

Capítulo 3

Experimentos y sistemas de adquisición de datos en física de partículas

En este capítulo se presentan los conceptos de experimento y de sistema de adquisición de datos dentro del contexto de la física experimental. Estas nociones son fundamentales en la comprensión de la problemática y en el modelaje de la solución.

3.1. Experimentos en física de partículas

Un experimento de física de partículas está conformado por el equipo experimental y el sistema de adquisición de datos (DAQ). El equipo experimental es el conjunto de dispositivos que conforman el experimento; en un experimento se generan señales análogas (pulsos) que corresponden a interacciones, por ejemplo la interacción de una partícula en un detector produce un pulso eléctrico. El DAQ permite digitalizar y almacenar las señales producidas en el experimento. Estos datos son el material de análisis del experimento para el físico, ya que mediante el estudio y procesamiento de esta información puede inferir patrones, fenómenos, entre otros.

Para el objeto de este trabajo se utiliza el estándar CAMAC para la implementación del DAQ dado que esta es la instrumentación para la cual se planteó el desarrollo de la aplicación.

3.2. Sistema de adquisición de datos

Un sistema de adquisición de datos es una configuración física y lógica que permite realizar captura de información de un experimento. Entiéndase por configuración física (*DAQ hardware*) al conjunto de dispositivos electrónicos que permiten la conversión de señales análogas en datos digitales para ser almacenados; CAMAC es una tecnología que provee esta funcionalidad. Por configuración lógica (*DAQ software*), al sistema que realiza el control del *DAQ hardware* y el transporte de los datos al computador para ser almacenados y posteriormente depurados.

3.2.1. DAQ hardware

CAMAC (*Computer Automated Measurement and Control*) es un estándar internacional de electrónica modular definido por el comité ESONE (*European Studies on Norms for Electronics*). CAMAC proporciona un esquema que permite la utilización de una amplia gama de instrumentos modulares (módulos) en un sólo bus de transmisión de datos, denominado *DATAWAY*. A través de un módulo se realiza el control y transmisión de las instrucciones y datos en el bus; también mediante una interface (ISA, PCI, SCSI) el módulo permite la conexión a un computador. A este dispositivo se le denomina *módulo controlador*. De esta manera CAMAC permite transferir información dentro y fuera de los módulos.

Un contenedor (*crate*) CAMAC, posee 25 estaciones (*Slots*) donde la estación 25 es de uso exclusivo del módulo controlador que se encarga del control y transmisión de la información entre los módulos y la interface del computador. Pueden configurarse varios contenedores CAMAC y de esta manera ampliar el número de estaciones disponibles [5].

Un módulo CAMAC es un dispositivo que convierte una señal análoga a un valor digital de acuerdo al tipo de medición que se quiera realizar. Los módulos más utilizados son:

- **TDC**: *Time to Digital Converter*, realiza mediciones de tiempo.
- **QDC**: *Charge to Digital Converter*, realiza mediciones de carga.
- **ADC**: *Amplitude to Digital Converter*, realiza mediciones de la amplitud de la señal de entrada.
- **SCALER**: Realiza la función de un contador de acuerdo a un intervalo de tiempo determinado.

3.2.2. DAQ software.

Es la integración de cuatro elementos:

- **Sistema operativo:** Administra el computador y sus componentes.
- **Driver:** Permite la interacción entre el computador y el DAQ hardware.
- **Software de administración y almacenamiento:** Maneja la información asociada a uno o varios experimentos; estructurar y compactar la información para ser almacenada.
- **Software de análisis y visualización:** Permite realizar el estudio correspondiente de la información. Cuenta con funciones matemáticas generales y específicas para la generación de gráficos, análisis numérico, entre otras tareas.

El *DAQ software* es empleado para realizar pruebas de equipos electrónicos, comparando la respuesta que captura el *DAQ* frente algún criterio particular y de esta manera seleccionar los dispositivos más adecuados para el montaje del experimento.

3.3. Estructura general de un experimento.

De acuerdo a los conceptos presentados y resumiendo, un experimento puede estructurarse de la siguiente manera (figura 3.1):

- **Configuración experimento:** Es la instrumentación que provee o genera un evento físico particular, ejemplo: un acelerador de partículas.
- **Detectores:** Son dispositivos que generan una señal (pulso) electrónica debido a una interacción.
- **Electrónica complementaria:** Son componentes adicionales que permiten amplificar, filtrar, limpiar, copiar, etc. las señales que provienen de los detectores, ejemplo: NIM (*Standar Nuclear Instrument Modules*)
- **CAMAC:** Es una interface de instrumentación modular. Digitaliza las señales que provienen de la electrónica complementaria, mediante los módulos en el *crate*, es decir conectados al *DATAWAY*. El módulo

controlador hace un barrido a través del bus de los módulos instalados, enviando al programa los datos en cada uno o del módulo especificado por el programa.

- **Software:** Es el encargado del control de lectura y transferencia de los datos del experimento a un medio de almacenamiento. Mediante el módulo controlador realiza la transferencia de los datos de los módulos al computador, almacenando los datos correspondientes. Permite acceso a la información mientras se realiza el muestreo del experimento (*online*) o cuando éste ya ha concluido y toda la información se encuentra almacenada(*offline*).

3.4. Captura de datos en un experimento y sistematización de la información

El intervalo de tiempo en el que se realiza la captura de datos de un experimento se denomina *ejecución*. Durante la ejecución se capturan y almacenan *eventos*. Un evento es el estado particular de los registros de los módulos. No todos los eventos son de interés para el experimentalista; éste configura mediante una señal que el evento es válido y debe ser almacenado, a este tipo de mensaje se le denomina *trigger*. Cada módulo tiene la capacidad de generar este tipo de mensajes en el bus informando al controlador, por ejemplo, que ha recibido una nueva señal. Dado este mensaje el módulo controlador genera un mensaje (LAM, *Look At Me*) informando al computador (programa) que en los módulos hay datos. En otras palabras, un *trigger* dispara el LAM para que el evento sea almacenado. El programa recibe el mensaje LAM y realiza la lectura mediante el módulo controlador, almacenando los datos correspondientes.

Durante la ejecución es necesario realizar un análisis de los datos que están siendo capturados (*Análisis Online*) con el propósito de supervisar el experimento; para ésta tarea se despliega en la interfaz del usuario un gráfico (i.e. histograma), una tabla o cualquier objeto.

Toda ejecución tiene un archivo asociado que contiene los datos recolectados, además un archivo histórico (*Bitácora*), en el cual se registra información de interés alrededor del experimento como: la configuración electrónica empleada, ejecuciones y los archivos asociados; Monitores (Gráficas, tablas, etc.); Notas de usuario, entre otros.

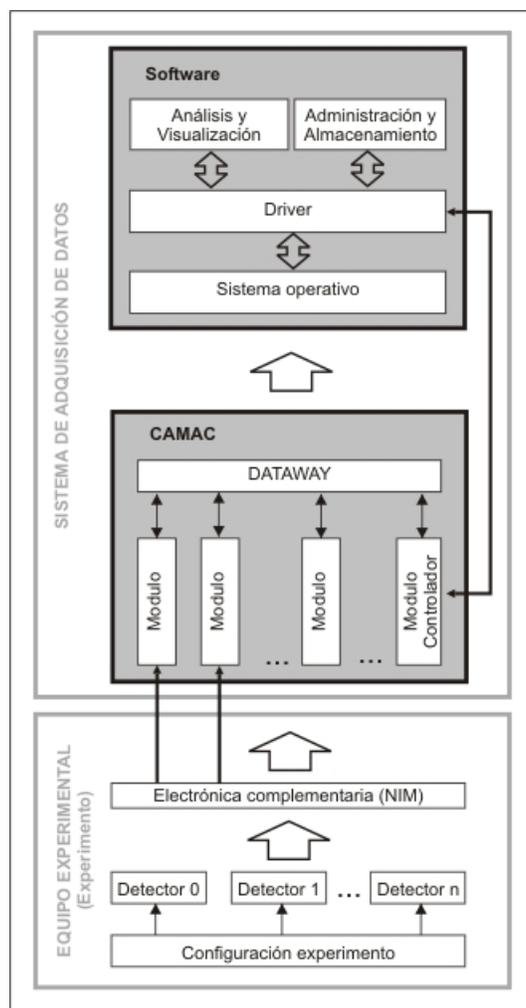


Figura 3.1: Estructura experimento

Capítulo 4

Estándar CAMAC

A continuación se resumen las definiciones y conceptos más importantes del estándar CAMAC que especifica la interacción con esta interface de instrumentación modular. La información de este capítulo es condensada y traducida de la sección 1.8, Communication lines, bus systems; de la referencia [1] y de la página de la referencia [14].

4.1. Introducción

CAMAC fue diseñado antes de 1970 y en la actualidad aun es usado en el área de física experimental. Desde su especificación son pocos los cambios que se le han realizado debido a la estabilidad que ha presentado por más de 30 años[3].

En Europa la especificación se encuentra dada por *CAMAC, A modular Instrumentation System For Data Handling*, EUR 4100 e, Marzo 1969. Y en América por *CAMAC standar* ANSI/IEEE 758-1979, *Modular Instrumentation and Interface Standards (CAMAC)*, ANSI/IEEE Std 583-1982.

CAMAC corresponde junto a otras tecnologías (como VME, MULTIBUS II, FASTBUS) a una arquitectura de instrumentación modular electrónica sobre un sólo bus de transmisión. Esto quiere decir que en lugar de realizar conexiones punto a punto con cada uno de los dispositivos de adquisición de datos (ADCs, TDCs, etc.) estos se encuentran conectados a un canal de líneas comunicación o bus. Este bus está compuesto por líneas de control, direccionamiento, transmisión de datos y strobe.

Los módulos como los TDCs, QDc, Scalers, etc. Son denominados módulos esclavos mientras los módulos controladores son denominados módulos maestros. El módulo maestro es el que permite la interacción con un com-

putador para transferir la información de los módulos a un dispositivo de almacenamiento.

4.2. Protocolo de bus

El conjunto de reglas para la transferencia de información entre el módulo maestro y el módulo esclavo se denomina protocolo de bus. La figura 4.1 ilustra la información que se intercambia entre maestro y esclavo.

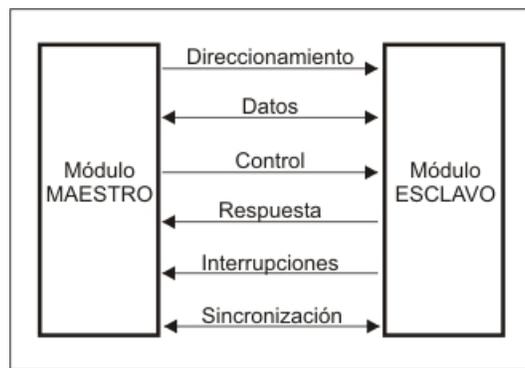


Figura 4.1: Protocolo de bus

- **Direccionamiento:** Es la manera como el módulo maestro localiza un módulo esclavo en el bus. CAMAC emplea un direccionamiento geográfico, es decir, la dirección de un módulo corresponde a la ubicación dentro del bus, número del *slot* en el *crate* (0-24).
- **Datos:** Es la transferencia de datos desde o hacia el módulo esclavo. Mediante el bloque de transferencia de datos, se transmiten los datos de los registros del esclavo hacia el computador.
- **Control:** Define la operación. CAMAC permite leer, leer y limpiar, leer el complemento, leer interrupción, limpiar interrupción, entre otras.
- **Respuesta:** Es un mensaje que emite el esclavo informando al maestro: operación exitosa, operación ilegal, no hay datos, entre otros.
- **Interrupciones:** Son mensajes que emite el esclavo ante hecho particular, ejemplo: se activo el *trigger*, los datos deben ser leídos; error en el bus, verificar; la transferencia del bus ha terminado; etc.

- **Sincronización:** Un ciclo es la secuencia de eventos de una operación sobre el bus. Los módulos realizan cualquier operación dentro de un intervalo de tiempo constante definido con una señal de reloj en el bus. En consecuencia cada ciclo se ejecuta en el mismo intervalo de tiempo. Esto permite la sincronización entre maestro y esclavo. Razón por la cual CAMAC es sincrónico.

En CAMAC una operación en el *DATAWAY* es una orden que emite el módulo controlador que incluye el número de la estación (N), una subdirección (UN) y un código de función (F). En la respuesta, el módulo generará una respuesta de aceptación (Respuesta X); Si el comando requiere transferencia de datos, se utilizarán las líneas de lectura (R) o escritura (W) del bus. Note que la acción de leer o escribir la realiza el módulo controlador, no los módulos esclavos [14].

4.3. Características generales de CAMAC

- Un *crate* tiene 25 *slots*. Pueden configurarse varios *crates* para ampliar el número de módulos disponibles.
- El tamaño del bloque de transmisión del bus es de 24 bits.
- Velocidad de transferencia es de 3 Mbytes por segundo.
- El módulo controlador tiene dominio sobre los módulos de su mismo *crate*
- Direccionamiento geográfico. Cada *slot* tiene 16 subdirecciones.
- 32 funciones definidas para el módulo controlador, ejemplo F0 significa leer datos de un esclavo; F16 significa escribir datos en esclavo.
- Cada módulo esclavo emite una respuesta Q que indica el estado del módulo.
- Interrupciones (LAM). Cada módulo puede emitir un mensaje LAM y este puede verificarse a través de la función F8.

4.4. Instrucciones CAMAC

Una instrucción es un conjunto de señales sobre las líneas del *DATAWAY* en donde se especifica la subdirección del módulo o de los módulos (a través

de las líneas de direccionamiento del bus) y la función a ser realizada (a través de las líneas de función del bus) . Las señales se mantienen sobre el `DATAWAY` en el tiempo de duración de la operación. Estas están acompañadas por una señal sobre la línea de disponibilidad (*Busy line*) indicando a todos los dispositivos que hay una operación en ejecución.

Para una instrucción se define:

- **Número de estación (N)**: Corresponde al número del *slot* en el *crate*. La numeración se realiza de izquierda a derecha, en decimal comenzando por la estación 1.
- **Subdirección (A0 - A15)**: Es una referencia en el módulo, por ejemplo, un canal de un QDC. Cada módulo cuenta con 16 subdirecciones numeradas en decimal de 0 a 15.
- **Función (F0 - F31)**: Indica la operación a realizar en la subdirección A del módulo N: leer, escribir, entre otras. Se definen 32 funciones numeradas en decimal desde 0 hasta 31: lectura(F0-F7), control(F8-F15) y (F24-F31), escritura(F16 - F23).
- **Señal Strobe (S1 y S2)**: 2 señales *strobe* son generadas en secuencia o por líneas separadas del bus. Estas señales se emplean para transferir la información entre *plug-in* de las unidades vía el *DATAWAY* o iniciar el funcionamiento en las unidades.

Datos:

Hasta 24 bits de datos pueden ser transferidos en paralelo entre el módulo controlador y un módulo determinado. Se proveen 2 líneas independientes (lectura y escritura) conservando las 2 direcciones de transferencia.

- **Escritura (W1 - W24)**: El controlador u otra fuente de datos genera señales sobre la línea W del bus al inicio de cualquier operación de escritura. Las señales W señalan el alcance de un estado estable ante S1 y se mantiene a menos que sea modificado por S2.
- **Lectura (R1 - R24)**: Estas señales son establecidas en las líneas R del bus cuando el modulo reconoce una instrucción de lectura. Las señales W señalan el alcance de un estado estable ante S1 y se mantiene a menos que la fuente de datos sea cambiada por S2.

Información de estado:

La información de estado está dada por las señales LAM (L), disponibilidad (Busy B), Comando aceptado (X) y respuesta (Q).

- **Look-At-Me (L):** Mientras no exista una operación en proceso (B) cualquier modulo puede general esta señal con el propósito de comunicar que algo ha ocurrido. Mediante la instrucción *clear LAM*, se restablece el estado del módulo.
- **Busy (B):** Esta señal se emplea como mecanismo de control entre operaciones (semaforo).
- **Comando aceptado (X):** Es un mensaje que informa que la instrucción fue aceptada por el módulo ($X=1$).
- **Respuesta (Q):** Indica el estado de un atributo seleccionado de un módulo.

Control:

Las señales de control comunes operan en todos los módulos conectados a ellos sin la necesidad de direccionarse separadamente por una instrucción. Para proporcionar protección contra señales ilegítimas las señales Inicializar (Z) y Limpiar (*Clear C*) deben estar acompañadas por S2.

- **Inicializar (Z):** Esta señal tiene prioridad absoluta por encima de otras señales. Pone todas las unidades en estado básico restableciendo todos los registros. Las unidades que generan señales Z también deben general S2 y B para poder lanzar la señal.
- **Inhibit (I):** La presencia de esta señal bloquea cualquier actividad, ejemplo la lectura de datos.
- **Clear (C):** Esta señal limpia el valor de los registros. Las unidades que generan señales C también deben general S2 y B para poder lanzar la señal.

Bibliografía

Libros

- [1] R. Frühwirth, M. Regler, R. K. Bock, H. Grote, D. Notz. *Data Analysis Techniques for High-Energy Physics*. Cambridge monographs on particle physics, nuclear physics and cosmology, Segunda edición. 2000.
- [2] *Proceedings of the 2nd International Data Acquisition Workshop on Networked Data Acquisition Systems*. World Scientific Publishing Co. Pte. Ltd. 1997.
- [3] S. Dhawan, C. Hubbard, T. Radway, R. Sumner. *An introduction to FASTCAMAC (60 Megabytes/sec in CAMAC)* from Proceedings of the 2nd International Data Acquisition Workshop on Networked Data Acquisition Systems, Poster Session. World Scientific Publishing Co. Pte. Ltd. 1997.
- [4] *Instrumentation in Elementary Particle Physics: The VII ICFA School*. The American Institute of Physics, 1998.

Artículos y publicaciones

- [5] J. Toledo, F.J. Mora, H. Müller. *Past, present and future of data acquisition systems in high energy physics experiments*. *Microprocessors and Microsystems* 27 (2003) 353-358.
- [6] R.V. Rivas. *A camac data acquisition system based on PC-Linux*. *Nuclear Instruments and Methods in Physics Research A*. 483 (2002) 830-832.
- [7] E. Leonardi, G. Organtini. *A PC/Linux-based Data Acquisition System*.

- [8] P. Kodys, J. Broz, Z. Dolezal. *Small Program For Fast Data Acquisition Of CAMAC and Other Devices On PCI ISA bus*. Intitute of Particle and Nuclear Physics Charles University, Faculty of Mathematics and Physics.
- [9] T. Sefzik, M. Drochner, P. Wüstnet, K. Zvoll. *A DataAcquisition System for Small Experiments*.

Manuales y guías de usuario

- [10] LUCID Data Acquisition and Analysis System. *User's Guide*. Ver. 2.0
- [11] *The ROOT Users Guide 4.08*. Julio de 2004.
- [12] *Qt 3.3 Whitepaper*.
- [13] *The Go4 Analysis Framework Introduction V2.8* Septiembre de 2004.

Enlaces Internet

Estándar CAMAC

- [14] *An introduction to CAMAC*. [OnLine]. Disponible:
<http://www-esd.fnal.gov/esd/catalog/intro/introcam.htm>
- [15] *FermiTools* [OnLine]. Disponible:
<http://fermitools.fnal.gov/abstracts/camac/abstract.html>
- [16] *IEEE - Nuclear Engineering Standards* [OnLine]. Disponible:
<http://standards.ieee.org/catalog/olis/nuclear.html>
- [17] *CAMAC (IEEE-583) Hardware Functions* [OnLine]. Disponible:
http://www.certif.com/spec_manual/ref_2_4_5_2.html

Especificaciones de hardware

- [18] *PCI to Ultra SCSI Controller* [OnLine]. Disponible:
http://www.lsilogic.com/products/scsi_ics/lsi53c860e.html
- [19] *Product Summary - Model 73A SCSI Crate Controller* [OnLine].
Disponible:
<http://www.jorway.com/prod01.htm>

Herramientas de desarrollo

- [20] *ROOT, An Object-Oriented Data Analysis Framework*. [OnLine].
Disponible:
<http://root.cern.ch/>
- [21] *Qt-Designer* [OnLine]. Disponible:
<http://www.trolltech.com/products/qt/>
- [22] *Qt interface to ROOT*. [OnLine]. Disponible:
<http://www-linux.gsi.de/~go4/qtroot/html/qtroot.html>
- [23] *Using the Qt3.0 designer to create ROOT applications*. [OnLine].
Disponible:
http://ssz.by.ru/programming/cpp/qt_unix/qt_2.htm

Sistemas de adquisición de datos

- [24] *MARaBOU, A MBS and ROOT Based Online/Offline Utility*. [On-Line]. Disponible:
<http://www.bl.physik.tu-muenchen.de/marabou/html/>
- [25] *Go4, GSI Object Oriented On-line Off-line system*. [OnLine].
Disponible:
<http://www-w2k.gsi.de/go4/>

Implementaciones C++

- [26] *A CAMAC device driver for Linux* [OnLine]. Disponible:
<http://www-zeuthen.desy.de/~ole/camac/camac.html>
- [27] *HADES C++ classes* [OnLine]. Disponible:
<http://www.d4m.de/wissenschaft/classes/>
- [28] *The complete list of C++ classes for the ETA project* [OnLine].
Disponible:
http://www.triumf.ca/project_ETA/complete_class_list.html
- [29] *C/C++ Libraries* [OnLine]. Disponible:
<http://www.physik.rwth-aachen.de/group/IIIphys/CMS/tracker/en/software/clibraries.html>
- [30] *HEP Group, DAQ-related documents* [OnLine]. Disponible:
<http://www.hep.phy.cam.ac.uk/computing/daqdocs.html>

[31] *Experimental Particle Physics* [OnLine]. Disponible:
<http://hepwww.ph.qmw.ac.uk/index.html>