# **LAGO Official Electronics guide**

Laboratorio Detección de Partículas y radiación

Centro Atómico Bariloche

Version 1.0 30 October 2011

This guide is composed of 4 documents:

- 1. Electrónica LAGO: Guía de conexión de hardware Describing the hardware of the electronics board (pp 2-8)
- 2. Interfaces LAGO Describing the extra sensors (P/T and GPS) (pp 9-15)
- 3. New LAGO electronic: Getting started How to use the new electronics and get data (pp 17-22)
- 4. Data analysis of the high bandwidth LAGO electronics How to analyse the new LAGO dataset (pp 23-30)

## Electrónica LAGO: Guía de conexión de hardware

Miguel Sofo Haro<sup>\*</sup> Laboratorio de Detección de Partículas y Radiación Centro Atómico Bariloche

28 de octubre de 2011

#### Resumen

Cada sitio de LAGO consta de tres detectores Cherenkov. Cada detector esta formado por un tanque de agua, un fototubo de 9 pulgadas y una base encargada de generar la alta tension y polarizar el fototubo. Los pulsos de salida de cada fototubos son digitalizados y procesados por la electrónica, para el posterior almacenamiento y distribución. Esta nota describe como interconectar a nivel de hardware cada uno de estos bloques.

### 1. Introducción

La figura 1 muestra un esquema de bloques de LAGO. En cada tanque de agua esta ubicado un fotomultiplicador (PMT) y una base que genera la alta tension y polariza el PMT. Las tensiones de alimentación de las bases se obtienen conectando las bases a la placa digitalizadora. La electrónica puede digital izar hasta tres señales de ánodo.

La tensión de alimentación general es de 12V (mas de 1A) y se conecta a la placa NEXYS-2 como se explica en la sección 3.



Figura 1: Diagrama de bloques de la electrónica de LAGO. La alimentación general del sistema se obtiene conectando a la NEXYS-2 una fuente externa de 12V de mas de 1A.

En la sección 2 se muestra como obtener los pulsos de ánodo de los PMT y las tensiones necesarias para hacer funcionar las bases.

En las secciones 3 y 4 se muestra como utilizar el kit de desarrollo NEXYS-2 y la placa digitalizadora respectivamente.

La sección 5 da los detalles de conexión de las bases a la electrónica.

<sup>&</sup>lt;sup>\*</sup>miguelsofoharo@gmail.com

## 2. El fototubo y las bases

Actualmente se utiliza una variedad de fototubos, pero en su mayoría son Photonis XP1802 (9") [1] y en menor medida los: Hamamatsu R5912 (8") [1] y ElectronTube 9353 (8") [1]; todos rescatados del arreglo de ingeniería del observatorio Pierre Auger. La figura 2 muestra el circuito de polarización del PMT Hamamatsu R5912. Todos se polarizan con tensiones HV positivas y con un circuito similar al de la figura 2, mas detalles de los circuitos de polarización se encuentran en [2].



Figura 2: Circuito de Polarización del PMT Hamamatsu R5912

Las bases de los PMT, también son rescatadas del arreglo de ingeniería del observatorio Pierre Auger. La alta tension se genera con una fuente EMCO C20, cuya salida va hasta 2000V. En LAGO, son importantes tres conectores de las bases: el conector SMA donde salen los pulsos del ánodo (indicado en la figura 3), el conector DB15 donde se suministran las tensiones de alimentación y de control de la tension de alta (indicado en la figura 4) y el conector circular azul donde se conecta el PMT.



Figura 3: Bases de los PMT de LAGO. En rojo se marca el conector SMA donde salen los pulsos del ánodo del PMT. En el conector circular azul se conecta el PMT.



Figura 4: Conector DB15 plano de las bases de los PMT de LAGO.

En la tabla 1 se resumen las conexiones que se deben realizar sobre el conector DB15 de la figura 4. En la sección 5 se explica como tomar todas las tensiones desde la placa digitalizadora.

Numana da Din	A dende se coneste?
Numero de Pin	A donde se conecta:
1	Se debe conectar a $+3,3V$
2	Se debe conectar a $-3,3V$
3	Se debe conectar a $+5V$
4	Se debe conectar a $+12V$
5	Se debe conectar a la tension de control de HV, puede ir desde $0V$ a $+2,5V$
6	No Conectar (NC)
7	No Conectar (NC)
8	No Conectar (es el pin3 del sensor de temperatura AD592 que se encuentra en la base)
9	Se debe conectar a GND
10	Se debe conectar a GND
11	Se debe conectar a GND
12	Se debe conectar a GND
13	Se debe conectar a GND
14	No conectar (NC)
15	No conectar (es el pin 1 del AD592)

Cuadro 1: Conexiones del conector DB15 de las bases de LAGO

## 3. La placa NEXYS-2

La placa NEXYS-2 es un kit de desarrollo de FPGA de la empresa Digilent [3], y se utiliza para tratar los pulsos digitalizados. La figura 5 muestra el kit NEXYS-2 y que conectores son importantes en LAGO.



Figura 5: Kit de desarrollo NEXYS-2 de Digilent. En rojo se indican que conectores y jumpers deben tenerse en cuenta en LAGO.

En el jack **J3** se conecta la alimentación general de 12V (mas de 1A), **RESPETANDO LA POLARIDAD QUE SE INDICA EN LA PLACA**.

Con el jumper **JP7** se indica la fuente de alimentación, para este caso, se debe colocar en la posicion **WALL** que corresponde a la alimentación por el jack J3.

Con el switch **SWP** se enciende la placa. En la placa se indica la posición de ON.

El conector **USB mini-B** es el puerto **USB** que se conecta a la PC, tanto para programar la FPGA como para realizar la transferencia de datos (pulsos).

Una vez cargado el firmware de la FPGA, el jumper **JP9** se debe pasar de la posición JTAG a la posición **ROM**.

En el conector J1 se conecta la **placa digitalizadora**, en la sección 4 se presenta la placa digitalizadora.

Para alimentar la placa digitalizadora con los 12V del jack J3, SE DEBE COLOCAR EL JUMPER JP4.

## 4. La placa digitalizadora

La figura 6 muestra la placa digitalizadora, se conecta a la placa NEXYS-2 (ver sección 3) por el conector J8 (conector Hirose FX2-100S-1.27DS(71)).



Figura 6: Placa Digitalizadora.

La placa puede digitalizar simulataneamente hasta tres canales de pulsos. La digitalización se realiza con 10bits a 40Msps (Msps = millones de muestras por segundo) con conversores AD9203 de Analog Devices. En los conectores SMA-HEMBRA **J1**, **J2** y **J3**, se conectan los pulsos de anodo que viene de las bases de los PMT.

Con los jumper **JP43**, **JP17** y **JP18** se ajusta la tension de referencia de los ADC en 1V o 2V. Con **J43** en la posición de la imagen 6, se puede digitalizar con 10bit pulsos de hasta 2V en **J1**, y en la posición opuesta, de hasta 1V con igual numero de bits. Idem para JP17 y JP18. JP17 se utiliza para los pulsos en J2 y JP18 en J3.

Los conectores RJ45: **J5**, **J6** y **J7**, proveen las tensiones de alimentación de las bases (las que se deben conectar en el conector DB15 de la figura 4) y la tension de control de la fuente de alta.

J5 corresponde al canal 1, J5 al canal 2 y J6 al canal 3.

La figura 7 muestra que tension corresponde a cada pin del conectores RJ45 J5 (los conectores J6 y J7 tiene igual configuración de tensiones).



Figura 7: Canales de Slow Control. En rojo se indican las tension disponible en cada pin.

La tension de control **Vcontrol** se utiliza para controlar la tension de alta que se aplica al PMT, va de 0 a 2.5V en 255 pasos. Para 2.5V de **Vcontrol**, la salida de la fuente EMCO C20 es de 2000V (la máxima aplicable a los PMT).

Los jumper no mencionados se deben mantener en la posición que aparece en la figura 6.

### 5. La conexión de las bases a la electrónica

Para conectar una base de PMT a la placa digitalizadora se necesitan dos cables: un cable coaxil RG-58 de 50 $\Omega$  de no mas de 20m con cada extremo terminado en conectores SMA-MACHO, y un cable de red UTP (de calidad para exteriores) DERECHO, esto quiere decir que en cada extremo se respeta el mismo orden de colores en el conector RJ45.

Un extremo del cable coaxial se conecta en el conector J1 (o en J2 o J3, dependiendo del canal que se este usando) de la placa digitalizadora , y el otro extremo, al conector de anodo de la base (ver figura 3).

Un extremo del cable UTP se conecta al conector J5 de la digitalizadora, y el otro extremo se conecta a la base utilizando la placa que convierte de RJ45 a DB-25 plano que se muestra en la figura 8.



Figura 8: Placa para convertir un RJ45 a DB15 plano

Una opción a la placa que convierte de RJ45 a DB15, es armar un cable como el de la figura 9. Un extremo es un conector RJ45 que se conecta a la digitalizadora, y el otro extremo, un conector DB15-HEMBRA plano para cable. El cable se arma correctamente si se hace corresponder, uno a uno, las tensiones indicadas en la figura 7 con las de la tabla 1 del conector DB15 de la base.



Figura 9: Cable de RJ45 a DB15

La figura 10 muestra conectada la base a la placa digitalizadora.



Figura 10: Coneccion de la base a la placa digitalizadora

## 6. La conexión a la PC

La conexión con la PC se realiza con la interfaz USB, del conector  ${\bf USB}$  mini-B de la NEXYS-2 (ver figura 5).

## Referencias

- [1] GAP note 99-045: Surface Detector PMT test.
- [2] Philips Photonics (Photonis), Photomultiplier tubes, principles and applications, chapter 5, 1994.
- [3] www.digilentinc.com

**Interfaces LAGO** 



# Interfaces LAGO

L. Horacio Arnaldi Laboratorio de Detección de Partículas y Radiación Centro Atómico Bariloche San Carlos de Bariloche, Argentina

#### Abstract

Se describen los pasos para la correcta instalación y funcionamiento de las interfaces de la nueva electrónica de Lago desarrollada en el laboratorio DPR.

### Introducción

Este artículo pretende explicar de forma detallada la forma en que deben instalarse las interfaces de la nueva electrónica de Lago. Lo que aquí se describe hace referencia a la última versión, a la fecha de creación de este documento, tanto del hardware como el software de la nueva electrónica de Lago desarrollada en el Laboratorio DPR de Bariloche.

## Desarrollo

Como primera medida se describirá la interfaz para el receptor GPS y luego la interfaz entre la lógica implementada en la FPGA y el sensor de presión y temperatura HP03 de Hope RF.

## **Fuente Down Converter DC-DC**

Debido a que existen dentro del mismo sistema distintos bloques que funcionan con diferentes tensiones se diseñó una fuente conmutada down-converter DC-DC para proveer de energía al módulo receptor GPS.

En el proyecto Lago se utilizan los receptores GPS Oncore y la placa diseñada para proveerle energía también incorpora líneas de señal para conseguir una integración lo más rápida posible evitando el uso de cables innecesarios.

Es importante saber que la placa Nexys2 debe estar alimentada con una fuente de tensión de 12 V regulados y que dicha fuente debe ser capaz de suministrar por lo menos 1 Ampere de corriente. Se corre riesgo de dañar las placas y/o los fototubos si no se presta especial atención a que la fuente cumpla con las condiciones antes mencionadas.

La fuente diseñada toma energía a través de la línea de 12V que también provee energía a la



Nexys2 y entrega 5V regulados al módulo receptor GPS. Se debe colocar el jumper tal y como indica la siguiente figura para habilitar el paso de energía desde la línea de 12V hacia la fuente conmutada.





La siguiente figura ilustra el esquema de la fuente conmutada diseñada para alimentar el módulo GPS Oncore.



Figura 2. Esquema de la fuente conmutada







Figura 3. PCB de la fuente

La misma se alimenta a través de la línea de 12V que también provee energía a la Nexys2 y entrega 5V regulados al módulo receptor GPS.

## Interfaz para el sensor de presión y temperatura

El sensor de presión y temperatura utilizado es el HP03S de Hope RF. Se monta el sensor en una placa pequeña que sirve de interfaz acon la Nexys2. En la siguiente figura se puede ver el montaje en el conector denominado JB de la Nexys2.



Figura 4. Montaje del sensor HP03 en la placa Nexys2



## **Interfaces LAGO**

Es importante destacar que debe colocarse una resistencia de 10 Kohms entre +V y la pata SDA del sensor HP03 para que la comunicación entre la FPGA y el sensor funcione correctamente. En la siguiente figura se puede apreciar el detalle de la resistencia colocada en posición.



Figura 5. Detalle de la resistencia a colocar

### Archivo .ucf

El archivo con extensión **.ucf** es el que determina la disposición final para la configuración de las patas de la FPGA. En dicho archivo pueden encontrarse las asignaciones para los pines de entrada y salida.

A continuación se transcribe el archivo utilizado para el proyecto LAGO.

```
#Copyright 2011 - Lab DPR (CAB-CNEA). All rights reserved.
#Redistribution and use in source and binary forms, with or without
#modification, are permitted provided that the following conditions
#are met:
#
#
    1. Redistributions of source code must retain the above
#
       copyright notice, this list of conditions and the following
       disclaimer.
#
#
#
    2. Redistributions in binary form must reproduce the above
#
       copyright notice, this list of conditions and the following
```



#

#### **Interfaces LAGO**

```
#
      provided with the distribution.
#THIS SOFTWARE IS PROVIDED BY LAB DPR ''AS IS'' AND ANY EXPRESS OR IMPLIED
#WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
#MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
#IN NO EVENT SHALL LAB DPR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
#INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
#IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
#POSSIBILITY OF SUCH DAMAGE.
#
#The views and conclusions contained in the software and documentation
#are those of the authors and should not be interpreted as representing
#official policies, either expressed or implied, of Lab DPR.
#
# This file is a general .ucf for Nexys2 rev A board
# To use it in a project:
# - remove or comment the lines corresponding to unused pins
# - rename the used signals according to the project
# Signals Led<7>Led<4> are assigned to pins which change type from s3e500 to
# other dies using the same package
# Both versions are provided in this file.
# Keep only the appropriate one, and remove or comment the other one.
# Clock pin for Nexys 2 Board
NET "clk"
               LOC = "B8" ; # Bank = 0, Pin name = IP L13P 0/GCLK8, Type =
GCLK, Sch name = GCLK0
NET "clk"
          TNM NET = clk;
TIMESPEC TS clk = PERIOD "clk" 20 ns HIGH 50%;
# OnBoard USB controller
NET "eppAS" LOC = "V14"; # Bank = 2, Pin name = IP L23P 2, Type = INPUT,
Sch name = U-FLAGA
NET "eppDS"
                LOC = "U14"; # Bank = 2, Pin name = IP_L23N_2, Type = INPUT,
Sch name = U-FLAGB
```

disclaimer in the documentation and/or other materials



NET "eppRnW" LOC = "V16"; # Bank = 2, Pin name = IP, Type = INPUT, Sch name = U-FLAGCLOC = "N9" ; # Bank = 2, Pin name = IO L12P 2/D7/GCLK12, Type NET "eppAck" = DUAL/GCLK, Sch name = U-SLRD NET "eppDB<0>" LOC = "R14"; # Bank = 2, Pin name = IO L24N 2/A20, Type = DUAL, Sch name = U-FD0NET "eppDB<1>" LOC = "R13"; # Bank = 2, Pin name = IO L22N 2/A22, Type = DUAL, Sch name = U-FD1NET "eppDB<2>" LOC = "P13"; # Bank = 2, Pin name = IO L22P 2/A23, Type = DUAL, Sch name = U-FD2NET "eppDB<3>" LOC = "T12"; # Bank = 2, Pin name = IO L20P 2, Type = I/O, Sch name = U-FD3NET "eppDB<4>" LOC = "N11"; # Bank = 2, Pin name = IO L18N 2, Type = I/O, Sch name = U-FD4NET "eppDB<5>" LOC = "R11"; # Bank = 2, Pin name = IO, Type = I/O, Sch name = U-FD5 NET "eppDB<6>" LOC = "P10"; # Bank = 2, Pin name = IO L15N 2/D1/GCLK3, Type = DUAL/GCLK, Sch name = U-FD6 NET "eppDB<7>" LOC = "R10"; # Bank = 2, Pin name = IO L15P 2/D2/GCLK2, Type = DUAL/GCLK, Sch name = U-FD7 # Leds NET "led<0>" LOC = "J14"; # Bank = 1, Pin name = IO L14N 1/A3/RHCLK7, Type = RHCLK/DUAL, Sch name = JD10/LD0 NET "led<1>" LOC = "J15"; # Bank = 1, Pin name = IO L14P 1/A4/RHCLK6, Type = RHCLK/DUAL, Sch name = JD9/LD1 NET "led<2>" LOC = "K15"; # Bank = 1, Pin name = IO L12P 1/A8/RHCLK2, Type = RHCLK/DUAL, Sch name = JD8/LD2 NET "led<3>" LOC = "K14"; # Bank = 1, Pin name = IO L12N 1/A7/RHCLK3/TRDY1, Type = RHCLK/DUAL, Sch name = JD7/LD3 LOC = "R4" ; # Bank = 3, Pin name = IO/VREF 3, Type = VREF, NET "led<4>" Sch name = LD7? s3e500 only **#PWM Signals** NET "hv2" LOC = "A16"; # Bank = 0, Pin name = IO LO1N 0, Type = I/O, Sch name = R-IO39NET "hv3" LOC = "B16"; # Bank = 0, Pin name = IO L01P 0, Type = I/O, Sch name = R-IO40# Buttons NET "reset" LOC = "B18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = BTN0# FX2 connector ch1 NET "adc clk1" LOC = "B4";



## **Interfaces LAGO**

NET	"ch1<9>"	LOC	=	"A4";
NET	"ch1<8>"	LOC	=	"C3";
NET	"ch1<7>"	LOC	=	"C4";
NET	"ch1<6>"	LOC	=	"B6";
NET	"ch1<5>"	LOC	=	"D5";
NET	"ch1<4>"	LOC	=	"C5";
NET	"ch1<3>"	LOC	=	"F7";
NET	"ch1<2>"	LOC	=	"E7";
NET	"ch1<1>"	LOC	=	"A6";
NET	"ch1<0>"	LOC	=	"C7";
# FX	2 connector cl	n2		
NET	"adc_clk2"	LOC	=	"F8";
NET	"ch2<9>"	LOC	=	"D7";
NET	"ch2<8>"	LOC	=	"E8";
NET	"ch2<7>"	LOC	=	"E9";
NET	"ch2<6>"	LOC	=	"C9";
NET	"ch2<5>"	LOC	=	"A8";
NET	"ch2<4>"	LOC	=	"G9";
NET	"ch2<3>"	LOC	=	"F9";
NET	"ch2<2>"	LOC	=	"D10";
NET	"ch2<1>"	LOC	=	"A10";
NET	"ch2<0>"	LOC	=	"B10";
# FX	2 connector cl	n3		
NET	"adc_clk3"	LOC	=	"A11";
NET	"ch3<9>"	LOC	=	"D11";
NET	"ch3<8>"	LOC	=	"E10";
NET	"ch3<7>"	LOC	=	"B11";
NET	"ch3<6>"	LOC	=	"C11";
NET	"ch3<5>"	LOC	=	"E11";
NET	"ch3<4>"	LOC	=	"F11";
NET	"ch3<3>"	LOC	=	"E12";
NET	"ch3<2>"	LOC	=	"F12";
NET	"ch3<1>"	LOC	=	"A13";
NET	"ch3<0>"	LOC	=	"B13";
# MA	X5501 Interfac	ce		
NET	"cs_e2prom" LO	DC =	"В	:14";
NET	"cs_max5501" 1	LOC =	. "	A14";
NET	"spi_dout"	LOC	=	"C14";

#### **Interfaces LAGO**



NET "spi sck" LOC = "D14"; # HP03 Interface # It is in JB 6 pin expansion connector NET "pMCLK" LOC = "R18"; # Bank = 1, Pin name = IO L02P 1/A14, Type = DUAL, Sch name = JB2NET "pXCLR" LOC = "R16"; # Bank = 1, Pin name = IO LO3N 1/VREF 1, Type = VREF, Sch name = JB8NET "pSDA" LOC = "T18"; # Bank = 1, Pin name = IO\_L02N\_1/A13, Type = DUAL, Sch name = JB9NET "pSCL" LOC = "U18"; # Bank = 1, Pin name = IO\_L01P\_1/A16, Type = DUAL, Sch name = JB10# GPS Interface # It is in JC 6 pin expansion connector NET "tx uart" LOC = "G15"; # Bank = 1, Pin name = IO L18P 1, Type = I/O, Sch name = JC1 NET "pps port" LOC = "G13"; # Bank = 1, Pin name = IO L20N 1, Type = I/O, Sch name = JC3NET "rx uart" LOC = "H16"; # Bank = 1, Pin name = IO L16P 1, Type = I/O, Sch name = JC4

## New LAGO electronic: Getting started

H.G. Asorey M. Gómez Berisso Laboratorio de Detección de Partículas y Radiación Centro Atómico Bariloche

October 30, 2011

The aim of this document is describe a basic procedure to turn on the new LAGO electronic, perform a simple data aquistion and turn off securely.

#### 1 Turn on procedure

- 1. Set the power source to a voltage of 12.0 Volt and limiting the current to 1.0 Amp.
- 2. Check the voltage in the power connector itself, negative in the outer jacket, positive in the inner pin.
- 3. Be sure that:
  - (a) the FPGA power switch is placed in the OFF position
  - (b) the JP7 POWER SELECT jumper is places in the WALL position
  - (c) the RJ45-DB15 connector to the PMT is disconnected
  - (d) the USB cable is connected to the PC
- 4. Plug the power connector to the FPGA and turn ON. After that:
  - (a) three LED will turn on in sequence: RED (POWER), YELLOW (DONE), and GREEN
  - (b) two LED will turn on: LD0 and LD1
  - (c) one LED will start to blink one a second: LD7
  - (d) finally LD1 will turn off
- 5. Remember that the new front end electronic have two different forms to control the PMT high voltage (CTRL\_PMT):

Channel 1: Digital to analog converter (DAC)

Channel 2: Pulse with method converter (PWM)

Channel 3: Pulse with method converter (PWM)

The calibration curves of such values have to be established for each electronics, setting the HV and seeing the resulting control voltage (see below).

- 6. Run the following commands in order to check a proper link between FPGA and PC:
  - (a) Registers status: IN: ./lago -a OUT: #Trigger Level Ch1 = 1000 #Trigger Level Ch2 = 1000 #Trigger Level Ch3 = 1000 #Subtrigger Ch1 = 1000 #Subtrigger Ch2 = 1000 #Subtrigger Ch3 = 1000 #High Voltage 1 = 0 #High Voltage 2 = 0 #High Voltage 3 = 0 #GPS Time Mode = UTC Status from registers complete! (b) Temperature and preasure status: IN: ./lago -t OUT: : 27.0 C Temperature : 919.52 hPa Pressure Altitude : 811 m (c) GPS status: IN: ./lago -g OUT: Motorola Oncore GPS Receiver Data v0r9 Date : 10/28/2011 Time : 15:27:16 Frac Sec: 0.847266

Geogarphic location and Status Flags

Lat	: -41.11	17785 Loi	1 :-	-71.39854	12			
Height :	834.26	msl Veld	ocity: 0	.32 Km/h				
Heading	: 315.0	true DOM	PT : 0					
NVS	: 10	NTS	: 7					
	Sat1	Sat2	Sat3	Sat4	Sat5	Sat6	Sat7	Sat8
ID	13	23	4	7	10	20	2	8
CHTM	8	8	8	8	8	8	0	8
CNo	43	41	46	48	32	50	0	46
CHSF	162	162	162	0	162	162	32	162
Receiver	Status	Flag : 3	32					

7. Plug the RJ45-DB15 cable to Channel 1 in the front end electronic and, with PMT **DISCONECTED**, check all the voltages in the DB15 connector:

PIN	
1	$+3.3\mathrm{V}$
2	-3.3 V
3	$+5.0\mathrm{V}$
4	$+12\mathrm{V}$
5	CTRL_PMT ( $0.0 - 2.5 \text{ V}$ ), depending on HV set
6	NC
7	NC
8	NC
9	GND
10	GND
11	GND
12	GND
13	GND
14	NC
15	NC

8. Set the PMT high voltage to 1000 DAC counts: IN:

```
./lago -s hv1 1000
OUT:
```

```
#Trigger Level Ch1 = 1000
#Trigger Level Ch2 = 1000
#Trigger Level Ch3 = 1000
#Subtrigger Ch1 = 1000
#Subtrigger Ch2 = 1000
```

#Subtrigger Ch3 = 1000
#High Voltage 1 = 1000
#High Voltage 2 = 0
#High Voltage 3 = 0
#GPS Time Mode = UTC

Status from registers complete!

Check that the voltage in a DB15 - PIN 5 - CTRL\_PMT (0.0 - 2.5 V) is approximately 1.2 Volt. Check then the correspondance between HV1 and control voltage, and repeat for channels 2 and 3.

- 9. Set the PMT high voltage to 0 ADC counts.
- 10. Connect the PMT to the DB15 connector (cross your fingers!). The current consumption of the whole electronics rises 0.26 Amps.
- 11. Check the PMT response:
  - (a) Connect the DYNODE cable of the PMT to a scope and set the amplitude scale to 10 mV, time scale to 25 ns/div, and negative edge trigger to -2 mv.
  - (b) Set the PMT high voltage to 1000 DAC counts: IN:

```
./lago -s hv1 1000
OUT:
#Trigger Level Ch1 = 1000
#Trigger Level Ch2 = 1000
#Trigger Level Ch3 = 1000
#Subtrigger Ch1
                   = 1000
#Subtrigger Ch2
                   = 1000
#Subtrigger Ch3
                   = 1000
#High Voltage 1
                   = 1000
#High Voltage 2
                   = 0
#High Voltage 3
                   = 0
#GPS Time Mode
                   = UTC
```

Status from registers complete!

The current consumption of the whole electronics rises 0.27 Amps. At this point the PMT traces should be visible on the scope:



- 12. Connect the DYNODE cable of the PMT to the front end electronic.
- 13. Set the PMT trigger level: IN:

```
./lago -s t1 100
OUT:
#Trigger Level Ch1 = 100
#Trigger Level Ch2 = 1000
#Trigger Level Ch3 = 1000
#Subtrigger Ch1
                   = 1000
#Subtrigger Ch2
                   = 1000
#Subtrigger Ch3
                   = 1000
#High Voltage 1
                   = 1500
#High Voltage 2
                   = 0
#High Voltage 3
                   = 0
#GPS Time Mode
                   = UTC
```

Status from registers complete!

## 2 Start a simple data adquistion

1. Start the adquisition: IN:

./lago -f file\_output.dat

Until the adquisition is stoped (CRT+C or killing the process), LED LD0 and LD1 will blink.

## 3 Turn off procedure

1. Set the PMT high voltage to 0 ADC counts: IN:

./lago -s hv1 0 OUT:

```
#Trigger Level Ch1 = 1000
#Trigger Level Ch2 = 1000
#Trigger Level Ch3 = 1000
#Subtrigger Ch1
                = 1000
#Subtrigger Ch2
                   = 1000
#Subtrigger Ch3
                   = 1000
#High Voltage 1
                   = 0
#High Voltage 2
                   = 0
#High Voltage 3
                   = 0
#GPS Time Mode
                   = UTC
```

Status from registers complete!

- 2. Disconnect the DYNODE cable of the PMT to the front end electronic.
- 3. Disconnect the PMT to the DB15 connector.
- 4. Turn off the FPGA.
- 5. Disconnect the FPGA power cable.
- 6. Turn off the power source.

# Data analysis of the high bandwidth LAGO electronics

H. Arnaldi, X. Bertou, M. Sofo Haro Centro Atómico Bariloche, Argentina

October 30, 2011

#### Abstract

High speed ADC, fast FPGA and high bandwidth data transfer methods such as USB, allow to get a much larger set of data than the one usually available when using the single particle technique. In this note we describe the data provided by the new LAGO FPGA code and how it can be used to mimic the previous data set.

## **1** Introduction

The new electronics presents a great improvement from the previous version, based on the Auger EA boards. One of the main improvement is the large bandwidth available from the FPGA to the DAQ PC, through USB. Taking advantage of this large bandwidth, the new data provide pulse information for every trigger, amplitude information for subtriggers, allowing scaler data of arbitrary threshold to be computed offline.

## 2 Data

#### 2.1 Data files

The new data files start with a:

# v <x>

with x being the version number of the data format. Lines starting with "# #" are comments and can be ignored. The header, built with comments, should describe the data format.

The data format for all versions can be found in annex.

#### 2.2 Data concept

Given the high bandwidth, the FPGA will send the trace information as often as possible. 2 buffers get filled one after another and then are read through USB. A threshold can be set for each of the 3 channels, and whenever a pulse triggers a full trace for the 3 channels is sent, with 2 pre-trigger bins, the trigger bin, and 13 post-trigger bins. The time of the pulse trigger (in 40 MHz clock cycle) and a trigger counter are also added.

These data can be read at up to about 10 000 pulses per second. In order to allow operation at higher rate (for large detectors, high altitude and/or low geomagnetic cutoff), a subtrigger mode is available where only the maximum of the triggering pulse is transfered, together with the trigger time. This mode allows to work up to 150 000 pulses per second.

Every second, a block of extra data giving the real time, clock performance, temperature and pressure sensor data, etc., is added to the data flow.

The data file produced are obviously large, depending on the trigger rate. A reasonable detector can easily produce tens of GB per day. It is therefore likely that one will post-process these data and only keep a summary of them. In any case, at a site linked to the internet, one could imagine that whenever a GRB is detected by a satellite, the full data is kept to be analysed later on.

#### 2.3 Data extraction

Data from a short run of about 70 h with the Nahuelito detector in Bariloche will be used in this section. The resulting file, obtained by connecting the anode to channel 1 and the dynode to channel 2, is about 20 GB, compressed with bzip2 to about 3.2 GB. The data was taken with a threshold of 100 ADC on channel 2.

#### 2.3.1 Scaler mode

The new electronics has no scaler mode as this can be mimicked by the new DAQ. One would likely set for each channel the subtrigger to the lowest scaler wanted, and then the trigger levels to a reasonable higher value.

The resulting data can be then converted into the usual 5 ms scaler rates by counting pulses offline and using their timing to determine whenever a new 5 ms period started.

A simple code counting subtriggers, triggers, and triggers of channel 2 above 150 every second is given below:

```
bzcat nahuelito.bz2 | awk '{
if ($1!="#") {
    n++; # trace length
    if ($2>p) p=$2;
}
if ($1=="#" && $2=="t" && n==1) st++; #subtrigger
if ($1=="#" && $2=="t" && n==16) t++; #trigger
if ($1=="#" && $2=="t" && n==16 && p>150) bt++;
```

```
if ($1=="#") n=p=0;
if ($1=="#" && $2=="x" && $3=="h") {
   s++; # new second
   print s,st,t,bt;
   st=t=bt=0;
}}'
```



Figure 1: Trigger rate for Nahuelito for a 50h run, with one dot per second and the average every minute.

#### 2.3.2 Histogram mode

The new data allow trivially to make for example minute-long histograms of charge for all pulses above threshold, and of amplitude for all pulses above subthreshold.

The following code computes the amplitude and charge histogram of channel 2 of the data set, using one hour of data.

```
bzcat nahuelito.bz2 | awk '{
if ($1!="#") {
    n++; # trace length
    q2+=$2-50;
    if ($2>p2) p2=$2;
}
if ($1=="#" && $2=="t" && n==16) {
    q[q2]++;
    p[p2]++;
    q2=p2=0;
}
if ($1=="#") n=0;
if ($1=="#" && $2=="x" && $3=="h") {
    s++; # new second
```



Figure 2: Amplitude (left) and charge (right) histograms for one hour of Nahuelito data in Bariloche. The strong differential non-linearity of the ADC can be seen in the amplitude histogram.

#### 2.3.3 Muon decay

With the time of each pulse, it is easy also to look for the histogram of time diferences between pulses and get the muon decay time, as can be seen in figure 3, where the muon decay time is fitted to  $2037 \pm 6$  (stat) ns. The charge of the second pulse for muon decays can be used for calibration studies, while the first pulse gives indication of the stopping muons energy.

Presence of gaussian humps at a few microseconds in the histogram sign the existence of afterpulses, while sharp spikes in the histogram indicate HF pick-up noise.

The following code assumes only one detector is connected and outputs the pulse time difference histogram.

```
bzcat nahuelito.bz2 | awk '{
if ($1!="#") n++; # trace length
if ($1=="#" && $2=="t" && n==16) {
   ot=t;
   t=$4;
   if (ot && t>ot) dt[t-ot]++;
}
if ($1=="#") n=0;
```

```
}
END { for (i=0;i<2000;i++) print i,dt[i]+0; }'</pre>
```



Figure 3: Time between two pulses and muon decay fit, for about 70 hours of data from Nahuelito.

#### 2.3.4 Monitoring data

The monitoring data can be extracted in a straightforward way:

bzcat nahuelito.bz2 | grep "^# x s" | awk '{print \$4,\$6,\$8}'
and can be used to produce graphs similiar to figure 4.



Figure 4: Temperature (left) and pressure (right) during 50h of data taking in Bariloche (temperature is measured inside a building).

# 3 Future data analysis

The new complete data set allows to compute probabilities of getting many pulses in a short period of time in a way much more interesting than the previously fixed scaler counts in 5 ms bins of time. A note is prepared on this topic.

# A LAGO commands

## A.1 Basic commands

-a	Get all registers status
-s	Set registers
-f	Start DAQ and save data to file
-0	Start DAQ and send data to stdout
-d	Get GPS data
-t	Get Pressure and Temperature data

Registers: t1, t2, t3 Specify triggers 1, 2 and 3 st1, st2, st3 Specify subtriggers 1, 2 and 3 hv1, hv2, hv3 Specify high voltages ...

## A.2 Experts commands

-r Get a s	ingle register value
-p Putav	alue into a single register
Registers:	
t1, t2, t3	Specify triggers 1, 2 and 3
stl, st2, st3	Specify subtriggers 1, 2 and 3
hv1, hv2, hv3	Specify high voltages
tm	Specify Time Mode for GPS Receiver
	(0 - UTC, 1 - GPS)
Options:	
-c <# bytes>	Number of bytes to read/write
-b <byte></byte>	Value to load into register

# **B** Data format

## **B.1** version 1

# v 1	: indicates it is version 1 of the data format
# # xxxx	: line of comments
<n1> <n2> <n3></n3></n2></n1>	: line with values of the 3 ADC for a triggered
	pulse. It is a trigger with the full pulse if
	16 lines are found. If only one line is found,
	it is a subtrigger with the pulse maximum bin
# t <c> <v></v></c>	: end of a trigger
	gives the channel trigger ( <c>: 3 bit mask)</c>
	and $40MHZ$ clock count ( <v>) of the trigger time</v>
# c <c></c>	: internal trigger counter
# x f <v></v>	: 40 MHz frequency
# x r C1-DD <v></v>	: raw temperature and pressure sensor value
# x r D1 <v></v>	: raw temperature/pressure value
# x r D2 <v></v>	: raw temperature/pressure value
<pre># x h <hh:mm:ss></hh:mm:ss></pre>	: GPS time (every new second)
# x s <t> C <p> 1</p></t>	nPa <a> m : temperature <t>, pressure <p> and</p></t></a>
	altitude (from pressure) <a></a>