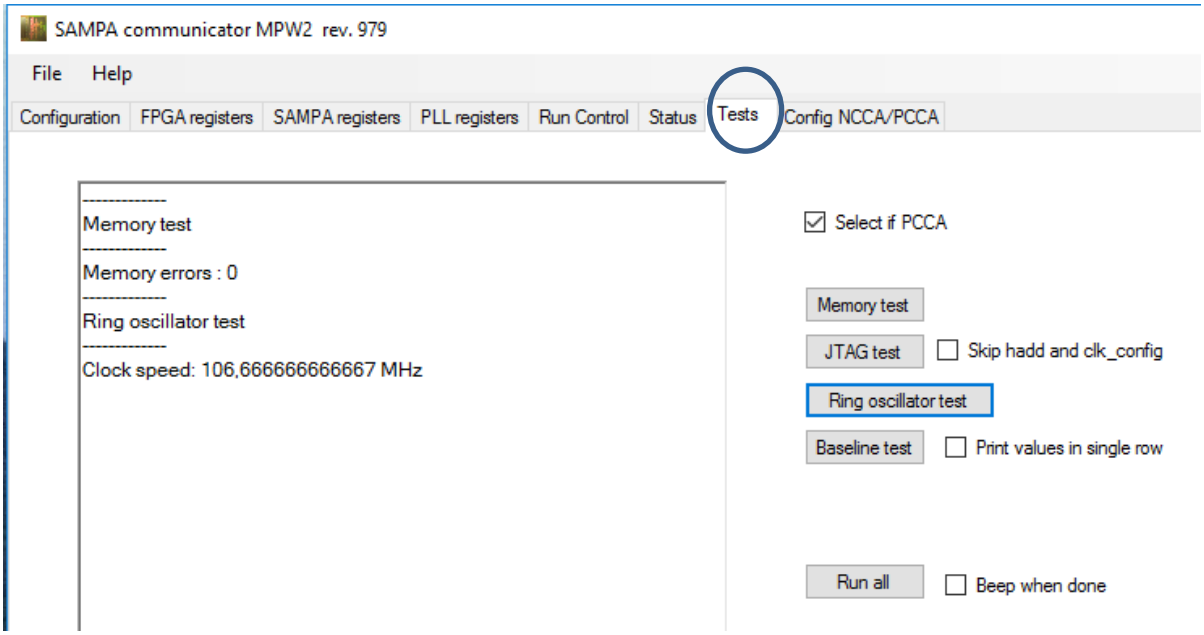


Figure 25 - Digital Tests

If all the test until now were executed and the results are good, the board is considerate a good sample and is suitable for tests with injection board.

Estimated time including the setup is around 1 hour for first sample until here.

SECOND PART – SAMPA V3/V4

To perform the items 1.12 to 1.18 of SAMPAV3andV4_TestPlanning, is important to have been done the tests of “first part”.

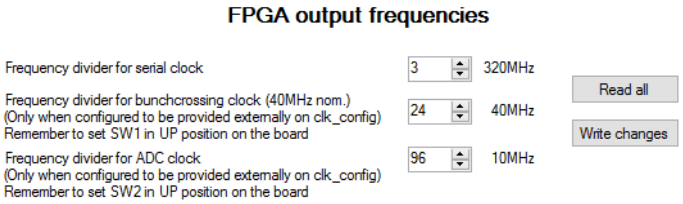
With the board connected and properly configured, follow the following steps sequence for evaluate the SAMPA, specially the front-end.

A. Chip readout behavior

This test is intended to confirm the proper connection between the FPGA and Board with different clock configurations, which are:

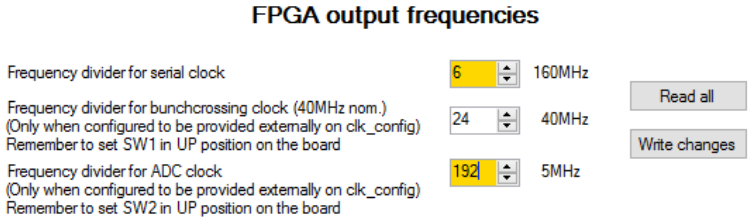
clkSOx**clkADC:**

- 320X10 MHz (Default) - *clk_config [6:0]: 0010011(JUMPERS)*



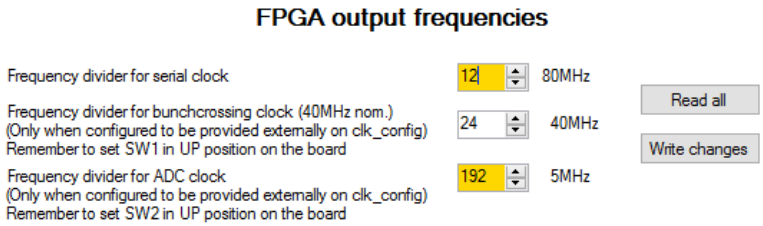
- 160X5 MHz (TPC)) - *clk_config [6:0]: 0000001 (JUMPERS)*

  To change the FPGA PLL, put the values shown below and click on Write Changes:



- 80X10 MHz (MCH) - *clk_config [6:0]: 0000000 (JUMPERS)*

  To change the FPGA PLL, put the values shown below and click on Write Changes:



✓ For this last case (80X10), enable only 1 serial link and make sure that it is synchronized.

“It has problem keeping sync at 80MHz because there is never sent any sync packets since you are then running with 1/4 the needed output bandwidth. Running at 80MHz you need either compression or turn off 3/4 of the channels.”

Is important to make sure that there is no loss of packets (check the Integrity of packets) and also do the measurements of the total current from VDD, VDDA and VREF with data acquisition. That means, see the consumption while you press to acquire of run control tab of SAMPA communicator.

✓ Is not expected changes on the values of power consumption seen on power supply.

B. Baseline Measurements and Noise

Using the default configuration of the SAMPA (320X10 MHz) and gain 30 mV, enable the 11 serial links and make sure that there is no packet loss. Open the SAMPA Analyzer.

Pay attention that will be create a root file *as SAMPA_test_17-08-22-15-51-43.root* in the directory that you open the SAMPA analyzer. The name is related to the date and time that the file is created.

In the Run control tab of SAMPA Communicator, click to Acquire.

From the window “Standard View” if SAMPA analyzer is possible to follow the values of Entries, Mean (that is the Baseline) and RMS (that is the noise in *ADC_counts*).

- ✓ *Take around 100000 entries for a good average.*

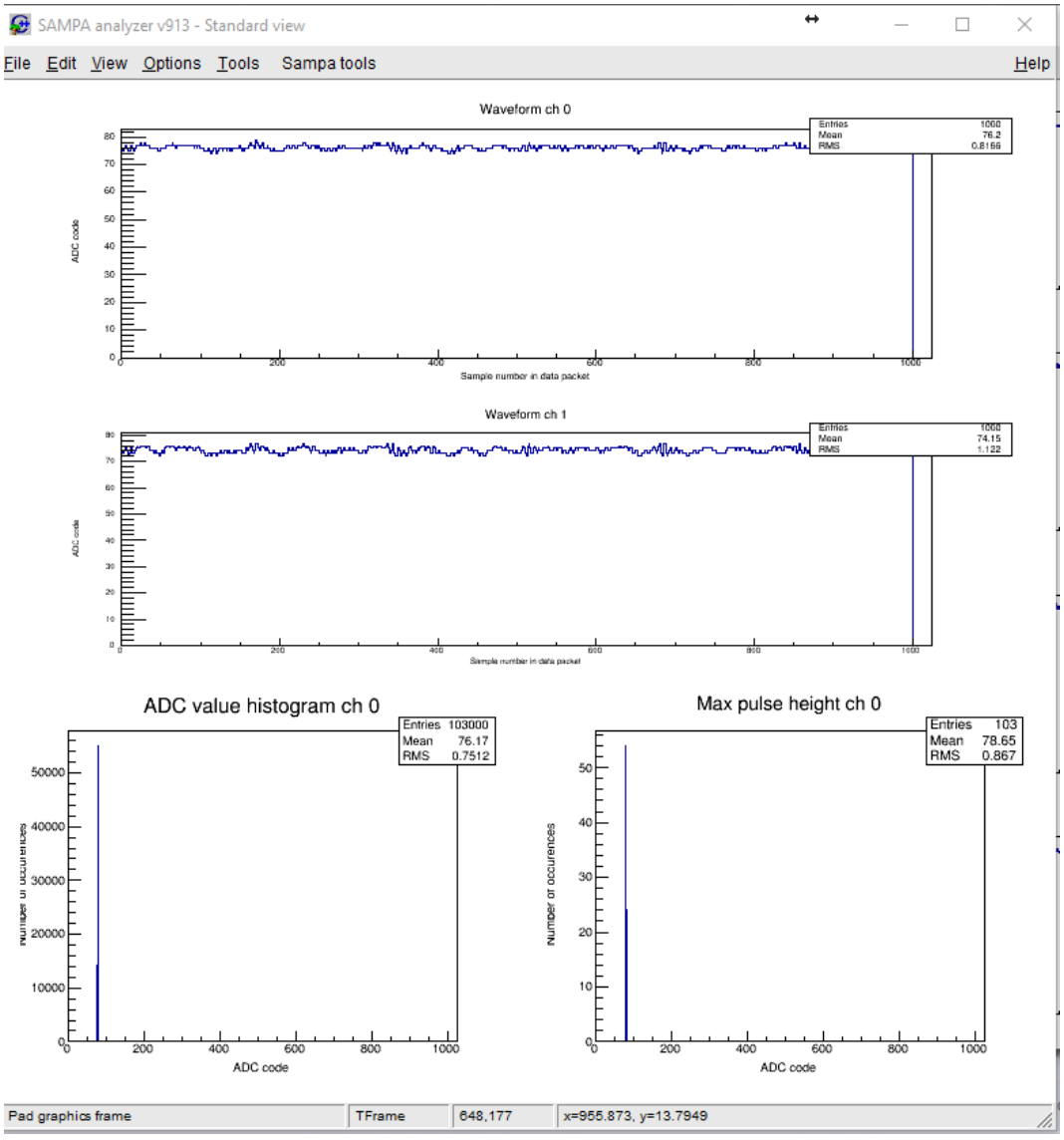


Figure 26 – Standard View of SAMPA analyzer

After that is acquired the suitable number of packets, you should to *Restart* the SAMPA analyzer or close the prompt to finalize the root file and manipulate it in order to get the values of baseline and noise for the 32 channels. See the figure 27 for restart.

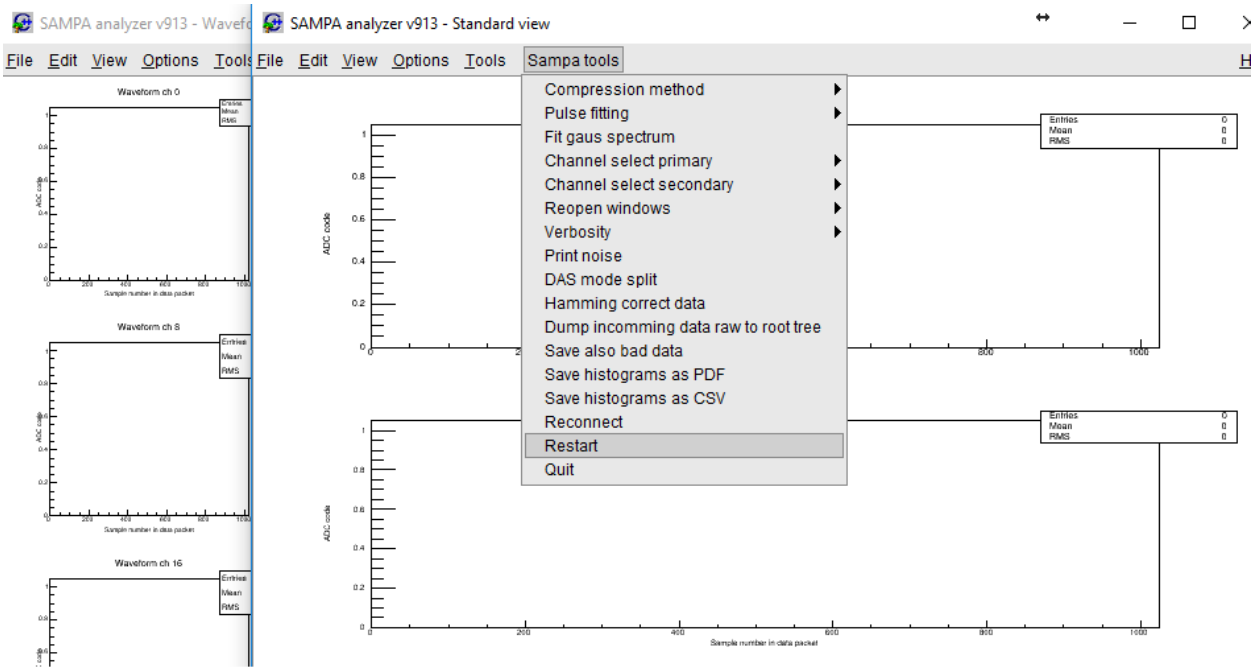


Figure 27 – Standard View of SAMPA analyzer - RESTART

To extract the information from the root file, see section 4.2.
The file **`Noise-anal_sampa_dataCh_fit.cpp`** is located in google drive:
https://drive.google.com/drive/u/0/folders/0ByQPJ-3aV8LYNlNrRWNtVC03NEE

(Internal_SAMPA_Documentation/SAMPA_MPW2_Evaluation_Board_and_Tests_Procedures)

To convert from ADC count to Electrons the is a expression:
ENC=ADC_count *2.14e-3 / (1.6e-19***Gain***1e15)

Example for a gain 30 mV
ENC=ADC_count *2,14*10^-3/ (1,6*10^-19*30*10^-3*10^15)

The gain is shown on section 3.3. This test will be performed with different configurations of gain, peaking time and clock configuration.

For reference, see the file noise_channels.ods on the google drive (same link before). There are a lot of results of SAMPA V2 NCCA board.

C. Linearity, Sensitivity and peaking time tests

These tests are divided on two parts:

1. The data acquisition varying the amplitude of input signal
2. The analysis of the collected data

To setup the system to acquire data, is necessary to couple the injection board to the NCCA/PCCA SAMPA board. Moreover, is necessary to configure the waveform generator and put the jumpers to enable the channel on the injection board. The figure 28 shows the setup.

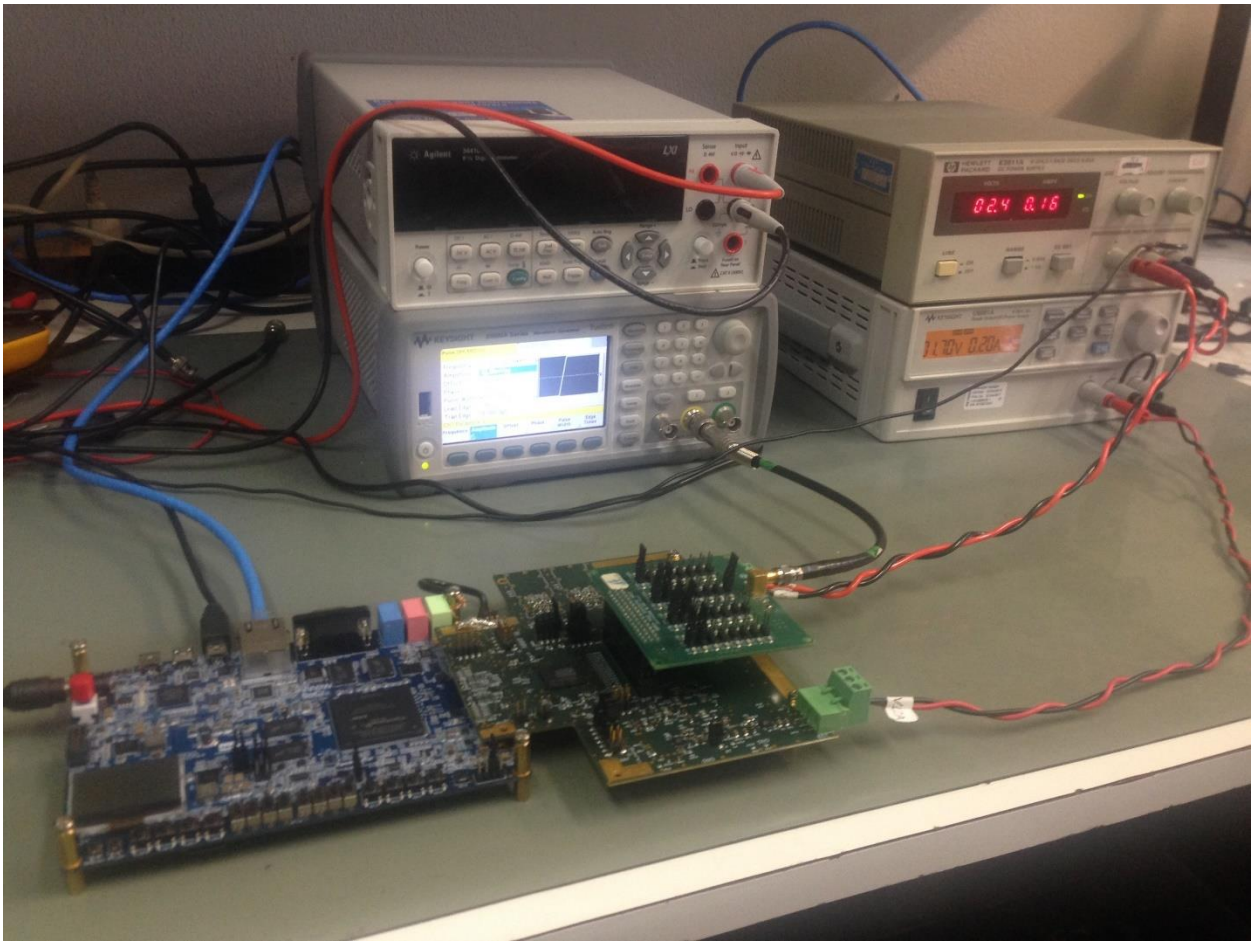


Figure 28 – Injection board coupled to the NCCA Board.

1. Data acquisition

a. Configure the SAMPA board to a specific mode TPC or MCH (Clock / Gain / Peaking Time/ Polarity);
b. Connect an injection board (see figure 3 and 28);
   There are different injection boards, see the file Injection_board_spec.xlsx on google drive (see above) for details;
c. Configure the waveform generator. Go to section 3.6 & figure 10;
d. Make sure there is NO packet loss;
e. Set the amplitude of input signal to the lowest value (see values below);
f. Acquire data (SAMPA Communicator);
g. Restart SAMPA Analyzer and rename the root file. Use the amplitude of input signal in the name. Ex: TPC_30_1mV.root;
h. Increase the value of the input signal following the values below and repeat f and g.

Input values:

- TPC Gain = 20 mV and CAP2:

1; 2; 4; 5; 9; 15; 27;33;45;51;63;69;79;83;87;90;96;99;101;104;105;108;114;120;124

- TPC Gain = 30 mV and CAP2:

1;2;4;5;10;14; 22;26;34;38;46;50;55;58;59;61;65;67;68;70;71;76;80;82

- MCH Gain = 4 mV and CAP8:

1; 2;4;5;10;14;22;26;34;38;46;50;60;65;69;71;77;80;83;87;89;92;94;97;100;103;104

- MCH Gain = 4 mV and CAP6:

25;10;20; 50;110;130;170;190;230;250;300;325;345;355;385;400;415;435;445;460;470; 485;500

(pay attention if signal is aready saturated)

In the SAMPA Analyzer – Standard view, look to Max pulse Height, if the Mean value is already close to 1024, probably the signal is saturated.

2. Analysis – Linearity

To perform the analysis of the root files and to plot a curve of linearity (amplitude of output x amplitude of input signal) is necessary to run some scripts. The figure 29 is an example of the plot, showing the plot for one channel.

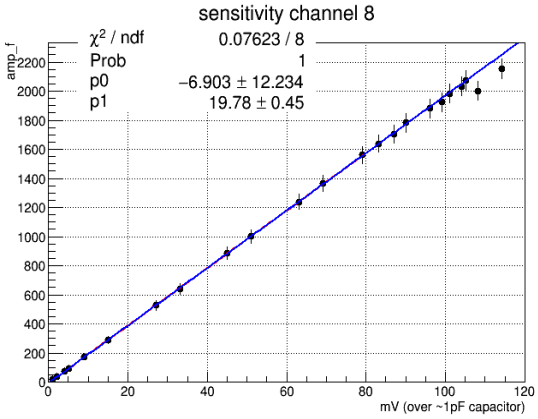


Figure 29 – Linearity plot

The scripts were uploaded to the folder MACROS on google drive:
(https://drive.google.com/drive/folders/0B3p8Q3P1dmewR2VydXlGNkJfNjQ)

The next steps were performed on the cluster of IF - USP.

Analysis of the root files using a script developed by Ganesh.

a. First of all, save the scrips on a personal area, e.g. /sampa/dcarvalho/SAMPA_ANALYSIS/MACROS
b. Save all the root files on a specific directory that is accessible, e;g:
   /sampa/dcarvalho/SAMPA_ANALYSIS/NCCA_V2_8/TPC_30_default_voltage
c. Edit the script ***submitGanesh.sh.*** Is necessary to set the property input and output directories:
   OUTPUTS:
   MAINDIR -> Where the output will be generated;
   DATADIR -> Where the .root to be analyzed were saved;
   LOGDIR -> Where the logs will be saved; (.e -> error file, .o -> output file).
   INPUTS:
   Yet on the ***submitGanesh.sh,*** there is a *for* that set the input values. Is necessary to define these values in accord to the .root files generated.
   Besides that, is necessary to set the FILEIN1, that will to load each .root file

```
 export FILEIN1=
# export FILEIN1="$DATADIR/TPC_30_${i}mVpp.root"
#for MCH_4_CAP8:
#export FILEIN1="$DATADIR/MCH_4_cap8_${i}mVpp.root"
#for MCH_4_CAP6:
#export FILEIN1="$DATADIR/MCH_4_Cap6_${i}mVpp.root"
#for TPC_20_cap2:
export FILEIN1="$DATADIR/TPC_20_${i}mV.root"
#for TPC_30_cap2:
#export FILEIN1="$DATADIR/TPC_30_${i}mV.root"
```

   Pay attention, the script uses the name that was defined of the root file and varies the value of the input signal, according to the for – e.g: TPC_30_${***i***}mV.root
   RUN:
   To run the script just do ***./submitGanesh.sh*** on the terminal (you should do it from the directory were the script is).

   After you run this script, several **i**mV folders will be created on the $MAINDIR.

To generate the graphics:
After the processing of the .root files, is necessary to run the script ***submitAnalysis2.sh*** to generate the sensitivity and linearity analysis.
First, is necessary to edit the script in order to set the properly directories:

MAINDIR -> Where the output will be generated (to use the same of the ***submitGanesh.sh);***
RESUDIR-> New folder that will be create the sensitivity files;
LOGDIR -> Where the logs will be saved;

The line *export FILEOUT="$RESUDIR/sensitivity_${i}.root"* define where the files will be created. For each YY value of input will be create a file sensitivity YY.root.

PAY ATTENTION: Besides the definition of the directories, the script ***submitAnalysis2.sh*** also has a ***for*** as the ***submitGanesh.sh*** script. The for must be with the same values.

As mentioned, the script ***submitAnalysis2.sh*** results in several files as sensitivity_YY.root. Is necessary to add them in order to be processed. The final file should be named as TotalSensitivity.root. For that, run, on the prompt, the follow comand inside $RESUDIR:

hadd -f TotalSensitivity.root sensitivity*.root

To create the plot as figure 29 and as figure 30, is necessary to run a C program, that depends of the configuration of the experiment. If the SAMPA ASIC was configured with a gain 20, uses the file plot_sensitivity_20.C, if 30, uses plot_sensitivity_30.C.

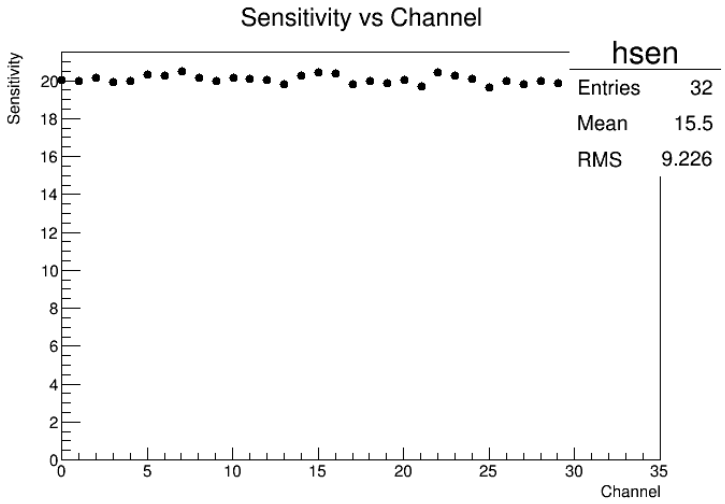


Figure 30 – Sensitivity plot

To do that, copy the refenciated C program to the folder RESUDIR , entry ***root*** on the prompt (inside the RESUDIR), and than, run the follow command:

.x plot_sensitivity_30.C or .x plot_sensitivity_20.C