



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA



Manual de Uso "Librería CaenDigitizerSiLab" SiLab

Autor: Jairo González,
Practicante Verano 2018.
jairo.gonzalez.13@sansano.usm.cl

Supervisor: Pablo Ulloa

Desarrollador de la Librería Orlando Soto

Versión: 2.0.0 **Fecha:** Martes 30 de enero 2018

Índice

1. Acerca del Digitizer	2
2. Acerca de la Librería	2
3. Requerimientos	3
3.1. Digitizer	3
3.2. Especificaciones de Hardware	3
3.3. Sistemas Operativos Recomendados	3
3.4. Drivers	4
3.5. Librerías	5
3.6. Software	6
4. Librería CaenDigitizerSiLab	9
4.1. Código Fuente	9
4.2. Dependencias	9
4.3. Compilación y Ejecución	9
4.4. Métodos	10
4.5. Ejemplos de Prueba	14
4.6. Configuraciones	15
4.6.1. Número de Muestras por Evento	15
4.6.2. Número de eventos por medición	15
4.6.3. Número de grupos de eventos por medición	15
4.6.4. Polarización, Rango Dinámico y Offset	15
4.6.5. Trigger	16
5. Análisis de Datos	18
6. Sitios y Documentos de Interés	19

1. Acerca del Digitizer

A grandes razgos, un digitizer es una herramienta que permite tomar muestras de varias señales de voltaje a través de un conversor análogo digital.

Para el caso del digitizer CAEN DT5730, se tiene una tasa de muestreo de 500MS/s (mega samples per second), lo que equivale a un tiempo de 4ns entre cada muestra. Con esta información, es posible calcular una ventana de adquisición, que se refiere a la cantidad ideal de muestras necesarias para contener por completo un evento. Un evento sería una señal de interés, por ejemplo una señal generada por un detector de partículas. En general, para capturar un evento, se suele configurar un umbral de voltaje (threshold) que al ser superado por la señal, gatilla el inicio de la medición (trigger).

Para mayor información, se recomienda visitar los siguientes vínculos:

- Sitio web: [caen.it](http://www.caen.it)
- CAEN DT5730 Site: <http://www.caen.it/csite/CaenProd.jsp?parent=14&idmod=757>
- First Installation Guide to Desktop Digitizers and MCA <http://www.caen.it/csite/CaenProd.jsp?showLicence=false&parent=14&idmod=757>
- CAEN DT5730 DT5725 User Manual: <http://www.caen.it/servlet/checkCaenManualFile?Id=12386>

2. Acerca de la Librería

La librería CaenDigitizerSiLab fue creada con el objetivo de integrar la adquisición de datos con el análisis de estos, utilizando un digitizer CAEN DT5730 para la adquisición de datos y el framework ROOT para el análisis y procesamiento de la información. Actualmente la librería es compatible con digitizers DT5730 y DT5724.

La librería fue desarrollada por Orlando Soto, ingeniero de SiLab del CCTVal, de la Universidad Técnica Federico Santa María.

Este documento fue redactado por Jairo González, estudiante en práctica de Ingeniería Civil Electrónica, bajo la supervisión de Pablo Ulloa, ingeniero de SiLab.

3. Requerimientos

3.1. Digitizer

La librería está hecha para ser utilizada con el digitizer CAEN DT5730 14-bit @ 500 MS/s 8 canales. Se recomienda leer el documento CAENDT730 User Manual del digitizer antes de utilizarlo. El manual está disponible en el sitio web del fabricante, en la página correspondiente al modelo del digitizer, en la sección 'Documentation'.

3.2. Especificaciones de Hardware

Al probar la librería, se utilizó un computador portatil HP-430, procesador Pentium P6200 @2.13GHz de 64bits, 2 núcleos.

3.3. Sistemas Operativos Recomendados

Debian

Se recomienda la utilización de Debian como sistema operativo ya que la librería fue desarrollada en este entorno. Específicamente, la librería fue probada con el sistema operativo Debian 9.3.0 Estable xfce.

Instalación de Debian

Primero, se recomienda descargar la versión completa, estable y más actual de Debian desde el sitio oficial debian.org, en la sección "Getting Debian". La imagen .iso descargada durante la realización de este documento corresponde a Debian 9.3.0 Estable xfce.

Una vez descargada la imagen .iso, se puede montar en un pendrive del tamaño adecuado siguiendo los pasos indicados a continuación:

Para hacerlo desde linux, primero hay que conectar el pendrive al equipo. Luego es necesario encontrar el identificador del pendrive, por ejemplo ejecutando por consola el comando "dmesg", donde probablemente las últimas líneas harán referencia a la unidad USB como /dev/sdX", donde 'X' corresponde al número identificador que estamos buscando. Una vez identificado el número de la unidad, es posible ejecutar por consola los siguientes comandos:

- cp debian.iso /dev/sdX
- sync

Donde 'X' corresponde al número identificador de la unidad. Con estos comandos se copiará y montará la imagen en el pendrive.

Una vez montada la imagen en el pendrive, es necesario apagar el equipo en el cual se instalará el sistema operativo. Luego, se conecta la unidad USB y se enciende el equipo. Se espera que el equipo arranque desde la unidad USB, pero dependiendo del fabricante y la configuración de la BIOS puede ser necesario presionar F9 inmediatamente después de encenderlo (para escoger dispositivos de arranque en equipos HP) u otra tecla especial para configurar el arranque del equipo.

Asumiendo que el equipo inició desde el dispositivo USB, se verá en pantalla la interfaz de instalación del nuevo sistema operativo. Se recomienda leer en detalle las opciones y tener cuidado al intentar modificar opciones avanzadas o particiones de disco. Se recomienda también instalar los paquetes recomendados por el instalador. Una vez finalizada la instalación, será posible iniciar el nuevo sistema operativo.

Para más detalles, leer la guía oficial de instalación de Debian: <https://www.debian.org/releases/stable/installmanual>.

Es posible instalar Debian en otra partición del disco, para que así pueda coexistir con otro sistema operativo. Es posible encontrar tutoriales relativos a este tema buscando en internet.

Otros Sistemas Operativos

Se ha probado la librería en otros sistemas operativos linux sin tener éxito, debido a problemas al incluir algunas dependencias y librerías mediante el Makefile. Aún así, es posible utilizar la librería compilando directamente desde ROOT con la ayuda de un script .sh, aunque es un método no recomendado.

3.4. Drivers

Para que el digitizer pueda comunicarse con el computador, es necesario instalar los drivers del fabricante. Estos drivers se pueden encontrar en el sitio web del fabricante, en la página correspondiente al modelo del digitizer, en la sección "Software/Firmware". Dado que se está trabajando en linux, es necesario descargar el driver correspondiente a este sistema operativo. Para descargarlo es necesario registrarse en el sitio web. Para instalar los drivers, basta extraer los archivos, ingresar a la carpeta y leer el archivo README, que indica las instrucciones. Estas instrucciones suelen ser ejecutar lo siguientes comandos consecutivos:

- make
- make install

Es posible que existan problemas en la instalación por algunas dependencias faltantes. Normalmente el administrador de paquetes indica los pasos a seguir para solucionar este

problema.

Para probar que los drivers están correctamente instalados, es necesario continuar instalando los requerimientos listados en este documento y luego utilizar algún software del fabricante (por ejemplo CAENUpgrader) o uno de los ejemplos de prueba en la sección "Librería CaenDigitizerSiLab".

3.5. Librerías

Tanto para el uso de software del fabricante como para el uso de la librería, es necesario instalar las librerías del fabricante. Estas librerías están disponibles en la categoría Products>Firmware/Software>Software Libraries. Es importante instalar las librerías en el orden indicado en este documento, debido a las dependencias que existen entre ellas.

Para probar que las librerías están correctamente instaladas después de seguir las instrucciones que se indican a continuación, será necesario continuar instalando los requerimientos listados en este documento y luego utilizar algún software del fabricante (por ejemplo CAENUpgrader) o uno de los ejemplos de prueba en la sección "Librería CaenDigitizerSiLab".

CAENVMELib

Para instalar esta librería, es necesario descargarla desde la página correspondiente, ingresando a la sección Download. Para descargarla es necesario registrarse en el sitio web. El archivo a descargar corresponde al indicado para sistemas operativos linux. Una vez descargado el archivo, es necesario descomprimirlo e ingresar a la carpeta generada para leer el archivo README, en el cual se detallan los pasos de instalación. Estos pasos de instalación suelen ser los siguientes:

- Tener permisos de administrador
- Ingresar a la carpeta lib, ubicada en la carpeta generada al descomprimir el archivo
- Abrir una consola en la carpeta lib y ejecutar "sh install.o" "sh install_x64" dependiendo si el sistema es de 32 o 64 bits respectivamente

CAENComm

Para instalar esta librería, es necesario descargarla desde la página correspondiente, ingresando a la sección Download. Para descargarla es necesario registrarse en el sitio web. El archivo a descargar corresponde al indicado para sistemas operativos linux. Una vez descargado el archivo, es necesario descomprimirlo e ingresar a la carpeta generada para leer el archivo README, en el cual se detallan los pasos de instalación. Estos pasos de instalación suelen ser los siguientes:

- Tener permisos de administrador
- Ingresar a la carpeta lib, ubicada en la carpeta generada al descomprimir el archivo
- Abrir una consola en la carpeta lib y ejecutar "sh install.o" "sh install_x64" dependiendo si el sistema es de 32 o 64 bits respectivamente

CAENDigitizer

Esta instalación es sutilmente diferente a las dos anteriores.

Para instalar esta librería, es necesario descargarla desde la página correspondiente, ingresando a la sección Download. Para descargarla es necesario registrarse en el sitio web. El archivo a descargar corresponde al indicado para sistemas operativos linux. Una vez descargado el archivo, es necesario descomprimirlo e ingresar a la carpeta generada para leer el archivo README, en el cual se detallan los pasos de instalación. Estos pasos de instalación suelen ser los siguientes:

- Tener permisos de administrador
- Ingresar a la carpeta generada al descomprimir el archivo
- Abrir una consola en la carpeta y ejecutar "sh install.o" "sh install_x64" dependiendo si el sistema es de 32 o 64 bits respectivamente

3.6. Software

Para el correcto funcionamiento de la librería es necesario instalar el software para análisis de datos ROOT v6 o superior. La versión utilizada durante la realización de este documento fue ROOT v6.12.04 Source - Production Version.

Se recomienda además instalar software para comunicación (CAEN WaveDump) y configuración (CAENUpgrader) del digitizer, disponible en el sitio web del fabricante.

ROOT

ROOT es un framework para procesamiento y análisis de datos descrito en C++. La librería está integrada con ROOT para así lograr analizar fácilmente los datos capturados por el digitizer. El sitio web de este framework corresponde a root.cern y la guía oficial del mismo puede encontrarse en Documentation>User's Guide.

Para descargar ROOT, se debe acceder al sitio web en la sección Download>Download ROOT. Se recomienda descargar el código fuente (source) de la 'Production Version' más actual, para así compilarlo directamente en el equipo.

Antes de instalar ROOT, es necesario instalar algunas librerías y software de terceros. El detalle de estos prerequisites está disponible en <https://root.cern.ch/build-prerequisites>. Particularmente para el caso de Debia, basta ejecutar la siguiente línea en consola:

- `su -c 'apt-get install git dpkg-dev cmake g++ gcc binutils libx11-dev libxpm-dev libxft-dev libxext-dev'`

Una vez instalados los prerequisites, se puede proceder con la instalación de ROOT. Los pasos a seguir se encuentran detallados en User's Guide>28.5 Installing the Source. También se encuentra detallado el proceso de instalación en el archivo 'Install' de la carpeta 'README' incluida en la carpeta principal del archivo descargado (visible al descomprimirlo). Principalmente, los pasos son los siguientes:

- Ir a la carpeta donde se quiere dejar la carpeta de ROOT y descomprimir el archivo descargado con el comando `'tar zxvf root_vx.xx.xx.source.tar.gz'`, donde las 'x' corresponden a los números de la versión descargada.
- Crear una carpeta para construir ROOT, por ejemplo con el comando `'mkdir ROOTv6'`
- Ingresar al directorio recién creado, por ejemplo con el comando `'cd ROOTv6'`
- Ejecutar el comando `'cmake ../root_vx.xx.xx'` donde el nombre posterior al slash corresponde al nombre de la carpeta que se creó al descomprimir el archivo descargado.
- Ejecutar el comando `'cmake -build . -jN'` donde 'N' corresponde al número de núcleos del procesador que se desean utilizar para procesar los archivos.
- Abrir el archivo `.bashrc`, por ejemplo con el comando `'nano ~/.bashrc'` y agregar en la última línea la siguiente instrucción: `'source /PATH/ROOTv6/bin/thisroot.sh'`, donde 'PATH' corresponde a la ubicación en la que se encuentra la carpeta ROOTv6 recién creada.
- Probar la instalación ejecutando ROOT con el comando `'root'`

En algunos casos, existen problemas al compilar ROOT debido a que faltan prerequisites o no se encuentran. Es importante reparar estos problemas antes intentar instalarlo nuevamente.

Opcional: Software del Fabricante

CAEN dispone de software para configurar el digitizer (Configuration Tools) y para tomar mediciones con él (Readout Software). Este software está disponible en Products > Firmware/Software> Digitizer Software.

En la categoría de Configuration Tools, se recomienda instalar CAENUpgrader. Este software posee variadas opciones, como actualizar el firmware del digitizer u obtener información

sobre la versión del digitizer. Esta última función es útil para probar si las librerías y drivers del digitizer están correctamente instaladas. Es posible descargarlo en la sección 'Download' de su página y existe además un manual sobre como utilizarlo en la sección 'Documentation'.

En la categoría de Readout Software, se recomienda instalar CAEN WaveDump. Este software se utiliza por consola y permite adquirir datos, guardarlos y graficarlos de manera muy básica. Es posible descargarlo en la sección 'Download' de su página y existe además un manual sobre como utilizarlo en la sección 'Documentation'.

4. Librería CaenDigitizerSiLab

4.1. Código Fuente

El código fuente de la librería se encuentra disponible en GitHub en el siguiente link: https://github.com/orsosa/CaenDigitizerSiLab/tree/jairo_branch. Cabe destacar que en este documento se trabaja sobre la la branch 'jairo_branch'.

4.2. Dependencias

La librería depende del framework ROOT y sus librerías propias. Depende también de la librería CAENDigitizer, disponible en el sitio web del fabricante. Esta última librería depende de la librería CAENComm, la que a su vez depende de la librería CAENVMLib, también disponibles en el sitio web del fabricante.

Se da por sabido que se necesitará contar con C++, sus compiladores y todas las librerías que ello implique.

4.3. Compilación y Ejecución

Debian

Para utilizar la librería, se debe compilar el script que contiene la rutina de adquisición de datos a realizar. Para este documento, se trabajará en base al archivo 'example.cxx'.

Antes de comenzar la compilación, es necesario seguir los siguientes pasos:

- Abrir el archivo .bashrc para editarlo, por ejemplo con el comando 'nano ~ /.bashrc'
- Escribir 'export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:"PATH"', donde 'PATH' corresponde a la dirección de la carpeta en la que se encuentran los archivos.

Finalmente, para compilar el script basta ingresar a la carpeta que contiene los archivos y ejecutar por consola el comando 'make'.

Para ejecutar el script, se debe ejecutar el comando './example', donde 'example' corresponde al nombre del archivo correspondiente al script. Si se ejecuta el archivo de ejemplo, se espera que el script se conecte al digitizer, calibre el equipo, mida la temperatura de cada canal en grados celcius e intente tomar 100 eventos de 100 muestras con trigger por software. Una vez que termina, vuelve a calibrar y a tomar la temperatura.

Para salir del script, se debe presionar la combinación de teclas 'ctrl+c'.

No recomendado: Otros sistemas operativos

En caso de no contar con Debian, es posible ejecutar la librería con un script que contenga el siguiente código:

```
1 #!/bin/bash
2
3 root -l -b -e 'gSystem->AddLinkedLibs("-L. -lCAENDigitizer"), ...
    gROOT->ProcessLine(".L CaenDigitizerSiLab.cxx+"), ...
    gROOT->ProcessLine(".L main.cxx+"), gROOT->ProcessLine(".x ...
    main.cxx"), gROOT->ProcessLine(".q") '
```

Para ejecutarlo hay que copiar el código en un archivo y guardarlo con formato .sh. Luego, se debe ejecutar el comando './script.sh', donde 'script' corresponde al nombre con el que se guardó dicho archivo. Cabe destacar que es necesario nombrar como 'main()' a la función principal del archivo .cxx que se desea compilar, y a su vez este debe llamarse 'main.cxx'.

4.4. Métodos

newFile(const char* name)

Crea y abre el archivo en el que se guardarán los datos. 'name' corresponde al nombre del archivo, el cual debe terminar en '.root'. Una vez que se hayan escrito los datos en el archivo, es necesario cerrarlo con closeLastFile().

closeLastFile()

Cierra el archivo creado por newFile().

setModel(int32_t model)

Permite configurar el modelo de digitizer a utilizar (5730 o 5724). Por defecto corresponde al '5730'.

setTriggerPolarity(int32_t mode)

Configura trigger con canto de subida o de bajada. Para canto de bajada, utilizar 'mode' igual a '0'. Para canto de subida, utilizar 'mode' igual a '1'.

init()

Inicia la comunicación con el digitizer y configura los canales. Debe ejecutarse después de configurar el modelo de digitizer a utilizar y después de ejecutar TriggerPolarity().

getInfo()

Entrega informacion sobre versión de firmware del digitizer.

readEvents(int32_t events, bool automatic, int32_t start_event)

Permite adquirir los datos con el digitizer. La variable 'events' corresponde al número total de eventos por grupo medido. La variable 'automatic' indica si será con trigger por software (automático) o selftriggered (no automático). La variable 'start_event' indica el número del evento que corresponde al evento inicial del grupo a medir.

readEvents(int32_t maxEvents, bool automatic, int32_t start_event, double tm)

Permite adquirir los datos con el digitizer. La variable 'maxEvents' corresponde al número máximo de eventos por grupo medido. La variable 'automatic' indica si será con trigger por software (automático) o selftriggered (no automático). La variable 'start_event' indica el número del evento que corresponde al evento inicial del grupo a medir. La variable 'tm' indica el tiempo durante el cual se realizará la lectura. La lectura termina una vez que se alcanza el tiempo indicado en 'tm' o si se alcanza el número máximo de eventos.

storeData()

Guarda los datos adquiridos en un archivo llamado 'data_from_digitizer.root', en formato de tupla de ROOT, donde cada entrada de la tupla corresponde a una muestra.

setCoincidence(int32_t ch0)

Configura qué pares de canales participarán de la coincidencia. El parámetro 'ch0' corresponde al canal con número par, para identificar el par de canales consecutivos. Para cambiar la función utilizada para generar el trigger (AND, OR, etc), se debe modificar el método, que por defecto está configurado en AND. Para mayor información, consultar el manual de uso del digitizer.

setMajorCoincidence(int32_t blkmask, int32_t wd, int32_t level)

Permite configurar el nivel de coincidencia y los canales que participarán en el trigger general, donde 'blkmask' corresponde a los canales, 'wd' corresponde a la duración de la ventana de coincidencia en unidades de ciclos del reloj para el trigger (8ns), y 'level' corresponde al nivel de coincidencia, el cual debe ser menor a la cantidad de canales habilitados para este propósito. Para mayor información, consultar el manual de usuario y la documentación de los registros del digitizer.

setTrigmv(float mV)

Configura el threshold del trigger. Se configura al nivel entregado a la función, en milivolts. El threshold indica el nivel de señal ante el cual se toma una muestra.

readTempAll()

Lee la temperatura de cada canal y la imprime en pantalla. Solo disponible para DT5730.

setPolarizationType()

Configura el rango dinámico del ADC.

setNSamples(int32_t nsamples, int32_t trigger_size)

Configura una cantidad 'nsamples' de muestras por cada evento. Si se ejecuta pero no se especifica un 'nsamples', se configuran 100 muestras por evento. Si no se ejecuta esta función, el valor por defecto son 100 muestras por evento. Además, el parámetro 'trigger_size' indica el tamaño del post-trigger de la ventana de adquisición, expresado en porcentaje respecto al número de muestras.

calibrate()

Calibra el digitizer. Es importante calibrar antes de medir. Solo disponible para DT5730.

storeTempAll()

Guarda la temperatura de cada canal en un archivo. Solo disponible para DT5730.

getPedestal()

No efectua acción alguna aún.

startSWAcq()

Comienza la adquisición de datos. Es llamada dentro de la función readEvents().

waitTempStabilization()

Espera a que los niveles de temperatura del digitizer estén estables

getTempMeanVar()

Calcula la temperatura media y su varianza para cada canal, guardandolo en arreglos.

setTrigADC(int32_t trigthresh)

Configura un threshold 'trigthresh' para el trigger, con su valor indicado en niveles del ADC.

finish()

Termina la comunicación con el digitizer y limpia el buffer de adquisición de datos. Es importante ejecutarlo cuando se termine de trabajar con el digitizer.

4.5. Ejemplos de Prueba

Para probar la librería, se puede programar un script simple con funciones básicas, como por ejemplo `getInfo()` o `getTempAll()`. Realizar una prueba de este estilo permite comprobar que la instalación de los drivers y las librerías se realizó correctamente.

El ejemplo adjunto a continuación permite imprimir en consola la información de versión y firmware del digitizer.

```
1  #include "CaenDigitizerSiLab.h"
2  #include <signal.h>
3
4  CaenDigitizerSiLab *dig;
5
6  //Funcion para cerrar correctamente el script con 'ctrl+c'
7  void signalHandler(int)
8  {
9      dig->storeData();
10     dig->stopSWAcq();
11     delete dig;
12     bench->Show("example");
13     delete bench;
14     exit(0);
15 }
16
17
18 int example()
19 {
20     signal(SIGINT , signalHandler);
21
22     //Se crea un objeto para medir con el digitizer
23     dig = new CaenDigitizerSiLab();
24     dig->setModel(5730); //se escoge el modelo
25     dig->setTriggerPolarity(1); //se configura el trigger
26     dig->init(); // se inicializa el digitizer
27
28     //Se solicita la informacion de firmware al digitizer,
29     //para luego imprimirla en pantalla
30     dig->getInfo();
31
32     return 0;
33 }
```

4.6. Configuraciones

Como se observa en el archivo 'example.cxx' incluido en el repositorio, se incluye un archivo 'config.conf' mediante el cual se pueden configurar la mayoría de los parámetros del digitizer. La lectura de este archivo se incluye en la primera sentencia 'if' descrita en el archivo de ejemplo. Es posible prescindir de este archivo y configurar manualmente los parámetros deseados.

A continuación se lista como configurar algunos parámetros básicos para adquirir datos utilizando la librería.

4.6.1. Número de Muestras por Evento

Para modificar el número de muestras por evento, es necesario utilizar el método `setN-Samples(int32_t nsamples, int32_t trigger_size)`, donde 'nsamples' corresponde al número de muestras deseadas por evento y 'trigger_size' indica el tamaño del post-trigger de la ventana de adquisición, expresado en porcentaje respecto al número de muestras.

4.6.2. Número de eventos por medición

Para modificar el número de eventos por medición, se puede setear una variable llamada 'events' de tipo entero, luego asignarle el valor de eventos deseados y finalmente entregarla al método `readEvents()` en su primer parámetro.

4.6.3. Número de grupos de eventos por medición

Es posible tomar varios grupos de eventos. Para esto, es posible declarar una variable llamada 'NBunch' y asignarle el número de grupos de eventos que se desean. Luego, es necesario realizar un ciclo 'for' para realizar la lectura y guardar los datos, una vez por cada NBunch. Esto implica que el tercer parámetro del método `readEvents()` corresponderá al número del grupo actual (índice del ciclo 'for') multiplicado por el número de grupos de eventos.

4.6.4. Polarización, Rango Dinámico y Offset

El digitizer CAEN DT5730 posee un ADC de 14 bits de resolución, al cual se le puede configurar mediante software el valor máximo de voltaje de las señales adquiridas (2Vpp o 0.5Vpp). Además, es posible ajustar dónde se encuentra el cero en los niveles del ADC mediante un offset con un DAC de 16 bits.

La librería se encuentra actualmente diseñada para funcionar con el valor máximo de voltaje de 2Vpp para digitizers DT5730 y no se encuentra implementada la opción para cambiar a 0.5Vpp. De igual forma, para el caso de digitizer DT5724, se encuentra diseñada para

funcionar con el valor máximo de voltaje de 2.25Vpp.

Por defecto, para digitizers DT5730, la librería está configurada para medir señales entre -1V y 1V (`kPolarizationType = 0`, bipolar), es decir, con un offset que sitúa al 0V del ADC a la mitad del rango. Para cambiar a un rango de 0V a 2V (unipolar positivo), se llama al método `setPolarizationType(1)`, lo cual modificará la polarización y ajustará el cero del ADC en el nivel más bajo. Para ajustar un rango desde -2v a 0V (unipolar negativo), se llama al método `setPolarizationType(-1)`, lo cual modificará la polarización y ajustará el cero del ADC en el nivel más alto.

Importante: Hay que tener en consideración que al configurar polarización 1 y -1 no será posible medir los valores extremos 2V o -2V respectivamente, ya que se configuró de manera intencional un offset tal que la referencia del sistema (0V) sea captada de manera correcta. Configurar un rango muy acotado (por ejemplo 0V a 2V exactos) podría causar que la referencia no sea muestreada si sus niveles reales cambian levemente de manera que se sitúen sutilmente fuera del rango. También es importante considerar que existe un error del orden del 5 % en la configuración de offset para valores extremos, además de existir un offset de voltaje de entrada arbitrario en cada canal, por lo que puede ser necesario modificar los niveles de la señal para que se vean reflejados los valores reales específicamente para los casos de polarización unipolar positiva y negativa.

Para mayor información, consultar el manual del usuario del digitizer en la sección 'Functional Description, Analog Input Stage', consultar la documentación de los registros del digitizer en la sección "DC Offset" (disponible en la página del digitizer), y consultar la documentación de la librería CAENDigitizer en la referencia del método 'Set/GetChannelDCOffset', disponible en la página de la librería.

4.6.5. Trigger

Al momento de realizar mediciones con el método `readEvents()`, se puede configurar su segundo parámetro con el valor 'true' para realizar trigger automático por software, o 'false' para realizar el trigger cuando la señal supere un threshold arbitrario configurable.

Para configurar el threshold, se debe llamar al método `setTrigmV(float mv)`, donde 'mV' corresponde al nivel de voltaje en milivolts para el cual se realizará el trigger. Cabe mencionar que este nivel se adapta automáticamente a la polarización escogida para el ADC, por lo que solo hay que tener en mente los valores reales de la señal que se quiere medir (asumiendo que el rango y polarización son los correspondientes).

Para modificar si el trigger se ejecuta en canto de subida o de bajada según el threshold configurado, es necesario ejecutar el método `setTriggerPolarity()`, donde '1' corresponde a detección por canto de subida y '0' corresponde a detección por canto de bajada.

Por último, es posible configurar cuáles canales gatillarán el trigger. Para esto, es necesario modificar el método `readEvents()` del archivo `'CaenDigitizerSiLab.cxx'`, buscando la función `'CAEN_DGTZ_SetChannelSelfTrigger()'`, en la cual su tercer parámetro indica en binario los canales que activarán el trigger, donde el LSB corresponde al canal cero. Para más información, consultar la documentación de la librería CAENDigitizer y el manual de usuario.

5. Análisis de Datos

Para trabajar con los datos adquiridos por el digitizer, se debe ejecutar root en el directorio donde se encuentre el archivo generado por el script, cuyo nombre corresponde a 'data_from_digitizer.root'. Este archivo contiene una o más tuplas con los datos muestreados, donde cada entrada corresponde a una muestra. La tupla tiene una columna para cada canal, una para el tiempo de cada evento (de 1 hasta 'n', donde 'n' corresponde al total de muestras por evento), una para el número de evento, y otra para el identificador de cada fila (muestras desde 0 hasta 'n'*'m', donde además 'm' representa el número total de eventos).

Por ejemplo, para graficar el primer evento captado por el canal cero, se pueden ejecutar los siguientes comandos:

```
1 root
2 file = new TFile("data_from_digitizer.root")
3 data->Draw("Ch0:time", "event==0", "*l")
```

Finalmente, para cambiar la manera en que se guardan los datos se debe modificar el metodo init(), además de adaptar los métodos de lectura, escritura y referencias en cualquier otro método de ser necesario. Si se prefiere no modificar la librería, existe la opción de crear un script que recorra la tupla y la guarde en otra estructura de datos, como por ejemplo guardándola en un árbol común.

6. Sitios y Documentos de Interés

Debian

- Sitio web: debian.org
- Versión utilizada en este documento: <https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/>
- Cómo flashear memoria usb con imagen .iso: <https://www.debian.org/releases/stable/amd64/ch04s03.html.en>
- Manual de Instalación: <https://www.debian.org/releases/stable/installmanual>

CAEN Digitizer DT5730

- Sitio web: caen.it
- CAEN DT5730 Site: <http://www.caen.it/csite/CaenProd.jsp?parent=14&idmod=757>
- First Installation Guide to Desktop Digitizers and MCA <http://www.caen.it/csite/CaenProd.jsp?showLicence=false&parent=14&idmod=757>
- CAEN DT5730 DT5725 User Manual: <http://www.caen.it/servlet/checkCaenManualFile?Id=12386>
- Digitizer Software: <http://www.caen.it/csite/Product.jsp?parent=38&Type=Product>
- Software Libraries: <http://www.caen.it/jsp/Template2/Function.jsp?parent=43&idfuns=99>
- CAENDigitizer Library Documentation: <http://www.caen.it/servlet/checkCaenManualFile?Id=12638>
- CAEN DT5730 Registers: <http://www.caen.it/servlet/checkCaenManualFile?Id=12730>

ROOT

- Sitio web: root.cern
- User's Guide: <https://root.cern.ch/root/html/doc/guides/users-guide/ROOTUsersGuide.html>
- Download ROOT: <https://root.cern/download-root>