

AUTOMATIZACIÓN DEL POSICIONAMIENTO DE UN DETECTOR Y ADQUISICIÓN DE DATOS PARA UN DIFRACTÓMETRO DE RAYOS X

N. Martínez , J. Runco , A. Ghisolfi

Dto. de Física, Universidad Nacional de La Plata. CICpBA, CONICET.

CC 67 1900 La Plata Argentina.

e-mail: electro@venus.fisica.unlp.edu.ar

El presente trabajo describe la implementación del control y automatización del posicionamiento del detector y adquisición de datos, de un difractómetro de rayos X.

El sistema forma parte del equipamiento de la cámara de reacción de un acelerador de iones de 400 KV que posee el Dto. de Física de la UNLP, y está montado de manera de poder hacer difracción sobre muestras que están simultáneamente siendo tratadas con el acelerador. La instalación permite además hacer análisis de rayos X en función de la temperatura por lo que el sistema de adquisición deberá ser capaz de controlar y registrar estas variables.

El sistema ha sido desarrollado a partir de una computadora tipo PC que realiza las funciones de adquisición y control, y soporta la correspondiente interfaz.

I. DESCRIPCIÓN GENERAL

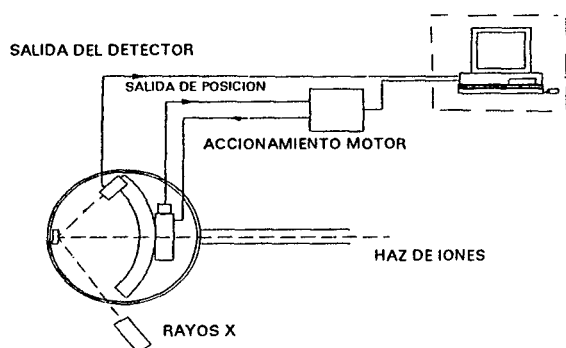


figura 1

La figura 1 muestra el esquema general del sistema. La cámara de reacción se encuentra en vacío y el generador de rayos X es externo a la cámara, conectado a la misma a través de una ventana de berilio. El goniómetro está montado dentro de la cámara y es movido por un motor paso a paso, la posición es monitoreada a través de un encoder angular (sensor de posición).

La geometría determinada entre otras cosas por el soporte del material a irradiar, limita el ángulo de barrido aproximadamente a 270° .

El detector, montado sobre el goniómetro, es un tubo fotomultiplicador, con su correspondiente fuente de alta tensión y preamplificador ubicados fuera de la cámara.

La interfaz desarrollada va conectada dentro de la PC a un slot standard EISA, (imagen recuadrada en la figura 1).

II. DESCRIPCIÓN DE LA INTERFAZ

Con el fin de automatizar la medida las distintas funciones a realizar son:

- Posicionar el detector: con todas sus variantes como por ejemplo llevarlo a una posición determinada al iniciar la experiencia, avanzar de una posición a otra en "pasos" y como consecuencia de lo mencionado determinar la posición del detector, con una resolución dada por el sensor de 5 centésimas de grado.

- Coleccionar datos: implica contar los pulsos recibidos del detector. Considerando una frecuencia del orden de 30.000 cuentas por segundo y un tiempo de adquisición por posición de entre 0.1 seg. y 100 seg., la implementación del sistema de conteo no puede ser implementado por software y deberán ser utilizados contadores externos manejados por el programa de aplicación.

- Medir el tiempo de adquisición en una determinada posición, o hacer un barrido continuo adquiriendo datos. Para ello utilizamos uno de los canales del clock de la PC que produce una interrupción a la CPU. El intervalo de tiempo entre dos de esta interrupciones puede ser fijado escribiendo en un registro de 16 bits que actúa como divisor del clock. Dado que el registro es de 16 bits, el divisor más grande es 65535 y como la frecuencia del clock es 1,193180 MHz, el mínimo número de veces es de 18,2 veces por segundo. Analizando estas interrupciones, es posible medir el tiempo transcurrido. Es importante considerar que las características del experimento realizado permiten utilizar el hardware disponible sin necesidad de implementar un clock externo. El reloj utilizado es independiente de la velocidad del procesador y toda PC cuenta con un reloj de estas características.

En caso de realizar en la experiencia una adquisición continua, la máxima velocidad de la misma estará limitada por el tiempo de lectura de los contadores, como

habíamos mencionado anteriormente 24 bits, por tanto este tiempo estará limitado por $3T_{reloj} + 6T_{reloj}$ del procesador, es decir una lectura de 16 bits y otra de 8 bits, siendo este tiempo dependiente de la velocidad de la máquina.

- Almacenamiento de los datos para su posterior procesamiento: almacenarlos en memoria, salvar los espectros en disco, enviar los datos a una impresora y graficarlos en pantalla. Para cumplimentar estas funciones la solución propuesta fue la siguiente: implementar la interfaz sobre un sistema de cómputo tipo PC.

El diagrama en bloques de la interfaz se muestra en la figura 2.

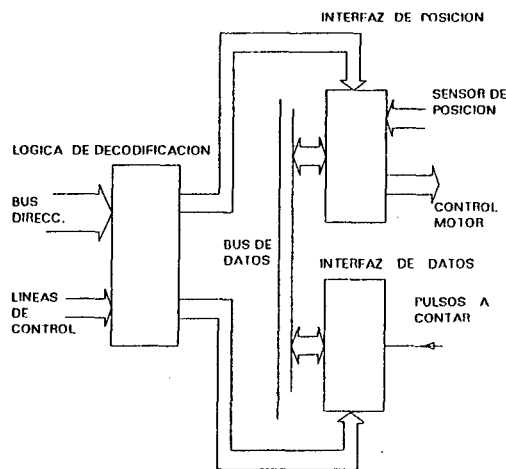


figura 2.

El diagrama muestra tres bloques fundamentales:

- Interfaz de posición: consta de los circuitos necesarios para leer las señales del sensor de posición y proporcionar la adecuada excitación del motor.

- Interfaz de datos: circuitos para contar los pulsos recibidos.

- El bloque de lógica de decodificación permite seleccionar adecuadamente cada uno de estos dos bloques dentro del mapa de I/O de la PC. Se utilizan además las líneas de control IOR e IOW que se activan con una operación de lectura o escritura en una posición de I/O, y a MEMR y MEMW que se activan con una operación de lectura y escritura en una posición de memoria.

La interfaz ocupa 8 posiciones de 8 bits cada una de I/O, dentro del mapa de la PC. El número de posiciones ocupadas están dadas por las necesidades para cumplir las funciones mencionadas anteriormente:

16 bits para obtener la posición del detector, con una resolución de 5 centésimas de grado, (2 posiciones).

4 bits para el control del motor (1 posición).

24 bits para obtener el estado de los contadores (3 posiciones).

Reset de los contadores (1 posición).

Habilitación de los pulsos a contar (arranque y parada: 1 posición).

Las direcciones de I/O utilizadas por la interfaz son las siguientes:

- 300 h posición parte baja
- 301 h posición parte alta
- 302 h excitación motor
- 303 h contadores parte baja
- 304 h contadores parte media
- 305 h contadores parte alta
- 306 h reset de los contadores
- 307 h habilitación de los pulsos a contar

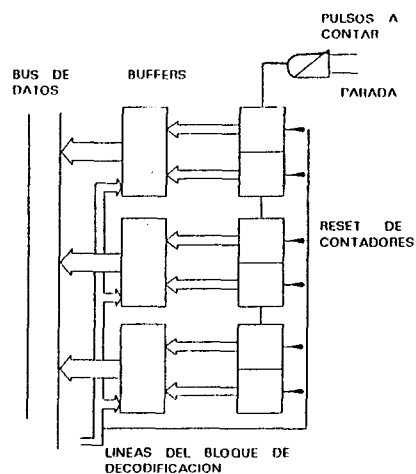


figura 3.

La interfaz de datos como se muestra en la figura 3., está formada por los contadores que determinan la cantidad de pulsos recibidos para una dada posición. A través de los buffers estos contadores están conectados al bus de datos de la PC, además, a los mismos llegan líneas de control para detener el conteo y ponerlos a cero.

Por último, el funcionamiento del programa de aplicación sería el siguiente: al comenzar la experiencia el detector es llevado a una posición inicial que corresponde a uno de los extremos del goniómetro. Una vez situado en dicha posición se ingresan los parámetros de medida:

Posición inicial (distinta de la anterior) y posición final de la experiencia.

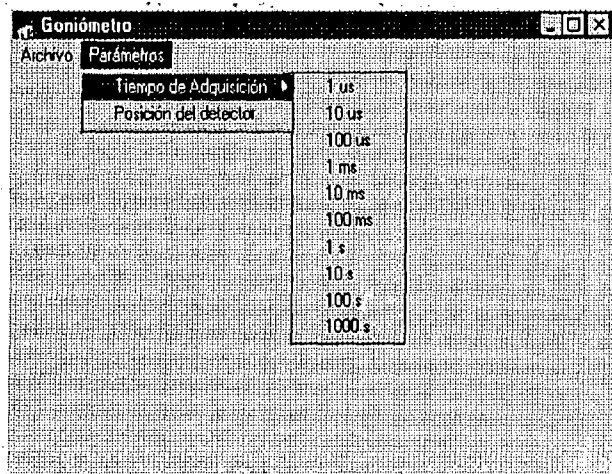
Avance entre estas dos posiciones: grado, décima de grado, centésimas de grado.

Tiempo de adquisición desde 0.01 seg. hasta 1000 seg.

Así comenzada la experiencia, se repite una secuencia de operaciones: posicionar al detector/ adquirir datos/ almacenarlos en memoria.

El número de cuentas de cada posición puede ser de hasta 2^{24} correspondiendo a cada una de ellas tres bytes en memoria, así el dato queda almacenado en tres números: parte alta, parte media y parte baja, luego pueden ser salvados en disco, graficados en pantalla y enviarlos a una impresora.

La siguiente figura muestra la interfaz gráfica que permite al usuario interactuar con el equipo



III. CONCLUSIONES

Se implementó un sistema de control y adquisición automático, que nos permite posicionar al detector con una resolución de 5 centésimas de grado de grado, en un margen de 270° limitado por la geometría del soporte de la muestra.

El tiempo de adquisición en cada posición puede seleccionarse desde 0.01 seg. hasta 1000 seg.

El número de cuentas de cada posición puede ser de hasta 2^{24} , correspondiendo a cada una de ellas 3 bytes de memoria como lugar de almacenamiento.

El sistema así implementado, se adapta para ser utilizado además, para una experiencia de correlación con otras variables como por ejemplo el seguimiento automático de pico de un difractograma para variaciones de temperatura de la muestra.

REFERENCIAS:

- 1- L. Eggebrecht, Interfacing to IBM Personal Computer, Sams, 1991.
- 2- R. Jourdain, Programmer's Problem Solver, Brady Publishing, 1992.
- 3- R. Simon, Windows 95 Win32 Programming API Bible, Waite Group Press, 1996.
- 4- F. Charte, Delphi. Programación Avanzada en Delphi 1.0 y Delphi 2.0, Anaya, 1996.
- 5- Microsoft Windows 95 Resource Kit, Microsoft Press, 1995.