



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

Diseño y programación de un sistema de control para un tubo de rayos-x con monitoreo de presión y temperatura por computadora

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO Y ELECTRÓNICO

P R E S E N T A:

MARIO ALBERTO CORONA CADENAS



DIRECTOR DE TESIS:

DRA. GUERDA MASSILLON J-L

CIUDAD UNIVERSITARIA 2014

Agradecimientos

Este trabajo y toda la carrera en general son parte de un esfuerzo que fue sustentado por muchas personas.

Primero que nada, le agradezco a Dios por haberme dejado concluir esta fase de mi vida. Sin Él, no hubiera podido concluir. Parece fácil, y tal vez concluir esta etapa fue fácil para muchas personas, tal vez no, por lo que a mí respecta, fue un paso con mucho esfuerzo que tuve que dar, por eso le agradezco a Él inmensamente, porque me permitió darlo.

Las personas más cercanas jugaron un papel muy importante en esta travesía, y por eso es necesario agradecer el inmenso apoyo que me mostró mi familia. En primer lugar me gustaría agradecerle a mi madre, porque fue su apoyo tanto económico como moral que me permitió sostenerme. También mi a mi padre, por sus constantes oraciones por mí, por su apoyo incondicional que recibí de él. A mi hermana Mabel, que a lo largo de este viaje me acompañó, sus ánimos siempre me dieron fuerza y los libros que me regaló me brindaron apoyo, por eso le agradezco bastante. A mi hermana Mariela, por todo el apoyo que mostró conmigo, por el don de la creatividad que Dios le ha dado, pues esa creatividad me inspiró en los proyectos que realicé.

Los amigos, los verdaderos amigos siempre son importantes en todo proyecto que se realiza en la vida, por eso quiero agradecerle a cada uno de ellos. Primero que nada le quiero agradecer a Nabila, pues llegó a mi vida para concluir esta etapa, siempre me apoyó incondicionalmente, siempre me alentó a seguir y dar lo mejor de mí por eso quiero agradecerle todo su apoyo. A mis amigos que desde CCH me han acompañado. A Albert, mi buen amigo desde hace muchos años. A Miguel, quién siempre me apoyó y me aconsejó a lo largo de este proceso. A Octavio, por su constante ayuda tanto en lo académico como en la amistad. A Angel, por toda la ayuda que recibí de él. A Fernanda, por la gran amistad que ha mostrado y el gran interés que muestra para que exista una mejor educación.

A mis compañeros de trabajo en el Instituto de Física. A la Dra. Guerda pues más que lecciones académicas me enseñó cómo se deben hacer las cosas en todo aspecto de la vida, y claro, por su gran apoyo que me mostró al realizar este proyecto, pues sin su apoyo no hubiera concluido este proyecto. Al M. I. Tirso, por su amistad y apoyo. Al M. I. Hesiquio por la ayuda que brindó para la realización de la mesa de experimentos. Al M. en C. Eduardo López pues fueron sus consejos los que mejoraron este proyecto. Al M. en C Cesar Ruiz por la ayuda brindada. A mis compañeros, Alexis, Iván, Christian, Naxi, Abel, Alain, Nestor y a todos los que no he nombrado pero que saben que jugaron un papel muy importante en este trabajo.

Al instituto de Física de la UNAM por la gran oportunidad que me brindó para colaborar con ellos y por dejarme hacer uso de sus instalaciones. A los proyectos PAPIT “IN105813” y CONACyT “127409”.

Índice

1. Introducción	1
1.1 Objetivo general	4
1.2 Objetivos particulares	4
2. Conceptos básicos de Dosimetría	5
2.1. Producción de rayos-X.	5
2.1.1 Rayos-X de frenado o Bremsstrahlung	6
2.1.2. Rayos-X característico.	8
2.2. Procesos de interacción	9
2.2.1 Efecto Fotoeléctrico	9
2.2.2 Efecto Compton	10
2.3. Dosimetría de rayos-X de bajas energías con cámara de ionización.....	10
2.3.1 Cámara de ionización como dosímetro de referencia.	11
2.3.2 Factores de Corrección.-	14
2.3.3 Incertidumbres	16
2.4. Comunicación Serial.....	18
2.4.1. Comandos Estándar	19
2.5. Programación en LabVIEW.	19
2.5.1 Rutinas y Subrutinas.	20
2.5.2. Controles e Indicadores.....	20
2.5.3. Estructuras.....	20
2.5.4. Manejo de comunicación serial	22
2.5.5. Sincronización.....	23
2.6. Sistemas Embebidos	23

2.6.1- Microcontrolador	23
3. Materiales y Desarrollo.....	25
3.1. Características de los equipos disponibles en la sala de irradiación con rayos-x.....	25
3.2. Materiales Comprados y Construidos.....	29
3.3. Diseño del programa de control de los equipos de dosimetría.	30
3.3.1 Comunicación con los dispositivos.....	34
3.3.2 Comandos y configuración de dispositivos.	36
3.3.3 Adquisición de datos.....	38
3.3.4 Datos de salida	41
3.4. Diseño del controlador de la mesa de experimento	43
4. Resultados	46
4.1. Construcción del sistema de control por computadora.....	46
4.1.1 Funcionamiento del sistema de control.....	47
4.1.2 Prueba de verificación del sistema	52
4.2. Control de la mesa.	54
5. Discusiones	55
6. Conclusiones	57
Referencias.....	58
Apendice.....	60
A.	60

Índice de Figuras

Figura 1.1 Primera radiografía de la historia.....	1
Figura 1.2 Cámara de Ionización Exradin A12 del IFUNAM	3
Figura 2.1 Estructura básica de un tubo de rayos-X	5
Figura 2.2 Diagrama eléctrico básico de un tubo de rayos-X	6
Figura 2.3 Rayo-X Bremsstrahlung [3].....	7
Figura 2.4 Espectro de radiación.....	8
Figura 2.5 Cámara de ionización.....	12
Figura 2.6 Variación de volumen en cámara de ionización	13
Figura 2.7 Electrómetro.....	14
Figura 2.8 Trama en la comunicación RS323 asíncrona.....	18
Figura 2.9 Controles en LabVIEW.	20
Figura 2.10 Estructura Sequence.....	21
Figura 2.11 Estructura Event.....	22
Figura 2.12 Arquitecturas en un microcontrolador	24
Figura 2.13 Microcontrolador PIC18F4550.....	24
Figura 3.1 Electrómetro SUPERMAX.....	26
Figura 3.2 Termómetro Fluke 1523	27
Figura 3.3 Druck DPI 142.....	27
Figura 3.4 Fuente para tubo de rayos-X YSMART 160E/1.5.....	28
Figura 3.5 Estructura de los comando de la fuente de rayos-X.....	28
Figura 3.6 Diagrama de conexión para el cable de comunicación de la fuente de rayos-X	30
Figura 3.7 Primera versión de interfaz.	31
Figura 3.8 Diagrama de flujo general del programa.	32
Figura 3.9 Diagrama de flujo para identificar los dispositivos	33
Figura 3.10 Ventana de configuración de comunicación automática	33
Figura 3.11 SubVi para comunicación.....	35
Figura 3.12 Comunicación dentro del ciclo de adquisición de datos.....	36
Figura 3.13 Diley en el ciclo principal.....	36
Figura 3.14 Cambio de unidades.....	37

Figura 3.15 Creación de los comandos para configurar el electrómetro.....	38
Figura 3.16 Modos de colección del electrometro.	39
Figura 3.17 Envío del comando para iniciar la fuente	39
Figura 3.18 Diagrama de flujo del ciclo de adquisición de datos	40
Figura 3.19 Estado de la fuente.....	41
Figura 3.20 Código donde señala las pruebas que tuvieron algún error	42
Figura 3.21 Mesa de experimentos de dosimetría de bajas energías.	43
Figura 3.22 Secuencia a seguir para mover un motor	44
Figura 3.23 Control para la mesa de dos ejes diseñada en el IFUNAM	44
Figura 3.24 Diseño del controlador de la mesa	45
Figura 4.1 Versión final de la interfaz.....	46
Figura 4.2 Última versión de la interfaz.....	47
Figura 4.3 Error que señala que no se encuentran conectados los dispositivos	48
Figura 4.4 Modos de comunicación con los dispositivos.....	49
Figura 4.5 Mensaje de error que señala la existencia de un problema con el electrómetro.	49
Figura 4.6 Interfaz adquiriendo datos	50
Figura 4.7 Mensaje de error muestra que la puerta fue abierta durante el experimento.	51
Figura 4.8 Resultados finales después de un experimento.....	51
Figura 4.9 Reporte elaborado por la interfaz para la verificación de los datos.....	52
Figura 4.10 Controlador de la mesa de experimento.	54

1. Introducción

En 1895, durante un experimento con tubos de rayos catódicos (RC) (electrones con energía cinética tal que permiten moverse de un cátodo a un ánodo en un medio de vacío) para estudiar la luminiscencia que se producía dentro del tubo, el científico alemán Wilhelm Roentgen descubrió los rayos-X (también llamados fotones). Además, se dio cuenta de que existían materiales, como los metales, que los rayos-X no podían atravesar fácilmente. Después, experimentó con diferentes materiales y objetos como obstáculo entre el tubo de rayos-X y una película fluorescente; entre los que se encuentra la mano de su esposa, Bertha, la cual fue expuesta a radiación por 15 minutos. Dicho experimento dio como resultado la mano grabada en la película fluorescente en la que se observa los huesos así como el anillo que llevaba puesto Bertha. Así, se produjo la primera radiografía de la historia y el primer tubo de rayos-x (ver figura 1.1) [1-2].



Figura 1.1 Primera radiografía de la historia, la cual muestra la mano de la esposa del científico alemán Wilhem Röntgen [2]

Desde entonces, los rayos-x han sido usados como una herramienta fundamental para el diagnóstico y/o tratamiento de cáncer en medicina. Sin embargo, el uso de los rayos-X requiere de un conocimiento exacto y preciso de la cantidad de energía entregada al paciente, ya que un uso inapropiado puede resultar perjudicial para la salud.

La Dosimetría de la radiación es la rama de la Física que estudia la cantidad de energía impartida en la materia debido a procesos de interacción de la radiación ionizante. La energía

depositada se cuantifica mediante la dosis, la cual es la cantidad de energía por unidad de masa y tiene unidad de Gray ($1\text{Gy}=1\text{J/Kg}$) [3].

Dependiendo de la energía de los fotones, se clasifican en 3 grupos: a) bajas energías que van desde 10keV hasta 100keV ; b) energía intermedia que van desde 100keV a 500keV y c) los de altas que incluyen energías arriba de 500keV . Los fotones de bajas energías son muy usados en aplicaciones clínicas tales como braquiterapia, mamografía, tomografía computarizada o radiografía convencional. Pero, la dosimetría en estos campos de radiación representa un desafío debido a la complejidad de los procesos de interacción con la materia en este intervalo de energía. Por ello, es de vital importancia realizar investigación en estos campos de radiación con el fin de entregar una dosis correcta al paciente para proteger los tejidos sanos en cada tratamiento y evitar la exposición innecesaria a la radiación en caso de aplicaciones de diagnóstico [4]. Actualmente, en el Instituto de Física de la Universidad Nacional Autónoma de México (IFUNAM) se están realizando investigaciones sobre las características dosimétricas de materiales que se usan normalmente en aplicaciones clínicas, por lo que se tiene un gran interés para desarrollar técnicas de alta precisión en la medición de la energía impartida [5].

El principal instrumento de medición que se ocupa en este tipo de estudios es el dosímetro que es un medio con un volumen sensible capaz de proveer información relacionada con la energía que le fue depositada durante la irradiación. Generalmente, la dosis absorbida se cuantifica mediante la medición de una respuesta a través de variables físicas tales como carga producida en un volumen de aire (cámara de ionización), cambio de color de un polímero (películas de tinte radiocrómico), producción de nano partículas (gel) o emisión de luz de un cristal (dosímetro termoluminiscente) debido a la interacción de la radiación con la materia [5-6-7]. Independiente del dosímetro usado, dicha respuesta depende de la dosis absorbida y del tipo de radiación. Este trabajo, se enfoca principalmente en el uso de cámara de ionización, la cual consiste en un volumen definido de aire (Figura 1.2). Al interaccionar la radiación con la cámara de ionización, el aire se ioniza produciendo pares ion-electrón que son colectados por un electrómetro. Así, la carga colectada es convertida a dosis absorbida mediante el uso de un factor de calibración [3].



Figura 1..2 Cámara de Ionización Exradin A12 del IFUNAM

Las cámaras de ionización pueden tener diferentes formas como placas paralelas o cilíndricas y son diseñadas para usarse en diferentes medios, por ejemplo: agua y aire [3].

Existen dos tipos de laboratorio para calibrar estos equipos: Un laboratorio primario donde se calibra el equipo en condiciones muy controladas, mientras que un laboratorio secundario se emplea equipo ya calibrado en un laboratorio primario para poder calibrar otro equipo en condiciones ambientales. En dichos laboratorios, se obtiene un factor de calibración para las cámaras de ionización [3].

El proceso de la obtención del factor de calibración en un laboratorio primario se realiza bajo condiciones físicas que no son iguales a las que se emplea la cámara en la clínica o en un laboratorio de investigación, pues la densidad del aire cambia dependiendo de las variaciones ambientales. Por esta razón, es necesario realizar correcciones en las medidas [8]. Por ello, es necesario contar con medidas confiables y precisas de presión ambiental y temperatura que son tomadas en el momento que se emplea la cámara de ionización por lo que se requiere instrumentos de medición que puedan entregar resultados con la menor incertidumbre posible. Por ello, la necesidad de adquirir medidas de carga colectada, presión y temperatura en sincronía para así poder asegurar una menor incertidumbre cuando se realicen las correcciones [9]. Además se debe contar con un buen control del tubo de rayos-x e igualmente poder sincronizar su funcionamiento con la adquisición de las variables antes mencionadas.

Actualmente, de nuestro conocimiento, en la mayoría de los lugares en los que utilizan cámara de ionización para la calibración de haces de radiación no cuentan con un programa de monitoreo de presión y temperatura por computadora sino se hace todo de manera manual.

Recientemente, se ha reportado el desarrollo de un programa en LabVIEW capaz de entregar factores de calibración para distintas cámaras de ionización en el que se ha logrado optimizar el tiempo de adquisición y disminuir la incertidumbre de las medidas [10]. Sin embargo, este programa no sería útil para este proyecto, ya que fue diseñado para ser usado en un Laboratorio secundario, por lo que las condiciones de medidas son diferentes a las de un laboratorio de Investigación en una universidad.

1.1 Objetivo general

El objetivo general de este trabajo es desarrollar un programa que funcione como una interfaz a través de la cual se puedan comunicar un electrómetro, un termómetro, un barómetro y la fuente de rayos-X para adquirir datos provenientes de cada uno de ellos así como manipularlos. Dicha interfaz permitirá sincronizar la adquisición de datos y así reducir la incertidumbre en las mediciones, además de optimizar el tiempo entre cada experimento.

1.2 Objetivos particulares

- 1) Diseñar una interfaz usando el entorno de programación LabVIEW.
- 2) Programar la interfaz para adquisición de datos
- 3) Validar el sistema mediante experimentos
- 4) Diseñar y construir un controlador para la mesa de experimentos.

2. Conceptos básicos de Dosimetría

2.1. Producción de rayos-X.

Los rayos-X son producidos cuando electrones con muy alta energía cinética interactúan con la materia convirtiendo toda o parte de esa energía cinética en radiación electromagnética. Para poder realizar este proceso, son necesarios tres elementos principales: *fente de electrones*, *un electrodo objetivo* y *una fuente de poder externa*. El dispositivo que incluye estos tres elementos se conoce como tubo de rayos-X el cual es un tubo que se encuentra al vacío e incluye un cátodo (fuente de electrones), un ánodo (electrodo objetivo) y se conecta a una fuente de poder externa.

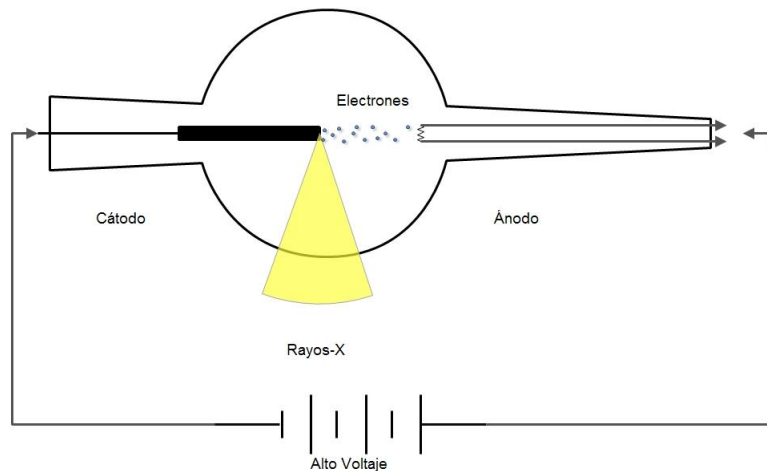


Figura 2.1 Estructura básica de un tubo de rayos-X

En la figura 2.1 se puede apreciar la estructura básica de un tubo de rayos-X con los tres elementos básicos [11]. Los electrones en el cátodo son acelerados con una diferencia de potencial que es la que determina la máxima energía que podrán tener los electrones y dicha energía es proporcional a la diferencia de potencial aplicada. Por ejemplo, si un tubo de rayos-x tiene una diferencia de potencial de 20kV, entonces la máxima energía que el electrón obtendrá es de 20keV. Los electrones chocan con el cátodo convirtiendo la energía cinética en distintas formas de energía. Cabe mencionar que la mayor parte de la energía de los electrones no se convierte en radiación

electromagnética, sino en calor, es un porcentaje muy pequeño de la energía que logra convertirse en radiación electromagnética.

El cátodo consiste en un filamento generalmente de tungsteno ya que requiere que tenga una temperatura muy alta para que pueda darse lo que se conoce como emisión termoiónica y poder liberar fácilmente los electrones. Para poder calentar el filamento se requiere de un voltaje bajo (generalmente 10v) y una corriente muy alta (6 a 7 A). En la figura 2.2 se puede apreciar un diagrama eléctrico del tubo de rayos-X considerando el cátodo [11].

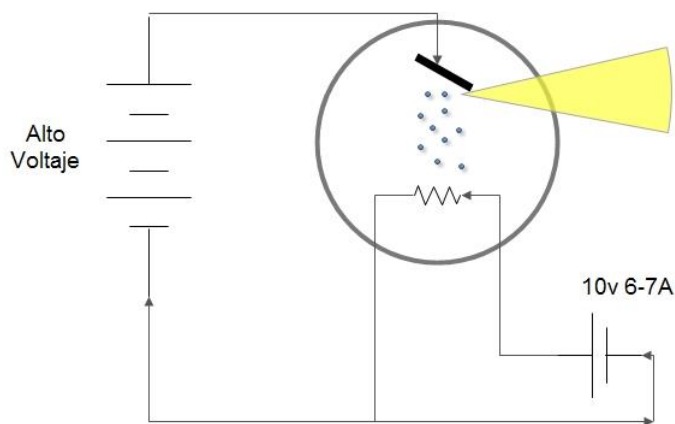


Figura 2.2 Muestra el diagrama eléctrico básico de un tubo de rayos-X

Cómo se mencionó anteriormente, la producción de rayos-X se debe a la conversión total o parcial de la energía cinética de un electrón en radiación electromagnética, esto puede lograrse gracias a diferentes procesos. En este trabajo se describirán dos principales procesos de producción de rayos-X.

2.1.1 Rayos-X de frenado o Bremsstrahlung

Los electrones son acelerados del cátodo con una energía cinética proporcional a la diferencia de potencial aplicada entre el ánodo y el cátodo. Una vez que son acelerados, estos electrones adquieren una trayectoria, cuando dichos electrones se acercan a un núcleo del átomo del objetivo (ánodo), el electrón frena debido a fuerzas coulombianas de atracción entre el núcleo

del átomo objetivo y el electrón, además que el electrón cambia su trayectoria. La energía cinética perdida en la desaceleración produce una onda electromagnética de igual energía que la energía cinética perdida. Esta radiación se le conoce como radiación de Bremsstrahlung que viene del alemán, y significa: radiación de frenado [11].

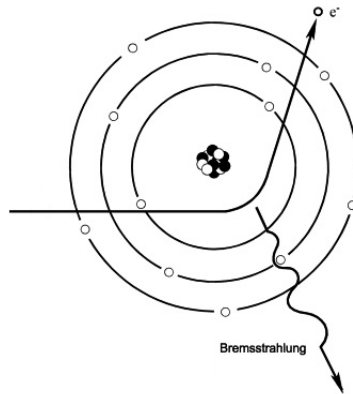


Figura 2.3 rayo-X Bremsstrahlung [3].

La distancia entre el núcleo del átomo y electrón determina la energía perdida, esto se explica con la ley de coulomb, que dice que la fuerza de atracción incrementa con el inverso al cuadrado de la distancia (ver figura 2.3). Una distancia relativamente grande producirá un rayo-X de baja energía, ya que existe poca fuerza de atracción, mientras que una distancia más cercana producirá un rayo-X con mayor energía. Cuando un electrón se impacta directamente con el núcleo del átomo, la energía del rayo-X producido será la misma que la energía cinética del electrón.

La probabilidad de que un electrón se impacte con el núcleo de un átomo es muy baja debido a que la estructura del átomo ocupa mucho espacio comparado con el tamaño del electrón así que la producción de rayos-X de alta energía decrece [11]. En la figura 2.4 se puede apreciar que a la mitad del espectro de radiación la intensidad decrece causada por los rayos-X Bremsstrahlung.

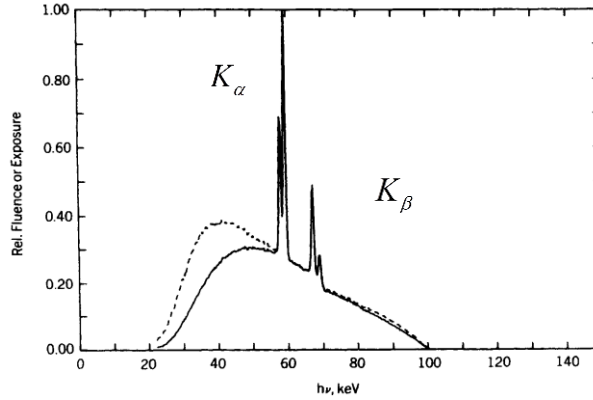


Figura 2.4 Espectro de radiación con un ánodo de tungsteno. Muestra energía contra la intensidad [3].

2.1.2. Rayos-X característico.

La producción de rayos-X característicos es el proceso en el que un electrón que es acelerado del cátodo se impacta con un electrón que orbita en una de las bandas de un átomo del ánodo dejando un hueco. Este hueco tiende a ser ocupado por un electrón de una banda superior y cuando se da la transición se libera energía. La energía liberada es característica del elemento objetivo y de la transición del electrón [3].

La energía cinética del electrón que proviene de un cátodo debe ser mayor o igual a la energía del electrón que se encuentra en la órbita de un átomo del ánodo para que pueda ser expulsado. Cuando un electrón de una capa superior llena la vacante de una órbita inferior, se libera un fotón característico. Este fotón tiene una energía igual a la diferencia de energías las dos bandas. Como la energía de cada banda es única en cada elemento, se producirán fotones con energías diferentes para cada elemento. El salto puede ser entre cualquier banda, pero esto determinará la energía del fotón [11]. Por ejemplo, si se tuviera un electrón con suficiente energía para desprender un electrón de la banda K de un átomo de tungsteno, cualquier electrón de cualquier banda superior sería capaz de ocupar este lugar como podría ser un electrón de la capa L [11]. Así la energía del fotón emitido estaría dado por la siguiente ecuación:

$$E_K - E_L = 69.5\text{keV} - 10.2\text{keV} = 59.3\text{keV}, \quad (1)$$

donde E_K y E_L son las energías de las bandas K y L de un átomo de tungsteno.

En figura 2.1.2.1 se puede observar los rayos-X característicos para un objetivo de tungsteno. Los subíndices α y β son designados para denotar donde ocurre la transición si de una banda adyacente o de otra banda. Por lo que K_α expresa que la energía del fotón se debió a una transición entre la banda K y la banda L, mientras que K_β expresará entonces que la transición se dio entre la banda K y la banda M, N u O.

2.2. Procesos de interacción

La radiación puede analizarse tomando los efectos que esta produce al interactuar con la materia. En el caso de los rayos-X se conocen como radiación ionizante. Dicha radiación, como su nombre lo indica, ioniza la materia, es decir, la radiación puede producir un par ion-electrón para modificar la estructura de la materia y a los electrones producidos se les conoce como electrones secundarios. Pueden ocurrir algunos fenómenos para que la radiación produzca iones, como lo son: *el Efecto Fotoeléctrico y el Efecto Compton* [3].

2.2.1 Efecto Fotoeléctrico

El efecto fotoeléctrico ocurre cuando un fotón incidente interacciona con un átomo. El fotón incidente transmite toda su energía a uno de los electrones orbitales causando que este se desprenda de su órbita adquiriendo energía cinética T igual a la diferencia de energías entre el fotón y la energía de ligadura del electrón [11]:

$$T = E_{\text{fotón}} - E_{\text{electrón}}, \quad (2)$$

donde $E_{\text{fotón}}$ y $E_{\text{electrón}}$ son las energías del fotón y del electrón, respectivamente.

Se crea una vacante en dicha orbita la cual es llenada por un electrón de un orbital superior produciendo un fotón más con energía T_n igual a la diferencia de energías entre los orbitales que se da la transición [11]:

$$T_n = E_K - E_L \quad (3)$$

2.2.2 Efecto Compton

El efecto Compton se da cuando un fotón incidente interactúa con un electrón de una orbital del átomo el cual tiene mucha menor energía que el fotón, con esto, se puede considerar una dispersión elástica, donde el fotón solo pierde una parte de su energía y esta se imparte al electrón. Después de cada interacción un fotón y un electrón son dispersados con un cierto ángulo respecto al fotón incidente. Produciendo así un par electrón-ion [3].

2.3. Dosimetría de rayos-X de bajas energías con cámara de ionización.

La dosis se puede definir como la cantidad de energía depositada en cada unidad de masa como resultado de la interacción de la radiación ionizante con la materia y tiene unidad de [3]:

$$D = \frac{d\varepsilon}{dm} \left[Gy = \frac{J}{kg} \right], \quad (4)$$

donde D es la dosis, ε es la cantidad de energía y m la masa.

La tasa de dosis absorbida se define como la cantidad de dosis en un intervalo de tiempo [3].

$$\dot{D} = \frac{dD}{dt}, \quad (5)$$

donde \dot{D} es la tasa de dosis y t el tiempo

Por otro lado la exposición radiométrica se puede definir como la cantidad de carga total de iones liberada por masa de aire seco, en condiciones estándar (presión= 1 [atm] y temperatura=20°C). Esto es para poder cuantificar la carga que se deposita en el medio y observar los efectos de las radiaciones ionizantes, se puede expresar la exposición radiométrica con la siguiente ecuación [3]:

$$X = \frac{dQ}{dm}, \quad (6)$$

donde X es la exposición radiométrica y Q es la carga total de iones liberada en la masa m .

Una vez que se tiene la exposición es muy fácil obtener la dosis absorbida haciendo uso de la siguiente ecuación [3]:

$$X = \frac{D}{k} [C/kg] \Leftrightarrow D = kX, \quad (7)$$

donde k es un factor de ajuste con valor de $k = 33.97 [J/C]$

Para realizar mediciones de dosis absorbida, se hace uso de lo que se conoce como dosímetro el cual puede ser cualquier dispositivo que sea capaz de entregar una lectura relacionada con la dosis absorbida en un volumen específico. Existen varios recursos para medir la dosis, por ejemplo, se pueden los cambios de color en películas radiocrómicas como una forma de medir la dosis [3]. Otro ejemplo de dosímetro es la cámara de ionización de la que se hablará un poco más adelante [10].

2.3.1 Cámara de ionización como dosímetro de referencia.

La cámara de ionización es el instrumento más usado como dosímetro de referencia ya que brinda mediciones precisas. Las cámaras de ionización se pueden encontrar de distintos diseños para diferentes aplicaciones, además pueden ser construidas bajo un diseño específico para una aplicación especial [3].

La cámara de ionización como dosímetro de referencia es bastante conveniente ya que se puede emplear el concepto de exposición radiométrica para medir la carga producida por radiación ionizante y así obtener una medida confiable de la dosis absorbida. Todas las cámaras de ionización se basan en el concepto carga que es la variable física medida para posteriormente obtener la dosis absorbida. Hay que señalar que la instrumentación no se encarga de medir carga directamente, sino que se mide la corriente eléctrica que existe entre dos placas con alto voltaje e integrando dicha corriente, se obtiene la carga.

Para las investigaciones de dosimetría de bajas energías que se están realizando en el IFUNAM, se están empleando cámaras de ionización, para emplearlas como dosímetro de referencia. A continuación se describe brevemente el funcionamiento de una cámara de ionización de aire libre de placas paralelas (ver figura 2.5).

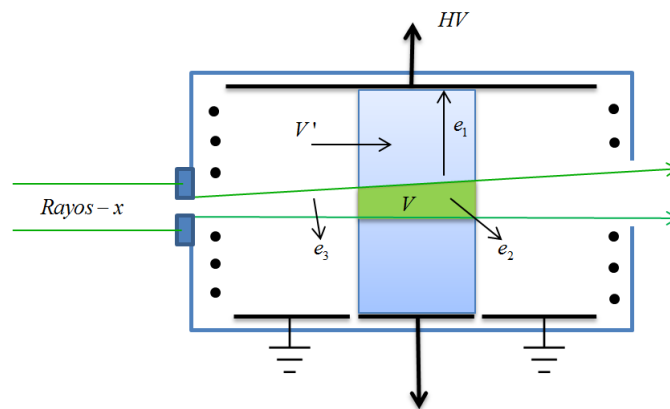


Figura 2.5 Cámara de ionización

Una forma de medir la carga producida por la radiación ionizante (exposición radiométrica) es haciendo uso de un campo eléctrico y del concepto de corriente eléctrica.

Ésta cámara está cubierta de plomo para excluir rayos-x provenientes de cualquier otra fuente que no sea la de interés. A la entrada de la cámara se encuentra un diafragma de tungsteno el cual es alineado con el haz de rayos-X.

En el interior de la cámara hay tres placas coplanaras que se encuentran localizadas en la parte inferior de la cámara de ionización y a un lado de la trayectoria del haz; además de una placa con alto voltaje situada del lado opuesto de las tres placas coplanaras. Todas las placas son paralelas al haz de rayos-X y equidistantes. La distancia del haz a las placas está diseñada para atrapar todos los electrones secundarios originados en el haz. Un alto voltaje entre las placas generará un campo eléctrico el cual ayuda en la colección de carga producida por la radiación [3].

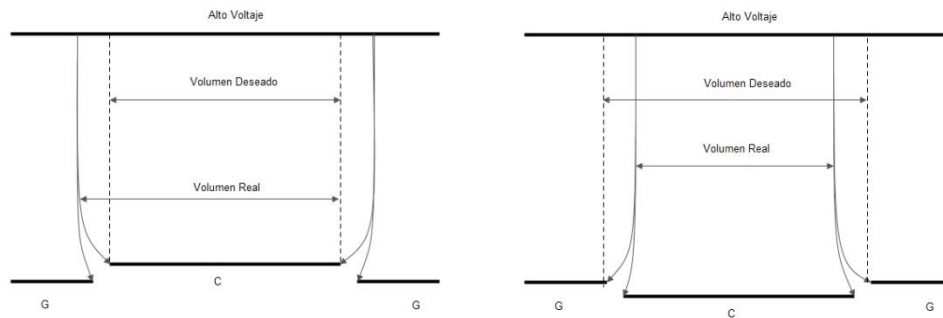


Figura 2.6 Muestra la variación de volumen debido a que las placas no se encuentran alineadas.

El hecho de que existan tres placas, y que en la del medio es donde se colecte la carga de interés, tiene una razón fundamental. Como se sabe, las líneas de campo (eléctrico o magnético) nunca se cruzan entre sí, esto es aprovechado ya que de esta manera se puede definir un volumen en la placa colectora. En este diseño es muy importante que las placas estén perfectamente equidistantes al haz de radiación, pues la mínima variación de distancia podría afectar al volumen en la placa colectora y causar incertidumbre en las medidas como se puede ver en la figura 2.6. Con un volumen definido es posible conocer la variación del volumen en función de la temperatura y la presión.

Se sabe que la corriente es una razón de cambio de carga en un intervalo de tiempo, en otras palabras:

$$i = \frac{dQ}{dt}, \quad (8)$$

donde i es la corriente que fluye a través de la cámara de ionización.

Despejando la carga se tiene:

$$Q = \int i dt, \quad (9)$$

De esta forma se puede medir la corriente e integrarla en un intervalo de tiempo, para así, conocer la carga que fue colectada.

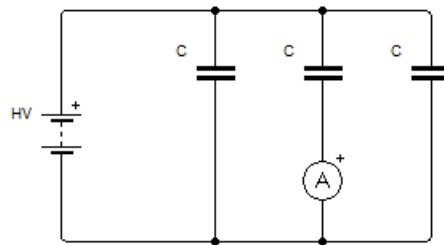


Figura 2.7 Electrómetro

Existen otros tipos de cámaras de ionización, pero el principio que rige a todas ellas, es el de campo eléctrico y corriente eléctrica [3]. Cabe señalar que la cámara de ionización se conecta a un dispositivo llamado electrómetro el cual se encarga de suministrar el alto voltaje a la cámara, mide la corriente y despliega la medida final. Haciendo un diagrama eléctrico, la cámara de ionización y el electrómetro se podrían representar como se muestra en la Figura 2.7.

2.3.2 Factores de Corrección.-

La densidad de aire en la cámara de ionización no es constante en cada prueba es por esta razón que se requiere de un factor que pueda corregir los cambios que puedan ocurrir.

Las cámaras de ionización son calibradas por un laboratorio primario o secundario el cual entrega un factor de calibración que es una relación entre la carga colectada y la dosis absorbida por el material.

En un laboratorio primario se realizan pruebas con variables físicas muy bien controladas, por lo que la presión y la temperatura son constantes mientras la cámara es calibrada, no obstante, la cámara no se usa en estas condiciones por lo que es necesario realizar una compensación por los posibles cambios de densidad debido a variación en la presión y en la temperatura [3-8]. La ley de los Gases Ideales establece que:

$$P \cdot V = n \cdot R \cdot T, \quad (10)$$

donde P es la presión, V es el volumen, n es el número de moles en el gas, T la temperatura y R es la constante ideal de los Gases Ideales.

Tenemos que:

$$R = \frac{P \cdot V}{n \cdot T}, \quad (11)$$

Se sabe que el volumen en una cámara de ionización permanecerá constante y que por lo tanto los cambios en la densidad de aire ocurrirán por cambios en la masa de aire por lo que en la ecuación se mantiene el volumen constante:

$$\frac{P_1}{n_1 \cdot T_1} = \frac{P_2}{n_2 \cdot T_2}, \quad (12)$$

La masa uno corresponde a la masa de aire con la que fue calibrada la cámara de ionización. Por otro lado masa dos es la masa de aire que se tiene en el laboratorio donde se realizarán prueba, y es la masa que se verá afectada por los cambios de presión y temperatura, por lo tanto, despejando el la masa dos tenemos que:

$$n_2 = n_1 \times \frac{P_2}{P_1} \times \frac{T_1}{T_2}, \quad (13)$$

$$n_2 = n_1 \cdot \left(\frac{1}{k_{TP}} \right), \quad (14)$$

$$\therefore k_{TP} = \frac{P_1}{P_2} \times \frac{T_2}{T_1}, \quad (15)$$

donde k_{TP} es el factor de corrección por presión y temperatura.

En el caso de las cámaras de ionización, la presión y la temperatura iniciales son la presión y la temperatura de referencia, es decir, las variables físicas con que fue calibrada la cámara de ionización: [8]

$$k_{TP} = \frac{P_{ref}}{P} \times \frac{T + 273.15}{T_{ref} + 273.15} \quad (16)$$

2.3.3 Incertidumbres

El resultado de una medida o un cálculo es solamente una aproximación del “valor verdadero” por lo que un valor de incertidumbre debe acompañar la medida para considerar que el resultado de la medición está completo. En general, la incertidumbre indica el intervalo en donde se encuentra el “valor verdadero”. No se debe confundir la incertidumbre con el error, pues el error se define como la diferencia entre el valor calculado y/o medido con el “valor verdadero”, el cual es imposible de conocer [12].

La incertidumbre se divide en dos categorías de acuerdo con el método que se usa para conocer su valor:

Tipo A: Incertidumbre que es evaluada y analizada con métodos estadísticos.

Tipo B: Incertidumbre que es evaluada con otros términos.

La desviación estándar es una incertidumbre tipo A. Es una medida que cuantifica que tanto se separan los datos de su media aritmética (también conocido como promedio), es decir, es una medida de dispersión. Se puede entender a la desviación estándar como el promedio de la diferencia de los datos con respecto a su media aritmética. La importancia de esta medida es grande, pues cuando se interpretan los datos, no es suficiente contar con el promedio, también se requiere de conocer su distribución para así tener una mejor idea de su comportamiento en la realidad [13].

La desviación estándar se puede calcular con la siguiente ecuación:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}, \quad (17)$$

donde

s = desviación estándar x_i = Dato i
 \bar{x} = Promedio de datos n = número de datos

La incertidumbre tipo B es basada en apreciaciones científicas y se incluye toda la información disponible como la siguiente:

- Mediciones previas
- Conocimiento general o experiencia con el comportamiento de algún material
- Especificaciones técnicas de equipo.
- Información proveniente de la calibración de algún equipo.

La incertidumbre estándar combinada refleja la contribución de todas las incertidumbres que intervienen para la obtención de una medida sin importar si dichas incertidumbres son A o B. La incertidumbre estándar combinada está dada por la siguiente ecuación [13]:

$$\delta q = |q| \sqrt{\left(\frac{\delta x}{x}\right)^2 + \dots + \left(\frac{\delta z}{z}\right)^2 + \left(\frac{\delta u}{u}\right)^2 + \dots + \left(\frac{\delta w}{w}\right)^2}, \quad (18)$$

donde: u, w, x, z son las variables que intervienen en la obtención de un mesurando

Por último, la incertidumbre expandida se define como la incertidumbre multiplicada por un factor de cobertura, k , el cual aumenta la incertidumbre pero por otro lado también aumenta el nivel de confianza de la medida. Normalmente, el valor del factor de cobertura esta en el rango de 1 a 3. El valor del factor de cobertura se escoge dependiendo del nivel de confianza que se desea, existen tablas donde se indica el valor de k para ciertos niveles de confianza, por ejemplo, $k=2$ define un nivel de confianza de aproximadamente 95% [12].

2.4. Comunicación Serial

La comunicación serial es aquella en que se transmite datos digitales bit a bit en contraste con la comunicación paralela que transmite bits simultáneamente. Existen diferentes interfaces de comunicación serial como la I2C o RS-232 [14].

La interfaz RS232 que viene de *Recommended Standard* es una interfaz de comunicación serie de datos binarios entre dos dispositivos, normalmente un “Data Terminal Equipment (DTE)” y un Data communication Equipment (DCE). Las características principales son:

- Velocidad máxima de 116kps (taza de baudaje)
- Longitud máxima de cable de 25 metros.
- Tensión máxima de $\pm 25v$
- Impedancia 3 a 7 $k\Omega$
- Modo de operación simple
- Transmisión síncrona o asíncrona

Para la comunicación asíncrona se tienen tramas de datos siguiendo un esquema como el que se muestra en la Figura 2.8.

Start	Datos (5-8 bits)	Paridad	Stop
-------	------------------	---------	------

Figura 2.8 Trama en la comunicación RS232 asíncrona

Para lograr la comunicación RS232, existen diversos conectores. Sin embargo, el más usado es el conector de 9 pines, conocido como DB-9. Existen varias configuraciones para el cable y el conector DB-9: loopback, null modem y directo. La configuración adoptada para la construcción del cable depende de la comunicación requerida [15].

2.4.1. Comandos Estándar

Los comandos Standard Commands for Programmable Instruments (SCPI) son usados en la comunicación RS232 e IEEE488 para lograr controlar diferentes instrumentos. El SCPI especifica cierta sintaxis, estructura y formato de datos y usa nemónicos para poder facilitar su uso. En SCPI, los comandos son agrupados en diferentes subsistemas los cuales representan un bloque funcional del instrumento. Estas agrupaciones se realizan con cierta jerarquía comenzando desde el subsistema más general, hasta el más particular. Algunos subsistemas son: CALCulate, CALibration, CONTROL, DIAGnostic, SYSTem y UNIT [14]. Un ejemplo de comando SCPI es: SYSTem:COMMunicate:SERial:BAUD 9600, el cual para poder ser usado se convierte en los siguientes nemónicos: SYST:COMM:SER:BAUD 9600

2.5. Programación en LabVIEW.

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) es un lenguaje de programación gráfico y un entorno de desarrollo. Este entorno fue creado para optimizar el diseño de aplicaciones gráficas comparado con otros lenguajes de programación como C, C++, etc [14].

En este entorno se trabaja con dos ventanas: a) Diagrama de Bloques que muestra el código interno del programa y b) Panel frontal que muestra la interfaz visible al usuario. Una aplicación en LabVIEW normalmente está conformada por controles, indicadores, funciones, VIs (rutinas), subVIs (subrutinas) y estructuras. Además, se pueden incluir variables globales y locales. A continuación se describirán algunos de estos conceptos.

2.5.1 Rutinas y Subrutinas.

Los programas creados en LabVIEW se conocen como *Virtual Instruments (VI)*. Un VI puede contener un subVI que es básicamente una subrutina que contiene fragmento de código y puede ser utilizado para reducir el código en un VI y así poder entender al programa de una forma más clara, o bien se puede utilizar como un código que se usará en distintas partes de un programa [14].

2.5.2. Controles e Indicadores

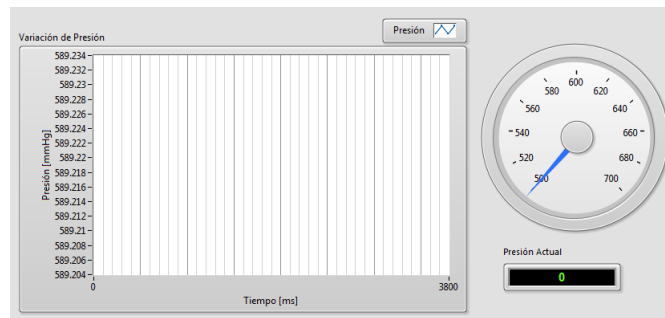


Figura 2.9 Se muestra las distintas formas de un control en LabVIEW.

Los controles e indicadores son usados para ingresar e imprimir datos. Estas herramientas aparecen tanto en el panel frontal como en el diagrama de bloques y se utilizan principalmente para la entrada y salida de datos. Los controles e indicadores dependen del tipo de variable que se está usando y pueden mostrar desde un simple carácter hasta una gráfica [14]. En la figura 2.9 se presenta un ejemplo de ellos.

2.5.3. Estructuras.

En LabVIEW, las estructuras se utilizan definir bloques de códigos. A continuación se explicaran brevemente las estructuras de mayor importancia en la elaboración de este proyecto.

La programación en LabVIEW se realiza mediante la interconexión de estructuras, funciones, controles, indicadores y variables. A diferencia de los lenguajes de programación escritos (C, C++, etc), LabVIEW no ejecuta las instrucciones de manera secuencial, sino bajo un concepto conocido como *dataflow*. Dicho concepto establece que se puede ejecutar una función solamente si se dispone de todos los datos de entradas. Esto a veces no es tan útil porque si se tiene dos funciones en condición de ser ejecutadas, es difícil definir cuál de ellas se ejecuta primero. Para ello se tiene la estructura Sequence [15].

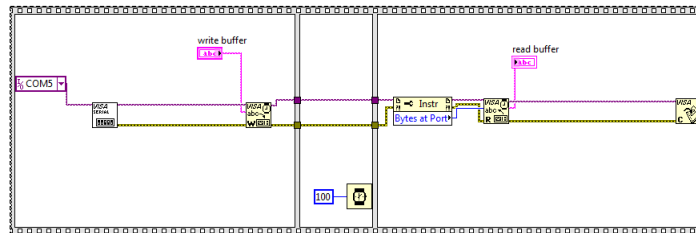


Figura 2.10 Estructura Sequence.

La estructura *sequence* se comporta de la misma forma que una película antigua de video, la cual está conformada por marcos (frames) donde cada marco va seguido detrás de otro. En cada *frame* hay una parte del código que se ejecuta de manera secuencial y se ejecuta de izquierda a derecha (ver figura 2.10).

Al igual que lenguajes de programación de texto en LabVIEW, en ciertas ocasiones se requiere ejecutar o no código y esto depende de una condición. Para esto existe la estructura llamada *Case* que tiene una entrada y se pueden agregar distintos casos e incluso un caso *default*. La elección de cada caso se condiciona a una entrada de datos, la cual puede ser de cualquier tipo [15].

LabVIEW también cuenta con ciclos condicionados que funcionan de manera muy similar a otros lenguajes de programación. Los principales ciclos de repetición en LabVIEW son las estructuras *for* y *while*. Se utiliza para repetir una parte de código tantas veces como la condición les permita. La estructura *while* está condicionado por un dato booleano y la *for* por un valor entero. En este último, se define el número de veces que se repetirá el ciclo [14-15].

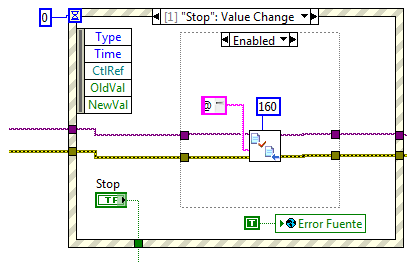


Figura 2.11 Muestra la estructura Event

Existe una estructura llamada *event* y es muy parecida a la estructura *case* ya que también cuenta con subdiagramas. La diferencia que tiene con la estructura *case* es que *event* congela la ejecución del programa que está en curso al momento que ocurra algún evento y ejecuta el código tomando en cuenta o el flujo anterior (ver figura 2.11). Para ello, se requiere contar con una lista de los eventos posibles, y esto puede ser desde apretar un botón en el panel frontal, el cambio de estado de un botón, etc [15].

2.5.4. Manejo de comunicación serial

Existen dos formas de establecer la comunicación serial con LabVIEW; una es mediante VI's muy específicos para dicha comunicación, y la otra es haciendo uso de la herramienta *Visual Instrument Software Architecture (VISA)* que fue usada en el proyecto [15]. La aplicación VISA fue desarrollada por distintos fabricantes y proporciona un software estándar para las operaciones de lectura, escritura y manejo de eventos. Lo que hace a VISA una gran herramienta, es el hecho de que se puede cambiar de estándar de comunicación sin que el programa principal se modifique demasiado. LabVIEW implementó VI's que funcionan con el estándar visa, y se llaman NI-VISA que puede establecer comunicación entre GPIB, comunicación serie, PXI, VXI o Ethernet. Los VI's usados para la comunicación VISA son: *VISA Configure Serial Port*, *VISA Write*, *VISA Read*, *VISA Close*, *VISA Bytes at Serial Port* [14]. El *VISA Configure Serial Port*, como su nombre lo indica se encarga de la configuración de los parámetros necesarios en la comunicación serie, estos son: El puerto a usar, baud rate, data bits, paridad y flow control. Tanto *VISA Write* como *VISA Read* son usados para escribir datos en el buffer de un dispositivo, así como para leer

datos del buffer. El último es necesario indicarle cuantos bytes serán leídos por lo que se hace uso de *VISA Bytes at Serial Port* que devuelve el número de bytes en el buffer [15]. Por último se tiene *VISA Close* Se usa para cerrar la sesión de VISA y así desocupar el puerto.

2.5.5. Sincronización

En ciertas aplicaciones se requiere sincronizar algunas tareas, como la adquisición de datos; ya que como se mencionó anteriormente la ejecución de un programa se da mediante el concepto *dataflow*. Para ello, existen diversas herramientas que permiten la sincronización de distintas formas. La que se empleó en este proyecto fue la función llamada “*rendezvous*” [15]. Para usar esta herramienta se requiere, antes que nada, configurar el número de tareas a sincronizar, posteriormente, agregar un *Wait a Rendezvous*, con esto se congela la ejecución hasta que se haya esperado el número de tareas que se ingresó inicialmente. Cuando el número de tareas es confirmado con las tareas que esperan, el flujo del programa continuara [15].

2.6. Sistemas Embebidos

Un sistema embebido es una agrupación de distintos elementos electrónicos interconectados de los cuales el más importante es el procesador. A diferencia de las computadoras personales, un sistema embebido se encarga de realizar una tarea muy específica por lo que su hardware está adaptado para desarrollar dicha tarea [16]. Ejemplos comunes de sistemas embebidos se pueden tomar de la vida diaria; desde un reproductor de música hasta el sistema de monitoreo de un automóvil [17]. Los sistemas embebidos juegan un papel muy importante en la investigación, ya que con ellos se puede realizar distintas labores.

2.6.1- Microcontrolador

Un microcontrolador es un circuito integrado programable que contiene un microprocesador, periféricos así como memoria. Este dispositivo también se conoce como

microcomputadora. Se puede encontrar con dos tipos de arquitecturas: Harvard o Von Neumann (ver figura 2.12). La arquitectura Von Neumann es aquella que tiene la memoria del programa junto con la memoria de datos, mientras que la arquitectura Harvard tiene las memorias separadas [16].

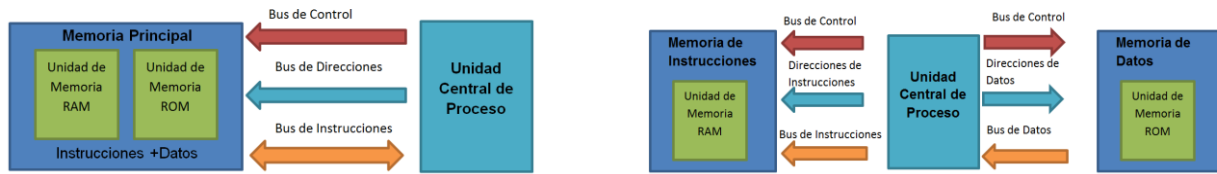


Figura 2.12 a) Arquitectura Von Neumann

b) Arquitectura Harvard.

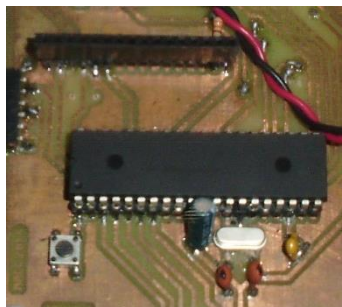


Figura 2.13 Microcontrolador PIC18F4550

Las características de los microcontroladores como el tamaño de bus, la memoria, periféricos, etc, pueden variar dependiendo del fabricante. La elección de cada uno de ellos se realiza en base de la aplicación y puede ser desde el control de una puerta automática hasta la computadora de vuelo de un satélite [17].

3. Materiales y Desarrollo

3.1. Características de los equipos disponibles en la sala de irradiación con rayos-x

Como se mencionó en el capítulo anterior, para poder realizar investigaciones de dosimetría se pueden emplear distintos métodos de medición. En este proyecto, se usa un electrómetro con una cámara de ionización como dosímetro de referencia. Además de estos instrumentos fundamentales de medición, son requeridos otros más, ya que se debe realizar un ajuste en las medidas que toma en cuenta los cambios de densidad del medio. Dichos cambios se dan por la variación de presión y temperatura, por lo que es necesario medir estas variables. A continuación, se enlistan los instrumentos utilizados:

Electrómetro SUPERMAX. Se trata de un electrómetro digital de pantalla táctil (ver 3.1) figura y cuenta con tres distintos modos de medición: a) continuo, el cual puede medir desde que se indica el inicio de la medición hasta que el usuario detenga la misma; b) Tiempo de colección, que mide un tiempo predefinido, desde un segundo hasta 24 horas, y c) Disparador que comienza la medición cuando detecta un cierto nivel de carga dentro de la cámara de ionización. Este dispositivo es ideal para la Dosimetría de bajas energías ya que cuenta dos escalas de medición, nC y pC y cuenta con una resolución de $0.001nC$ y $0.001pC$ para cada escala, respectivamente. Además, tiene una incertidumbre en las mediciones de $\pm 0.25\%$ lo cual lo hace ideal para mediciones de bajas energías. Este electrómetro cuenta con un puerto serie que usa el estándar RS232 el cual lo hace muy conveniente para poder manipularlo y adquirir mediciones desde un controlador que puede ser un PC o un microcontrolador. Cabe señalar que a pesar de que se controla por medio de comandos enviados por el puerto serie usando RS-232, estos no son comandos estándar, son comandos definidos por el fabricante. Los comandos de este dispositivo que fueron usados en este proyecto fueron los siguientes:



Figura 3.1 Electrómetro SUPERMAX

- <GID> Obtiene el nombre del electrómetro “SUPERMAX”
- <GCx> Obtiene la carga colectada para una canal específico
- <DZ> limpia parámetros del electrómetro llevándolo a cero
- <GZS> Obtiene el estado cuando se está llevando a cero
- <SRGx y> Ingresa la escala para un canal específico
- <SBSx y> Ingresa el voltaje para un canal específico
- <SCT x> Ingresa el tiempo de colección
- <STRC> Se inicia la medición
- <STPC> Se detiene la medición

Termómetro Fluke 1523 (figura 3.2). Este instrumento fue elegido gracias a su alta precisión en las medidas. Además cuenta con una resolución de 0.001°C que lo hace ideal para calibrar equipos y puede ser usado tanto en la industria como en laboratorio. También cuenta con varias funciones definidas como lo es la integración de la temperatura en un periodo de tiempo. Este instrumento cuenta también con una interfaz de comunicación RS232 y puede ser controlado por comandos estándar SCPI.



Figura 3.2 Termómetro Fluke 1523

El set de comando de este dispositivo que fue usado en este proyecto se enlista a continuación:

- READ? Solicita la información de la temperatura en el momento previo a la petición.
- IDN? Pide al termómetro se identifique
- CLS Borra cualquier dato guardado en la memoria del termómetro

Barómetro Druck DPI 142. Se trata de un dispositivo de alta precisión de la marca General Electric, el cual cuenta con una incertidumbre en la medición de 0.03% sobre cada medida (figura 3.3). Cuenta con distintas unidades de medición, además de unidades aeronáuticas y realiza las medidas a través de un transductor resonante de silicio. También cuenta con interfaz de comunicación RS-232 para la comunicación con otros dispositivos, contando con comandos SCPI. En la siguiente lista se describen los comandos usados:



Figura 3.3 Druck DPI 142

- :UNIT <name> Selecciona las unidades que se ocuparán
- INST: SENS: PRES? Solicita la medición de la presión.
- CLS Borra cualquier dato guardado en la memoria
- IDN? Pide al barómetro que se identifique

Fuente de Rayos-X YSMART 160E/1.5. Este dispositivo es el que se encarga de suministrar la potencia requerida en el tubo de rayos-X, puede suministrar una potencia máxima de 960W (figura 3.4). Cuenta con una selección de voltaje que puede ir desde 10kV hasta 160kV con incrementos de 1kV. Su corriente de operación es entre 2.0 mA y 6.0 mA. En la figura 3.1.4 se muestra la fuente usada en este trabajo. Adicionalmente, posee un panel de control en la parte frontal para seleccionar el voltaje, la corriente y el tiempo de disparo e iniciar y apagar el disparo.



Figura 3.4 Fuente para tubo de rayos-X YSMART 160E/1.5

Cuenta con una interfaz de comunicación RS232 con un conector hembra lemo. En este instrumento, los comandos de comunicación no son SCPI y tienen su propia estructura. A continuación se explica brevemente dicha estructura como se muestra en la figura 3.5.

Start Char 1 byte	Header			Data n bytes	Tail 1 byte
	Length 3 bytes	ID (src) 2 bytes	CMD 1 byte		
@ (0x40)	009 (0x30, 0x30, 0x39)	RB ⁵ (0x52, 0x42)	B (0x42)	0	EOR (0x1A)

Figura 3.5 Estructura de los comando de la fuente de rayos-X

La estructura del comando consta de una cabecera+datos+Cola donde todos los caracteres tienen una longitud de 8 bits y están contenidos en la tabla ASCII. En la cabecera se incluye un carácter de inicio (@, 0x40), la longitud del comando descrita en tres caracteres (en cada carácter se pone el número correspondiente), el identificador cuyo comando de envío es “RB” y el tipo de

fenólicas de doble cara, una pantalla LCD, pulsadores y resistencias. El microcontrolador es de 8 bits con arquitectura Harvard con una memoria de 2kB en RAM y cuenta con un puerto USB.

Para lograr la comunicación se ocuparon 3 cables con configuración null-modem con conectores DB-9 macho y hembra. Para la fuente también se ocupó un cable con configuración null-modem, pero con un conector DB-9 macho y conector LEMO FGG.0B.307.CLAD52Z que es el tipo de conector que se requiere para este dispositivo. En la figura 3.6 se muestra el diagrama de conexión para el cable de la fuente.

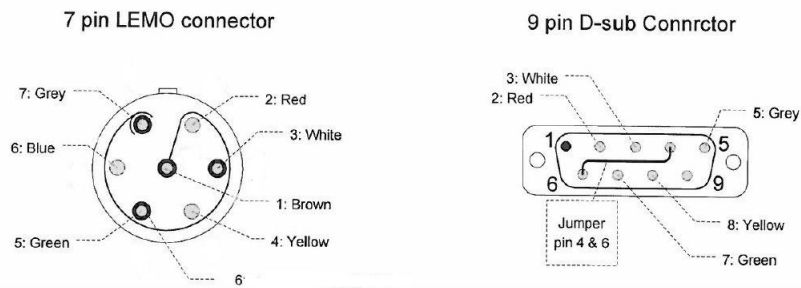


Figura 3.6 Diagrama de conexión para el cable de comunicación de la fuente de rayos-X

Por otra parte, para el control de la mesa se requirió de un cable con dos conectores DB25 que se encarga de llevar las señales de control al driver de los motores.

3.3. Diseño del programa de control de los equipos de dosimetría.

La programación de la interfaz de este proyecto se llevó a cabo en el entorno de programación LabVIEW versión 2012. Se usó el concepto subVI's para manejar un código más organizado y estructurado. Se construyó una propuesta de interfaz y, una vez desarrollada, se creó el código correspondiente. En la figura 3.7 se presenta una primera versión de la interfaz. En la propuesta de la interfaz se incluyeron los siguientes aspectos:

- Tener controles para indicar el voltaje y la corriente deseada en el experimento así como el tiempo que dura.
- Poseer indicadores que muestren las variables de presión, temperatura, carga colectada, y temporizador en todo momento
- Contar con unidades dinámicas donde sea requerido.
- Tener con ventanas o marcos que puedan cambiarse entre sí para mostrar distintos eventos: una ventana que muestre la situación de la fuente de rayos-X; una ventana para mostrar el estado actual de la presión. También, se cuenta con una ventana que muestra el estado de la temperatura; otra que muestra el estado de la carga colectada y controles para manipular el electrómetro, y finalmente se cuenta con otro marco que muestra la carga corregida así como el factor de corrección.
- Contar con dos botones, lo cuales sirven para iniciar el experimento y para detenerlo. Hay que mencionar que estos botones son de gran importancia, y deben ser colocados en un lugar estratégico.

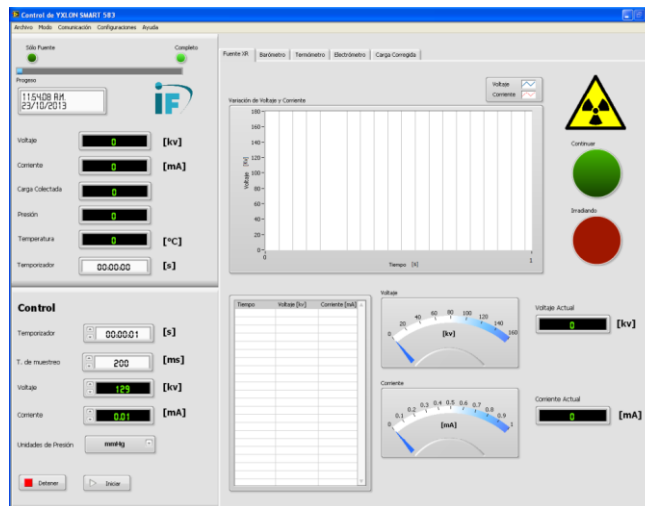


Figura 3.7 Primera versión de interfaz.

Una vez teniendo la interfaz, se procedió a realizar el código y se hicieron modificaciones donde fue necesario. A continuación, en la figura 3.8 se presenta un diagrama de flujo donde se muestra la idea general del programa principal, el cual será explicado por partes.

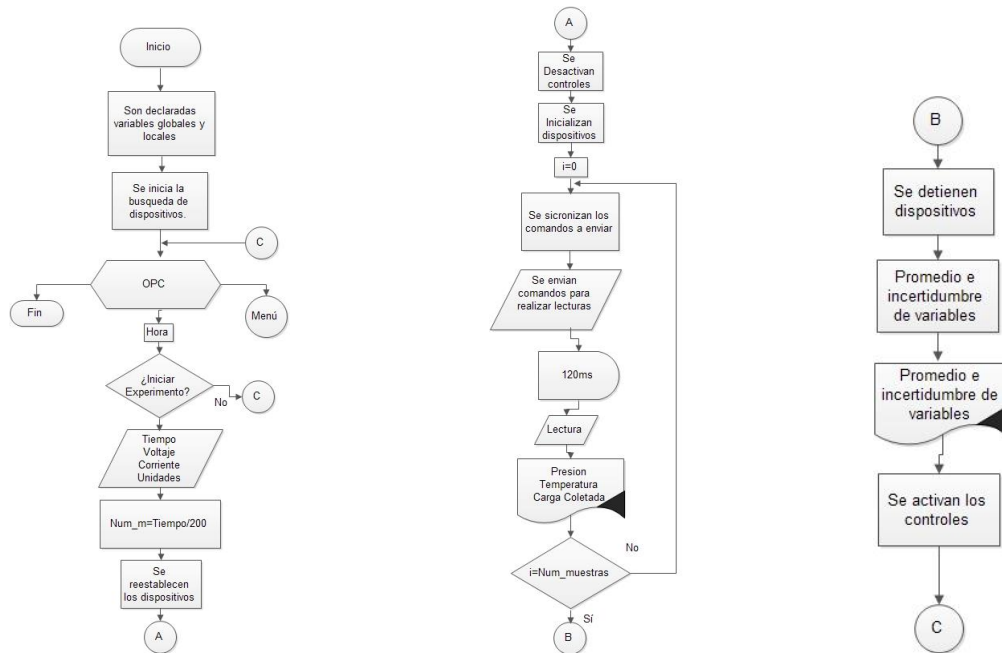


Figura 3.8 Diagrama de flujo general del programa.

En este proyecto fue necesario declarar variables globales por lo que se crearon dos archivos: a) se incluyen variables de comunicación, y b) se incluyen variables generales. Las variables de comunicación son globales, pues se usan en distintos subVI's. Estas variables se usan para configurar la comunicación serie por lo que se crearon variables donde se guardan los siguientes parámetros: puerto de comunicación, tasa de baudaje, bits de datos y paridad; creando una variable por cada parámetro. Se crearon variables locales de cada control e indicador ya que fue indispensable usarlos en distintos lugares del programa.

Para realizar la búsqueda de los dispositivos, se creó un subVI que verifica los puertos de comunicación disponibles; una vez encontrados se envían comandos para identificar cada dispositivo. Si la respuesta corresponde a la esperada, se guarda el puerto en la variable global de comunicación relacionada con el dispositivo. Es necesario mencionar que se envían comandos de una matriz, donde la posición de cada elemento corresponde a un dispositivo, así se conoce el dispositivo en cuestión. Se muestra el diagrama de flujo básico en la figura 3.9

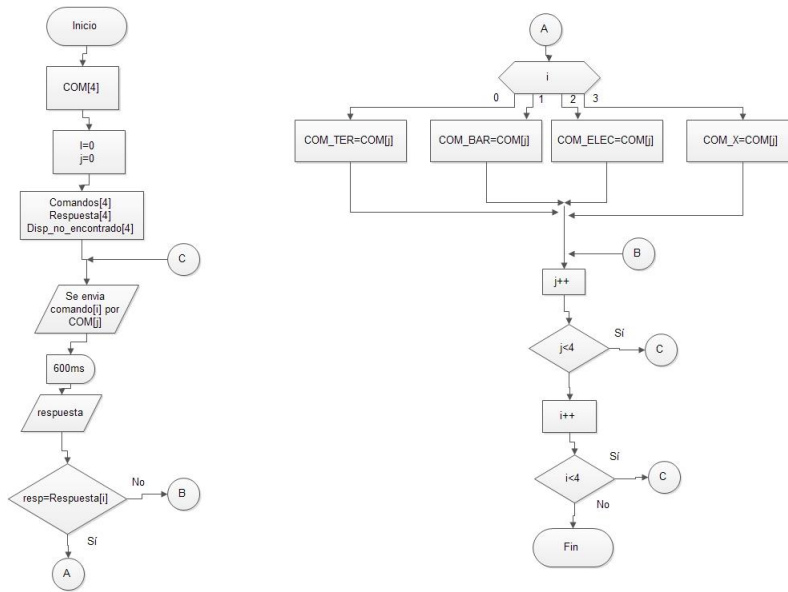


Figura 3.9 Diagrama de flujo para identificar los dispositivos

Un menú situado en la parte superior de la interfaz fue agregado donde se muestran distintas opciones que pueden ser desde utilizar solamente la fuente o todos los instrumentos, generar un reporte en Excel, opciones de comunicación y salir. En el caso de las comunicaciones, existen dos opciones disponibles: comunicación manual y automática. En la primera se seleccionan todos los parámetros manualmente, mientras que en la segunda se requieren de 3 parámetros: tasa de baudaje, bits de datos y paridad. En la Figura 3.10 se muestra la ventana donde se configuran los puertos y se realiza la búsqueda automática.

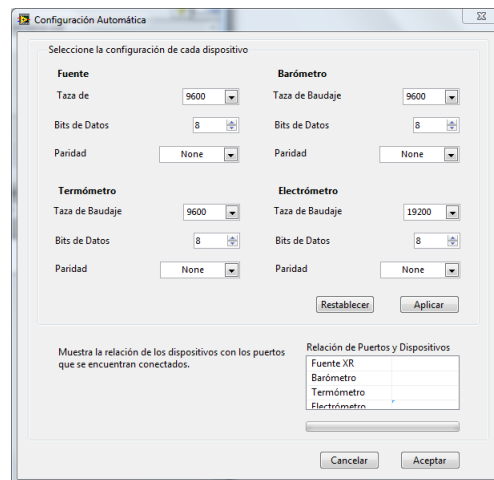


Figura 3.10 Ventana de configuración de comunicación automática

Antes de iniciar el experimento el usuario debe seleccionar las variables de voltaje y corriente de la fuente, las unidades de presión para el barómetro, la escala deseada del electrómetro y del tiempo de exposición usando los controles correspondientes. La primera tarea que se realiza una vez presionado el botón de inicio es calcular el número de muestras que se tomarán dividiendo el tiempo que se ingresó entre la frecuencia de muestreo. También se reestablecen los instrumentos.

Una vez que el experimento se inicia no se podrá hacer uso de ningún control hasta que este termine, o bien, sea cancelado. Para cumplir esto se debe desactivar los controles y activar únicamente el botón de paro. Se emplean las propiedades de cada control para desactivarlos y dejarlo “invisible” al usuario.

También se inicializan todas las variables locales además de ajustar las escalas de los instrumentos virtuales. En este punto también se establecen las unidades a ocupar durante el experimento enviando comandos de configuración de unidades donde sea necesario. Cuando se realiza esta configuración el programa verifica si no existen errores en los instrumentos; estos errores pueden ser problemas de comunicación o problemas internos de los instrumentos. Por ejemplo, que no se encuentre bien conectada la sonda del termómetro o que la puerta de plomo se encuentre abierta. Si existen problemas, el programa se encarga de notificarlos y cancela el experimento. Si todo está en orden se inicia el electrómetro, inmediatamente después se inicia la fuente.

Una vez comprobado que todo está bien, el flujo del programa entra en el ciclo donde serán adquiridos los datos. Cuando se sale del ciclo se restablecen los controles y el botón de paro es bloqueado. También se cierran las comunicaciones y se limpian las variables de error para poder realizar un nuevo experimento.

3.3.1 Comunicación con los dispositivos

Como ya se dijo anteriormente, el tipo de comunicación que se usó en este proyecto es la comunicación serial RS-232. A continuación se describirá de manera general como se realizó la comunicación con los dispositivos. En casi todo el programa se emplearon subVI's para lograr la

comunicación, estos subVI's requieren de los parámetros de configuración y del mensaje a enviar. Como resultado, se tiene el mensaje almacenado en el buffer. En la Figura 3.11 se muestra la estructura de este subVI, se puede apreciar que inicia la comunicación, envía el mensaje, tiene una espera, lee el buffer, limpia errores y cierra la comunicación. Es muy importante que este subVI cierre la comunicación, pues de lo contrario el programa entraría en conflicto al no desocupar el puerto. Además de este subVI, que es el que predomina más en el programa, existen otros y dependen de la tarea requerida. Por ejemplo hay casos en que solo es necesario enviar el comando sin necesidad de recibir respuesta.

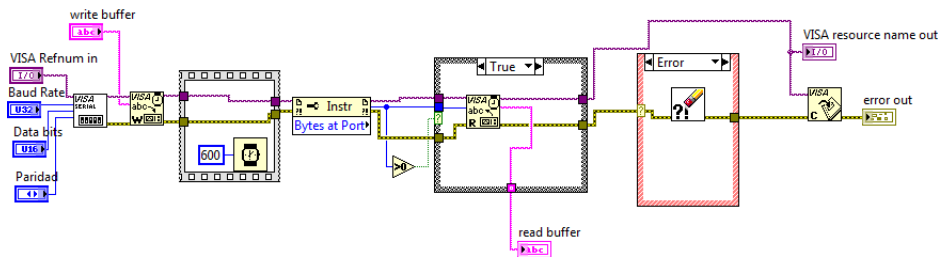


Figura 3.11 Muestra el subVI llamado *W_AND_R.vi* que se usa para enviar mensajes a través del puerto serie.

Sin duda alguna, la parte más importante en el programa es la comunicación cuando se adquieren los datos de los diferentes dispositivos. La estructura que se encarga de cumplir esta tarea es diferente a las estructuras de comunicación que se han mencionado. Todo empieza con la configuración de la comunicación de VISA, esta se realiza antes de entrar al ciclo de adquisición. Una vez entrando al ciclo, un aspecto de gran importancia es la sincronización, pues es indispensable enviar los comandos lo más sincronizado posible. Para la sincronización se ocupó del subVI llamado *rendezvous*. Lo primero que se crea es un *rendezvous* de tamaño 4 pues son cuatro dispositivos a los que se les envían comandos de forma sincronizada: electrómetro, barómetro, termómetro y fuente de rayos-X. Una vez creado el *rendezvous* se crea un subVI para cada dispositivo dónde tiene como característica principal el subVI de espera del *rendezvous*, el cual es seguido de la herramienta de escritura de VISA para que de esta manera se envíen los comandos en sincronía. En la figura 3.12. se observa esta estructura.

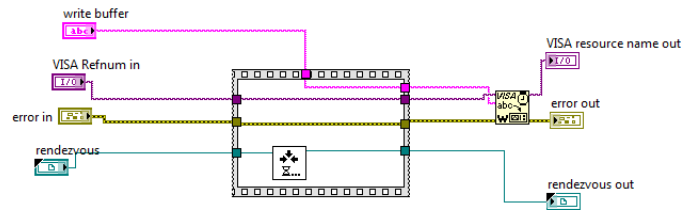


Figura 3.12 Comunicación dentro del ciclo de adquisición de datos

Una vez que se envían los comandos, el flujo del programa se detiene durante 120ms antes de leer el buffer donde se almacenó la respuesta de cada dispositivo. En la figura 3.13 se aprecia el bloque de código que realiza esta tarea. Después de la espera se debe leer el buffer, para esto se puede hacer ocupando el subVI de visa de lectura. Por último, se debe cerrar el puerto una vez que se ha salido del ciclo de comunicación.

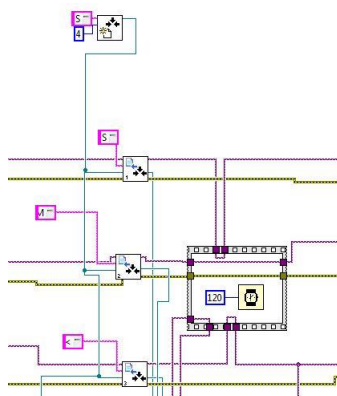


Figura 3.13 No se lee ningún dato del buffet hasta que hayan transcurrido 120ms.

3.3.2 Comandos y configuración de dispositivos.

Los comandos en este proyecto juegan un papel muy importante, ya que con ellos se manipulan los dispositivos. Los comandos se usan en varias partes de programa: al identificar los dispositivos, al reiniciarlos, al configurarlos, al configurar el inicio de las medidas (para el electrómetro y la fuente de rayos-X) y en la adquisición de datos. Se debe recordar que solo el

termómetro y el barómetro utilizan comandos SCPI mientras que la fuente de rayos-X y el electrómetro utilizan comandos con su propia estructura.

En la búsqueda de los dispositivos los comandos son enviados a los dispositivos esperando una respuesta conocida para cada dispositivo, la respuesta varía en cada dispositivo, puede ser desde el número de serie, modelo y fabricante del dispositivo hasta una lista detallada del mismo.

Para configurar los dispositivos de deben definir las unidades a usar y el tiempo de duración del experimento. Esto solo se realiza para el barómetro y el electrómetro pues son los únicos que lo requieren. La configuración que se realiza para el barómetro solamente es el cambio de unidades. Lo primero que se hace una vez que el usuario selecciona el tipo de unidades, es guardarla en una variable global pues se usará en distintas partes del programa. Una vez creada esta variable se ingresa en una estructura *case* pues se debe agregar un texto específico para crear el comando correspondiente. En la Figura 3.14 se puede observar la estructura.

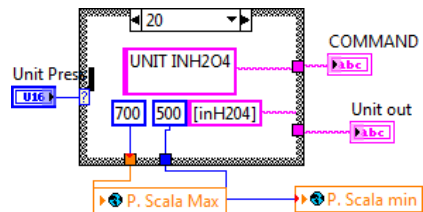


Figura 3.14 Muestra las variables modificadas cuando se realiza un cambio de unidades

Además de la creación del comando, el subVI crea las escalas para los medidores de la interfaz y una variable de texto para identificar las unidades que serán usadas. Cuando se crea la variable, se puede enviar el comando de configuración de unidades.

Para el caso del electrómetro, es un poco más complejo ya que se deben realizar configuraciones de tiempo, según el modo de colección de datos, intervalo de medición y seleccionar el modo de operación.

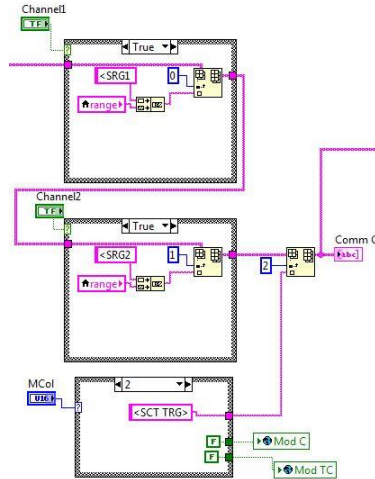


Figura 3.15 Muestra la creación de los comandos que se ocuparan para configurar el electrómetro.

En este caso, una vez que el usuario ingresa las variables, el subVI de configuración se encarga de guardar todo lo necesario en variables globales. También realiza una conversión de número decimal a un string para así poder formar la parte del comando que se requerirá para dicha acción. Cada comando que se crea, se guarda en una cadena de strings para después pasar a un ciclo donde serán enviados los comandos pues se deben enviar dos comandos por canal y un comando para elegir el modo de colección (figura 3.15).

En el subVI encargado de la configuración del electrómetro, también se realiza una inspección, pues el usuario debe escoger al menos un canal para poder utilizar dicho dispositivo, si no se escoge ningún canal, se advierte al usuario.

3.3.3 Adquisición de datos

Antes de pasar al ciclo de adquisición de datos es necesario configurar el electrómetro y la fuente de rayos-X, pues estos instrumentos registran el tiempo de exposición. Pevio al inicio del experimento, en el electrómetro se verifica que no existan errores en todos los instrumentos para posteriormente iniciar el proceso de medición enviando un comando. Cabe destacar que esta última acción se realiza dependiendo del modo de operación que haya sido seleccionado previamente. Si se seleccionó el modo continuo, es necesario que el usuario inicie y finalice el electrómetro cuando

el deseo en cualquier parte a lo largo de la adquisición, por lo que se agrega una estructura de *case* que depende del estado de los botones que se encuentran en la sección de la interfaz del electrómetro, donde uno es para iniciar la colección y el otro es para pararlo. El código que realiza estas operaciones se muestra en la figura 3.16.

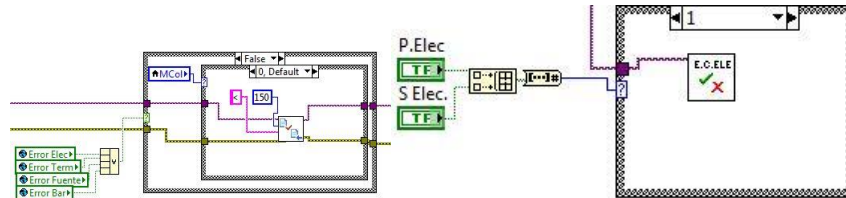


Figura 3.16 a) Muestra la parte de código para iniciar el electrómetro b) se muestra la parte de código dentro del ciclo de adquisición donde se puede iniciar o parar el electrómetro si es que se eligió el modo de colección continuo.

Para iniciar la fuente solo se requiere de enviar el comando señalado por el fabricante, indicando el voltaje y la corriente de operación. El comando es construido en la etapa de configuración de los dispositivos y enviado antes de entrar a ciclo de adquisición. Al igual que el electrómetro, se verifican errores para poder continuar con la prueba (figura 3.17).

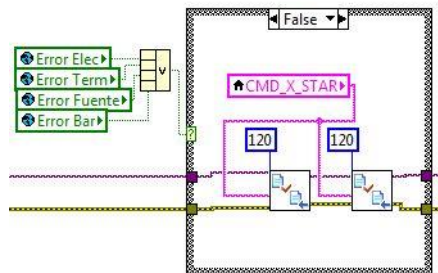


Figura 3.17 En esta parte del programa se envía el comando de inicio para la fuente el cual incluye los parámetros con los que operará la fuente de rayos-X

El inicio y la detención de colección de datos por el electrómetro depende del modo de tiempo de adquisición. Para el modo continuo de adquisición, es necesario enviar el comando de paro seguido de un comando de inicio, esto último no iniciará al electrómetro, sino que borrará la última lectura. Los instrumentos se detienen con la acción de un comando, esto se realiza fuera del ciclo de adquisición, sin embargo puede existir el caso en el que el usuario detenga el experimento

antes de tiempo, por lo que estos instrumentos son detenidos dentro del ciclo de adquisición en una estructura de eventos (ver Figura 3.18)

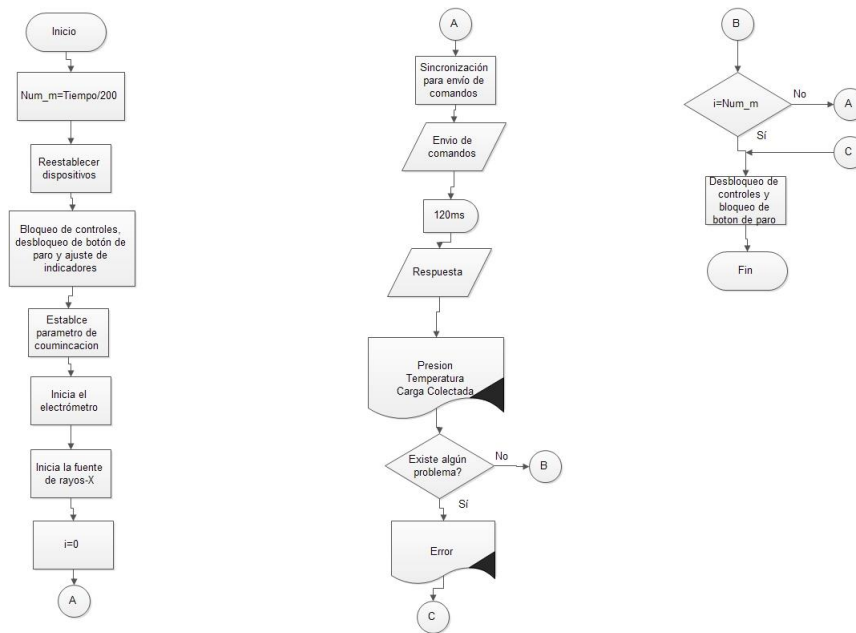


Figura 3.18 Diagrama de flujo del ciclo de adquisición de datos

Los dispositivos se encargan de adquirir los datos continuamente, sin embargo, habrá ocasiones en los que los datos no se reciban correctamente, debido a factores internos del equipo o simplemente por fallas en la transmisión. Para evitar estos problemas, antes de que el dato sea aceptado como un dato válido, se revisa para ver si llegó correctamente. La inspección se realiza midiendo el tamaño de la cadena, pues se espera un tamaño fijo para cada variable. Si el tamaño de la cadena que se recibió corresponde con el esperado, el dato es aceptado y guardado, por el contrario, si la longitud no corresponde con la esperada el dato se rechaza y se mantiene el dato anterior para esa adquisición.

Mientras que los demás dispositivos están adquiriendo datos en el ciclo de adquisición, el programa verifica el estado de la fuente de rayos-X. Esto se logra con un comando que puede devolver una respuesta con hasta 16 posibles interpretaciones, la cuales pueden ser anomalías en el tubo de rayos-X como calentamientos del ánodo o el cátodo o ambos, que la puerta de seguridad se encuentre abierta, que la llave de seguridad se encuentre bloqueada, entre otros. Esta acción es

de vital importancia pues debe advertir al usuario por si algún error llegase a ocurrir. Entonces en el subVI donde se recibe la respuesta del comando se despliega un mensaje al usuario por cada error que pueda ocurrir (ver figura 3.19).

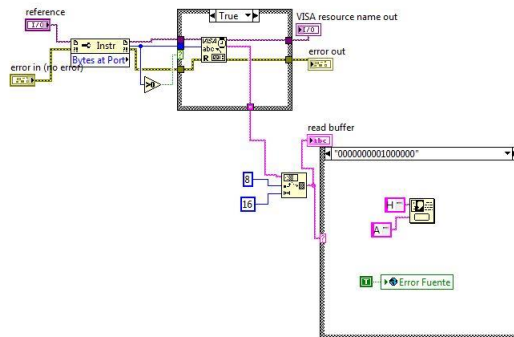


Figura 3.19 Muestra las posibles respuestas del comando donde se pregunta a la fuente por su estado y la estructura case mostrando las acciones a seguir.

En este mismo bloque se pregunta al tubo por el voltaje y corriente suministrados al tubo de rayos-X, para así tener una mejor visualización de la operación del tubo de rayos-X. Ambas tareas se realizan en este bloque, por lo que es necesario alternar los comandos. Cuando el contador del ciclo de adquisición es un número par el programa pide el estado de la fuente, mientras que cuando es un número impar pregunta por el voltaje y corriente.

3.3.4 Datos de salida

Cuando se sale del ciclo de adquisición, el programa realiza el promedio de todas las muestras que se obtuvieron así como la desviación estándar de cada una de las variables físicas medidas. Esto se logra con las herramientas estadísticas que con las que cuenta LabVIEW. Ya se ha mencionado la importancia en la corrección de datos por presión y temperatura. Cuando se tienen los datos promediados se obtiene el factor de corrección correspondiente. Para ello, se elaboró un subVI que obtiene dicho factor. Contando con el promedio de cada variable así como su desviación estándar, el programa debe realizar un análisis de incertidumbre en las medidas, por lo que se usa la información proporcionada por el fabricante relativa a la incertidumbre que tienen los dispositivos. Así se emplean todos los datos y se obtiene un valor para la incertidumbre del

valor obtenido. Para cada variable se obtiene su incertidumbre utilizando la desviación estándar obtenida y la incertidumbre de medición proporcionada por el fabricante de cada dispositivo. Las incertidumbres se calcularon según la ecuaciones 18 con un factor de cobertura igual a 1 que corresponde a un nivel de confianza de 68%. Se requiere de un factor que corrija las variaciones de densidad dentro de la cámara de ionización. Estas correcciones, como se explicó anteriormente, se obtienen con la presión y la temperatura. En este programa se recaban todos los datos y se promedian. Una vez promediados los datos se obtiene el factor de corrección mediante la ecuación 16.

En cada experimento se debe contar con un reporte de todos los datos que se adquirieron para su posterior análisis. Se deben guardar todos los datos obtenidos en cada experimento así como las variables. Para guardar todos los datos desde el primer experimento hasta el último, se emplean matrices de datos. Se crean matrices como variables globales, pues estas se ocuparán cuando se cree el reporte. Las variables que se guardaran son: voltaje y corriente de la fuente de rayos-X, las variables de presión, temperatura, carga y carga corregida cada una con sus incertidumbres y unidades, el factor de corrección, la carga corregida así como su incertidumbre, también se agrega una variable de notas, la cuales se pueden agregar si el usuario así lo desea; además se añade una matriz de errores que ayudará a identificar experimentos fallidos. Las matrices son llenadas al final del programa, cuando se sale del ciclo de adquisición. El reporte en Excel se crea cuando el usuario así lo solicita, desde el menú, o presionado ctrl+E. Para la creación del reporte se emplearon herramientas especiales que LabVIEW tiene para trabajar documentos en Excel y Microsoft.

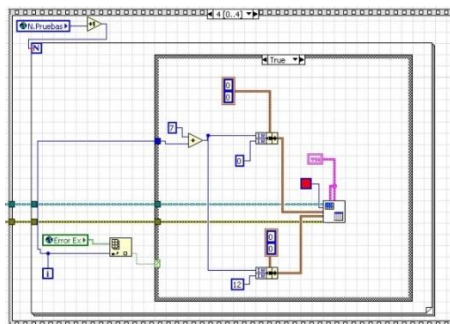


Figura 3.20 Código donde señala las pruebas que tuvieron algún error

La estructura básica es crear el archivo de Excel y acomodar las matrices por columnas en el reporte (ver figura 3.20). Pero antes, se ingresa el título del archivo así como los escudos de la UNAM y del Instituto de Física, además de la hora en la que se creó el reporte. Cuando todo ha sido acomodado se entra a un ciclo *for* donde se pregunta por cada elemento de la matriz de errores, si existiera una anomalía todo el renglón se marcará con color rojo, indicado así el error.

3.4. Diseño del controlador de la mesa de experimento

La mesa que se usará para colocar objetos para su estudio, fue desarrollada en el Instituto de Física. Esta mesa se puede desplazar en dos ejes por medio de dos motores (ver figura 3.21).



Figura 3.21 Mesa de experimentos de dosimetría de bajas energías.

Los motores que se utilizaron para el desplazamiento de la mesa requieren para su funcionamiento unos controladores especiales. Se pueden emplear los controladores para poder mover los motores, pero el movimiento es con parámetros definidos como la velocidad, rigidez, etc. Para evitar esto, fue necesario construir un controlador externo que envía una señal al controlador indicando cuando moverse. Al controlador se le definen parámetros que no se podrán cambiar con el controlador externo. Las señales que envía el controlador externo al controlador del motor son: clear, sentido, pulsos. Existen dos señales que ingresan al controlador externo e indican cuando la

mesa ha llegado a algún límite en los ejes. El diagrama de tiempos que se debe seguir para poder mover cada motor es el que se muestra en la figura 3.22

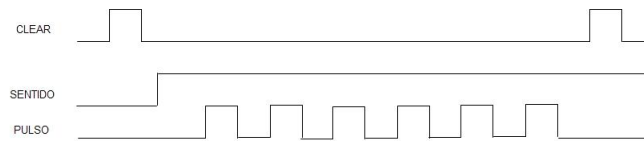


Figura 3.22 Secuencia a seguir para mover un motor

Como se puede apreciar en los motores, un pulso en CLEAR debe ser enviado en primer lugar, para posteriormente establecer el sentido de giro del motor e iniciar el tren de pulso que moverá al motor, finalmente se envía otro pulso para limpiar errores. Para lograr esta trama, se desarrolló un programa en C usando el compilador CCS. Se ocuparon los puertos: A, C, D y E para la interfaz de los pulsadores, indicadores, así como para las señales de salida, y el puerto B se ocupa para conectar una pantalla de cristal líquido que indica las acciones que se están realizando. Se empleó el timer0 del microcontrolador para la generación de los pulsos. El controlador tiene un total de 6 botones, cuatro para elegir la dirección y sentido en que se mueve cada eje, uno para entrar al menú y un último funciona como botón de acceso (entrada).

Se programó un menú con distintas opciones en las que el usuario puede elegir. Estas opciones son: “Ajuste Grueso”, “Ajuste fino”, “Ir al origen”, “salir del menú”. Este menú es rotatorio (ver figura 3.23). La programación del microcontrolador, así como los esquemáticos se incluyen en los apéndices de este trabajo.



Figura 3.23 Control para la mesa de dos ejes diseñada en el IFUNAM

El diseño del controlador se divide en dos partes, una placa fue construida para los botones y otra para desplegar los datos y todo el control de la mesa. En la figura 3.24 se presentan los diseños del hardware.

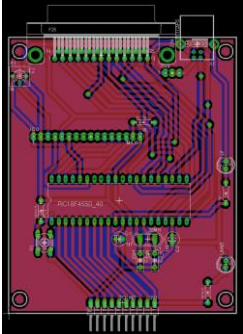
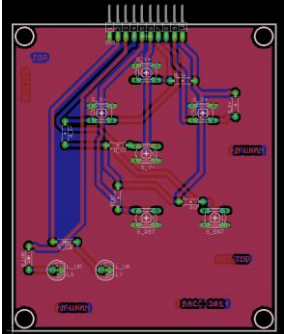


Figura 3.24 a) Diseño del controlador de la mesa



b) Panel de botones para el controlador de la mesa

4. Resultados

4.1. Construcción del sistema de control por computadora

Los resultados obtenidos en este proyecto se describen a continuación. Se muestran resultados en la versión final de la interfaz, resultados de optimización en tiempo y reducción en la incertidumbre. La versión de la interfaz que se obtuvo como resultado se muestra en la Figura 4.1.

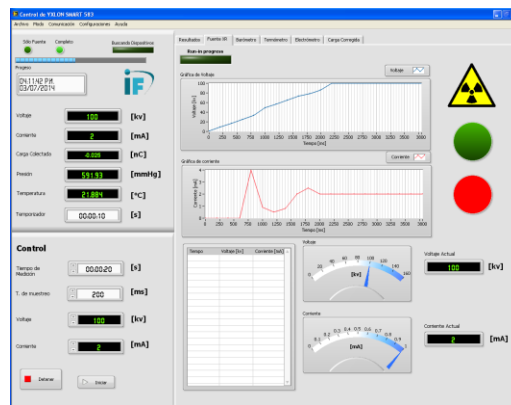


Figura 4.1 Versión final de la interfaz

Esta interfaz puede manipular el tiempo de cada experimento, sincronizando la adquisición de cada experimento. También tiene la capacidad de manipular el tubo de rayos-X.

4.1.1 Funcionamiento del sistema de control

A continuación en la figura 4.2 se explica cada una de las partes que integran la interfaz:



Figura 4.2 Última versión de la interfaz

- 1. Control.** En esta parte de la interfaz el usuario debe ingresar los valores generales para su experimento los cuales son el tiempo de adquisición, el tiempo de muestreo que puede ir de 200 a 1000 ms; el voltaje y la corriente de la fuente de rayos-X.
- 2. Indicadores Generales en Tiempo Real.** En esta sección se muestran los valores en tiempo real de los instrumentos cuando la adquisición ha sido iniciada. Además se incluye un cuadro donde se muestra la fecha y hora; la cual es mostrada en todo momento. La fecha y la hora es muy importante ya que se pueden hacer registros manuales independiente de la interfaz.
- 3. Marcos.** Son mostrados diferentes marcos relacionados a cada instrumento, además se añade uno más para la carga corregida y otro para mostrar los resultados finales (promedio e incertidumbre). Cabe mencionar que en el barómetro y en el electrómetro se pueden

manipular otros parámetros como las unidades a usar o el modo de colección del electrómetro.

- 4. Menú.** En el menú el usuario puede encontrar aplicaciones para la configuración de los dispositivos. También en el menú se generan los reportes de Excel. Una ayuda es añadida para que el usuario comprenda cada parte del dispositivo.

Cuando se inicia la interfaz muestra una barra de led que está encendida, pues mientras está permanezca encendida indicará que se está buscando los dispositivos que están conectados. Al final de la búsqueda la barra se apagará y mostrará, si es que es necesario, los dispositivos que no se encuentran conectados como lo muestra la Figura 4.3

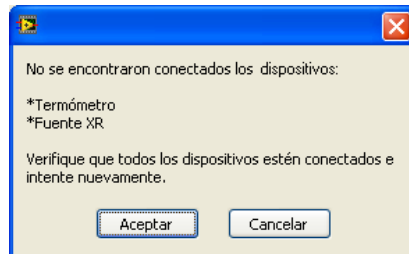


Figura 4.3 Este error señala que no se encuentran conectados 2 dispositivos: termómetro y electrón,

La asignación de los dispositivos a los puertos se puede hacer de manera manual así como automática. En ambos modos el usuario debe ingresar los parámetros de configuración de la comunicación: velocidad, tasa de baudaje, bits y paridad (ver figura 4.4). Cuando el usuario ingresa estos parámetros, estos se guardan para mantener la configuración para un próximo experimento.

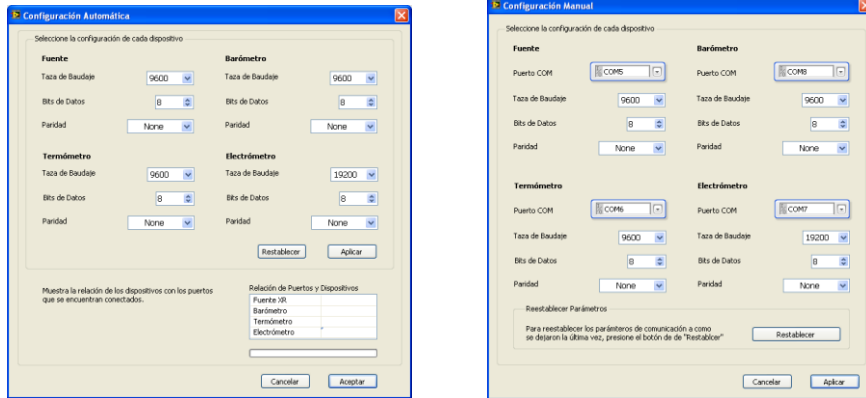


Figura 4.4 Modos de comunicación con los dispositivos: a) modo automático (busca los dispositivos en los puertos), b) Manual (el usuario ingresa manualmente los puertos).

Una vez que se apaga la barra indicadora de búsqueda el usuario puede ingresar los parámetros para poder llevar a cabo el experimento. Estos parámetros se ingresan en el panel principal y se pueden ingresar también, en los marcos correspondientes al barómetro y al electrómetro

Cuando el usuario ha ingresado todos los parámetros de inicio puede presionar el botón para iniciar el experimento. Mientras que esto sucede el programa se encuentra indicando la hora y la fecha. Si existe un error al no detectar conectado algún instrumentos, o fallas en los sensores de los mismos, la interfaz lo mostrará con un mensaje especificando el error, como se puede apreciar en la Figura 4.5

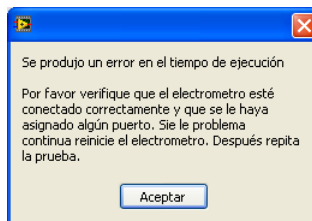


Figura 4.5 Muestra un mensaje de error al señalar que existe un problema con el electrómetro.

Existe la posibilidad de que se presenten problemas ya sea en la fuente o en el tubo de rayos-X. A continuación se da una lista de los posibles problemas que pueden surgir:

1. El usuario ha presionado el botón de paro desde la fuente.
2. Error en la lámpara del panel.

3. Error con la lámpara del tubo.
4. Error en el voltaje de salida
5. Error en la corriente de salida.
6. El ánodo y el cátodo se encuentran con una temperatura muy elevada.
7. La presión interna del tubo de rayos-X es demasiado alta.
8. La puerta se encuentra abierta o la llave de seguridad se encuentra en modo de no disparo.

Si todo se encuentra en orden, la interfaz inicia con la adquisición y va desplegando los datos (ver figura 4.6). Hay que mencionar que se debe señalar al usuario por medio de la vista y el oído que se está exponiendo radiación, por lo que al programa se agregó indicadores visuales tipo leds que indican cuando se está llevando a cabo un disparo, además de esto, cuando el experimento está en marcha una alerta sonora indica el disparo. Se puede activar o desactivar la alerta sonora, pero es recomendable, por protección, mantenerla activa en cada experimento.

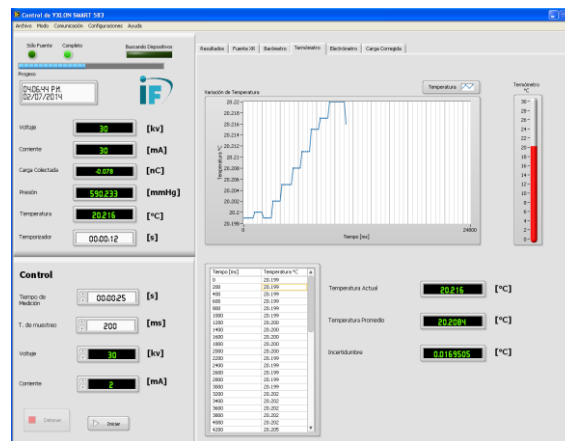


Figura 4.6 Interfaz adquiriendo datos

El experimento se llevará a cabo siempre y cuando todo se encuentre en orden, es decir, no ocurra algún error inesperado durante la prueba. Si un error se llegará a presentar, la interfaz lanzará un mensaje advirtiéndolo dicho error. Un error típico que podría presentarse es que la puerta se haya abierto durante la adquisición de datos y se haya tenido que cancelar el experimento, en la Figura 4.7 podemos ver un ejemplo de ello.

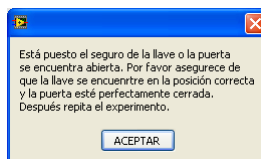


Figura 4.7 Mensaje de error muestra que la puerta fue abierta durante el experimento.

Si todo está bien el experimento continuará hasta finalizar el tiempo programado por el usuario hasta mostrar los datos finales en el marco de resultados. Se muestra en un marco para recalcar el resultado de la carga ya corregida así como su incertidumbre (ver figura 4.8).



Figura 4.8 Resultados finales después de un experimento.

El usuario puede realizar tantas pruebas como él lo desee. Un reporte en Excel se puede crear cuando el usuario así lo desee. Cuando el reporte en Excel sea creado, desplegará todos los resultados de los experimentos llevados a cabo. En este archivo se señala la hora y la fecha de cuando se creó el archivo y se despliega en tablas todos los datos. Si algún error surgió durante el experimento, todos los datos correspondientes a ese experimento son señalados iluminando las celdas correspondientes de color rojo (ver figura 4.9).

Reporte de Adquisición
02/07/2014 05:44:30 p.m.

Tiempo [s]	Voltaje [kV]	Corriente [mA]	Presión [mmHg]	Incertidumbre[mmHg]	Temperatura- C	Incertidumbre- C	Carga [pC]	KTP	Carga Corregida[pC]	Incertidumbre[pC]	NOTAS
5	30	2	590.141051	0.017759	19.94912	0.015197	-0.026	1.278879	-0.032251	-0.000025	
5	30	2	590.162807	0.017689	19.84836	0.015223	-0.025	1.278394	-0.03196	-0.000025	
5	30	2	590.168749	0.017828	19.861	0.015179	-0.024	1.278452	-0.030683	-0.000023	
5	30	2	590.158512	0.017749	19.81404	0.015402	-0.025	1.278252	-0.031956	-0.000025	
5	30	2	590.144544	0.017851	19.881	0.015679	-0.025	1.278574	-0.031964	-0.000025	
5	30	2	590.150805	0.017828	19.94932	0.015057	-0.025	1.278859	-0.031971	-0.000024	
5	30	2	590.159517	0.017767	19.96452	0.016025	-0.016	1.278906	-0.029463	0.000016	
10	30	2	590.156817	0.017538	19.88516	0.015395	-0.068	1.278478	-0.088537	0.000067	
10	30	2	590.165615	0.017769	19.91174	0.028004	-63.64	1.278663	-81.3741	0.11445	
10	30	2	590.17564	0.017989	20.0421	0.03277	-62.98	1.27921	-80.564635	0.131728	
10	30	2	590.180885	0.017879	20.21152	0.028043	-63.13	1.279938	-80.802464	0.080127	
10	30	2	590.183316	0.017824	20.4189	0.021147	-66.79	1.280635	-72.738644	0.176339	
20	100	4	589.80312	0.018359	20.1667	0.026286	-632.4	1.2803	-886.419592	1.15897	Cámara Inestable
20	100	4	589.802546	0.017634	20.28533	0.022867	-691.6	1.28081	-886.251793	0.939044	
20	100	4	589.784923	0.017948	20.02613	0.048383	-694.9	1.279888	-889.46335	2.148937	
20	100	4	589.793803	0.018026	19.89736	0.041987	-689.9	1.279408	-882.663457	1.8626	
20	100	4	589.795764	0.017986	19.89753	0.02644	-686.6	1.279228	-878.317905	1.869447	
20	100	4	589.809899	0.018067	19.8335	0.022063	-687.4	1.279210	-878.547311	0.987386	
20	100	4	589.819412	0.017995	19.75617	0.023817	-687.2	1.278734	-878.746106	1.055383	
20	100	4	589.84008	0.017786	19.85689	0.024269	-686.2	1.279126	-877.735949	1.072824	
20	100	4	589.846876	0.018167	20.06915	0.038286	-686.9	1.280041	-879.260104	1.67739	
20	100	4	589.844714	0.017774	20.25963	0.029942	-686.6	1.280877	-879.459249	1.299771	
20	100	4	589.830115	0.017778	20.04649	0.022107	-689.9	1.279978	-881.77086	0.93724	
20	100	4	589.840358	0.017697	20.05849	0.024969	-686	1.28	-878.079943	1.09053	

Figura 4.9 Reporte elaborado por la interfaz para la verificación de los datos.

4.1.2 Prueba de verificación del sistema

En la tabla 4.1.2.1, se presenta el resultado de un experimento llevado a cabo con los equipos mencionados con el fin de verificar la eficiencia de la interfaz. Para ello, los datos fueron obtenidos convencionalmente, es decir obtener datos de cada instrumento individual para después hacer el análisis. Simultáneamente, todos los equipos fueron sincronizados y conectados a la interfaz para obtener medidas directas. El experimento se hizo para voltajes de 30 kV and 100kV y corriente de 2 mA y 4 mA del tubo de rayos-x y tiempo de exposición entre 5s a 20s. La carga colectada fue obtenida con una cámara de ionización conectada al electrómetro mencionado arriba. Como se puede observar, aun cuando el valor de carga colectada es casi la misma independiente del voltaje y corriente del tubo, el tiempo de exposición, la incertidumbre obtenida usando la interfaz es mucho menor que la obtenida de manera manual.

Tabla. 4.1 Carga medida y corregida con su incertidumbre asociada usando cada equipo de manera individual y obtenida por la interfaz. Las incertidumbres corresponden a un factor de cobertura $k=1$

			Calculo manual		Datos Adquiridos por la interfaz	
Tiempo [s]	Voltaje [kV]	Corriente [mA]	Carga corregida (pC)	Incertidumbre (pC)	Carga corregida (pC)	Incertidumbre (pC)
5	30	2	-33.250	0.086	-33.25	0.025
5	30	2	-30.681	0.080	-31.96	0.025
5	30	2	-30.682	0.080	-30.68	0.023
5	30	2	-31.956	0.083	-31.96	0.025
5	30	2	-40.914	0.107	-31.96	0.025
5	30	2	-31.971	0.083	-31.97	0.024
5	30	2	-20.462	0.053	-20.46	0.015
10	30	2	-86.936	0.227	-86.94	0.067
10	30	2	-81.374	0.233	-81.37	0.114
10	30	2	-80.564	0.240	-80.56	0.131
10	30	2	-80.802	0.217	-80.8	0.080
10	30	2	-72.739	0.196	-72.74	0.075
20	100	4	-886.479	2.499	-886.5	1.158
20	100	4	-886.255	2.429	-886.3	0.999
20	100	4	-889.464	3.092	-889.5	2.148
20	100	4	-882.66	2.882	-882.7	1.862
20	100	4	-878.322	2.486	-878.3	1.169
20	100	4	-878.658	2.406	-878.6	0.987
20	100	4	-878.753	2.431	-878.7	1.059
20	100	4	-877.737	2.441	-877.7	1.072
20	100	4	-879.263	2.759	-879.3	1.677
20	100	4	-879.452	2.555	-879.5	1.299
20	100	4	-881.784	2.406	-881.8	0.972
20	100	4	-878.08	2.437	-878.1	1.090

4.2. Control de la mesa.

Para el control de la mesa se obtuvo un dispositivo que tiene varios botones: cuatro de ellos son para las direcciones, uno para ingresar a un menú y otro para indicar entrada (ver figura 4.10).



Figura 4.10 Controlador de la mesa de experimento.

En el menú de este dispositivo se incluye el tipo de ajuste que se desea: ajuste grueso o ajuste fino. También se incluye una opción para llevar a la mesa al origen, es decir, a la posición límite de la mesa. La última opción es salir del menú.

Este control cuenta también con dos leds que indican el movimiento actual de la mesa, que puede ser en el eje x o en el eje z.

5. Discusiones

Como se menciona arriba, la eficacia de la construcción y programación de la interfaz se verificó mediante la obtención de datos manualmente y por la interfaz. La tabla 4.1 presenta dichos resultados. Como se puede notar, estadísticamente no existe diferencias significativas (0.001%-0.013%) entre los dos métodos de obtención de datos absolutos. Mientras que la incertidumbre estándar combinada obtenida manualmente fue reducida por factores hasta de 3.5 veces más pequeña cuando se usa la interfaz. Esta disminución tan drástica en la incertidumbre obtenida puede asociarse al factor humano en el que se requiere una atención especial para lograr sincronizar todos los equipos al mismo tiempo. Por ejemplo, para obtener datos de manera manual, se requiere iniciar el tubo de rayos-x y al mismo tiempo tratar de iniciar a coleccionar las cargas, monitorear la presión y la temperatura para después determinar un promedio. Esta situación provoca un incremento considerable en la incertidumbre de las medidas. Por otra parte, al usar la interfaz, se elimina el factor humano y se logra una mejor sincronización de los equipos de manera simultánea. Además, estadísticamente se puede obtener un número mayor de datos en un intervalo de tiempo muy corto que permite disminuir la incertidumbre tipo A, lo cual es completamente imposible en modo manual. Los resultados obtenidos con la interfaz sugieren que se pueden determinar la tasa de dosis en el tubo de rayos-x con una exactitud aceptable como se esperaba al iniciar este proyecto.

Como trabajo futuro en la interfaz, se propone añadir algunos subprogramas para poder expandir su uso a otros equipos diferentes a los usados aquí. Lo que se propone para llevar a cabo esto, es pedir al usuario que ingrese los comandos que se usaron en este programa, como los de adquisición para los distintos instrumentos, comandos para configuración, etc. La aplicación debe mostrar una lista donde son pedidos los comandos uno por uno y especificando su uso. También debe indicar el dispositivo nuevo que se esté agregando. Finalmente se debe agregar todos los instrumentos nuevos a una base de datos para así registrar los dispositivos disponibles, para que el usuario pueda elegir entre todos los dispositivos que fueron usados.

Por otro lado, gracias al control de movimiento de la mesa de experimento, actualmente se tiene un mejor control de la posición de las muestras o detectores con respecto al eje del haz de

rayos-x. El control permite al usuario poder mover la mesa con distintos movimientos y velocidades. Esto ayuda a tener mas precisión en el tipo de movimiento y velocidad que el usuario necesita para optimizar el posicionamiento de las muestras.

Otra mejora que podría ser añadida es desarrollar una aplicación dentro del programa donde se pueda controlar la mesa con el control que fue desarrollado. El control diseñado cuenta con el hardware necesario para la comunicación USB por lo que solo se necesitaría la aplicación. En esta aplicación se tendrá la opción de control desde la computadora así como una base de dato donde se registren posiciones usadas. Esto es para que pueden ser ocupadas para experimentos posteriores.

6. Conclusiones

En este trabajo se diseñó una interfaz que cumplió con el propósito de este proyecto. Pues se cumplieron los objetivos al optimizar el proceso de realización de cada experimento. Se obtuvieron notables cambios tanto en la incertidumbre (hasta de 3.5 veces más pequeña cuando se usa la interfaz) como en el tiempo entre cada experimento. Con estas mejoras se pueden asegurar mejores resultados en el análisis de los experimentos que servirán para estudios de Dosimetría.

El usuario que opere este programa básicamente solo necesita conectar los dispositivos, definir los parámetros que él requiera para su experimento (voltaje de la fuente, corriente, unidades de presión) y pulsar iniciar. Al finalizar una serie de experimentos todos los cálculos como las variables adquiridas se reportan en un archivo en Excel, por lo que el usuario no debe preocuparse por realizar estos cálculos. La interfaz también cuenta con un cambio para ser utilizada con los demás instrumentos o únicamente la fuente.

Referencias.

- [1]. A Agarwal 2008 *Nobel Prize winners in physics* (New Delhi: APH Corporation)
- [2]. A Michette, P Fauntsh 1996 *X-Rays: The First Hundred Years* (New York: John Wiley & Sons Inc)
- [3]. F H Attix 2004 *Introduction to Radiological Physics and Radiation Dosimetry* (New York: Wiley)
- [4]. G Massillon-JL, D Cueva-Prócel, P Diaz-Aguirre, M Rodriguez-Ponce 2013 Dosimetry for Small Fields in Stereotactic Radiosurgery Using Gafchromic MD-V2-55 Film, TLD-100 and Alanine Dosimeters *PloS. one* **8** (5), e63418
- [5]. G Massillon-JL, ST Chiu-Tsao, I Domingo-Muñoz, MF Chain 2012 Energy dependence of the new Gafchromic EBT3 film: dose response curves for 50 kV, 6 and 15 MV X-ray beams *International Journal of Medical Physics, Clinical Engineering and Radiation Oncology* **1**, 60-65
- [6]. G Massillon-JL, L Zuñiga Meneses The response of the new MD-V2-55 radiochromic film exposed to ^{60}Co gamma rays *Med. Phys* **55** (18), 5437
- [7]. G Massillon-JL, I Gamboa-deBuen, AE Buenfil, MA Monroy-Rodriguez, ME Brandan, 2008 Tm (TLD-300) thermoluminescent response and glow curve induced by γ -rays and ions *Nucl. Instr. and Meth. in Phys. Res. B* **266** 772–780
- [8]. C M Chair, C W Coffey, L A DeWerd, C Liu, R Nath *et al.*, “AAPM protocol for 4-300 kV x-ray beam dosimetry in radiotherapy and radiobiology”, *Med. Phys.* **28** (6) 868-893 (2001)
- [9]. M Stellini 2009, “Evaluation of Uncertainty and Repeatability in Measurement: two application studies in Synchronization and EMC Testing” available in <http://paduaresearch.cab.unipd.it/1338/1/thesis-Stellini.pdf> (accessed 2009).
- [10]. C Weiss, K Al-Frouh, O Anjak 2011 Automation of radiation dosimetry using PTW dosimeter and LabVIEWTM *Nucl. Instr. and Meth. in Phys* **1**, 4
- [11]. J T Bushberg, J A Seibert, E M Leidholdt Jr and J M Boone 2001 *The Essential Physics of Medical Imaging* (Philadelphia: Lippincott Williams & Wilkins)
- [12]. B N Taylor, C E Kuyatt 1994 Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results *Natl. Inst. Stand. Technol. Tech. Note 1297* p**24**

- [13]. R J Tylor 1987 *Measurements, An introduction to error analysis. The study of uncertainties in Physical.* (California: University Science Book)
- [14]. J Fernandez d-R 2013 *LabVIEW Programación para sistemas de Instrumentación* (Mexico: Alfaomega)
- [15]. J R Lajara, J Pelegrí. 2011 *LabVIEW entorno gráfico de Programación.* (Mexico: Alfaomega)
- [16]. J Yiu 2014 *The definitive guide to ARM Cortex-M3 and Cortex-M4 processors* (Oxford: Elsevier)
- [17]. J Valvano. 2014 *Embedded Systems: Introduction to ARM Cortex-M Microcontrollers.*(Texas: Valvano)
- [18]. G Massillon-JL 2010 Dosimetry in steep dose-rate gradient radiation fields: A challenge in clinical applications. Eleventh Mexican Symposium on Medical Physics *AIP Conference Proceeding* **1310** 23–28

Apendice

A.

El programa se desarrolló en CCS para PIC's, utilizando el PIC18F4550. Se diseñó una biblioteca especial para el control de la mesa. Aquí se presenta el programa principal y a continuación las bibliotecas añadidas.

```
////////////////////////////////////  
////////////////////////////////////  
//Este programa  
//LLama las funciones principales para el movimiento de la mesa  
// 02/03/14  
////////////////////////////////////  
////////////////////////////////////  
#include <18F4550.h> //Biblioteca  
principal  
#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL5, CPUDIV1, VREGEN //Fuses  
necesarios para el oscilador y otras cuestiones. Cristal 8MHz!!  
#use delay(clock = 48000000)  
#include "mesa10.c"  
  
void main(){  
    init_mesa();  
    mesa_show_distance();  
    for (;;) {  
        mesa_get_ax_dir();  
        mesa_test_menu();  
        mesa_show_distance();  
    }  
}
```

A continuación se presenta la biblioteca mesa10.c:

```
#include "lcd_b_18_pin_3.c"  
////////////////////////////////////DEFINIR  
PINES////////////////////////////////////  
#use fast_io(A)  
#define LED_Y PIN_A0  
#define LED_X PIN_A1  
#define MENU PIN_A3  
#define ENTER PIN_A2  
#define Y_N PIN_A4  
#define Y_P PIN_A5  
#use fast_io(E)  
#define X_P PIN_E1
```

```

#define X_N    PIN_E0
#define AX_X   PIN_C0
#define X_DIR  PIN_C1
#define X_CLR  PIN_C2
#define AX_Y   PIN_D0
#define Y_DIR  PIN_D1
#define Y_CLR  PIN_D2
////////////////////////////////////CONSTANTES////////////////////////////////////
////////////////////////////////////
#define LIMIT_MENU 6
#define WIDTH_P    500
#define PULSES_CM  1000
#define PULSES_MM  100
#define PULSES_UM  10
#define FREQ_TIMER 0x00

////////////////////////////////////VARIABLES
GLOBALES////////////////////////////////////
int ax_to_move;
int ax_sense;
int ax_clr;
int ax_dir;
int menu_opc = 0; //Opciones posibles en el menu
int opc_selected = 0; //Opcion seleccionada
long pulses_X = 0;
long pulses_Y = 0;
long max_pulses = 0;

////////////////////////////////////FUNCIONES////////////////////////////////////
////////////////////////////////////
#INT_TIMER0
void int_tmr0(void){
    output_toggle(ax_to_move);
    switch (ax_sense){
        case X_P:
            pulses_X++;
            if (pulses_X >= max_pulses&&max_pulses != 0)
                disable_interrupts(INT_TIMER0);
            break;
        case X_N:
            pulses_X--;
            if (pulses_X <= max_pulses&&max_pulses != 0)
                disable_interrupts(INT_TIMER0);
            break;
        case Y_P:
            pulses_Y++;
            if (pulses_Y >= max_pulses&&max_pulses != 0)
                disable_interrupts(INT_TIMER0);
            break;
        case Y_N:
            pulses_Y--;
            if (pulses_Y <= max_pulses&&max_pulses != 0)
                disable_interrupts(INT_TIMER0);
            break;
    }
    set_timer0(FREQ_TIMER);
}

```

```

}
////////////////////////////////////TECLADO:
void init_mesa(void){
    ///CONFIGURACIÓN DE PUERTOS
    set_tris_a(0xFC);          //PURTO A ENTRADA y A0 , A1 como slaidas (LED_X LED_Y)
    set_tris_e(0xFF);          //PURTO E ENTRADA
    set_tris_c(0x00);          //PURTO C SALIDA
    set_tris_d(0x00);          //PURTO D SALIDA
    setup_adc_ports(NO_ANALOGS); //Puerto TODO digital
    ///CONFIGURACIÓN DE TIMER0
    setup_timer_0(T0_INTERNAL | RTCC_DIV_64 | RTCC_8_bit); //Configuración
//del timer0 --Reloj intermo y preescalamiento a 2--
    set_timer0(FREQ_TIMER); //Carga el timer0 con 0x00
    enable_interrupts(GLOBAL); //Habilita interrupciones globales
    disable_interrupts(INT_TIMER0);
    output_a(0x00);
    output_c(0x00);
    output_b(0x00); //deshabilita interrupción por timer0

    lcd_init();
}
int mesa_get_x_p(void){
    if (input(X_P) == 0){
        delay_ms(30);
        if (input(X_P) == 0)
            return 1;
    }
    return 0;
}
int mesa_get_x_n(void){
    if (input(X_N) == 0){
        delay_ms(30);
        if (input(X_N) == 0)
            return 1;
    }
    return 0;
}
int mesa_get_y_p(void){
    if (input(Y_P) == 0){
        delay_ms(30);
        if (input(Y_P) == 0)
            return 1;
    }
    return 0;
}
int mesa_get_y_n(void){
    if (input(Y_N) == 0){
        delay_ms(30);
        if (input(Y_N) == 0)
            return 1;
    }
    return 0;
}
int mesa_get_enter(void){
    if (input(ENTER) == 0){
        delay_ms(30);
        if (input(ENTER) == 0)
            return 1;
    }
}

```

```

    }
    return 0;
}
int mesa_get_menu(void){
    if (input(MENU) == 0){
        delay_ms(30);
        if (input(MENU) == 0)
            return 1;
    }
    return 0;
}

void mesa_show_distance(void){
    lcd_gotoxy(1, 1);
    printf(lcd_putc, "P_X= %5Lu      ", pulses_X);
    lcd_gotoxy(1, 2);
    printf(lcd_putc, "P_Y= %5Lu      ", pulses_Y);
}

//////////////////////////////////MENU//////////////////////////////////
//////////////////////////////////
int mesa_get_pad(void){
    if (mesa_get_x_p() == 1)
        return X_P;
    if (mesa_get_x_n() == 1)
        return X_N;
    if (mesa_get_y_p() == 1)
        return Y_P;
    if (mesa_get_y_n() == 1)
        return Y_N;
    return 0;
}

void mesa_get_opc_menu(void){           //Avanza en las diferentes opciones del menú
    if (mesa_get_pad() == X_P){
        while (mesa_get_pad() == X_P);
        if (menu_opc <= LIMIT_MENU)
            menu_opc++;
        if (menu_opc == LIMIT_MENU + 1)
            menu_opc = 0;
    }
    if (mesa_get_pad() == X_N){
        while (mesa_get_pad() == X_N);
        if (menu_opc >= 0)
            menu_opc--;
        if (menu_opc == -1)
            menu_opc = LIMIT_MENU;
    }
}

void mesa_test_menu(void){             //Pregunta si se apretó el botón de MENU
//de ser así elegije una opción
    int stay_menu = 1;
    if (mesa_get_menu() == 1){
        menu_opc = 0;
        lcd_putc("\f");
        lcd_gotoxy(1, 1);
        printf(lcd_putc, "      MENU      ");
    }
}

```

```

while (stay_menu == 1){
    lcd_gotoxy(1, 2);
    mesa_get_opc_menu();
    switch (menu_opc){
    case 0:
        printf(lcd_putc, "Ajuste Grueso ");
        if (mesa_get_enter() == 1){
            opc_selected = menu_opc;
            stay_menu = 0;
        }
        break;
    case 1:
        printf(lcd_putc, "Ajuste en cm ");
        if (mesa_get_enter() == 1){
            opc_selected = menu_opc;
            stay_menu = 0;
        }
        break;
    case 2:
        printf(lcd_putc, "Ajuste en mm ");
        if (mesa_get_enter() == 1){
            opc_selected = menu_opc;
            stay_menu = 0;
        }
        break;
    case 3:
        printf(lcd_putc, "Ajuste en um ");
        if (mesa_get_enter() == 1){
            opc_selected = menu_opc;
            stay_menu = 0;
        }
        break;
    case 4:
        printf(lcd_putc, "Ir al origen ");
        if (mesa_get_enter() == 1){
            opc_selected = menu_opc;
            stay_menu = 0;
        }
        break;
    case 5:
        printf(lcd_putc, "Calibrar control ");
        if (mesa_get_enter() == 1){
            opc_selected = menu_opc;
            stay_menu = 0;
        }
        break;
    case LIMIT_MENU:
        printf(lcd_putc, "Salir del menu ");
        if (mesa_get_enter() == 1){
            stay_menu = 0;
        }
        break;
    }
}
lcd_putc("\f");
//mesa_show_distance();
}
}

```

```

//////////CONTRLMESA//////////
void mesa_set_clr(void){
    output_high(ax_clr);
    delay_us(WIDTH_P);
    output_low(ax_clr);
    delay_us(WIDTH_P);
}
void mesa_tren_pulsos(void){
    mesa_set_clr();
    switch (ax_sense){
    case X_P:
        output_high(ax_dir);
        delay_us(WIDTH_P);
        max_pulses += pulses_X;
        enable_interrupts(INT_TIMER0);
        while (pulses_X<max_pulses){
            mesa_show_distance();
        }
        break;
    case X_N:
        output_low(ax_dir);
        delay_us(WIDTH_P);
        max_pulses = (pulses_X - max_pulses);
        enable_interrupts(INT_TIMER0);
        while (pulses_X>max_pulses){
            mesa_show_distance();
        }
        break;
    case Y_P:
        output_high(ax_dir);
        delay_us(WIDTH_P);
        max_pulses += pulses_Y;
        enable_interrupts(INT_TIMER0);
        while (pulses_Y<max_pulses){
            mesa_show_distance();
        }
        break;
    case Y_N:
        output_low(ax_dir);
        delay_us(WIDTH_P);
        max_pulses = (pulses_Y - max_pulses);
        enable_interrupts(INT_TIMER0);
        while (pulses_Y>max_pulses){
            mesa_show_distance();
        }
        break;
    }

    output_low(ax_to_move);
    mesa_set_clr();
}
void mesa_calibrar(void){
    int aux;
    ////////////PRIMERO DEBE IR AL ORIGEN
    lcd_gotoxy(1, 1);
    printf(lcd_putc, "Eje a calibrar \nPresione X o Y");
    do{
        aux = mesa_get_pad();
    }
}

```



```

} while (aux == 0);
if (aux == X_P || aux == X_N){
    ax_to_move = AX_Y;
    ax_sense = Y_P;
    ax_clr = Y_CLR;
    ax_dir = Y_DIR;
}

//////////aquí

delay_ms(1500);
lcd_putc("\f");
mesa_show_distance();
opc_selected = 0;
}
void mesa_opc(void){
    switch (opc_selected){
    case 0:
        mesa_set_clr();
        delay_us(WIDTH_P); //Delay
//antes de establecer la dirección
        if (ax_sense == X_P || ax_sense == Y_P)
            output_high(ax_dir);
        else
            output_low(ax_dir);
        delay_us(WIDTH_P); //Delay
//antes de evir el tren de pulsos
        max_pulses = 0;
        enable_interrupts(INT_TIMER0);
        while (input(ax_sense) == 0){ //mientras esté
//oprimido el botón se mantendrá en el ciclo
            mesa_show_distance();
        }
        disable_interrupts(INT_TIMER0); //deshabilita
interrupción por timer0
        output_low(ax_to_move); //deja en cero el eje
        mesa_set_clr();
        break;
    case 1:
        max_pulses = PULSES_CM;
        mesa_tren_pulsos();
        while (input(ax_sense) == 0);
        break;
    case 2:
        max_pulses = PULSES_MM;
        mesa_tren_pulsos();
        while (input(ax_sense) == 0);
        break;
    case 3:
        max_pulses = PULSES_UM;
        mesa_tren_pulsos();
        while (input(ax_sense) == 0);
        break;
    case 4:
        break;
    case 5:
        mesa_calibrar();

```

```

        break;
    }
}
void mesa_get_ax_dir(void){
    switch (mesa_get_pad()){
    case X_P:
        ax_to_move = AX_X;
        ax_sense = X_P;
        ax_clr = X_CLR;
        ax_dir = X_DIR;
        output_high(LED_X);           //Prende LED_X
        mesa_opc();
        output_low(LED_X);           //Apaga LED_X
        break;
    case X_N:
        ax_to_move = AX_X;
        ax_sense = X_N;
        ax_clr = X_CLR;
        ax_dir = X_DIR;
        output_high(LED_X);           //Prende LED_X
        mesa_opc();
        output_low(LED_X);           //Apaga LED_X
        break;
    case Y_P:
        ax_to_move = AX_Y;
        ax_sense = Y_P;
        ax_clr = Y_CLR;
        ax_dir = Y_DIR;
        output_high(LED_Y);           //Prende LED_Y
        mesa_opc();
        output_low(LED_Y);           //Apaga LED_Y
        break;
    case Y_N:
        ax_to_move = AX_Y;
        ax_sense = Y_N;
        ax_clr = Y_CLR;
        ax_dir = Y_DIR;
        output_high(LED_Y);           //Prende LED_Y
        mesa_opc();
        output_low(LED_Y);           //Apaga LED_Y
        break;
    }
    if (opc_selected>3)
        mesa_opc();
}

```

La biblioteca para manipular la lcd fue originalmente Flex_LCD416.c la cual se encuentra la red. Se requirió de modificaciones para adaptar al hardware que se tiene. La biblioteca se renombró como lcd_b_18_pin_3.c. La biblioteca se muestra a continuación.

```

// Flex_LCD416.c
// --Modify by Mario Alberto Corona Cadenas
// --IFUNAM
// --02/02/14
// These pins are for my Microchip PicDem2-Plus board,
// which I used to test this driver.

```

```

// An external 20x4 LCD is connected to these pins.
// Change these pins to match your own board's connections.

#define LCD_DB4    PIN_B7
#define LCD_DB5    PIN_B6
#define LCD_DB6    PIN_B5
#define LCD_DB7    PIN_B4

#define LCD_RS     PIN_B2
#define LCD_RW     PIN_C5
#define LCD_E      PIN_B3

/*
// To prove that the driver can be used with random
// pins, I also tested it with these pins:
#define LCD_DB4    PIN_D4
#define LCD_DB5    PIN_B1
#define LCD_DB6    PIN_C5
#define LCD_DB7    PIN_B5

#define LCD_RS     PIN_E2
#define LCD_RW     PIN_B2
#define LCD_E      PIN_D6
*/

// If you want only a 6-pin interface to your LCD, then
// connect the R/W pin on the LCD to ground, and comment
// out the following line. Doing so will save one PIC
// pin, but at the cost of losing the ability to read from
// the LCD. It also makes the write time a little longer
// because a static delay must be used, instead of polling
// the LCD's busy bit. Normally a 6-pin interface is only
// used if you are running out of PIC pins, and you need
// to use as few as possible for the LCD.
// #define USE_RW_PIN 1 // La comente para no usar el pin R/W

// These are the line addresses for most 4x20 LCDs.
#define LCD_LINE_1_ADDRESS 0x00
#define LCD_LINE_2_ADDRESS 0x40
#define LCD_LINE_3_ADDRESS 0x10
#define LCD_LINE_4_ADDRESS 0x50

// These are the line addresses for LCD's which use
// the Hitachi HD66712U controller chip.
/*
#define LCD_LINE_1_ADDRESS 0x00
#define LCD_LINE_2_ADDRESS 0x20
#define LCD_LINE_3_ADDRESS 0x40
#define LCD_LINE_4_ADDRESS 0x60
*/

//=====

#define lcd_type 2 // 0=5x7, 1=5x10, 2=2 lines(or more)

int8 lcd_line;

```

```

int8 const LCD_INIT_STRING[4] =
{
    0x20 | (lcd_type << 2), // Set mode: 4-bit, 2+ lines, 5x8 dots
    0xc, // Display on
    1, // Clear display
    6 // Increment cursor
};

//-----
void lcd_send_nibble(int8 nibble)
{
    // Note: !! converts an integer expression
    // to a boolean (1 or 0).
    output_bit(LCD_DB4, !(nibble & 1));
    output_bit(LCD_DB5, !(nibble & 2));
    output_bit(LCD_DB6, !(nibble & 4));
    output_bit(LCD_DB7, !(nibble & 8));

    delay_cycles(1);
    output_high(LCD_E);
    delay_us(2);
    output_low(LCD_E);
}

//-----
// This sub-routine is only called by lcd_read_byte().
// It's not a stand-alone routine. For example, the
// R/W signal is set high by lcd_read_byte() before
// this routine is called.

#ifdef USE_RW_PIN
int8 lcd_read_nibble(void)
{
    int8 retval;
    // Create bit variables so that we can easily set
    // individual bits in the retval variable.
    #bit retval_0 = retval.0
    #bit retval_1 = retval.1
    #bit retval_2 = retval.2
    #bit retval_3 = retval.3

    retval = 0;

    output_high(LCD_E);
    delay_us(1);

    retval_0 = input(LCD_DB4);
    retval_1 = input(LCD_DB5);
    retval_2 = input(LCD_DB6);
    retval_3 = input(LCD_DB7);

    output_low(LCD_E);
    delay_us(1);

    return(retval);
}

```

```

#endif

//-----
// Read a byte from the LCD and return it.

#ifdef USE_RW_PIN
int8 lcd_read_byte(void)
{
    int8 low;
    int8 high;

    output_high(LCD_RW);
    delay_cycles(1);

    high = lcd_read_nibble();

    low = lcd_read_nibble();

    return((high << 4) | low);
}
#endif

//-----
// Send a byte to the LCD.
void lcd_send_byte(int8 address, int8 n)
{
    output_low(LCD_RS);

#ifdef USE_RW_PIN
    while (bit_test(lcd_read_byte(), 7));
#else
    delay_us(60);
#endif

    if (address)
        output_high(LCD_RS);
    else
        output_low(LCD_RS);

    delay_cycles(1);

#ifdef USE_RW_PIN
    output_low(LCD_RW);
    delay_cycles(1);
#endif

    output_low(LCD_E);

    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}
//-----

void lcd_init(void)
{
    int8 i;

    lcd_line = 1;

```

```

        output_low(LCD_RS);
#ifdef USE_RW_PIN
        output_low(LCD_RW);
#endif

        output_low(LCD_E);

        // Some LCDs require 15 ms minimum delay after
        // power-up. Others require 30 ms. I'm going
        // to set it to 35 ms, so it should work with
        // all of them.
        delay_ms(35);

        for (i = 0; i < 3; i++)
        {
            lcd_send_nibble(0x03);
            delay_ms(5);
        }

        lcd_send_nibble(0x02);

        for (i = 0; i < sizeof(LCD_INIT_STRING); i++)
        {
            lcd_send_byte(0, LCD_INIT_STRING[i]);

            // If the R/W signal is not used, then
            // the busy bit can't be polled. One of
            // the init commands takes longer than
            // the hard-coded delay of 50 us, so in
            // that case, lets just do a 5 ms delay
            // after all four of them.
#ifdef USE_RW_PIN
            delay_ms(5);
#endif
        }
    }

    //-----

    void lcd_gotoxy(int8 x, int8 y)
    {
        int8 address;

        switch (y)
        {
        case 1:
            address = LCD_LINE_1_ADDRESS;
            break;

        case 2:
            address = LCD_LINE_2_ADDRESS;
            break;

        case 3:

```

```

        address = LCD_LINE_3_ADDRESS;
        break;

    case 4:
        address = LCD_LINE_4_ADDRESS;
        break;

    default:
        address = LCD_LINE_1_ADDRESS;
        break;

    }

    address += x - 1;
    lcd_send_byte(0, 0x80 | address);
}

//-----
void lcd_putc(char c)
{
    switch (c)
    {
        case '\f':
            lcd_send_byte(0, 1);
            lcd_line = 1;
            delay_ms(2);
            break;

        case '\n':
            lcd_gotoxy(1, ++lcd_line);
            break;

        case '\b':
            lcd_send_byte(0, 0x10);
            break;

        default:
            lcd_send_byte(1, c);
            break;
    }
}

#ifdef USE_RW_PIN
char lcd_getc(int8 x, int8 y)
{
    char value;

    lcd_gotoxy(x, y);

    // Wait until busy flag is low.
    while (bit_test(lcd_read_byte(), 7));

    output_high(LCD_RS);
    value = lcd_read_byte();
    output_low(LCD_RS);

    return(value);
}
#endif

```