

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



Diseño e implementación de un sistema para conocer el estado del acero de los vaporductos por medio de una gráfica radiación versus posición angular

Instituto Costarricense de Electricidad

I.C.E

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Allen Fco. Ledezma Rojas

Cartago, Enero de 2007

Declaración de autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Palmares, 23 de enero de 2007

Allen L.R.

Allen Francisco Ledezma Rojas.

Cédula: 1-1167-0201

INSTITUTO TECNOLÓGICO DE COSTA RICA


ESCUELA DE INGENIERIA ELECTRONICA


PROYECTO DE GRADUACIÓN

TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

Firma

Marvin Hernández Cisneros
Profesor lector

Firma

Faustino Montes de Oca Murrillo
Profesor lector

Firma

Francisco Navarro Henríquez
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 23 de Enero del 2007.

Resumen

En el Proyecto Geotérmico de Miravalles se cuenta con tres tipos de tuberías superficiales que transportan la materia prima con la que se genera electricidad. Existen aproximadamente 55Km de tuberías distribuidas entre los proyectos Miravalles I, II, III y V. Sin embargo el estado actual del acero, principal material para la construcción de estas tuberías se desconoce. Este acero puede presentar algún tipo de corrosión que a futuro puede originar grietas en la tubería, lo que originaría un escape ya sea del fluido bifásico, del líquido geotérmico o del vapor seco, según la tubería que presente el daño.

Generando datos reales del estado del acero se pueden encontrar y reparar daños en las tuberías. Ya que el agrietamiento de las tuberías, según la gravedad del caso puede originar desde contaminación ambiental hasta la disminución de la producción de electricidad en la Planta.

Por tal motivo se realizó un sistema con el cual se obtiene en un software una gráfica de radiación versus posición angular y así poder detectar puntos en la tubería donde presenta problemas de corrosión.

Palabras claves

Corrosión, medidor Geiger, microcontrolador, radiación, vaporducto

Abstract

At the Miravalles Geothermic Project there are three types of superficial pipes, these pipes transport the raw material that is used to generate electricity. There are approximately fifty five kilometers of pipes distributed along the first, second, third and fifth Miravalles Projects. However, the actual state of the steel used as main material for the construction of the pipes is unknown. This steel can have some kind of corrosion that in the future can originate cracks in the pipes, this would result in the escape of the two phased fluid, the geothermal fluid or the dry-steam, depending on the pipe that shows the problem.

Generating real data about the state of the steel the damages in the pipes can be found and repaired. This is important because the cracking in the pipes can originate according to the gravity of the matter from environmental pollution up to even the shortening of production of electricity in the plant.

Taking into consideration the reasons explained above, a system that is able to obtain a graph that shows the behavior of radiation versus the angular position in the pipe was implemented, this in order to be able to detect the exact location in which the pipe is having corrosion problems.

Keywords

Corrosion, Geiger measurer, microcontroller, radiation, steam pipeline

ÍNDICE GENERAL

Capítulo 1: Introducción.....	11
1.1 Problema existente e importancia de su solución	11
1.2 Solución seleccionada.....	13
Capítulo 2: Meta y objetivos	20
2.1 Meta	20
2.2 Objetivos generales.....	20
2.3 Objetivos específicos.....	20
2.3.1 Objetivos de documentación	21
2.3.2 Objetivos de implementación	21
Capítulo 3: Marco teórico	22
3.1 Descripción del sistema o proceso a mejorar	22
3.2 Descripción de los principales principios físicos, de software y electrónicos relacionados con la solución del problema.	24
3.2.1 Radiación gamma (γ) [1], [2], [3].....	24
3.2.2 Puerto serie RS-232 [13].....	25
3.2.3 I2C [14].....	26
3.2.4 Contador Geiger [15].....	27
3.2.5 Puerto serie en java [16].....	27
Capítulo 4: Procedimiento Metodológico	29
4.1 Reconocimiento y definición del problema.....	29
4.2 Obtención y análisis de información	29
4.3 Evaluación de las alternativas y síntesis de una solución.....	30
4.4 Implementación de la solución	30
4.5 Reevaluación y rediseño	31
Capítulo 5: Descripción detallada de la solución	32
5.1 Análisis de soluciones y selección final.....	32
5.2 Descripción del hardware.....	35
5.3 Descripción de Software.....	48
5.3.1 PIC principal.....	48
5.3.2 PIC secundario	57
5.3.3 Java	67
Capítulo 6: Resultados y Análisis.....	74
Capítulo 7: Conclusiones y Recomendaciones	98
7.1 Conclusiones	98
7.2 Recomendación	100

Bibliografía	101
Apéndices 103	
A.1 Glosario, abreviaturas y simbología.....	103
A.2 Manuales de usuario	104
A.2.1 Manual de usuario del programa en Java.....	104
A.3 Protocolos de medición	106
A.3.1 Protocolo de medición para el control externo	106
A.4 Información sobre la institución	107
A.4.1 Descripción de la institución	107
A.4.2 Descripción del departamento o sección en la que se realizó el proyecto	108
A.5 Secuencia de acontecimientos:	109
A.6 Programa PIC principal	112
A.7 Programa PIC secundario.....	120
A.8 Programa driver de memoria.....	129
Anexos 131	
B.1 Especificaciones para el uso de la EEPROM 24FC512:	131
B.2 Sobre radiación y cesio-137 [1], [2], [3].....	134
B.3 Corrosión	141
B.4 Descripción del contador Geiger GCA-04.....	143
B.5 Hojas de datos del PIC16F877A	145
B.6 Hojas de datos del Max232	146
B.7 Hojas de datos de la memoria EEPROM 24FC512	147
B.8 Hojas de datos del regulador de voltaje LM317	148
B.9 Hojas de datos del motor paso a paso.....	149
B.10 Hojas de datos del transistor de potencia NTE331y NTE332.....	150

ÍNDICE DE FIGURAS

Figura 1.1	Corte transversal de un vaporducto.....	11
Figura 1.2	Vaporducto con grieta en el acero.....	12
Figura 1.3	Ubicación de fuente y receptor en radial (1).....	14
Figura 1.4	Ubicación de fuente y receptor en radial (2).....	14
Figura 1.5	Ubicación de fuente y receptor, y su recorrido para el transversal.....	15
Figura 1.6	Ubicación de fuente y receptor, y su recorrido para el longitudinal.....	15
Figura 1.7	Radio interno y externo del vaporducto.....	17
Figura 1.8	Diagrama de bloques de la solución propuesta.....	17
Figura 3.1	Descripción del proceso de generación de electricidad en Miravalles.....	22
Figura 3.2	Radiación a través de la tubería.....	25
Figura 5.1	Inductancia y voltajes para el transformador.....	36
Figura 5.2	Comportamiento de la corriente para un puente rectificador.....	36
Figura 5.3	Conexión del regulador LM317.....	38
Figura 5.4	Motor paso a paso tipo bipolar.....	38
Figura 5.5	Secuencia par el manejo de un motor paso a paso bipolar.....	39
Figura 5.6	Puente H para el control de motor paso a paso.....	40
Figura 5.7	Circuito de control de bobina de apertura de la fuente.....	41
Figura 5.8	Bobina y pistón para apertura y cierre de la fuente de radiación.....	42
Figura 5.9	Conexión de memoria serial 24FC512.....	43
Figura 5.10	Conector DB9 del contador Geiger.....	43
Figura 5.11	Circuito para pulsador.....	44
Figura 5.12	Circuito de conexión del Max232.....	45
Figura 5.13	Circuito de conexión para PIC principal y secundario.....	47
Figura 5.14	Diagrama de flujo del ciclo <i>Do</i> del programa del PIC principal.....	53
Figura 5.15	Diagrama de flujo para la interrupción externa del PIC principal.....	54
Figura 5.16	Diagrama de la interrupción puerto B.....	55
Figura 5.17	Diagrama de flujo de la interrupción RDA.....	56
Figura 5.18	Espacio de memoria y dirección inicial para gráficas en la EEPROM externa... ..	58
Figura 5.19	Tiempos entre CPSs y bytes.....	61
Figura 5.20	Diagrama de flujo del ciclo <i>Do</i> del programa del PIC secundario.....	64
Figura 5.21	Diagrama de flujo de la interrupción RTCC.....	64

Figura 5.22	Diagrama de flujo de la interrupción RB	65
Figura 5.23	Diagrama de flujo para la interrupción RDA	66
Figura 5.24	Diagrama de flujo del programa principal	67
Figura 5.25	Diagrama de flujo para el subprograma de control	69
Figura 5.26	Diagrama de flujo para el subprograma sólo graficar	71
Figura 5.27	Diagrama de flujo para el subprograma para graficar	72
Figura 6.1	Gráfica de la salida del PIC principal para una bobina de motor paso a paso	74
Figura 6.2	Gráfica del intervalo de pasos para el motor.	75
Figura 6.3	Apertura de la fuente de radiación (abajo) debido a final de carrera (arriba), inicio de recorrido.	76
Figura 6.4	Cierre de la fuente de radiación (arriba) debido a final de carrera (abajo), fin de recorrido.	77
Figura 6.5	Cierre de fuente de radiación debido a dos pulsos de detener (paro de emergencia).	78
Figura 6.6	Dato enviado del PIC a la PC antes y después de pasarlo por el Max232	79
Figura 6.7	Pedido de dato de la PC y tiempo de respuesta del PIC.	80
Figura 6.8	Inicio de pasos del motor desde el control de la PC.	81
Figura 6.9	Inicio de pasos del motor desde control externo.	82
Figura 6.10	Detención de pasos del motor desde el control de la PC.	83
Figura 6.11	Detención de pasos del motor desde el control externo.	84
Figura 6.12	Salida del Geiger de un CPS, tiempos de duración y orden de bytes.	84
Figura 6.13	Encendido del Geiger y datos de CPS.	85
Figura 6.14	Salida del Max232 de un dato de CPS enviado al PIC.	87
Figura 6.15	Salida del Max232 del encendido del Geiger y datos de CPS.	87
Figura 6.16	Salida del pin SDA del PIC secundario para la EEPROM.	88
Figura 6.17	Salida del pin SCL del PIC para la EEPROM.	89
Figura 6.18	Solicitud de gráfica y verificación de datos.	90
Figura 6.19	Datos enviados y recibidos en Java por el puerto serial	91
Figura 6.20	Gráfica de Prueba de Posición versus Radiación	92
Figura 6.21	Solicitud de borrar datos de gráfica recién leída.	93
Figura 6.22	Gráfica de Prueba de Posición versus Radiación	94
Figura 6.23	Gráfica real de Posición versus Radiación.	95
Figura 6.24	Gráfica real de Posición versus Radiación.	95
Figura 6.25	Gráfica real de Posición versus Radiación.	96

Figura B.1.1 Formato del byte de control.....	131
Figura B.1.2 Asignamientos de bits para la secuencia de dirección	132
Figura B.1.3 Escritura de 1 bytes	132
Figura B.1.4 Escritura de una página de bytes	132
Figura B.1.5 Secuencia de lectura normal de un byte	133
Figura B.1.6 Secuencia de lectura de un byte en una dirección al azar	133
Figura B.1.7 Secuencia de lectura de varios bytes seguidos.....	133
Figura B.3.1 Absorción de película de óxido	141
Figura B.3.2 Nucleación y crecimiento de óxido.....	142
Figura B.3.3 Oxidación interna.....	142
Figura B.3.4 Formación de Cavidades, porosidades, microgrietas.....	142
Figura B.3.5 Formación de macrogrietas.....	142

ÍNDICE DE TABLAS

Tabla 1.1 Ventajas y desventajas de las soluciones para la primera etapa	16
Tabla 3.1 Descripción de puerto serie RS-232 para conector de 9 pines	26
Tabla 5.1 Caracteres reconocidos por el PIC principal	51
Tabla 5.2 Dirección y dato a almacenar en la EEPROM interna del PIC	58
Tabla 5.3 Valores del puerto debido a la interrupción RB una vez rotado e indicaciones al PIC	59
Tabla B.2.1 Constantes de emisividad más comúnmente utilizados en radiografía	138
Tabla B.2.2 Valores del factor RBE para diferentes los tipos de radiación	139
Tabla B.2.3 Características de fuentes de rayos gamma más comunes	141

Capítulo 1: Introducción

En este capítulo se presenta el problema existente y su entorno. De la importancia de su solución y cómo se pretende resolver.

1.1 Problema existente e importancia de su solución

En el Proyecto Geotérmico Miravalles “existen tres tipos de tuberías superficiales, llamados vaporductos, una que transporta la mezcla del líquido y vapor del pozo productor al separador ciclónico, otra que lleva el líquido geotérmico desde el separador ciclónico hasta el pozo de reinyección en caliente y la última que lleva el vapor seco desde el separador hasta la turbina en la Casa de Máquinas”. [8]

Estos vaporductos son “tuberías de acero suficientemente gruesas para no tener problemas de dispersión de calor; además, sobre la tubería se coloca una capa de un aislante térmico (fibra de vidrio) de dos pulgadas de ancho. Este aislante tiene como función impedir que el metal de la tubería irradie en la atmósfera una parte del calor que transita en los interiores. Sobre el aislante térmico se coloca una lámina muy delgada de aluminio para evitar que el aislante térmico se humedezca con las lluvias y se desgarré o se deteriore”. [8]. La estructura de un vaporducto se puede observar en la figura 1.1.

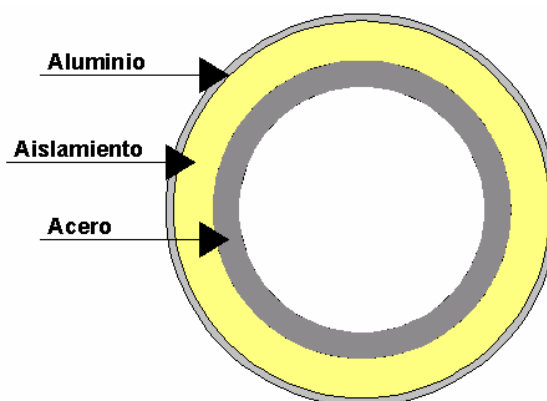


Figura 1.1 Corte transversal de un vaporducto

El problema existente es que en la actualidad no se tiene ninguna información del estado del acero en estos vaporductos. Pueden presentar algún tipo de corrosión con lo que a un corto o mediano plazo tendería a originarse grietas, dándose un escape de fluido bifásico, líquido geotérmico o vapor seco, según la tubería que presente el daño. Esto se presenta en la figura 1.2.

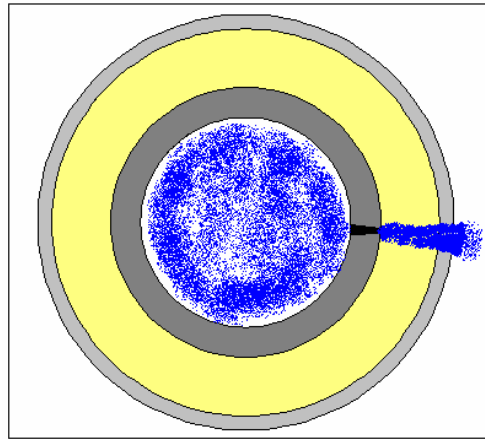


Figura 1.2 Vaporducto con grieta en el acero

Generando datos del estado del acero se pueden encontrar posibles puntos de corrosión, originando una reparación temprana. Se daría un mantenimiento preventivo en vez de un mantenimiento correctivo.

En el caso del líquido geotérmico, al perderse gran cantidad no sería nuevamente reinyectado al reservorio, “este tipo de reinyección se utiliza para evitar la contaminación del ambiente, y a la vez, prolongar la vida útil del recurso geotérmico”, [8]. En el caso del vapor seco, es el encargado de hacer mover la turbina para la generación de la electricidad, al perderse este vapor se da una disminución en la producción de electricidad por parte del proyecto geotérmico, por lo tanto, su eficiencia decae y se dejaría de percibir gran cantidad de dinero. En el caso del fluido bifásico, al tener tanto vapor como agua se daría una pérdida de ambos casos.

Otro percance, menos probable pero si de importancia, es el hecho de que un civil o trabajador del ICE se encuentre cerca en el momento de que se agrieta una tubería, ya que se generaría una pequeña explosión y el fluido como se encuentra a alta temperatura puede ocasionar alguna quemadura. Esto porque muchas tuberías durante los diferentes trayectos se encuentran cerca de caminos públicos; lo que puede ocasionar problemas legales.

1.2 Solución seleccionada

El ICE requiere el diseño e implementación de un sistema que sea capaz de obtener en una computadora una gráfica de radiación captada versus posición, esto para conocer el estado del acero en puntos específicos. Se debe usar el equipo de radiación que se tiene en el proyecto geotérmico para realizar el análisis de las tuberías. El equipo es una fuente y receptor de radiación que se habían comprado hace un tiempo atrás para otro proyecto, pero al final este no se realizó y quedó sin uso. Aunque esto puede estar sujeto a cambios ya que nadie ha manipulado el receptor y no se conoce su correcto funcionamiento.

Además, como se trata del uso de radiación gamma, este sistema se debe operar a una distancia segura. Así la persona encargada de las mediciones no estar expuesta a altos niveles de radiación que vaya a perjudicar su salud, ya que hay normas de seguridad que indican los niveles de exposiciones que una persona puede tener.

La solución de este proyecto se puede dividir en cuatro etapas, la primera es buscar la mejor manera de cómo hacer el recorrido a través de la tubería, la segunda es el diseño del montaje del sistema, la tercera es la parte electrónica con la cual se obtiene la gráfica y la cuarta es la interpretación de la gráfica para conocer el estado de la tubería.

Este proyecto cuenta con la participación de un estudiante de Ingeniería en Materiales del ITCR, él tiene como objetivo realizar el diseño del sistema de montaje del equipo analizador, además de la interpretación de la gráfica obtenida.

La solución planteada de cómo se va hacer el recorrido del equipo de radiación, primera etapa, es una decisión entre el responsable del proyecto en el ICE, el estudiante de Ing. en Materiales y mi persona. Se obtuvieron y discutieron cuatro formas de poder hacer el recorrido:

1. Radial (1)

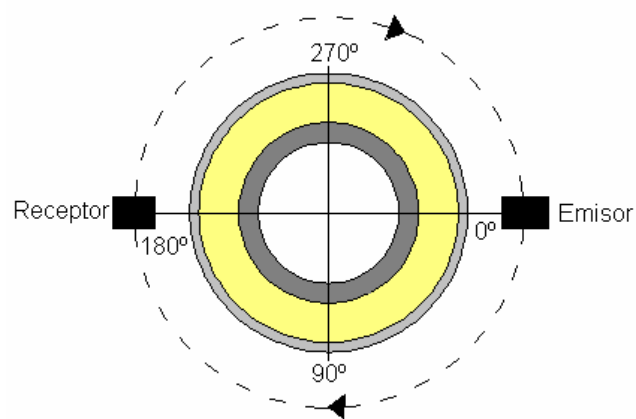


Figura 1.3 Ubicación de fuente y receptor en radial (1)

2. Radial (2)

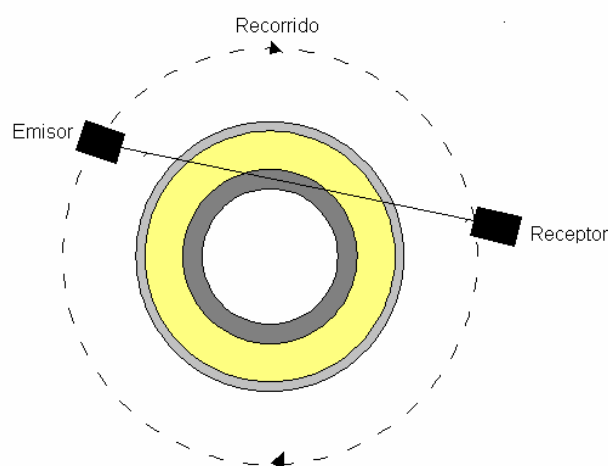


Figura 1.4 Ubicación de fuente y receptor en radial (2)

3. Transversal

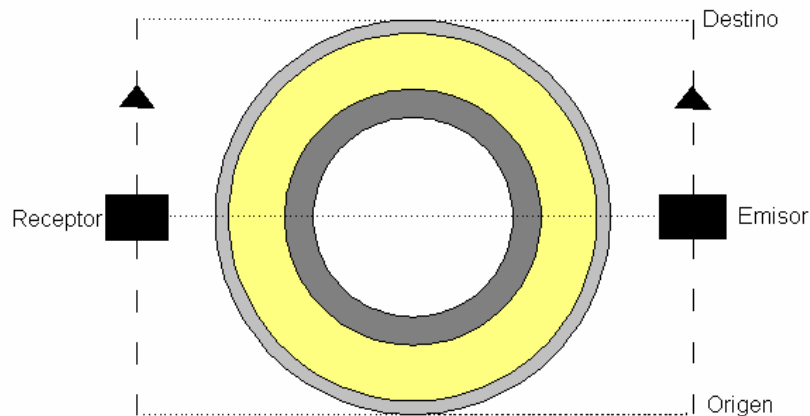


Figura 1.5 Ubicación de fuente y receptor, y su recorrido para el transversal.

3. Longitudinal

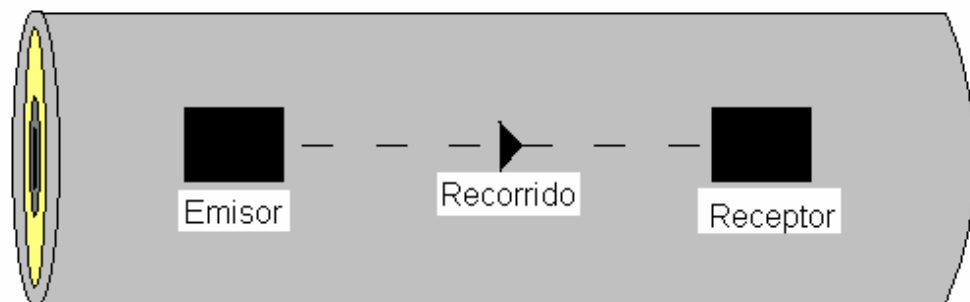


Figura 1.6 Ubicación de fuente y receptor, y su recorrido para el longitudinal

Cada propuesta originó diferentes soluciones tanto en el área de montaje como en el área de electrónica, por ejemplo movimiento del motor, qué motor usar, que control sería menos complicado, en cuál se darían datos más confiables, y muchos factores de ingeniería que llevarían a dar la mejor solución. Los diseños de la obtención de la gráfica y de la estructura partieron de este punto.

Tabla 1.1 Ventajas y desventajas de las soluciones para la primera etapa

Recorrido	Ventajas	Desventajas
Radial (1)	<ol style="list-style-type: none"> 1. Siempre radiación por el centro de tubería 2. Montaje no muy complicado 	<ol style="list-style-type: none"> 1. Diferentes radios 2. Tuberías por el suelo
Radial (2)	<ol style="list-style-type: none"> 1. Siempre radiación por el espesor del acero 	<ol style="list-style-type: none"> 1. Diferentes radios 2. Conocer la ubicación del acero 3. Tuberías por el suelo
Transversal	<ol style="list-style-type: none"> 1. No problemas a ras de suelo 2. Montaje no muy complicado 	<ol style="list-style-type: none"> 1. Diferentes radios 2. Diferentes alturas 3. Conocer ubicación de acero 4. No detectar bien punto de daño
Longitudinal y radial	<ol style="list-style-type: none"> 1. Cubre mayor área en una medición 	<ol style="list-style-type: none"> 1. Diferentes radios 2. Sistema de rieles para desplazamiento longitudinal 3. Problemas en los codos y sitios quebrados 4. Diseño complicado

En la tabla 1.1 se muestra la comparación entre las cuatro propuestas planteadas, en donde se establecen sus ventajas y desventajas. Se concluyó que el mejor método para hacer el recorrido es la radial (1). La mayoría de las tuberías se encuentran a una altura en la que sí se le puede hacer el análisis y se realizó un montaje para un único radio, correspondiente a un diámetro interno de aproximadamente 500mm y diámetro externo de aproximadamente 600mm, como se muestra en la figura 1.7

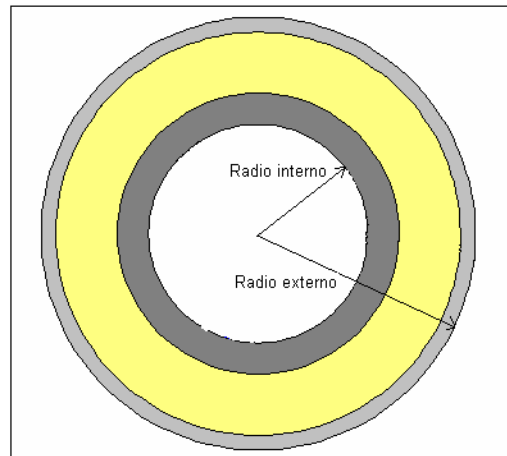


Figura 1.7 Radio interno y externo del vaporducto

Ya una vez conociendo el movimiento que se va a tener la estructura se plantea la siguiente solución:

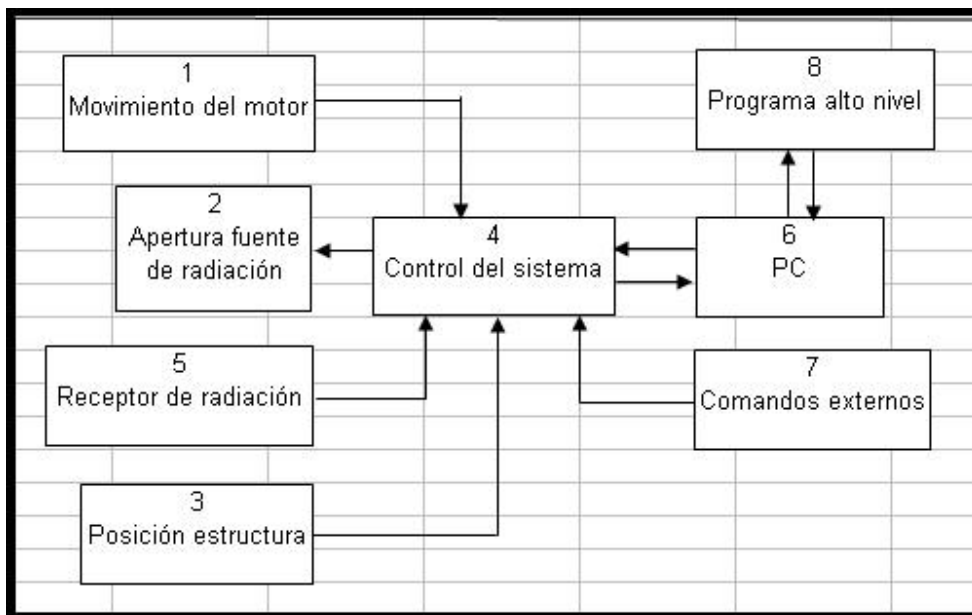


Figura 1.8 Diagrama de bloques de la solución propuesta

En la figura 1.8 se presenta el diagrama de bloques de la solución seleccionada.

En 1 se tiene un motor de pasos cuya función es mover la estructura completa; es importante mencionar que este motor tiene suficiente torque ya que la

fuentes de radiación es muy pesada, 13.1Kg. El diseño de la estructura se planteó de tal forma que el motor se mueva en conjunto con la estructura que mueve la fuente; con lo cual se buscó un motor liviano pero a la vez que tuviera buena fuerza para poder mover la estructura y que esta no tuviera problemas de estabilidad. La alimentación del motor se realizó por medio de dos puentes H, uno para cada bobina.

En 2 se tiene la apertura de la fuente de radiación en donde se realizó por medio de una bobina que al ser alimentada mueve un pistón que a su vez mueve la placa que cierra y abre el orificio del haz de radiación. La alimentación de esta bobina se realizó por medio de un relé que al ser alimentado hace que la bobina se energice.

En 3 se tiene la detección de la posición de la estructura, esto se hizo por medio del tiempo de duración del motor en dar el recorrido, este tiempo lo toma de la cantidad de muestras que genera el sensor de radiación, explicado más adelante, los pasos del motor siempre se da a la misma frecuencia, con lo cual el tiempo y la posición son directamente proporcionales.

En 4 se tiene el control del sistema, esto se realizó por medio de dos microcontroladores de la empresa Microchip, conocidos como PICs. Uno tiene el control de datos enviados de la PC o botones externos, así como la apertura y cierre de la fuente de radiación, tiempo transcurrido, pasos del motor y señales al segundo PIC. Y este segundo se encarga de recibir datos de la fuente de radiación, guardar en una memoria y luego enviar datos a una PC. Dentro de este mismo control está una memoria serial de alta capacidad de almacenamiento encargada de guardar datos de los niveles de radiación captados.

En 5 se tiene un receptor de radiación. Después de varias pruebas realizadas con el receptor que venía con la fuente no se obtuvo ninguna señal con lo que se tuvo que desechar y buscar una mejor opción para captar la radiación. Se seleccionó un medidor Geiger como sensor de radiación, este es capaz de comunicarse por medio de RS-232 con lo cual el PIC puede recibir los datos que se manejan en el Geiger.

En 6 se tiene una PC encargada de comunicar el control con el programa de alto nivel. En la solución se plantean dos posibilidades: llevar una computadora al campo con lo cual usar una Laptop o guardar los datos de las variables en una memoria y usar una Desktop.

En 7 se tienen unos pulsadores. Estos envían comandos al control del sistema y éste actúa de acuerdo a ellos. Estos se utilizarán cuando no se trabaje con la Laptop, los datos serán guardados en la memoria y se podrán hacer varias mediciones antes de ir graficar.

En 8 se tiene un programa en Java donde se presenta el gráfico radiación versus grados. Se comunica por el puerto serial RS-232 de la PC para la recepción y envío de datos. Cuando se trabaja con la Laptop en el programa de alto nivel el operador puede tener un manejo de las mediciones, desde movimiento del motor hasta la recepción y presentación de los datos. Cuando se trabaja con la Desktop el programa sólo se encarga de recibir los datos y graficarlos.

El control, como se mencionó, se va a llevar de dos maneras: La primera es cuando al campo se lleva una Laptop y desde el programa en Java se tiene un control sobre el sistema. Se hace un recorrido y se presenta la gráfica. Esto se hace cuando se quiere ubicar de inmediato una falla en la tubería, observando exactamente adónde se encuentra el daño. Con este control al detectar un daño se puede de inmediato hacer una segunda prueba en ese recorrido, detectar bien la ubicación de la corrosión y tomar las medidas receptoras. La segunda manera de control es cuando no se lleva o no se tiene una Laptop al alcance, de esta forma el operador puede hacer varias mediciones en diferentes recorridos y luego nada más llevar el circuito de control a una Desktop, descargar los datos en el programa y ver las gráficas respectivas; es importante en este caso llevar anotaciones de la ubicación donde se tomaron los datos de cada gráfica.

Capítulo 2: Meta y objetivos

En este capítulo se presenta tanto la meta como los objetivos, generales y específicos, del proyecto.

2.1 Meta

Diseñar e implementar un sistema para detectar a tiempo problemas de corrosión en el acero de los vaporductos, evitando así, agrietamientos que puedan causar escapes de la materia prima del proyecto geotérmico.

2.2 Objetivos generales

Diseñar e implementar un sistema que sea controlado desde una distancia segura a la fuente radioactiva, con el cual obtener una gráfica de radiación versus posición de un recorrido radial en un vaporducto de 600mm de radio externo.

2.3 Objetivos específicos

- Obtener los suficientes datos de radiación para lograr una gráfica lo más precisa posible.
- Determinar la distancia a la fuente de radiación, de acuerdo a las leyes internacionales, para que el operario no vea afectada su salud.
- Determinar la velocidad del motor, de acuerdo a la fuerza requerida para tener estabilidad en la estructura y además del periodo de lectura del sensor de radiación para obtener lecturas reales.
- Establecer comunicaciones entre los dispositivos: PICs, memoria externa y PC, de acuerdo a hojas de datos, para que puedan dar tiempos de respuesta lo más rápido posible.

2.3.1 Objetivos de documentación

- Realizar un manual de usuario para el uso del programa en Java.
- Realizar el protocolo para realizar las mediciones con el control externo.

2.3.2 Objetivos de implementación

- Realizar el montaje del sistema electrónico y mecánico para la evaluación de su funcionamiento.
- Realizar el montaje del sistema electrónico a la distancia escogida de la fuente para corroborar su correcto funcionamiento.

Capítulo 3: Marco teórico

En este capítulo se presenta los principales temas o conceptos relacionados implícita o explícitamente con la solución del problema.

3.1 Descripción del sistema o proceso a mejorar

La solución del problema no interfiere directamente en un proceso. Está orientado al análisis del estado de un sistema dentro de un proceso. Esto es, el análisis del sistema de vaporductos dentro todo el proceso de generación de electricidad en el Proyecto Geotérmico Miravalles.

Por lo cual es conveniente dar una descripción del sistema en el cual se inserta el proyecto a implementar.

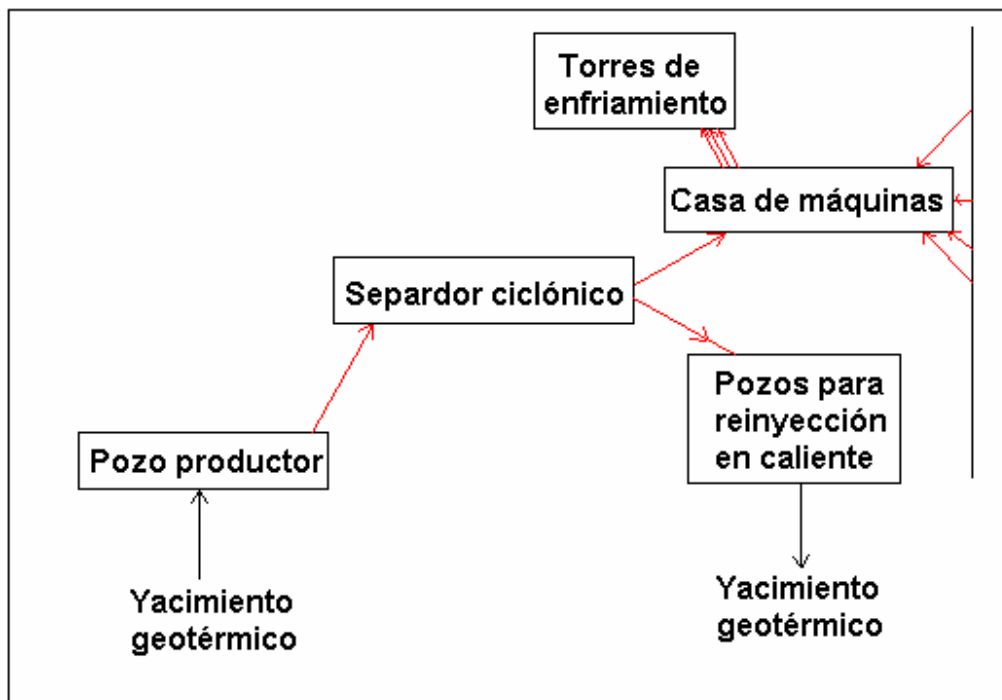


Figura 3.1 Descripción del proceso de generación de electricidad en Miravalles

En la figura 3.1 se presenta una descripción del proceso de generación de electricidad. Inicialmente se hace un estudio sobre un posible yacimiento geotérmico

y si los resultados son satisfactorios se hace un pozo entre los 1 y 3Km de profundidad. En este yacimiento se encuentra la materia prima para la generación de la electricidad, es un fluido que se encuentra en estado bifásico, mezcla de agua y vapor.

Es por medio de los pozos productores que se extrae este fluido y es llevado al separador ciclónico. El separador ciclónico es el encargado de separar el líquido y el vapor. Así el líquido caliente es llevado a pozos de reinyección en caliente y el vapor es llevado a la casa de máquinas.

Los pozos de reinyección en caliente se encargan de devolver el líquido al yacimiento geotérmico en una zona apta para tal efecto.

En casa de máquinas el vapor da movimiento a las turbinas que hacen mover los generadores para la producción de la electricidad. Después el vapor pasa a un condensador que se encarga de hacer pasar el vapor ya utilizado a líquido. De aquí pasa a las torres de enfriamiento.

Las torres de enfriamiento lo que hacen es enfriar el líquido por medio de grandes ventiladores. Parte de este líquido se utiliza para el condensador y el resto se reinyecta en frío a los yacimientos.

Como se observa en la figura 3.1 las líneas rojas corresponden a los vaporductos, los cuales se encargan de llevar los fluidos (líquido, vapor y bifásico) a través de todo el proceso. Es importante mencionar que existen varios pozos productores, y por lo tanto, varios separadores ciclónicos y pozos de reinyección. Es por esto, como se observa en la figura, que a la casa de máquinas llegan varios vaporductos con vapor para hacer mover las turbinas. Son aproximadamente 55Km de vaporductos los que se encuentran en estos momentos en operación en el Proyecto Geotérmico Miravalles, esto para los diferentes proyectos: Miravalles I, II, III y V.

Por lo tanto, para hacer el análisis de los vapores no se daría un paro en la planta. Mientras se realiza el estudio la planta puede estar generando electricidad normalmente.

3.2 Descripción de los principales principios físicos, de software y electrónicos relacionados con la solución del problema.

3.2.1 Radiación gamma (γ) [1], [2], [3]

Es un tipo de radiación electromagnética producida generalmente por elementos radiactivos y procesos subatómicos. Debido a su alta energía es un tipo de radiación ionizante, capaz de penetrar en la materia más profundamente que la radiación alfa o beta. Al no tener carga, los campos eléctricos y magnéticos no la afectan.

Su energía se mide en megaelectronvoltios (MeV). Un MeV corresponde a fotones gamma de longitudes de onda inferiores a 10^{-11} m o frecuencias superiores a 10^{19} Hz.

Lo que se quiere en este proyecto es mandar rayos gamma a través de la tubería. Se espera que el aluminio y la fibra de vidrio no atenúen en gran porcentaje esta radiación y que este la gran mayoría antes de hacer contacto con el acero. Por su gran poder de penetración se espera que del otro lado de la tubería llegue la suficiente energía para ser captada por el medidor Geiger y detectar puntos en donde por falta de material, debido a desgaste de acero, se vean picos de radiación.

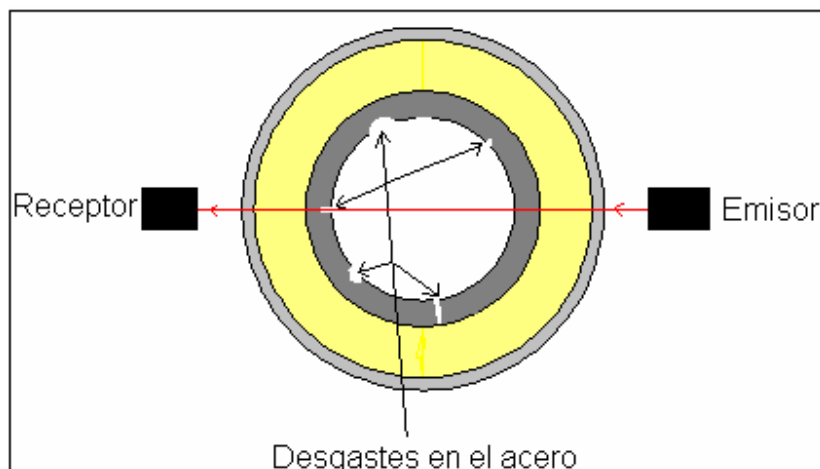


Figura 3.2 Radiación a través de la tubería

3.2.2 Puerto serie RS-232 [13]

La interfaz RS-232 está diseñada para distancias cortas, de unos 15 metros o menos, y para velocidades de comunicación bajas, de no más de 20kb. Aunque muchas veces se utiliza a mayores velocidades con un resultado aceptable

Se puede trabajar en comunicación asíncrona o síncrona y tipos de canal simplex, half duplex o full duplex. En un canal simplex los datos siempre viajarán en una dirección. En un canal half duplex los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada antes que los datos puedan viajar en la otra dirección. En un canal full duplex los datos pueden viajar en ambos sentidos simultáneamente. Una interfase full duplex puede obtenerse con solamente 3 cables. Una vez que ha comenzado la transmisión de un dato, los bits tienen que llegar uno detrás de otro a una velocidad constante y en determinados instantes de tiempo. Por eso se dice que el RS-232 es asíncrono por carácter y síncrono por bit.

Las señales con las que trabaja este puerto serie son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos. El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a unas velocidades determinadas. Después de la transmisión de los datos, le sigue un bit opcional de paridad, que

indica si el número de bits transmitidos es par o impar, para detectar fallos y después 1 o 2 bits de parada.

Para este proyecto la tasa de transferencia es de 9600bps y el cable de 10m. El dato es de 8 bits, sin paridad y con 1 bit de parada. La transmisión es full duplex.

Tabla 3.1 Descripción de puerto serie RS-232 para conector de 9 pines

Número de Pin	Señal	Descripción	E/S
1	-	Masa chasis	-
3	TxD	Transmit Data	S
2	RxD	Receive Data	E
7	RTS	Request To Send	S
8	CTS	Clear To Send	E
6	DSR	Data Set Ready	E
5	SG	Signal Ground	-
1	CD/DCD	(Data) Carrier Detect	E
4	DTR	Data Terminal Ready	S
9	RI	Ring Indicator	E

3.2.3 I2C [14]

I2C es un bus de comunicaciones serie. Su nombre viene de *Inter-Integrated Circuit*. Es un bus muy usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas empotrados.

La principal característica de I2C es que sólo usa dos hilos para transmitir la información: por uno van los datos y por otro la señal de reloj que sirve para sincronizarlos. También es necesaria una tercera línea, la señal de masa.

Tanto la línea de datos como de reloj de son drenador abierto, por lo que necesitan resistencias de pull-up.

Los dispositivos conectados al bus I2C tienen una dirección única para cada uno. También pueden ser *maestros* o *esclavos*. El dispositivo *maestro* inicia la transferencia de datos y además genera la señal de reloj, pero no es necesario que el *maestro* sea siempre el mismo dispositivo, esta característica se la pueden ir pasando los dispositivos que tengan esa capacidad. Esta característica hace que al bus I2C se le denomine bus multimaestro.

3.2.4 Contador Geiger [15]

Es un instrumento que permite medir la radiactividad de un objeto o lugar. Está formado normalmente por un tubo delgado metálico (cátodo) con un fino hilo metálico a lo largo de su centro (ánodo), el espacio entre ellos está aislado y relleno de un gas (Neón, Argón o más gas halógeno). El ánodo se encuentra a unos 500 V relativos con al cátodo.

Un Ion o electrón que penetra en el tubo (o se desprende un electrón de la pared por los rayos X o gamma) desprende electrones de los átomos del gas y, debido al voltaje positivo del hilo central, son atraídos hacia el hilo. Al hacer esto ganan energía, colisionan con los átomos y liberan más electrones, hasta que el proceso se convierte en una avalancha que produce un pulso de corriente detectable.

Al instrumento se le llama contador debido a que cada partícula que pasa por él produce un pulso idéntico, permitiendo contar las partículas normalmente de forma electrónica.

3.2.5 Puerto serie en java [16]

La comunicación se realiza por medio del API de Comunicaciones en Java. Este da un control total de todos los parámetros de los puertos serie: velocidad en baudios, bits de parada, paridad, bits por trama.

Este API utiliza para entrada y salida los streams. Los streams pueden extenderse para proporcionar control sobre el flujo y los umbrales. Por ejemplo, puede ser necesario lanzar un aviso cuando haya 15 caracteres en el buffer, o cuando solamente queden 15 espacios vacíos para almacenar caracteres. El control del flujo es importante cuando dos dispositivos conectados a través de un interfaz no pueden mantenerse uno al otro. Sin el control del flujo, pueden darse condiciones alteradas, tanto por exceso como por defecto. En condiciones en que haya

problemas por exceso de información, se recibirán datos mientras todavía se están procesando otros, con lo cual habrá pérdidas de información; y en el caso contrario, la información estará presente pero no disponible. Normalmente estas condiciones solamente se dan a nivel hardware, a nivel de la USART (Universal Synchronous Asynchronous Receiver Transmitter), que es el dispositivo físico que convierte los bytes a una señal ondulada coincidiendo con la velocidad en baudios. El API de Comunicaciones Java utiliza el modelo de eventos que JavaSoft ha introducido desde el lanzamiento del JDK 1.1 para proporcionar información de las líneas de señal que van cambiando y del estado del buffer.

Capítulo 4: Procedimiento Metodológico

En este capítulo se describe las etapas del método de diseño en ingeniería seguidas para llegar a la solución del problema.

4.1 Reconocimiento y definición del problema

Para el reconocimiento del problema se tuvo una reunión con un Ingeniero del Proyecto Geotérmico Miravalles, en las oficinas de SETEC ubicadas en el mismo proyecto. Se planteó la idea de hacer un sistema para poder conocer el estado del acero en los vaporductos ya que estos llevan varios años de vida útil sin interrupción. Por principios químicos el acero al estar expuesto a altas temperaturas, como en este caso, se oxida más rápidamente, con lo cual no se tiene con certeza cómo están las tuberías internamente por lo que se requiere un sistema capaz de generar datos que indique en qué condiciones se encuentran.

Estos vaporductos iniciaron su operación en marzo de 1994 para Miravalles I, agosto de 1998 para Miravalles II, marzo del 2000 para Miravalles III y junio del 2003 para Miravalles V.

Se explicó el uso de la radiación gamma que tiene gran capacidad de penetración en materiales y cómo se tenía pensado hacer el análisis con una fuente de este tipo. De esta manera la solución a este sistema está sujeta al uso de una fuente radiactiva, con lo cual debe considerarse desde un inicio el tema de seguridad en donde la distancia para controlar el sistema a la fuente es un punto importante.

4.2 Obtención y análisis de información

En el proyecto geotérmico se tiene bastante experiencia en el uso de la radiación ya que es por medio de ella que se hacen pruebas a diferentes partes de máquinas, desde tornillos hasta aspas. Se hacen pruebas para ver la calidad en otro tipo de tuberías. Estos análisis se hacen por medio de placas radiográficas, sin embargo, resulta muy costoso hacer tantas placas y además se debe esperar alrededor de 30 minutos para cada análisis.

Es de esperar que al atravesar el haz de radiación a través de la tubería y captándola del otro lado se obtenga una idea del estado del acero. En el caso de se presente corrosión se daría un incremento en la radiación del otro lado, ya que el haz pasaría por menos material y al obtener datos casi constantes de radiación en una tubería indicaría un vaporproducto sin presencia de corrosión.

4.3 Evaluación de las alternativas y síntesis de una solución

Antes de plantear alternativas de solución se divide el proyecto en etapas, cada una con resultados concretos. A partir de aquí se toman posibles soluciones para cada etapa, dando a su vez costos de componentes para cada solución.

Otro punto importante en la evaluación de la solución son componentes que deben ser comprados fuera del país ya que esto implica un atraso en el proyecto y por lo tanto en la entrega final.

Aplicando los conocimientos adquiridos de los cursos teóricos de la carrera y aplicando la experiencia de los laboratorios se toma la propuesta que se piensa va a tener mejores resultados.

4.4 Implementación de la solución

Primeramente se debe hacer pruebas con el receptor de radiación que se dispone en este momento. Por medio de las hojas de fabricante y pruebas de laboratorio se debe llegar a la conclusión si con este se obtendrá la respuesta esperada o si hay que comprar un sensor de radiación.

Por medio de simulaciones y pruebas de laboratorio se verifica el correcto funcionamiento de cada etapa. Una vez que se obtienen los resultados esperados de cada etapa se hace la interconexión de todas y se procede a hacer simulaciones y pruebas de campo para así evaluar el buen desempeño de todo el sistema y corroborar una correcta solución al problema.

Al final se evalúa los resultados prácticos con los resultados esperados para determinar si la solución corresponde exactamente a un gráfico de radiación versus

posición angular. Esto debido a, como se dijo anteriormente, que en el Proyecto de Miravalles están familiarizados con la radiación y se tiene una idea de cómo es una gráfica cuando se presenta y no corrosión.

Es importante mencionar que para la evolución de las diferentes etapas del sistema se trabajan con vaporductos que no están en el campo, sino con muestras de ellos que pueden ser llevados a un laboratorio, estar trabajando en él, hacer todas las pruebas necesarias y una vez comprobar su funcionamiento poder ir al campo.

Una vez comprobado el buen funcionamiento del sistema los resultados del proyecto se presentan en una exposición al encargado del proyecto, personas involucradas y afines a este, para dar una evaluación de los logros alcanzados.

4.5 Reevaluación y rediseño

Valorar el diseño y compararlo con el diseño replanteado, si se espera que la gráfica pueda ser obtenida en un menor tiempo es una mejora bastante significativa. Además es muy importante para toda empresa o institución el comparar costos y beneficios de la solución planteada y las soluciones que se replantean.

También el sistema de montaje es para un radio determinado, por lo que sólo se pueden hacer mediciones para radios iguales o menores o este, por lo que sería bastante bueno encontrar un método para el cual se pueda ajustar el sistema según el radio.

Capítulo 5: Descripción detallada de la solución

En este capítulo se explica el diseño de la solución implementada, haciendo referencia a los criterios de diseño que fueron considerados.

5.1 Análisis de soluciones y selección final

Primeramente se dividió la solución por bloques y se determinó a qué debía llegar cada. Así en cada bloque se buscaron posibles soluciones y al final se escogió la que tenía mejor proyección.

La primera etapa pensada fue el control del sistema. Aquí se pensó de inmediato en el uso de un microcontrolador, ya que es capaz de controlar varios dispositivos periféricos al mismo tiempo, tiene alta velocidad de respuesta y es de muy alta fiabilidad. El microcontrolador escogido fue el PIC16F877A de Microchip, debido a que es uno de los más conocidos y se había trabajado con él en cursos anteriores. Sus principales características es su arquitectura RISC (Reduced Instruction Set Computer) con 35 instrucciones de programación. Una velocidad de operación de 20MHz lo que la hace relativamente rápida, obteniendo una respuesta en corto plazo ante posibles interrupciones. Interrupción externa, de datos recibidos por el pin Rv y del puerto B lo que genera una buena comunicación con el exterior. Tiene la capacidad de comunicación con el puerto I2C (*Inter-Integrated Circuit*) y USART (Universal Synchronous Asynchronous Receiver Transmitter), lo que hace que se pueda comunicar tanto con una PC como con una memoria eeprom serial de este tipo.

Es un microcontrolador de 40 pines, 8K x 14 palabras de memoria flash de programa, 368 bytes de memoria de datos tipo Ram y 256 bytes de datos tipo EEPROM. Estas características la hace que tenga la capacidad suficiente para guardar en el PIC el programa para este proyecto. Se debe de recordar que las microinstrucciones de programa se guardan en la memoria Flash y los datos que se van generando mientras se ejecuta el programa se guardan en la memoria Ram.

Seguidamente para el almacenamiento de los datos se determinó el uso de una memoria serial con protocolo de comunicación I2C. Primeramente se tenía que determinar entre el uso de una memoria paralela o tipo serial. La decisión del uso de la memoria serial se debió a dos situaciones: la primera está en la cantidad de patillas que se debían de utilizar en el PIC para el manejo de datos y direcciones para una memoria en paralelo. Para una memoria de 8Kbytes, se necesitan 13 patillas para la dirección A0...A12 y 8 patillas para los datos D0...D7. Además de ello 3 patillas para el control, indicándole a la memoria si es una lectura o escritura. La segunda situación es causa inmediata de la primera, es sobre la complejidad en la programación del PIC para el manejo de la información en la memoria, ya que para la dirección se deben de concatenar dos puertos, buscar qué patillas de ese puerto están libres, usar variables tipo long para incrementar la dirección y luego darle ese valor a las patillas correspondientes. Para la lectura de los datos leer patillas por paralelo, guardarlas en una variable tipo int y luego enviarlas a la PC.

En cambio, para el uso de la memoria serial, gracias a su protocolo I2C sólo es necesario el uso de dos patillas del PIC, que están debidamente marcadas, ya que como se mencionó el PIC tiene este mismo protocolo. Una patilla da la dirección y los datos (SDA), y la otra da la señal de reloj (SCL) para la sincronización. La complejidad en el uso de esta memoria está en la secuencia que pide este protocolo para la comunicación entre el maestro y el esclavo.

Con esto se utilizó la memoria 24FC512 de Microchip, entre las principales características es la compatibilidad con I2C, ciclo de escritura máximo de 5ms y 64Kbytes de memoria tipo EEPROM. Con esto el manejo de los datos es más sencillo ya que en el PIC se captura inmediatamente el dato en formato int para luego ser enviado a la PC, además el envío de la dirección es más sencillo y no complica mucho el programa del PIC.

Para el receptor de radiación se determinó el uso de un contador Geiger, esto por recomendación del Ing. Leonel Siles Loaiz, encargado de control de calidad en temas de radiología. Se tenía anteriormente el receptor de radiación, pero al

realizarle varias pruebas se observó que no daba ninguna señal medible, por lo que se tuvo que descartar esta opción. Primeramente se realizaron pruebas a la fuente de radiación para medir la dosis entregada y con esta buscar un contador que estuviera cercano a este valor. Por medio de un contador Geiger que se tenía en la empresa se midió que la máxima dosis de la fuente es de aproximadamente 7mSv/h cerca de la salida y sin ningún obstáculo. También se realizaron pruebas, con este mismo Geiger, cuando la radiación atravesaba tuberías de prueba y se midieron dosis máximas de 250uSV/h con lo cual la atenuación es bastante significativa.

Después de buscar las ofertas en el mercado se encontró un Contador Geiger Digital modelo GCA-04 de Imagees SI. La dosis que maneja ese contador está por debajo de la máxima de la fuente, 6.5mSV/h; sin embargo, según las dosis medidas cuando la radiación atraviesa la tubería este contador anda muy por encima de estos valores. Entre las principales características por las cuáles se escogió este contador están lectura de radiación en tiempo real en count/second (cuentas por segundo) equivalentes en mSv/hr (unidades de medida del SI para niveles de radiación), medida que se puede realizar cada segundo. El contador tiene certificación de calibración con niveles de lecturas de radiación. Capacidad de 9360 CPS (counts per second) equivalentes a 6.5mSv/hr. Detector de radiación gamma sobre los 7KeV. Trabaja con una batería de 9V. Comunicación con protocolo RS232 al exterior.

De esta forma el PIC puede recibir los datos de niveles de radiación en formato digital por medio del puerto serial y la actualización de los datos se realiza cada segundo.

Para la apertura de la fuente de radiación se determinó usar una bobina que al energizarse por acción de campo magnético atrae un pistón que mueve a la vez el cierre de la fuente. Se tenía la posibilidad de usar un pequeño motoreductor, sin embargo el acople al cierre de la fuente era muy complicado, se debía de hacer una pequeña cremallera y por factores de seguridad se desechó esta idea.

Para el movimiento de la estructura se escogió el uso de un motor paso a paso. Tiene la ventaja ya que tiene movimientos precisos, con él se puede detener la

estructura en cualquier momento, el control se encarga de dar la dirección de giro. Tienen la posibilidad de quedar enclavado en cualquier posición y así la estructura no perdería estabilidad.

Para el control externo se escogió el uso de unos pulsadores, de esta forma debido a las interrupciones del PIC al pulsar el PIC determina cual fue y realiza la instrucción correspondiente.

Finalmente para el programa de alto nivel se escogió Java ya que es un programa bastante avanzado en donde tiene las clases que se requerían para este proyecto: graficar y comunicación con puerto RS-232. En este programa se puede realizar una interfaz con el usuario para el uso de botones para peticiones de control y de graficación. Por ser un programa de alto nivel que se había utilizado con anterioridad, así como la cantidad de información y ejemplos que se puede encontrar fue el programa seleccionado par este proyecto.

5.2 Descripción del hardware

Primeramente se diseño la parte de alimentación en donde se debía de obtener tres voltajes en CD a partir del voltaje de red 120VAC. Estos voltajes corresponden a 5Vcd para el control, 26Vcd para el motor de paso a paso y 14Vcd para la bobina de apertura de la fuente de radiación.

Es importante mencionar los consumos que se dan cuando la bobina y el motor están trabajando, se midió que el motor de pasos consume aproximadamente 1.5A por bobina energizada, con lo que el consumo máximo es de 3A. Para la bobina de apertura se midió un consumo de 2A cuando está energizada. De esta forma la corriente máxima que se le exige a la red es de aproximadamente 5A, esto porque el consumo de la parte de control es muy baja, aproximadamente 70mA.

Se escogió uso de un transformador con varios bobinados de salida. Con este se tenían tres bobinas en el secundario. Al alimentarlo con 120VAC en el primario

dos bobinas del secundario daban 9VAC y una 11VAC, con lo cual se le hicieron pruebas con carga para ver si el voltaje no se caía debido a la corriente de salida y se determinó que el transformador si aguantaba la carga requerida. Así al unir los dos bobinados de 9VAC en la salida se obtiene un total de 18VAC, esto para alimentar el motor. De esta forma ya se tenía la primera parte en al etapa de alimentación.

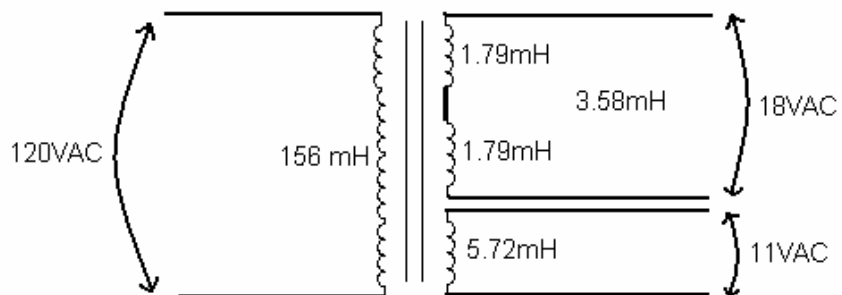


Figura 5.1 Inductancia y voltajes para el transformador

Después de esto cada voltaje AC, 18V y 11V, se llevaron independientemente a dos rectificadores de onda completa, esto para convertir las señales de CA a CD. Como se observa en la figura 5.2 por el posicionamiento de los diodos para ambos semiciclos de la onda la corriente sale por la misma patilla lo que la convierte en el voltaje positivo de una fuente CD, convirtiendo a la otra en por donde entraría la corriente para cerrar el circuito.

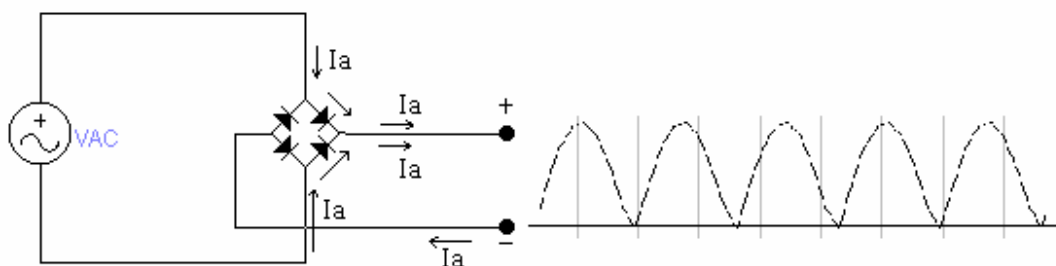


Figura 5.2 Comportamiento de la corriente para un puente rectificador

Para convertir la señal en CD pura se paso por un capacitor electrolítico. Primero el capacitor se va cargando hasta el valor pico de la señal, a partir de este instante el capacitor empieza a descargarse, dependiendo del valor de capacitor esta descarga se da más rápidamente. Con un valor alto de capacitancia este tendrá más capacidad de acumular energía por lo tanto se descargará más lentamente, de esta forma a pesar que la onda está decreciendo el capacitor se encuentra cargado; como los valores pico de señal se dan cada 16.6ms (60Hz) el capacitor siempre estará cargado y la señal será un voltaje CD puro. En paralelo a este capacitor se lo coloca una resistencia de descarga del capacitor, esta tiene un valor alto para que consuma poca corriente para cuando las bobinas están trabajando, pero cuando se apaga el sistema el capacitor se descargan poco a poco por esta resistencia.

Así, estos voltajes de corriente alterna se pasaron por dos rectificador NTE5313 cuya máxima corriente es de 8A, de esta forma se consigue que esté muy por encima de la corriente máxima que pasa por cada uno. Se colocó para cada rectificador un capacitor de 2200uF de 63V, con esto se tiene la CD pura y el valor de la resistencia es de 100KΩ. De esta forma se obtuvieron voltaje de CD de $11x\sqrt{2} = 15.5V_{cd}$ para la bobina y $18x\sqrt{2} = 25.5V_{cd}$ para el motor paso a paso.

Para obtener el voltaje del control se utilizó un regular de voltaje ajustable LM317. Puede manejar corrientes de carga de hasta 1.4A y puede ser ajustado para obtener en la salida voltajes entre 1.2V a 37V, de esta forma se hizo el cálculo de las resistencias para que la salida fueran 5V. En la figura 5.3 se observa la forma de conexión de este integrado, se colocó en R1 150Ω y en R2 480Ω y de esta forma $V_o=5.2V$, que está dentro del rango de voltaje del PIC.

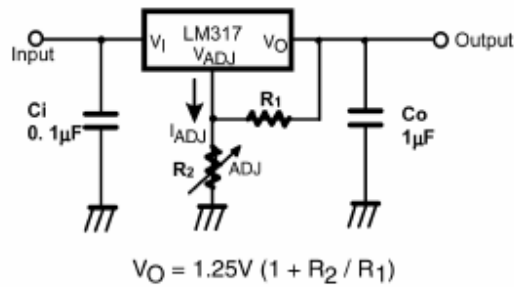


Figura 5.3 Conexión del regulador LM317

El tipo de motor paso a paso utilizado es del tipo bipolar, mostrado en la figura 5.4. Debido a que requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento se controla con un puente H. Se necesita un puente H para cada bobina del motor.

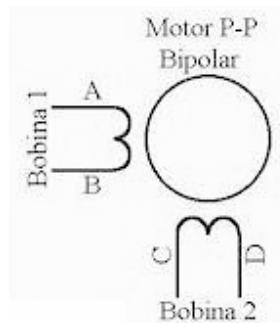


Figura 5.4 Motor paso a paso tipo bipolar

En la figura 5.5, tomado del sitio de Internet de acuerdo a apéndice 12, se muestra la secuencia para el movimiento de un motor paso a paso bipolar. Obsérvese que nunca están energizados al mismo tiempo los extremos de una misma bobina. De esta forma hay una rotación en el campo magnético del devanado del estator y genera un movimiento en el imán del rotor del motor dándole movimiento al eje.

PASO	TERMINALES			
	A	B	C	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

Figura 5.5 Secuencia par el manejo de un motor paso a paso bipolar

En la figura 5.6, se muestra el puente H, tomado del sitio de Internet de acuerdo a apéndice 11, con el cual se energiza una bobina del motor. Cabe destacar que para energizar la otra bobina del motor es con un puente H igual al de la figura 5.6, lo que cambia son las señales de control que vienen de otros pines del PIC.

El nombre de puente H viene de cómo están montados los transistores. Como se observó en la figura 5.5 para una misma bobina nunca se encuentran en alto al mismo tiempo las señales de control, así en la figura 5.6 D0 y D1 nunca van a tener valores de 5V al mismo tiempo. En la figura 5.6 se muestra el recorrido de la corriente para cuando D0=5V (verde) y D1=5V (rojo). Cuando D0=5V en el emisor del transistor 2n2222n correspondiente se tiene el mismo voltaje y en el colector se tiene 0V, lo que hace que el transistor T4 de tipo NPN se active y T3 de PNP también, de esta forma la corriente tiene un camino para llegar a tierra pasando por la bobina energizándola. En ese mismo momento como D1 esta en bajo el transistor T2 de tipo NPN no esta activado y por lo tanto no hay paso de corriente por él. Esta misma situación ocurre cuando D1=5V y D0=0V pero esta vez los transistores polarizados se invierten y el camino de la corriente es diferente lo que hace invertir el campo en esta bobina.

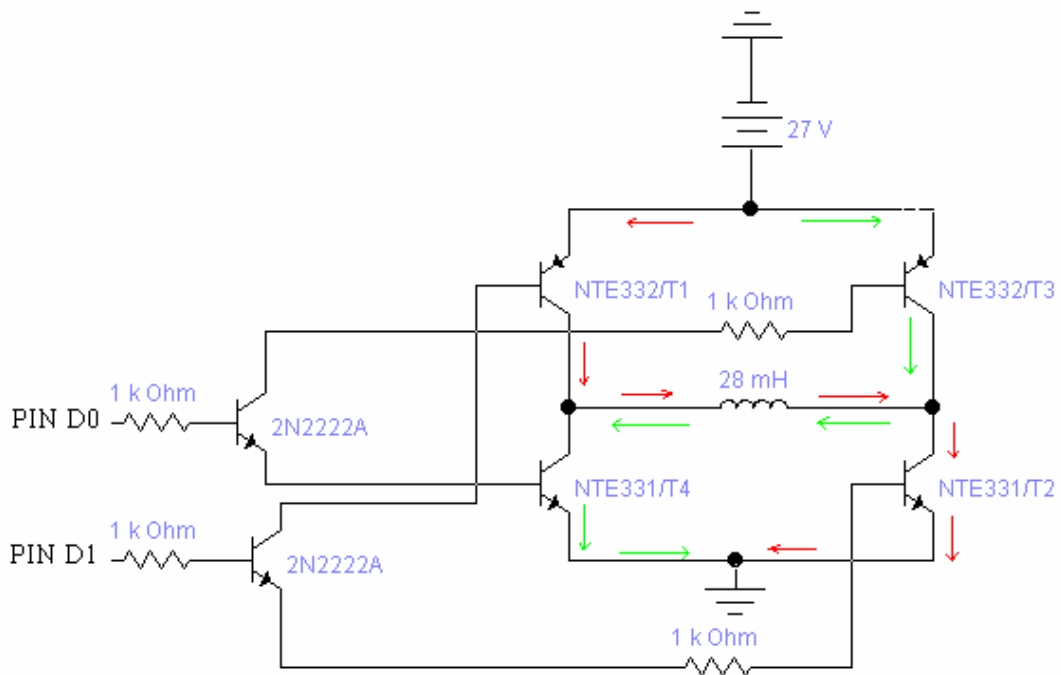


Figura 5.6 Puente H para el control de motor paso a paso

Para estos puentes H se utilizaron transistores de potencia para el manejo de las bobinas, NTE331 tipo NPN y NTE332 tipo PNP, cabe destacar que estos transistores son complementarios y la corriente máxima es de 15A, lo que supera con gran porcentaje la corriente máxima en cada bobinado; los transistores utilizados para polarizar la base de los transistores de potencia son lo conocidos 2n2222A tipo NPN. Es importante mencionar que en la figura 5.6, faltan los diodos de rodamiento libre, estos en el circuito se coloca un diodo para cada transistor de potencia, en el caso de los NPN el ánodo se coloca en el emisor y el cátodo en el colector; en cambio en los PNP el ánodo se coloca en el colector y el cátodo en el emisor. Estos diodos sirven para proteger al circuito, están inversamente polarizados porque al conmutar el circuito y tener una inductancia la corriente se dice que se invierte en la bobina y la regresa al circuito, al estar este diodo la corriente lo polariza y se descarga a través de él, así no se ve afectado el circuito, en este caso, los transistores.

En la figura 5.7 se tiene el circuito de control para la bobina de apertura de la fuente de radiación. Primeramente si la patilla E0 del PIC esta en alto deja la bobina del relé tiene paso a tierra con la cual se energiza y cierra el contacto normalmente abierto. Al cerrarse este contacto la bobina grande es energizada y por lo tanto mueve el pistón y abre la fuente de radiación. En la figura se observa el diodo de protección, en este caso como el sentido de la corriente no debe estar cambiando y sólo tiene una dirección, este diodo se coloca en paralelo a la bobina.

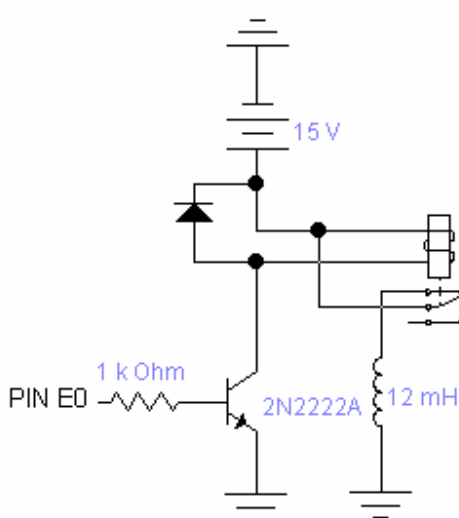


Figura 5.7 Circuito de control de bobina de apertura de la fuente

En la figura 5.8 se tiene el mecanismo de apertura, cuando la bobina es energizada por acción del campo atrae el pistón y este en el otro extremo hace abrir la fuente, al mismo tiempo el resorte es comprimido. Al desenergizarse por acción del resorte vuelve a la posición de equilibrio y por lo tanto la fuente de radiación se cierra.

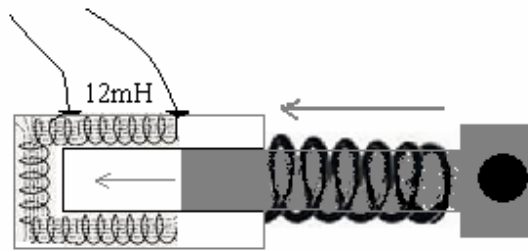


Figura 5.8 Bobina y pistón para apertura y cierre de la fuente de radiación

Para el almacenamiento de datos de radiación se utilizó la memoria serial 24FC512, esta es de la misma empresa del microcontrolador PIC, Microchip, con lo cual existe un protocolo de comunicación ya establecido entre ambos. Este protocolo de comunicación es I2C y se dan en las hojas de datos la forma cómo deben ser conectados estos dispositivos y el formato para la lectura y escritura. Esta memoria serial con protocolo I2C es la segunda de mayor tamaño, 64KBytes, que ofrece Microchip, y la de mayor capacidad disponible en el país. De esta forma se tiene una memoria con la cual se puede ir al campo y realizar varias mediciones sin antes tener que descargar los datos. Después de varias pruebas con una tubería de pruebas y de acuerdo a la fuente de radiación utilizada el Ing Leonel Siles Loaiza, encargado de control de calidad en temas de radiología, determinó que cada recorrido cubre aproximadamente 10cm de la tubería. De acuerdo a lo anterior, y a la cantidad de memoria utilizada para cada grafica, 2Kbytes ver figura 5.18, en la memoria se pueden almacenar 32 gráficas. Se puede ir al campo y hacer mediciones en aproximadamente 3.2 metros de tubería para luego ser graficados en el programa de alto nivel.

En la figura 5.9 se tiene la conexión para la memoria. Las patillas A0, A1 y A2 están a tierra debido a que esta memoria tiene la capacidad de trabajar con siete memorias más y estas patillas se encargan de seleccionar con cual memoria se está trabajando. Como únicamente se está trabajando con ella se ponen a tierra estas señales. La patilla WP (write protect) esta a tierra debido a que no se quiere tener protección en la escritura. La patilla SCL es la encargada de sincronizar la

transferencia de datos y el dispositivo. La patilla SDA da la dirección y el dato. Estas patillas son de colector abierto y por ello se coloca una resistencia a Vcc. Estas resistencias son conocidas como pull up en donde se conectan entre una señal lógica, SDA Y SCL, y Vcc, su función es asegurar que las señales no queden en estado flotante.

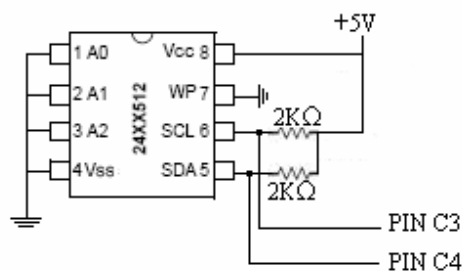


Figura 5.9 Conexión de memoria serial 24FC512

Para el medidor de radiación se tiene el contador Geiger. Este contador tiene salida serial RS-232 para PC disponible con conector DB-9. La salida es un número de dos bytes cada segundo que representa la previa cuenta por segundo. Primero se envía el byte bajo y luego el alto. La salida del contador es de 5V y se esperaría conectarlo directamente en el PIC pero sin embargo los parámetros de comunicación del Geiger son: 9600 bps, invertido, 8 bits de datos, no paridad y un bit de parada. Al estar la señal invertida el PIC reconocería mal en dato ya que cambiaría los 1 por 0 y 0 por 1. Por esto es necesario pasarlo por Max232 explicado más adelante.

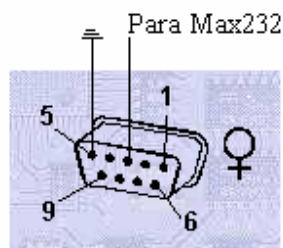


Figura 5.10 Conector DB9 del contador Geiger.

Para el control externo se utilizaron unos pulsadores. En la figura 5.11 se muestra la forma en que se conectó estos pulsadores. Cuando el pulsador esta

afuera en la señal del PIC se tiene 0V en cambio cuando el pulsador está adentro en la señal del PIC se tiene 5V, de esta forma al pulsar el PIC detecta 5V en alguna de sus patillas de interrupción y ejecuta de acuerdo al pulsador oprimido. Cabe destacar que se tienen tres pulsadores: uno para posicionar la fuente a cero, otro para dar movimiento de recorrido normal y el último para detener y paros de emergencia.

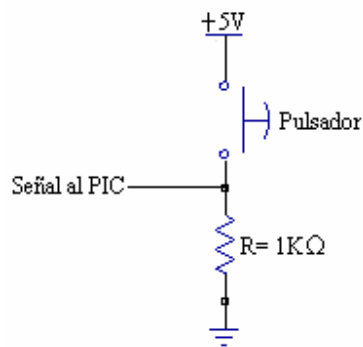


Figura 5.11 Circuito para pulsador.

Cuando se da el posicionamiento de la fuente a cero o cuando se da la vuelta completa al recorrido está siempre debe llegar al mismo lugar, es por ello que se debe colocar un final de carrera en la cremallera, esto para tener un control real sobre de adónde iniciar y adónde terminar, ya que si no se tiene esto cuando se da la gráfica de posición la referencia no sería la misma y no se tendría una buena ubicación del lugar de la tubería que corresponde a ese dato. Se colocó un sensor magnético en la cremallera y por medio de un imán colocado cerca del motor en movimiento. Al pasar el imán cerca del sensor por medio de su campo magnético cierra el contacto normalmente abierto del sensor con lo cual permite mandar una señal al PIC. Sería un esquema parecido al de la figura 5.11 sólo que en vez del pulsador está el sensor que cierra o abre los contactos.

Para la comunicación de la PC y el contador Geiger con el PIC fue necesario dos Max232. El PIC trabaja con niveles TTL niveles de 0 y 5V, en cambio las líneas del puerto serie 232 de la PC trabaja con voltajes de +12V y -12V. Siendo +12V lógica 0 y -12V lógica 1. De esta forma se podría pensar que la lógica de voltaje está

invertida y es por esto que la señal del contador Geiger, como se mencionó anteriormente invertida, se pasa por este circuito y así tener en el PIC los datos reales. Esto porque el Max232 se encarga de invertir la señal para hacer de un voltaje positivo de +5V sea un 1 lógico y 0V sea un 0 lógico. En la figura 5.12 se ve el circuito de conexión del Max-232 de acuerdo a las hojas de datos.

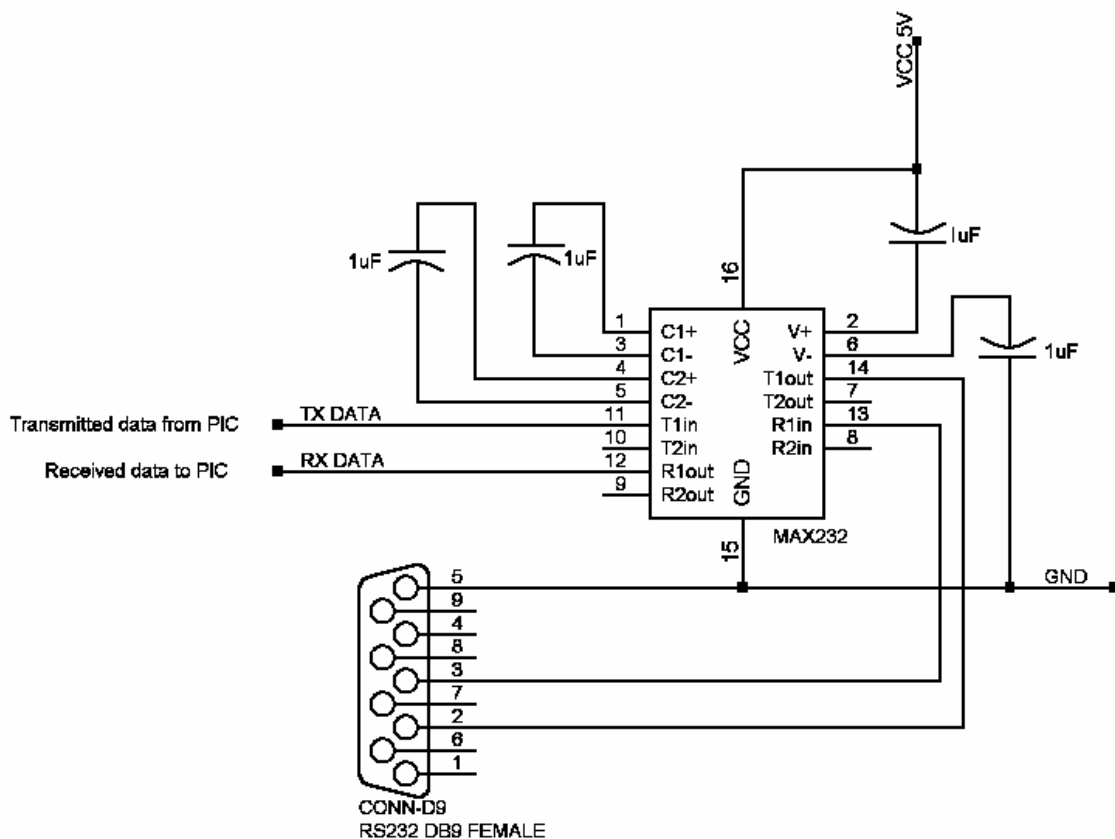


Figura 5.12 Circuito de conexión del Max232

En la figura 5.13 se muestra el circuito implementado para los PICs. Se deben de enviar dos datos al PIC: uno del contador Geiger para la radiación y el otro los datos de control de la PC. A nivel de software se trató de implementar en el PIC un programa en donde fuera capaz de recibir estos dos datos seriales. Sin embargo esto no se pudo lograr, sólo para el caso de transmisión sí fue capaz de enviar datos por dos patillas del PIC. Por esta razón se tuvo que implementar dos PICs: el primero para la recepción de datos de la PC para el control y el segundo para la recepción y

envío de datos a la PC de los niveles de radiación. De esta forma se implementó un PIC principal y uno secundario.

A nivel de hardware el PIC principal recibe comandos de tres pulsadores externos, envía señales de control para PIC secundario, abre y cierra la fuente de radiación, recibe señal de final de carrera para indicar posición de la fuente, envía señales de control para el movimiento del motor paso a paso, recibe dato de segundos dado por el Geiger y por último recibe datos de control por el puerto serial RS232 de la PC. El PIC se encarga de recibir datos por el puerto serial RS-232 del Geiger, almacena y datos en la memoria EEPROM por el puerto serial I2C, recibe datos del PIC y actúa de acuerdo a ellos.

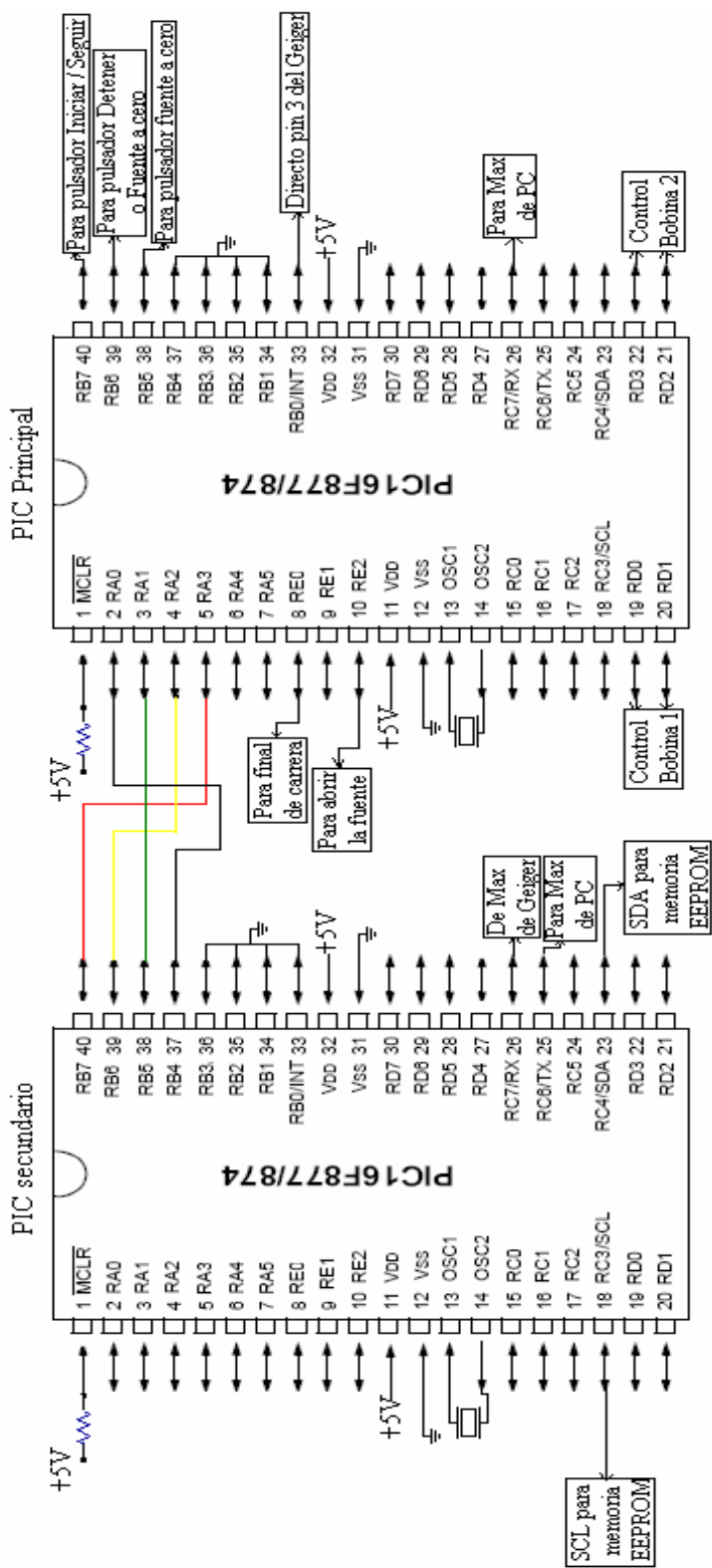


Figura 5.13 Circuito de conexión para PIC principal y secundario.

5.3 Descripción de Software

5.3.1 PIC principal

Las patillas utilizadas por el PIC son:

Puerto A: A0...A3 A0...A3 salidas de codificación para PIC2

Puerto B: B0 B0 entrada pulsos de 1seg

B5...B7 B5...B7 entradas botones de control

Puerto C: C7 C7 puerto serial, RV de la PC

Puerto D: D0...D3 D0...D3 salidas del motor stepper

Puerto E: E0 E0 entrada final de carrera

E1 E1 led siempre encendido

E2 E2 abre fuente radiación

Las interrupciones utilizadas por el PIC son:

#INT_EXT interrupción externa

#INT_RB Cambio en uno de los pines B4, B5, B6, B7

#INT_RDA SCI dato serie recibido

En el programa se tienen dos banderas principales: reversa y habilitar. La primera de ellas es para dar pasos al motor al inicio para posicionar la fuente a cero y la segunda se utiliza para indicar el inicio de recorrido hacia delante, se da el movimiento del motor y se reconocen datos por la interrupción externa.

De esta forma se tiene en la programación un *do* (hacer) que verifica siempre *while(TRUE)* el estado de estas dos bandera. Esto se hace por medio de unos *if* y *else* con los cuales a partir del estado se realizan acciones. Así en este *do* si habilitar es 1, entonces entra a un *while* (mientras) que la bandera terminó sea 0, esta

bandera se pone a 1 cuando finaliza el recorrido; o sea, mientras no se concluya el recorrido, y además, debido a una interrupción no se ponga habilitar en 0 (debido a botón detener) y detener en 1 (debido a paro de emergencia), se dan los pasos al motor por medio del puerto d. De esta forma en este *do* se tiene los pasos del motor hacia delante, estos pasos se realizan con un contador de para tener un control adecuado de pasos para el motor como se explicó en el hardware del motor. Así el motor va a estar girando hacia delante mientras habilitar=1 y detener=0, si cambian estos estado el programa hace un *break* (romper) del *while* y no se dan pasos. Si habilitar es 0 no hace nada.

En el caso de que reversa se pone a 1 entra en el *if*, se pone a 0 para hacer esto sólo una vez y entra en un *while* (mientras) que da pasos al motor por medio del puerto d pero esta vez en reversa, esto se hace igualmente por medio de un contador de control. Este *while* da pasos hasta que en pin E0 pase de estado 0 a 1, este pin es la estrada del final de carrera para posicionar la fuente al inicio siempre en la misma posición. De igual forma si hay un paro de emergencia el programa sale del *while*. Al salir del *while* se pregunta por medio de un *switch* el valor de detener, esto para abrir la fuente de radiación en el caso de detectar el final de carrera o mantenerla cerrada en caso de paro de emergencia; además de esto pone a 1 la bandera *fuente* para poder pulsar el botón de Inicio.

Como se dijo hay tres interrupciones, por medio de la directiva *#priority* se le da prioridad a las interrupciones en donde los elementos de mayor prioridad se colocan de primero. Con lo cual en este programa el orden de prioridad es rb, ext, rda.

La mayor parte del tiempo el programa principal le está dando pasos al motor. Al ocurrir una interrupción el programa la atiende y luego vuelve donde había quedado. En el caso de la interrupción RB se lleve a cabo es debido a que uno de los 3 botones externos se pulso. Al ocurrir esta interrupción se pregunta por el pin del PIC que está en alto, o sea el botón oprimido, ya que como se explicó en el hardware al ser oprimido un botón la patilla correspondiente se pone en alto. El pin B7

corresponde a Iniciar-Continuar, el pin B6 corresponde a Detener-Paro de emergencia y el pin B5 corresponde a posicionar la Fuente a Cero.

Por medio de banderas se lleva un control sobre los botones, como no poder oprimir un botón si antes haber oprimido otro. Así por ejemplo primero se debe de oprimir el botón Fuente a Cero, ya que Iniciar ni Detener tienen algún efecto en el programa. Es hasta cuando la fuente está correctamente posicionada que la bandera *fuentes* se pone en 1 y únicamente es el botón Iniciar que es aceptado. Además el botón Fuente a Cero es aceptado una única vez, sólo si se da paro de emergencia es de nuevo aceptado. Cuando el botón Fuente a Cero se resetean algunas banderas, la bandera reversa, que da pasos para atrás, se pone a 1 y le envía un control a PIC secundario para ubicar los punteros de memoria.

Una vez posicionado bien la fuente y oprimido el botón Iniciar la bandera *habilitar* se pone a 1 y además se le envía al PIC secundario la señal de inicio. Durante el recorrido hacia adelante puede ser que el operario detenga el sistema y con lo cual el botón Detener es detectado por el control sólo en el recorrido hacia delante. Cuando se oprime este botón la bandera *habilitar* se pone a 0 y se para el sistema, además del envío de la señal de detención al PIC secundario.

Si ocurre alguna emergencia y se debe dar un paro, al oprimir dos veces el botón detener ocurre un paro en el PIC. Así el primer pulso da la detención y si se vuelve a oprimir ocurre un paro de emergencia. Si ocurre esto se detienen los pasos del motor y se cierra la fuente de radiación, ya que el pin E2 se pone a 0. Ocurre además un reset de banderas y se le envía al PIC secundario de este paro.

Cada vez que es oprimido un pulsador de estos da un delay de 400ms para que el pulso sea detectado una vez, ya que por ejemplo si se da detener por un error del operario puede ser que el PIC detecte dos pulsos y se de un paro de emergencia.

En el caso de la interrupción externa se encarga de dar la señal de finalización del recorrido. Esta interrupción se lleva a cabo cada vez que el Geiger le envía un dato al PIC secundario, o sea, cada segundo. Si la bandera *habilitar* está en 1, motor

hacia delante, el PIC aumenta un contador y si este es mayor de 25, serían 25 s, pregunta cada interrupción por el estado de la entrada E0 que es la de final de carrera, de esta forma si el pin está en 1 es porque ya se completó la vuelta y entonces se envían las señales que le indican al PIC esta finalización. Se pone *habilitar* en 0, se da un reset de banderas y le indica al PIC secundario que ya no guarde más datos en la memoria y se cierra la fuente de radiación. Se da un tiempo de espera para preguntar por la entrada ya que si se pregunta de una vez por este estado el sensor está en alto debido al movimiento en reversa del motor así se da una espera para que el sensor pase al estado bajo.

En el caso que ocurra una interrupción RDA, debido a un carácter en el pin RCV se guarda el carácter y luego se hace un *switch* del carácter en donde se preguntan por los diferentes valores. Como se sabe estos pueden ser para control o para solicitud de datos para gráfica.

Tabla 5.1 Caracteres reconocidos por el PIC principal

Carácter	Debido a
c	Programa Java con control
l	Leer posición de memoria
o	Mandar otro dato a PC
m	Iniciar
j	Continuar
k	Detener
b	Borrar datos de memoria
p	Paro de emergencia
s	Fuente a cero
1	Tomar datos de gráfica 1
2	Tomar datos de gráfica 2
3	Tomar datos de gráfica 3
4	Tomar datos de gráfica 4
5	Tomar datos de gráfica 5

En la tabla 5.1 se muestran los caracteres enviados de la PC al PIC principal y qué indica cada uno.

Para el carácter *c* se le indica al PIC secundario que se va a utilizar el programa de Java con control, además por medio de la función

`disable_interrupts(INT_RB)` se le indica al PIC no reconocer ningún botón externo. También se da un reset de banderas.

El carácter *l* al ser detectado por el PIC le envía una señal de control al PIC secundario sobre leer una posición de memoria; no actúa sobre PIC principal. Igualmente esto sucede con el carácter *o* pero le indica al PIC secundario enviar otro dato de la memoria a la PC.

El carácter *m* es de control e indica Iniciar, pone la bandera *habilitar* en 1 y le indica al PIC secundario esta condición.

El carácter *j* es de control e indica continuar, actúa de la misma forma que carácter *m*.

El carácter *k* es de control e indica detener, pone la bandera *habilitar* en 0 y le indica al PIC secundario esta condición.

El carácter *b* le indica al PIC secundario sobre borrar los datos de memoria recién enviados para la gráfica.

El carácter *p* es de control e indica paro de emergencia, la bandera *habilitar* la pone en 0, la bandera *detener* en 1 y cierra la fuente de radiación, le indica al PIC esta condición.

El carácter *s* es de control e indica posicionar la fuente a cero, actúa de la misma forma sobre el PIC como cuando se oprime el botón Fuente a Cero.

Los caracteres *1*, *2*, *3*, *4* y *5* le indican al PIC secundario sobre que posiciones de memoria van a ser leídas.

En las figuras 5.14, 5.15, 5.16 y 5.17 se muestran los diagramas de flujo del programa del PIC principal.

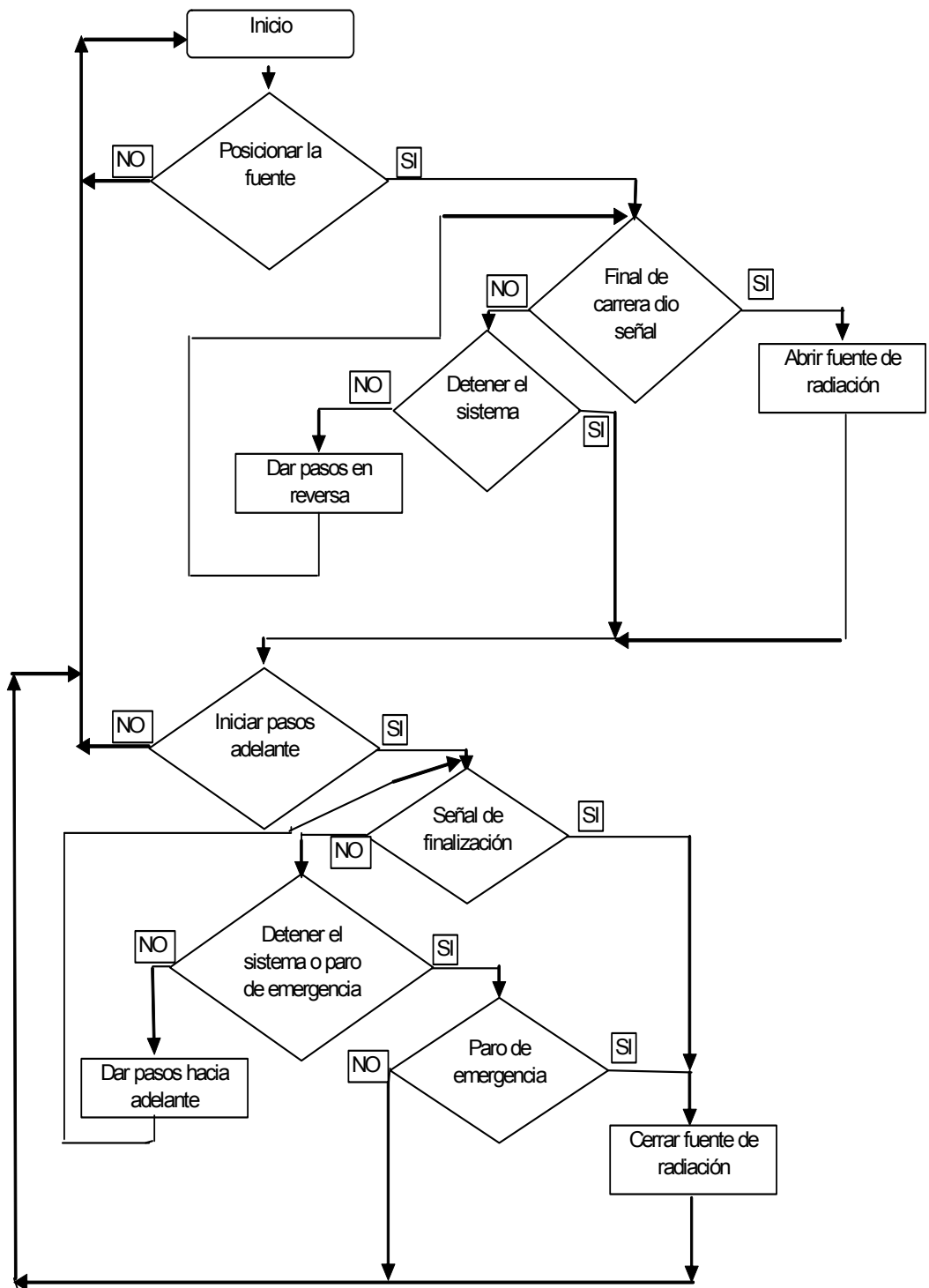


Figura 5.14 Diagrama de flujo del ciclo *Do* del programa del PIC principal

En la figura 5.14 se muestra el diagrama de flujo para bloque *Do*, como se aprecia una vez llegado al punto más bajo vuelve de nuevo al inicio, esto se hace por medio de `while(TRUE)`; así sólo en el caso de una interrupción el programa sale de este *Do*.

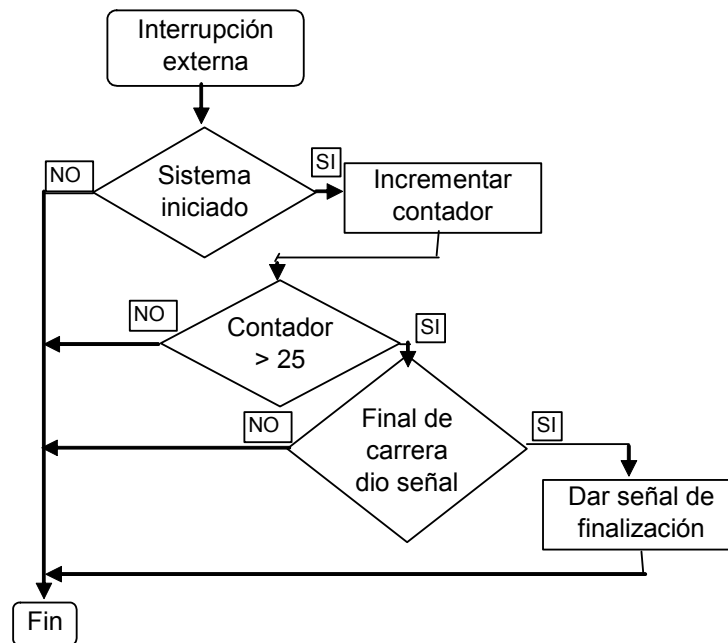


Figura 5.15 Diagrama de flujo para la interrupción externa del PIC principal

En la figura 5.15 se muestra el diagrama de flujo para la interrupción externa, obsérvese que la pregunta por el sensor se hace una vez iniciado el sistema y el contador ser mayor a 25; si estos casos no se cumplen termina la interrupción.

En la figura 5.16 se muestra el diagrama de flujo de la interrupción RB, se nota como primero se debe de oprimir el botón Fuente a Cero para poder oprimir Iniciar, y de cómo al dar Detener dos veces, caso que se puede dar estando el motor iniciado, ocurre un paro de emergencia.

En la figura 5.17 se muestra el diagrama de flujo de la interrupción RDA, se muestran los diferentes caracteres que reconoce el PIC la acción del PIC para cada uno.

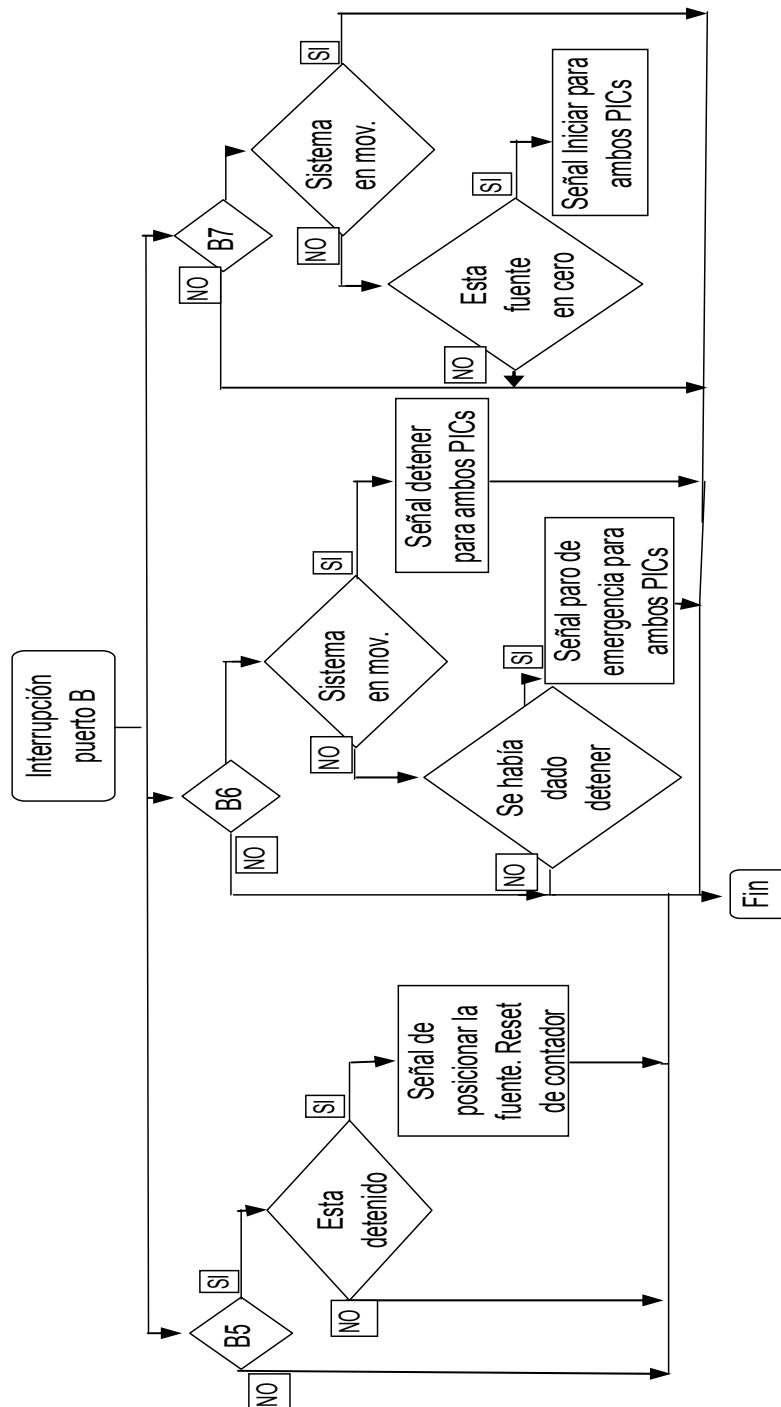


Figura 5.16 Diagrama de la interrupción puerto B

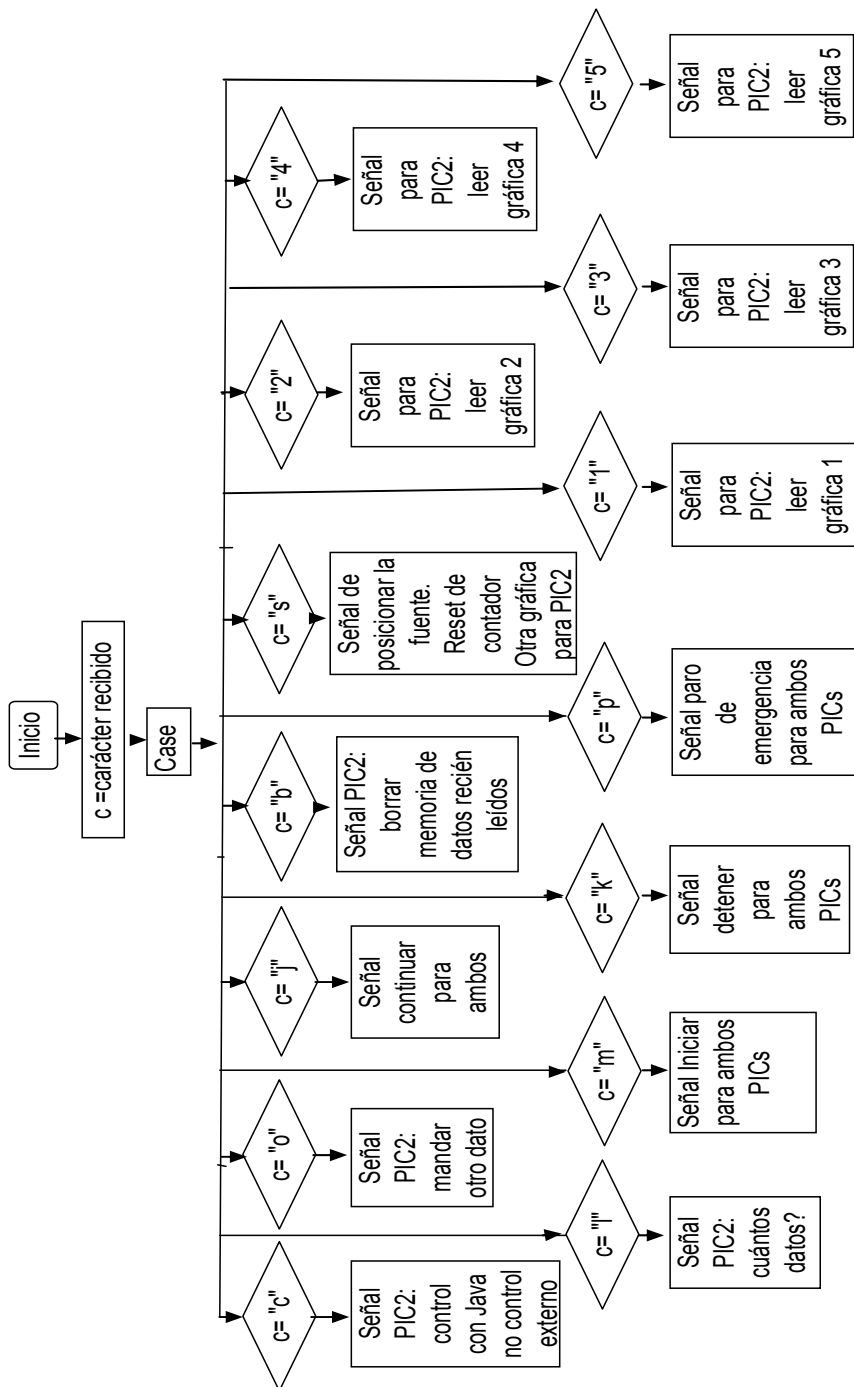


Figura 5.17 Diagrama de flujo de la interrupción RDA

5.3.2 PIC secundario

Las patillas utilizadas por el PIC son:

Puerto A: A0 A0 led siempre encendido
Puerto B: B4...B7 B4...B7 entradas botones de control
Puerto C: C3...C4 C3 SCL EEPROM externa, C4 SDA EEPROM externa
C6...C7 Puerto serial, C6 TX a la PC, C7 RV del Geiger

Las interrupciones utilizadas por el PIC son:

#INT_RTCC Desbordamiento del timer 0
#INT_RB Cambio en uno de los pines B4, B5, B6, B7
#INT_RDA SCI dato serie recibido

Este PIC se encarga de recibir datos del Geiger y almacenarlos en una memoria externa, además de luego enviar estos datos a la PC. Para cada gráfica se midieron aproximadamente 670 CPS con lo cual cada uno son 2 bytes en total son 1340 bytes con lo cual se dispuso de 2Kbytes de memoria de la EEPROM, además de ello se tiene un control de la cantidad de datos que son almacenados en memoria. La cantidad de datos almacenados en la memoria externa de cada gráfica se almacena en la memoria eeprom interna del PIC.

En la figura 5.18 se observa las posiciones de memoria en la eeprom externa de las gráficas, para cada una se utiliza 2Kbytes de memoria. En la tabla 5.2 se muestra las posiciones de memoria de la eeprom interna. Primeramente se lleva un control de la cantidad de graficas que se llevan en memoria, esto para cuando se va a realizar otra gráfica ubicar el puntero en la posición adecuada y no almacenar datos encima de los datos de otra gráfica. Por otra parte, como se mencionó son aproximadamente 670 CPS con lo cual se utilizan dos espacios de memoria para la cuanta de los datos. Así cada vez que el dato en la primera dirección llega a 255 se

aumenta en 1 el dato en la segunda dirección de esa gráfica. Luego se concatenan en Java y se tiene la cuenta final.

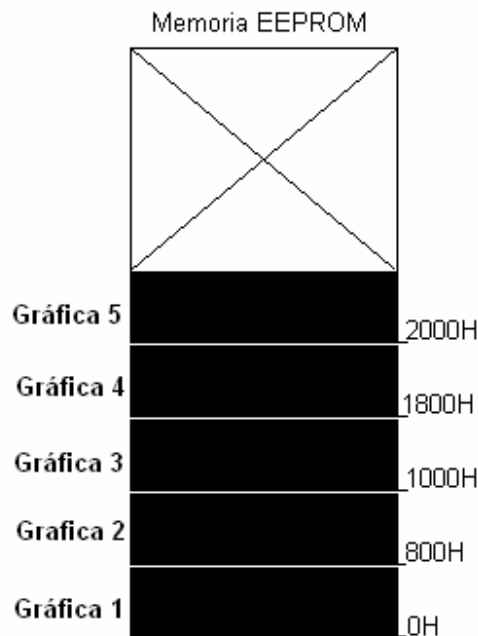


Figura 5.18 Espacio de memoria y dirección inicial para gráficas en la EEPROM externa

Tabla 5.2 Dirección y dato a almacenar en la EEPROM interna del PIC

Dirección de memoria	Dato almacenado
0 h	Cantidad de gráficas
1 h	Cantidad datos gráfica 1
2 h	Cantidad datos gráfica 1
3 h	Cantidad datos gráfica 2
4 h	Cantidad datos gráfica 2
5 h	Cantidad datos gráfica 3
6 h	Cantidad datos gráfica 3
7 h	Cantidad datos gráfica 4
8 h	Cantidad datos gráfica 4
9 h	Cantidad datos gráfica 5
A H	Cantidad datos gráfica 5

De esta forma al iniciarse el PIC lo primero que debe hacerse es colocar los punteros de las memorias en la posición adecuada. Esto se hace por medio de leer la posición 0 de la eeprom interna, ya que lleva la cantidad de graficas almacenadas. Así se lee este dato y por medio de un *switch* se pregunta por el valor y de acuerdo a

el se le dan valores a los punteros *gráfica* y *número*. Como se mencionó cada gráfica va a tener dos posiciones de memoria para la cantidad de datos y es por esto que el puntero *gráfica* varía de dos en dos. Esto sólo se realiza una vez y al inicio, cuando se alimenta el PIC.

Tabla 5.3 Valores del puerto debido a la interrupción RB una vez rotado e indicaciones al PIC

Valor del puerto b	Indica
0	Nada
1	Java con control
2	Enviar dato de cantidad
3	Enviar dato de radiación
4	Iniciar
5	Detener
6	Otra gráfica
7	Se completó la vuelta
8	Enviar cantidad de datos gráfica 1
9	Enviar cantidad de datos gráfica 2
10	Enviar cantidad de datos gráfica 3
11	Enviar cantidad de datos gráfica 4
12	Enviar cantidad de datos gráfica 5
13	Borrar datos grafica recién leída
14	Nada
15	Paro de emergencia

Igualmente se introdujo prioridades para las interrupciones para el PIC, el orden de prioridad es rb, rda, rtcc.

En el caso de que ocurra la interrupción RB, primero se lee el puerto_b y para tener un mejor control se rota 4 veces a la derecha, lo cual introduce ceros a la izquierda con cada rotación. Al final se tiene un valor de los bits B4...B7, lo cual se presentan en la tabla 5.3. Hay que recordar que esos 4 bits son enviados por el control del PIC principal.

En el caso de ser 0, no ocurre nada. En el caso de ser 1 indica que es Java con control, con lo cual cada vez se va a graficar de una vez, con lo cual siempre se posiciona al inicio los punteros únicamente en la gráfica 1. Además se da un reset de los contadores de datos.

En el caso de ser 2 le indica al PIC que debe de leer una posición de la eeprom interna y luego la envía por el puerto, como se sabe son dos bytes y es por

eso, que primero manda la primera posición de la cantidad y luego la segunda. Además una vez recibido este carácter almacena en la posición 0 de la memoria interna el valor de cero, esto porque cada vez que se va a una PC a descargar todas las gráficas y al volver al campo se debe de iniciar de nuevo en la gráfica 1.

En el caso de ser 3 le indica al PIC que debe leer y enviar a la PC dos bytes de la eeprom externa, esto porque como se sabe cada dato CPS son dos bytes con lo cual en Java se deben concatenar y mostrar en la gráfica el verdadero valor. De esta forma debido a los datos de la memoria para las lecturas a la eeprom se deben de dar delays para tener el byte correcto, además se con cada caso se aumenta el puntero para el próximo envió del valor 3 y de nuevo leer los bytes siguientes.

En el caso del valor 4 (iniciar) pone la bandera *habilitar* en 1 para almacenar datos en las memorias. Además da un reset de unas banderas para el primer CPS a almacenar.

En el caso del valor 5 (detener) pone la bandera *habilitar* a 0, con lo cual no se almacenan datos en memoria.

En el caso del valor 6 igualmente coloca los punteros de las memorias en las posiciones adecuadas, este valor se da por si se quiere hacer la misma medición del recorrido y almacenar los datos en otra posición sin tener que resetear todo el sistema.

En el caso del valor 7 (terminó la vuelta) lo que hace es aumentar el 1 el valor del dato contenido en la dirección 0 de la eeprom interna (cantidad de gráficas) ya que únicamente al completar la vuelta que esta cantidad se debe aumentar, ya que si hay un paro de emergencia de debe comenzar desde el inicio con lo cual posicionar los punteros de igual forma. Igualmente la bandera *habilitar* se pone a cero y se almacenan más datos.

En el caso del valor 8 es enviar los 2 bytes de la posiciones 1 y 2 de la eeprom interna y de esta forma indicar si esta gráfica tiene datos, gráfica 1.

En el caso del valor 9 es enviar los 2 bytes de la posiciones 3 y 4 de la eeprom interna y de esta forma indicar si esta gráfica tiene datos, gráfica 2.

En el caso del valor 10 es enviar los 2 bytes de la posiciones 5 y 6 de la eeprom interna y de esta forma indicar si esta gráfica tiene datos, gráfica 3.

En el caso del valor 11 es enviar los 2 bytes de la posiciones 7 y 8 de la eeprom interna y de esta forma indicar si esta gráfica tiene datos, gráfica 4.

En el caso del valor 12 es enviar los 2 bytes de la posiciones 9 y 10 de la eeprom interna y de esta forma indicar si esta gráfica tiene datos, gráfica 5.

En el caso del valor 13 en las posiciones del puntero gráfica y gráfica+1 el valor de 0, de esta forma se da un reset la cantidad de datos de la gráfica recién leída, con lo cual si se pide de nuevo ya sea los valores 8, 9, 10, 11 o 12 se van a mandar la cantidad de 0 con lo cual indica que esta gráfica no tiene datos o no se lee.

En el caso del valor 15 (paro de emergencia) lo único que se hace es poner la bandera *habilitar* en 0 para no almacenar datos.

La interrupción RDA tiene la tarea de indicar cuando un byte llega a la patilla RV del PIC, el contador Geiger cada segundo manda 2 bytes, el primero es el menos significativo y el segundo el más significativo, el tiempo entre bytes de aproximadamente 0.1ms, con lo cual el PIC debe ser lo suficientemente rápido para detectar estos dos bytes y luego almacenarlos.

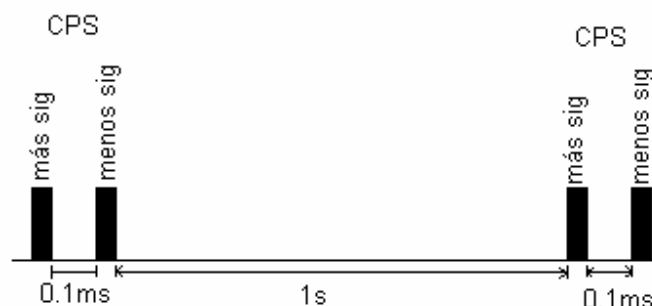


Figura 5.19 Tiempos entre CPSs y bytes

Como se puede observar en la figura 5.19 existe la posibilidad de que el comando de poner la bandera *habilitar* en 1 se dé entre el 0.1s cuando en el PIC esté llegando el byte más significativo. De esta forma desde un inicio se empezarían a almacenar mal los datos, ya que siempre se debe de almacenar el menos significativo de primero. De esta forma por medio de la interrupción RTCC es que se evita este problema. Así cada vez que pase la bandera del estado bajo a alto se permite esta interrupción.

Esta interrupción se lleva a cabo cada vez que se desborda el timer 0, por medio de la ecuación $20000000/(X*256*4)$ y tomando a X como 256 se tiene que este timer se desborda 76 veces por segundo. Lo que se hace es cuando se pone la bandera *habilitar* en 1, se pueden almacenar datos, y no se sabe cual byte llegará primero, si un menos significativo o un más significativo, es aceptar esta interrupción. Así se recibe el primer byte y al recibir el segundo se pregunta por la cantidad de desbordamientos que han transcurrido, por medio de un contador, entre los dos bytes; cercano a los 76 indica que primero se recibió un byte más significativo ya que debería ser un valor bajo que indicara que se recibió un byte menos significativo. Si se recibe un byte más significativo de primero lo que se hace es desecharlo y comenzar la cuenta a partir del byte menos significativo. Cabe destacar que esto sólo sucede en el cambio de estado de la bandera *habilitar* de 1 a 0, ya que esta interrupción si estuviera trabajando todo el tiempo le quitaría capacidad al programa.

La interrupción RDA almacena los datos recibidos sólo si *habilitar* está en alto. Como se mencionó los dos bytes recibidos están separados aproximadamente 0.1ms con lo cual no le daría tiempo al PIC almacenar en la memoria externa con cada interrupción, ya que la memoria externa ocupa aproximadamente 10ms para el almacenamiento de un byte. Lo que se hace es almacenarlos en dos variables y luego al saber que se recibió el más significativo se alienan en la memoria, primero el menos y luego el más significativo. Al ocurrir esto también se aumenta en 1 la cantidad de datos para la gráfica pero esta vez en la memoria interna del PIC, todo esto pasa entre el segundo de espera del próximo byte menos significativo.

En el programa cuando se da un almacenamiento de los datos se llama a la directiva 24fc512.c que corresponde al driver para la lectura y el almacenamiento en la eeprom. En ella se programa la inicialización y los métodos de lectura y escritura para el puerto serial I2C.

En la figura 5.20 se presenta el diagrama de flujo para el ciclo *Do* del programa, esto sólo se realiza una vez, ya que reinicio es 0 cuando se reinicia el PIC, los punteros toman los valores respectivos de acuerdo al dato almacenado en la primera posición de memoria, cantidad de datos.

En la figura 5.21 se presenta el diagrama de flujo para la interrupción del timer0, esta interrupción lo que hace únicamente es aumentar un contador cada vez que se desborda el timer0, esto es utilizado en la interrupción RDA.

En la figura 5.22 se presenta el diagrama de flujo para la interrupción del puerto b. Cada vez que se da la interrupción se lee el puerto y primeramente se rota 4 bits a la derecha para así tener los valores reales de los bits B1, B5, B6 y B7 enviados desde el PIC principal. A partir de esto se hace un case de los posibles valores y se toman decisiones a partir de estos.

En la figura 5.23 se presenta el diagrama de flujo para la interrupción RDA. Para el primer CPS cuando se da habilitar es necesario saber el tiempo transcurrido entre los 2 primeros bytes, esto para asegurarse que siempre se va a almacenar en la memoria primero el byte menos significativo de un CPS, y con esto no perder el orden de los datos. Para los otros bytes ya en orden, se almacenan en la memoria externa y por cada CPS se aumenta en uno la cantidad de datos para esta gráfica.

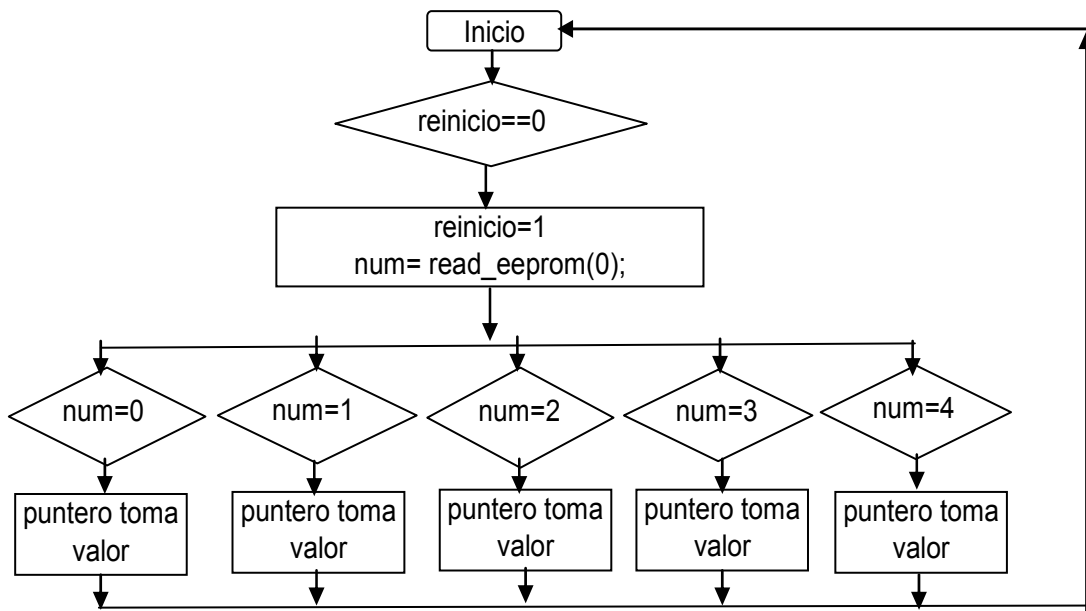


Figura 5.20 Diagrama de flujo del ciclo *Do* del programa del PIC secundario

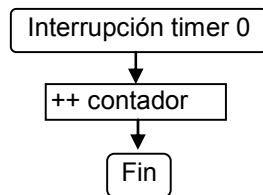


Figura 5.21 Diagrama de flujo de la interrupción RTCC

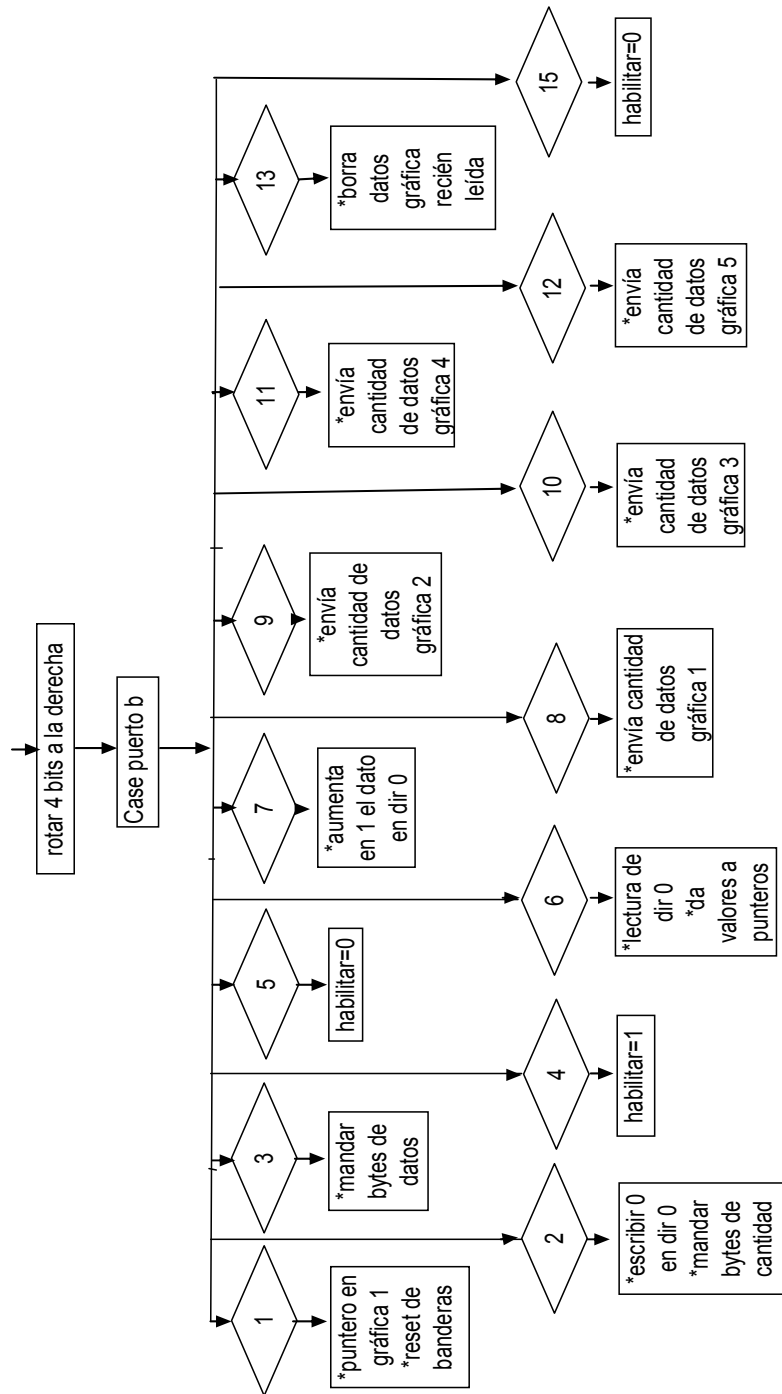


Figura 5.22 Diagrama de flujo de la interrupción RB

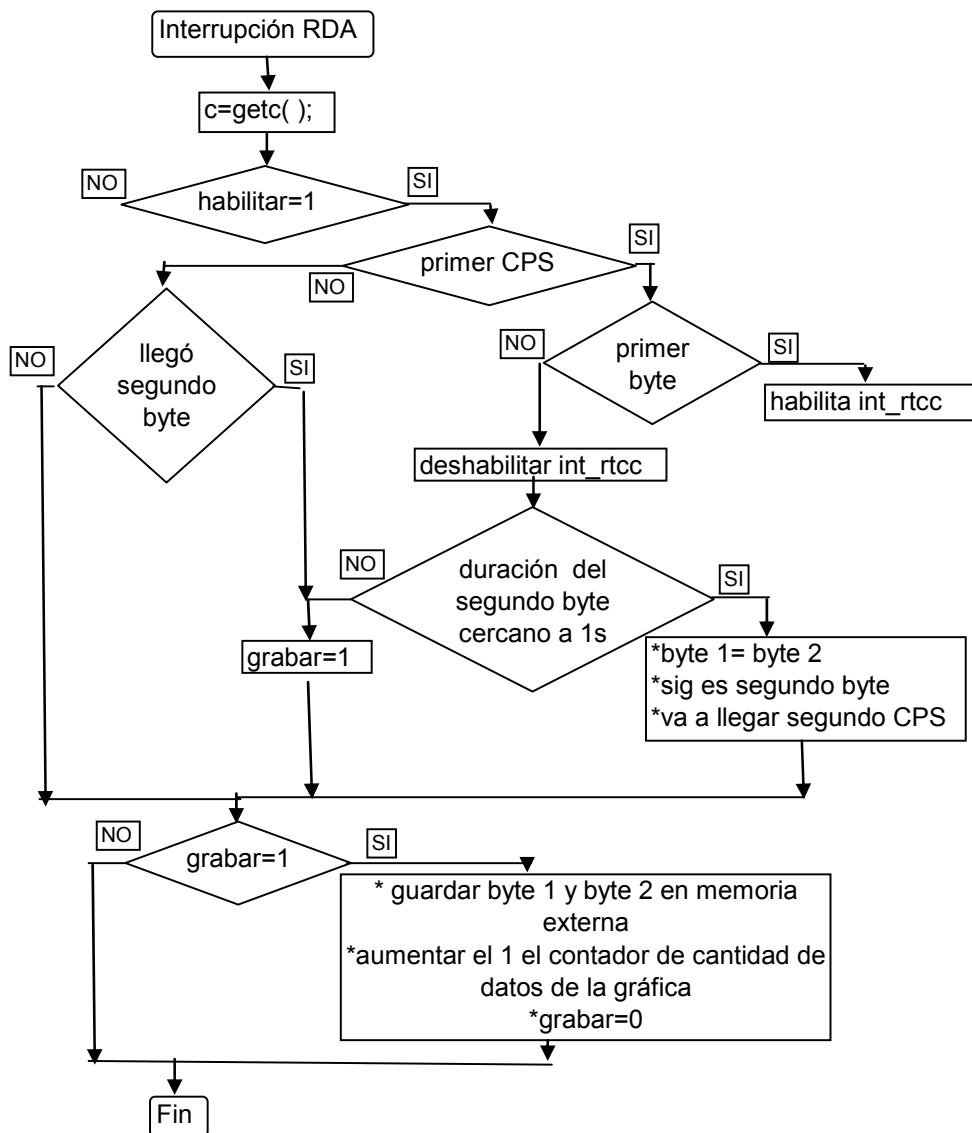


Figura 5.23 Diagrama de flujo para la interrupción RDA

5.3.3 Java

Primeramente se hizo una interfase en donde el usuario podía escoger entre la pantalla en donde viene la parte de control o la pantalla en donde va a graficar las gráficas obtenidas con el control externo. Esto se hizo por medio por medio de dos botones que al ser pulsados hacen una instancia a otra clase, en las cuales están sus respectivos controles.

En la figura 5.24 se presenta el diagrama de flujo del programa principal en donde se tienen tres botones y el usuario determina cuál se va a usar. En el caso de que se lleve la PC al campo entonces se escogería el programa con control, en el caso de que se realicen diferentes medidas con el control externo se utiliza el de sólo graficar. De esta forma se tiene la primera interfase con el usuario.

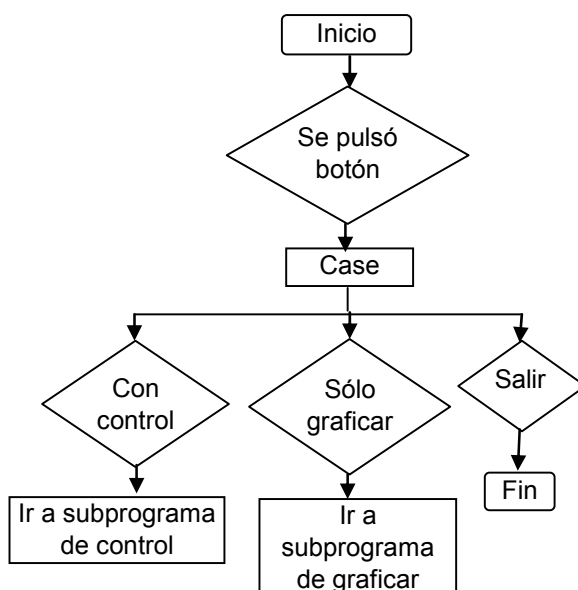


Figura 5.24 Diagrama de flujo del programa principal

En el caso del subprograma con control se solucionó de la siguiente manera. Primeramente se tienen 6 botones: "Fuente en posición inicial", "Iniciar mediciones", "Detener", "Continuar", "Graficar datos" y "Salir".

El botón *Fuente en posición inicial* es para indicarle al microcontrolador que debe posicionar la fuente de radiación en cero, el botón *Iniciar mediciones* es para iniciar el recorrido hacia delante, el botón *Detener* detiene el sistema, el botón *Continuar* sigue con el recorrido desde la misma posición de donde se encontraba antes de dar detener, el botón *Graficar datos* es para que el operario puede bajar los datos y graficarlos, ya sea hasta donde llegó la estructura o se haya completado el recorrido, y el botón *Salir* vuelve al programa principal.

Al inicio, apenas entramos a este subprograma se envía el carácter *c* por el puerto serial para indicarle al PIC que se trata de Java con control y de esta forma siempre se van a guardar los datos en la posición de la gráfica 1. Además de esto en un inicio, el usuario entra puede oprimir los botones *Fuente en posición inicial* y *Salir* ya que como con el control externo se debe de posicionar la fuente antes de hacer cualquier cosa. Esto se hace por medio de *setEnabled(false)* en el caso de los botones que no acortan ser oprimidos y *setEnabled(true)* en el caso que sí pueden ser oprimidos. Cuando el usuario oprime el botón *Fuente a posición inicial* lo que se hace es una instancia al subprograma que envía un carácter por el puerto serial. Este carácter corresponde a la letra *s*. Al dar el clic en ese mismo instante el único botón que se puede oprimir es *iniciar mediciones*, es responsabilidad del usuario detectar cuando el motor ha llegado a la posición de inicio.

Una vez llegado al inicio se pueden iniciar con las mediciones y lo que se hace es oprimir este botón. Al dar clic con este botón se manda por el puerto serial el carácter *m*. En este mismo momento el único botón que se puede oprimir es el botón *Detener*. De esta forma en el transcurso del recorrido puede dar clic en este botón o esperar que el recorrido termine. Si se da *Detener* los botones *Continuar* y *Graficar datos* son los únicos que puede ser oprimidos. *Continuar* es en el caso que de que el sistema no haya terminado el recorrido, además de ellos de nuevo el único botón que se puede oprimir es *Detener*, y *Graficar datos* es para cuando se quieren graficar los datos.

Para cuando se da detener se envía por el puerto el carácter *k*, para cuando se da continuar se envía el carácter *j* y para cuando es para graficar se envía el carácter *1*, a que estamos trabajando siempre en la gráfica 1; después de esto va al subprograma que gráfica.

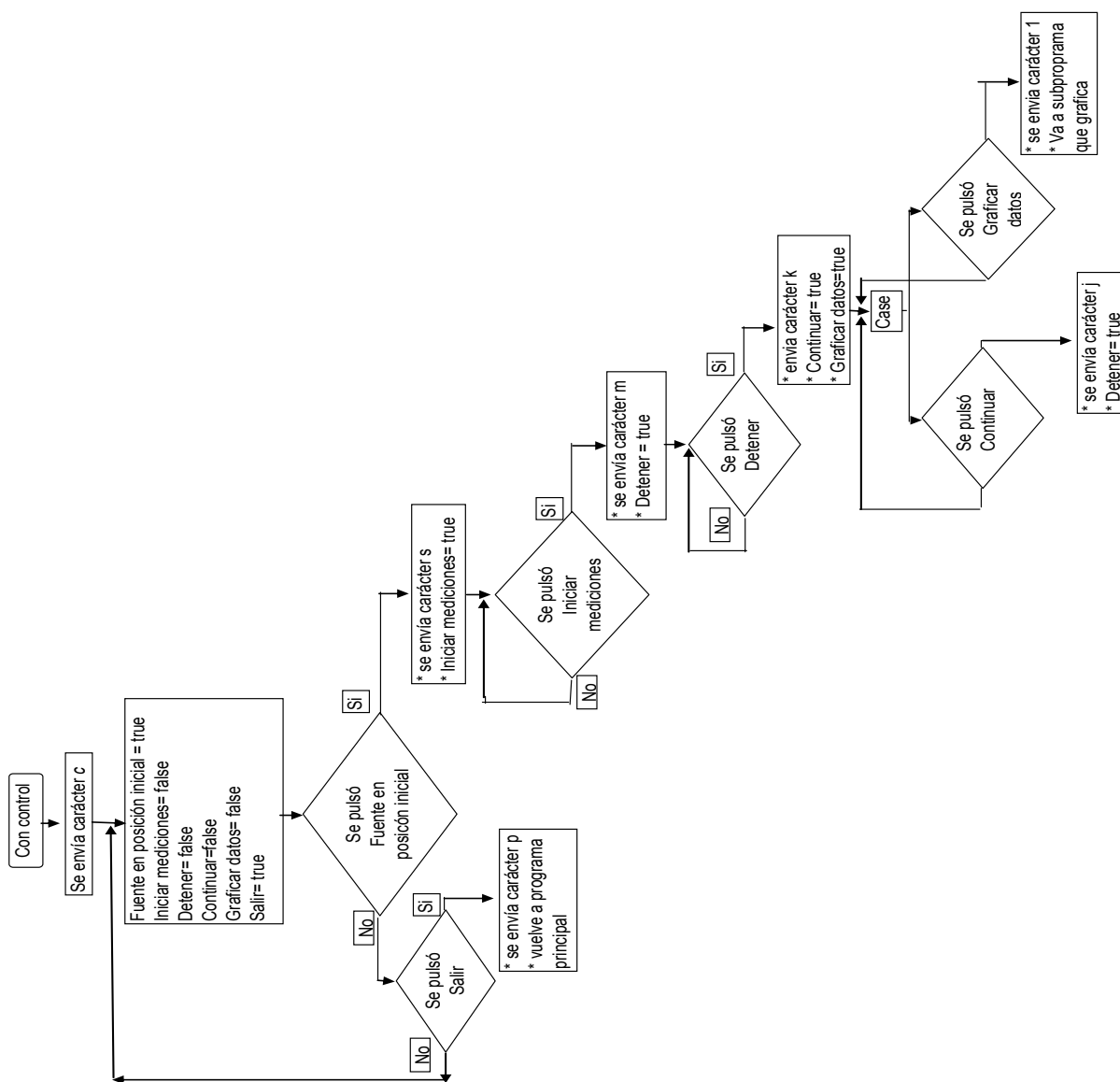


Figura 5.25 Diagrama de flujo para el subprograma de control

En la figura 5.25 se presenta el diagrama de flujo para el subprograma con control. Como ocurre la secuencia de los botones y como es hasta el final del recorrido en donde se pueden graficar los datos. En todo momento se puede oprimir el botón *Salir* con lo cual es similar al paro de emergencia con el control externo.

En el caso del subprograma de graficar, este se solucionó de la siguiente manera. Primeramente se tienen 6 botones: "*Gráfico 1*", "*Gráfico 2*", "*Gráfico 3*", "*Gráfico 4*", "*Gráfico 5*" y "*Salir*".

El nombre de cada botón indica cuales datos va a ser pedir al PIC para ser graficados. Al ser oprimido un botón, por ejemplo *Grafico 1*, se envía el carácter 1 por el puerto serial, en esto se recibe el byte menos significativo enviado por el PIC, de nuevo es enviado el carácter 1 para recibir el byte de cantidad más significativa. De esta forma estos dos datos son sumados y si la cantidad es mayor que 0 indica que esta grafica tiene datos e inmediatamente se llama al subprograma para graficar. Si por el contrario la cantidad sumada es 0 indica que esta grafica no tiene datos y por medio de *setEnabled(false)* este botón no puede volver a ser oprimido ya que sabemos que es en vano oprimirlo. Para el caso de ser la gráfica 2 es el carácter 2, en el caso de ser la gráfica 3 es el carácter 3, en el caso de ser la gráfica 4 es el carácter 4 y si es la gráfica 5 es el carácter 5. Si se da *Salir* lo único que se hace es volver al programa principal.

En la figura 5.26 se muestra el diagrama de flujo para el subprograma en donde únicamente se grafica. Acá se describe lo mencionado anteriormente en donde se verifica que la cantidad de datos de esta grafica sea mayor a 0 para iniciar con la operación de extraer los datos y graficarlos. Es importante mencionar que una vez dada la grafica y el usuario haberla cerrado vuelve a esta misma interfase en donde se pueden grafica las otras gráficas siguiendo el mismo procedimiento.

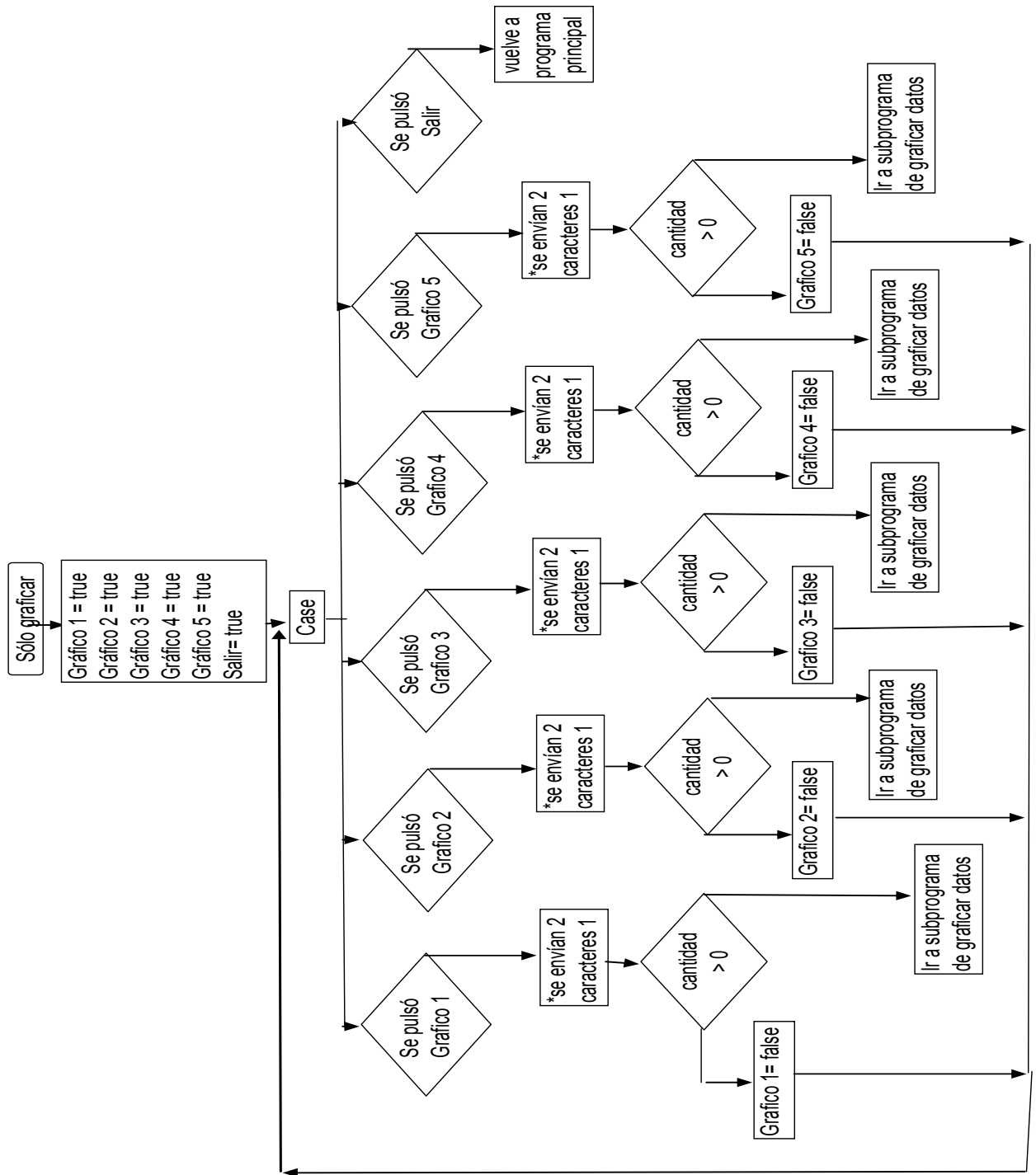


Figura 5.26 Diagrama de flujo para el subprograma sólo graficar

Para el caso del subprograma de para graficar lo que se hace es nuevo enviar, pero esta vez el carácter *I*, por el puerto serial, dos veces para saber de nuevo la cantidad de datos a graficar, esto para entrar a un ciclo *for* con el cual ir obteniendo cada CPS hasta llegar a la cantidad encontrada. Cada dentro del ciclo *for* se envía el carácter *o* para que el PIC principal le indique al secundario que debe enviar un dato de radiación; así dentro de este mismo ciclo se van guardando en un String, que al final se ve como una lista, y además de esto debido a la clase para graficar se van guardando en una serie(x, y): *series.add(angulo,numero)* que por medio de *XYSeriesCollection* pueden ser tomados como valores x, y para la gráfica. Así al final se sale del ciclo y se presentan tanto la lista, por medio de un *JTextArea*, como la gráfica por medio de *JFreeChart chart = ChartFactory.createXYLineChart ("Gráfica del estado de la tubería", "Posición (°)", "Radiación (mSv/h)", juegoDatos, PlotOrientation.VERTICAL,false, false, true)*. Esto se presenta en la figura 5.27 en la cual se observa el ciclo *for*, la cual no concluye mientras el contador no llega a la cantidad de datos encontrada enviando el carácter *I*, así se van guardando los datos para luego ser presentados en forma de lista y en una gráfica de radiación versus posición.

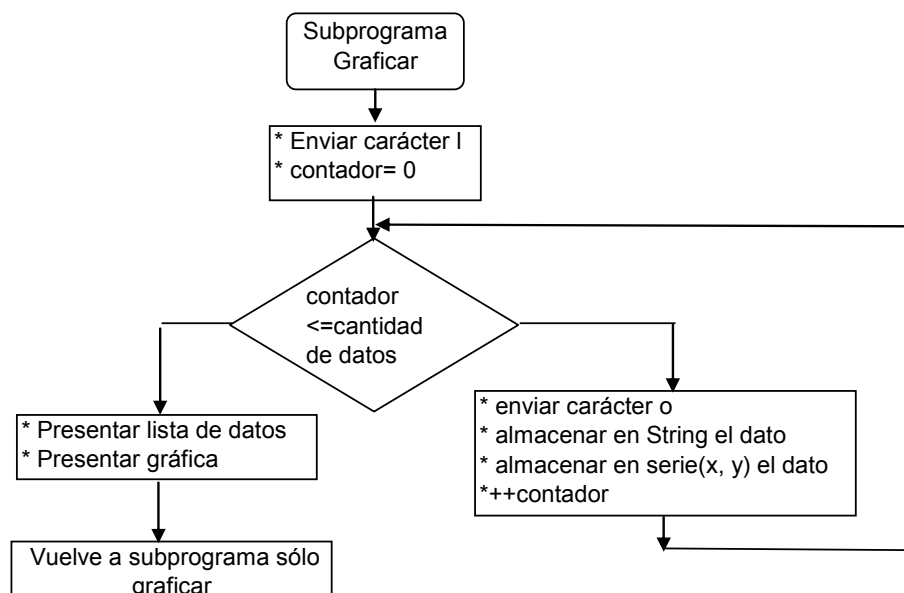


Figura 5.27 Diagrama de flujo para el subprograma para graficar

La forma en la cual se presentan los datos de radiación y posición es mediante la proporcionalidad que existe en el valor de 1 CPS en relación a las unidades internacionales de medida de la dosis de radiación mSV/h. Según los datos de fabricante del contador Geiger como se había expuesto anteriormente tiene la capacidad de detectar 9360 CPS, se especifica que 1 CPS es equivalente a 0.00052mSV/h.

Los dos bytes enviados desde el PIC secundario hacia el programa en Java correspondiente a un dato de radiación deben ser concatenados para ver que valor de CPS corresponde, entonces como el primer byte es el menos significativo sólo se almacena en una variable, el segundo byte se almacena en una variable y como es el más significativo se multiplica por 256, ya que corresponde a los bits 8...15 de dato, después de acá se suman las dos variables y se obtiene el dato de CPS. Es como que se concatenaran los 2 bytes y al final se obtiene una sola cantidad. Entonces ese valor encontrado se multiplica por el valor de 0.00052 y se obtiene en las unidades SI el valor de radiación para ese punto específico.

Hay que recordar que con cada CPS se obtiene el valor y de la gráfica. El valor x de ese punto se obtiene del tiempo de recorrido del sistema, el tipo de motor permite que los tiempos de recorrido siempre sean los mismos, se hicieron varias pruebas y se comprobó que siempre anda entre 11 minutos y 15 segundos, se tiene que para cada valor y encontrado se ha recorrido 0.533° de la estructura, entonces este es un valor constante que se suma al valor anterior para cada valor x . Así por ejemplo en el segundo 15 se tiene que la estructura va por los $15 \cdot 0.533 = 7.995^\circ$.

Capítulo 6: Resultados y Análisis

Según la comisión internacional de protección radiológica (CIPR), organismo establecido para recomendar prácticas seguras sobre el uso de radiación, la dosis máxima para una persona que trabaja con radiación es de $25 \mu\text{S} / \text{h}$ [17]. Se realizaron pruebas por medio del contador Geiger y se encontró que el operario debe controlar el equipo a 3 metros de la fuente de radiación; a esta distancia se midieron $20 \mu\text{S} / \text{h}$ y de esta forma se esta por debajo de la dosis máxima permitida.

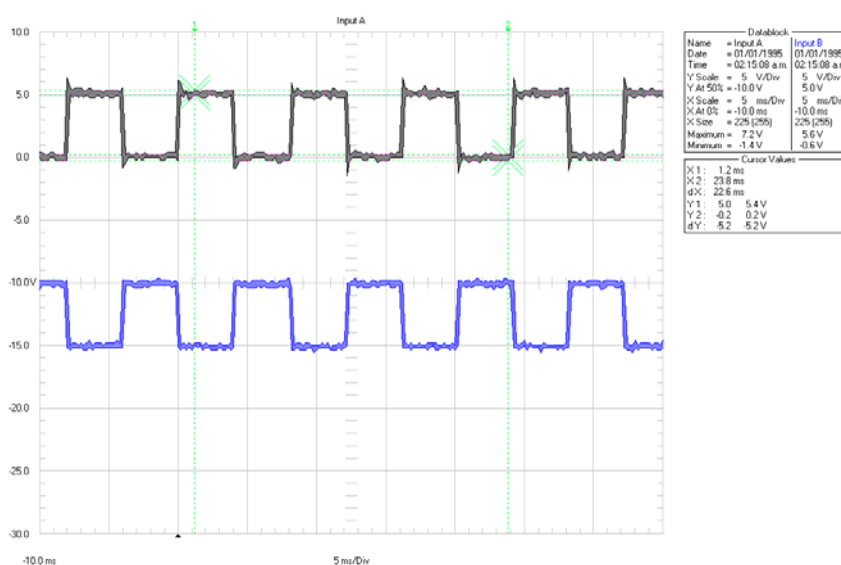


Figura 6.1 Gráfica de la salida del PIC principal para una bobina de motor paso a paso

La figura 6.1 corresponde a los pasos para una bobina del motor, corresponde a las patillas D0 y D1 del PIC principal, las salidas D2 y D3 tienen las mismas características. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.1 se tienen los pasos para una bobina del motor paso a paso, en ella se puede observar como en ningún momento una señal está en el mismo estado que la otra. Esto hace que sólo una terminal de una bobina esté dejando pasar la corriente en algún sentido y con esto el motor siempre va a girar dependiendo del orden de los pasos. En la figura 5.5 se presenta la secuencia de pasos para el

manejo de un motor de pasos bipolar en ella también se aprecia como las dos terminales para la misma bobina nunca están activas o desactivas al mismo momento. De esta forma se cumple lo que dice la secuencia de pasos.

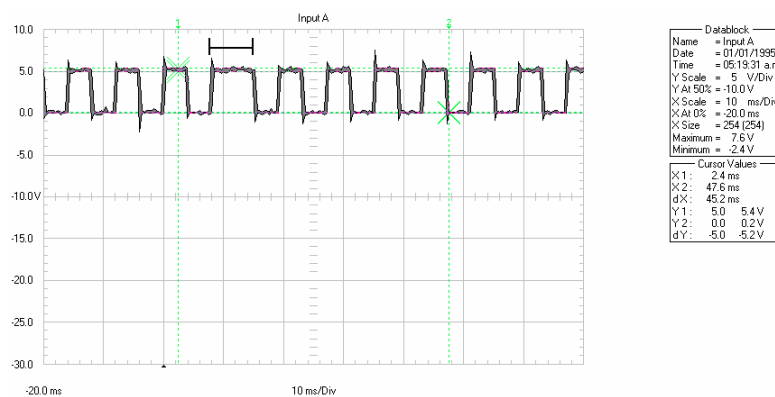


Figura 6.2 Gráfica del intervalo de pasos para el motor.

La figura 6.2 corresponde a la salida para los pasos de un motor, corresponde a cualquier patilla D0, D1, D2 o D3 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.2 se presenta el intervalo de pasos del motor. El periodo de la señal es de 8ms lo cual hace que cada paso del motor se hace en 4ms. Cabe destacar que hay un momento donde este periodo se hace más grande aproximadamente 11ms, esto sucede cuando el PIC principal da servicio a la interrupción externa. El PIC está dando pasos al motor y en un momento dado ocurre una interrupción externa entonces deja la patilla de los pasos en el estado actual y da servicio a la interrupción, lo cual demanda cierto tiempo, al volver de la interrupción se da cuenta que hay que bajarla y es por esto que se da un atraso del estado antes de la interrupción. Esto se da cada 1 s ya que el Geiger que envía datos CPS cada segundo es el que da la señal de interrupción. Esta interrupción provoca un leve retraso en la velocidad de pasos del motor pero no es considerable ya que la mayoría del tiempo el motor se va a estar moviendo a la misma velocidad.

Se midieron los tiempos de recorrido del sistema y el tiempo para que el sistema complete el recorrido es de aproximadamente 11 minutos y 15 segundos. La velocidad de los pasos del motor debe estar de acuerdo a las hojas de datos del motor, ya que a mayor velocidad de los pasos menor es el torque en el motor, y como se ha mencionado en el transcurso del documento este torque debe ser suficientemente alto para mover la estructura. Es por esto que se determinó que con un período mínimo de 8ms el motor tiene el suficiente torque para mover la estructura, esto de acuerdo a la gráfica del motor velocidad versus torque presentada en los anexos.

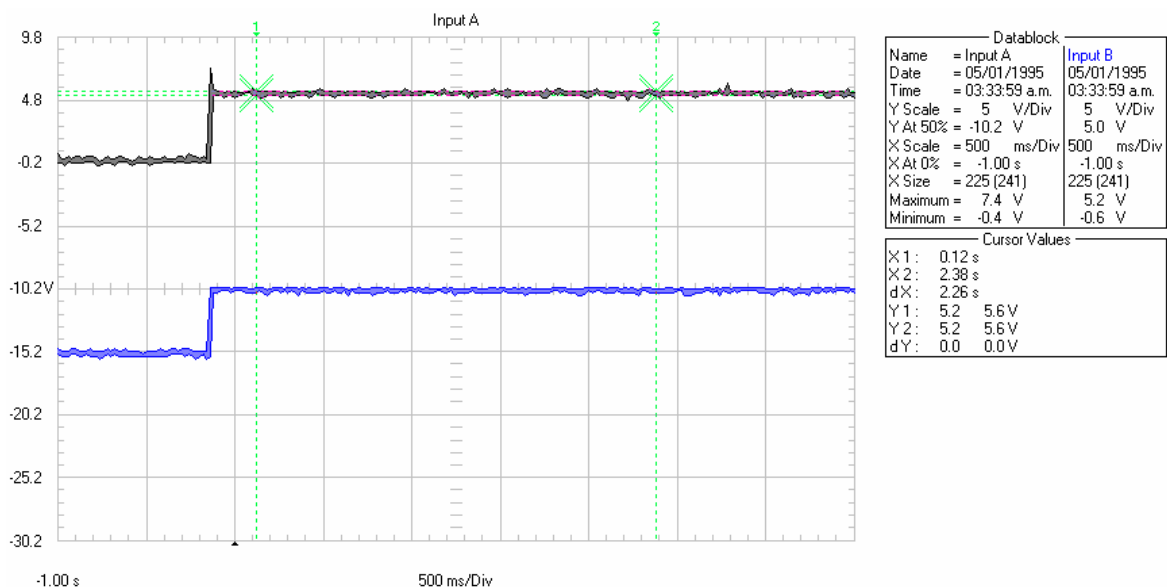


Figura 6.3 Apertura de la fuente de radiación (abajo) debido a final de carrera (arriba), inicio de recorrido.

La figura 6.3 corresponde a la apertura de la fuente de radiación debido al sensor final de carrera al inicio del recorrido. Corresponden a las patillas E2 y E0 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.3 se presenta la apertura de la fuente de radiación. Como se ha explicado anteriormente al inicio se debe posicionar la fuente de radiación en la posición cero, con lo cual se le dan pasos al motor hasta que el sensor final de carrera detecte esta posición, al llegar a ella inmediatamente la fuente de radiación

se abre para el inicio de las mediciones. La respuesta del PIC principal ante este echo, como se puede observar en la gráfica.

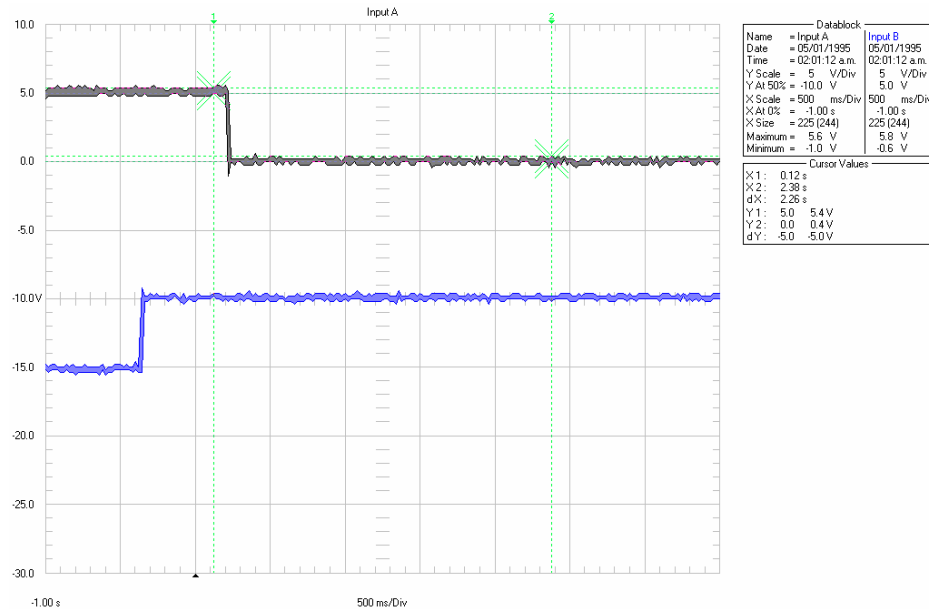


Figura 6.4 Cierre de la fuente de radiación (arriba) debido a final de carrera (abajo), fin de recorrido.

La figura 6.4 corresponde al cierre de la fuente de radiación debido al sensor final de carrera al final del recorrido. Corresponden a las patillas E2 y E0 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.4 se muestra lo contrario a la figura 6.3, ya que es el cierre de la fuente de radiación debido al sensor final de carrera. Cuando el recorrido se completa el sensor final de carrera lo detecta y pasa del estado bajo a alto y la fuente de radiación que ha pasado abierta durante todo el recorrido se cierra para seguridad del operario para colocar la estructura en otra posición.

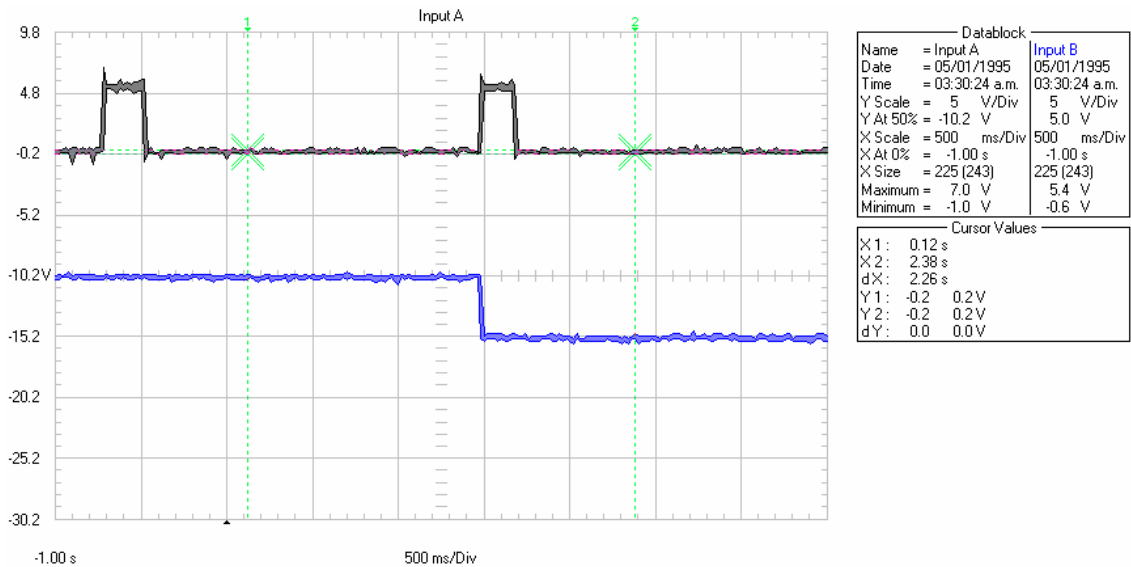


Figura 6.5 Cierre de fuente de radiación debido a dos pulsos de detener (paro de emergencia).

La figura 6.5 corresponde al cierre de la fuente de radiación por medio de pulsos de detener del control externo. Son las patillas B6 y E2 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.5 se presenta el cierre de la fuente de radiación debido a un paro de emergencia. Con el botón Detener del control externo se hace este paro de emergencia. Al pulsarlo una vez actúa sobre el sistema parando el motor de pasos, pero al volver a pulsar Iniciar el motor de nuevo se vuelve a mover hacia delante a partir de donde estaba; pero si se da detener de nuevo el sistema actúa sobre un paro de emergencia, con lo cual lo primero que debe hacerse es cerrar la fuente de radiación por seguridad. Se puede observar que el sistema actúa inmediatamente sobre el pin para cerrar la fuente.

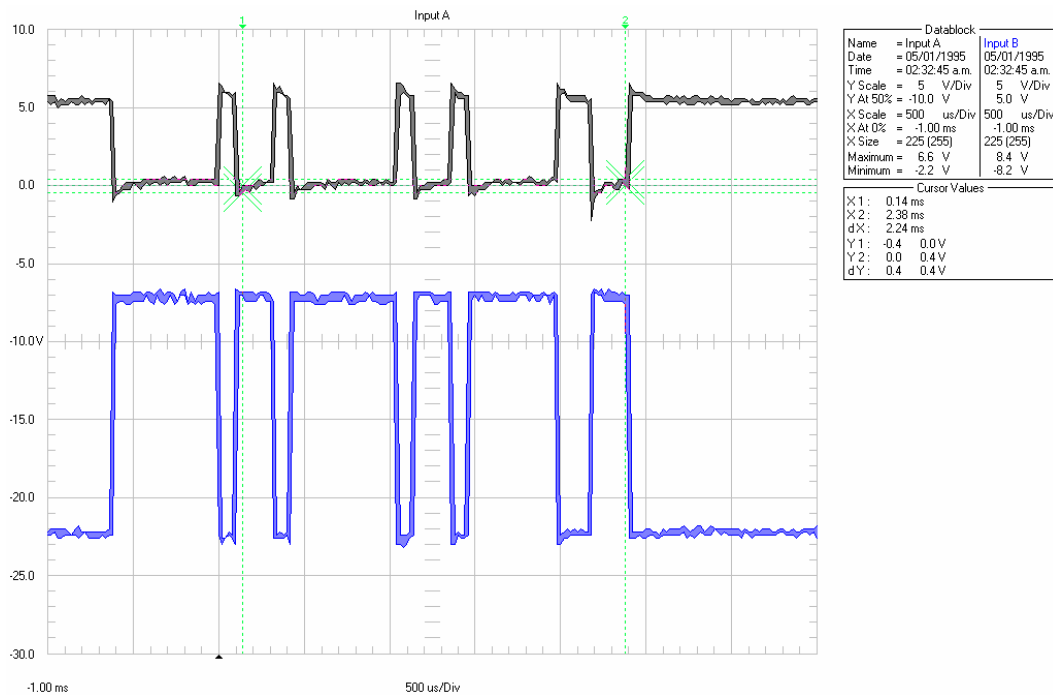


Figura 6.6 Dato enviado del PIC a la PC antes y después de pasarlo por el Max232

La figura 6.6 corresponde a un dato enviado del PIC secundario a la PC antes y después del Max232. Corresponden a las patillas 11 y 14 del Max232 de interfase con la PC. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.6 se tiene un dato enviado del PIC secundario a la PC, como se ha explicado anteriormente el protocolo RS-232 indica el valor -12V corresponde a un 1 lógico y el valor +12V corresponde a un 0 lógico. En esta figura se puede apreciar esta característica del protocolo ya que hay una inversión de estado para la señal. Se puede observar que a pesar de que la señal anda por el orden de 200us el Max232 es lo suficientemente rápido para enviar la correcta señal la PC.

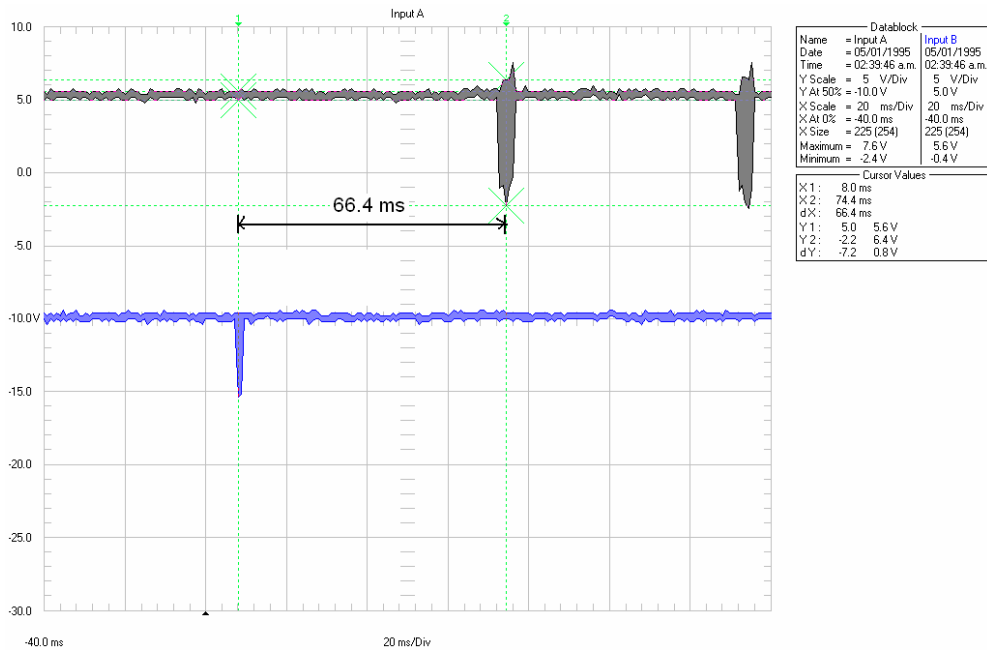


Figura 6.7 Pedido de dato de la PC y tiempo de respuesta del PIC.

La figura 6.7 corresponde a la solicitud de un dato de la PC al PIC principal y la duración del tiempo de respuesta del PIC secundario para enviar el byte. Corresponden a las patillas 11 y 12 del Max232 de interfase con la PC. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.7 se tiene la petición de pedir un dato CPS, 2 bytes, al PIC secundario. Esto primero llega al PIC principal y este se encarga por medio del puerto b decirle que envíe estos dos datos. El tiempo transcurrido desde la solicitud del dato a el PIC secundario enviarlo es de aproximadamente 66.4ms. En este caso corresponde al carácter o que actúa sobre la interrupción RDA del PIC principal y este le envía una señal que por medio de la interrupción RB el PIC secundario envía el dato. En la gráfica se puede observar el envío de los dos bytes, primeramente se envía el byte menos significativo y 50ms después el byte más significativo, esto para que le dé tiempo al programa en Java de almacenar el primer byte en una lista y volver al método para capturar el segundo byte.

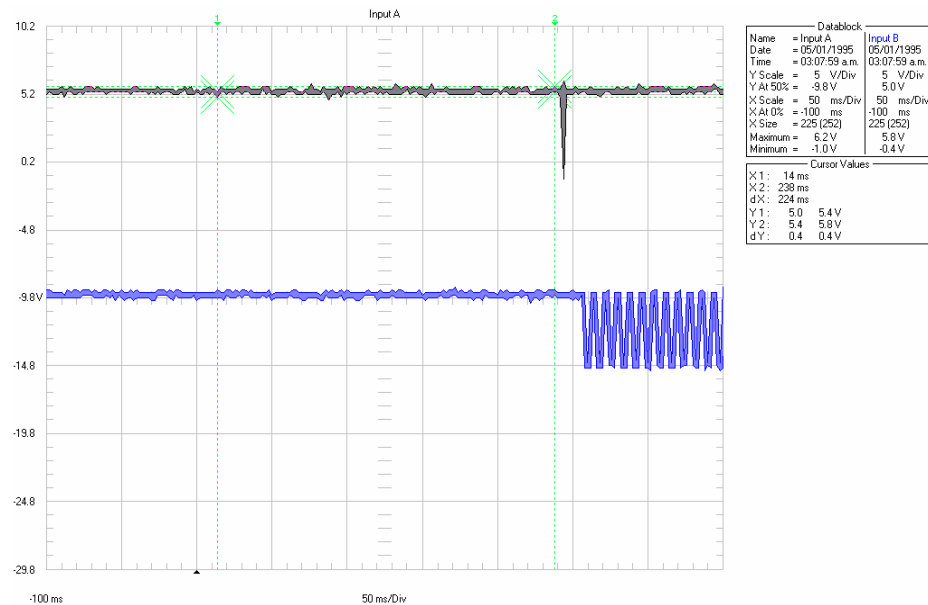


Figura 6.8 Inicio de pasos del motor desde el control de la PC.

La figura 6.8 corresponde al inicio de pasos del motor desde el control en Java. Corresponde a la patilla 11 del Max232 de interfase con la PC y D0 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.8 se observa el inicio de los pasos a partir de una señal enviada de la PC. Como se aprecia en un inicio la salida de la patilla está en alto, lo que indica que el motor se encuentra enclavado en una posición. Por medio de la patilla RV del PIC principal se recibe este carácter el cual corresponde al carácter *m* lo cual hace poner a 1 la bandera *habilitar* y esta da inicio a los pasos del motor hacia delante. La escala de tiempos del osciloscopio se puso un nivel de 50ms/Div para apreciar la respuesta del PIC principal.

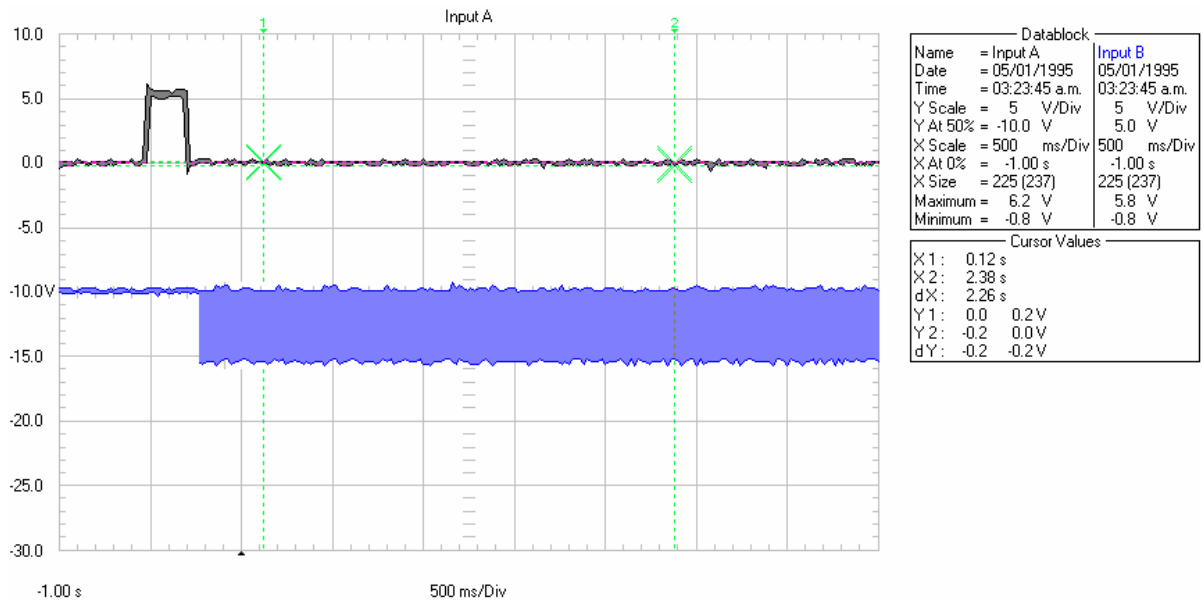


Figura 6.9 Inicio de pasos del motor desde control externo.

La figura 6.9 corresponde al inicio de pasos del motor desde el control externo. Corresponden a las patillas B7 y D0 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

De la misma forma en la figura 6.9 se observa el inicio de los pasos del motor pero debido al botón iniciar del control externo. La escala de tiempos para hacer esta medición es de 500ms/Div y es por esto que no se aprecian bien los pasos, que se dan cada 4ms. Pero lo importante acá es ver que el PIC actúa bien sobre el motor una vez pulsado este botón.

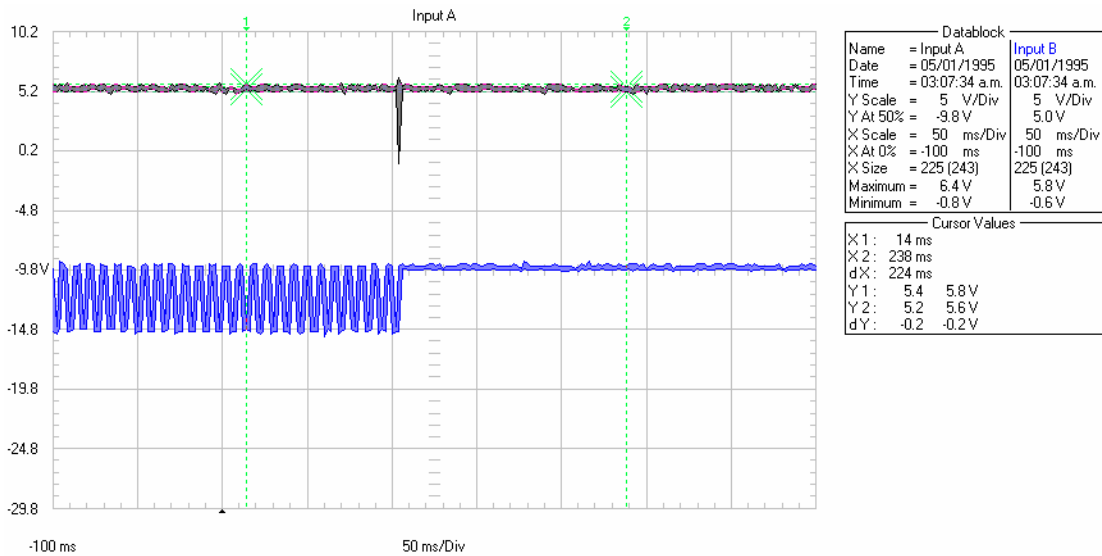


Figura 6.10 Detención de pasos del motor desde el control de la PC.

La figura 6.10 corresponde a la detención de pasos del motor desde el control en Java. Corresponde a la patilla 11 del Max232 de interfase con la PC y D0 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.10 se observa la detención de pasos del PIC hacia el motor debido al control desde Java. Se puede observar que a partir que llega el carácter a la patilla RV del PIC principal, éste deja de oscilar la pastilla y la pone en un estado, lo cual hace enclavar el motor en una posición. Este dato enviado de la PC al PIC principal corresponde al carácter *k* y lo cual hace poner a 0 la bandera *habilitar* y esta no deja dar más pasos.

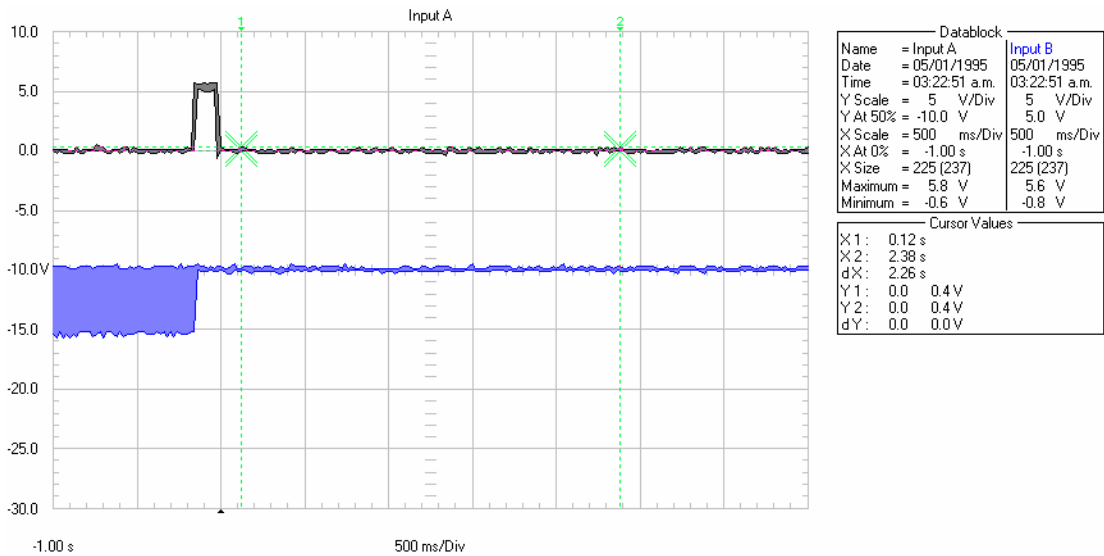


Figura 6.11 Detención de pasos del motor desde el control externo.

La figura 6.11 corresponde a la detención de pasos del motor desde el control externo. Corresponden a las patillas B7 y D0 del PIC principal. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

De igual forma en la figura 6.11 se da la detención de los pasos del motor por medio del botón Detener del control externo. Como se ha explicado anteriormente con el primer pulso a este botón se pone la bandera *habilitar* en 0, y enclava al motor.

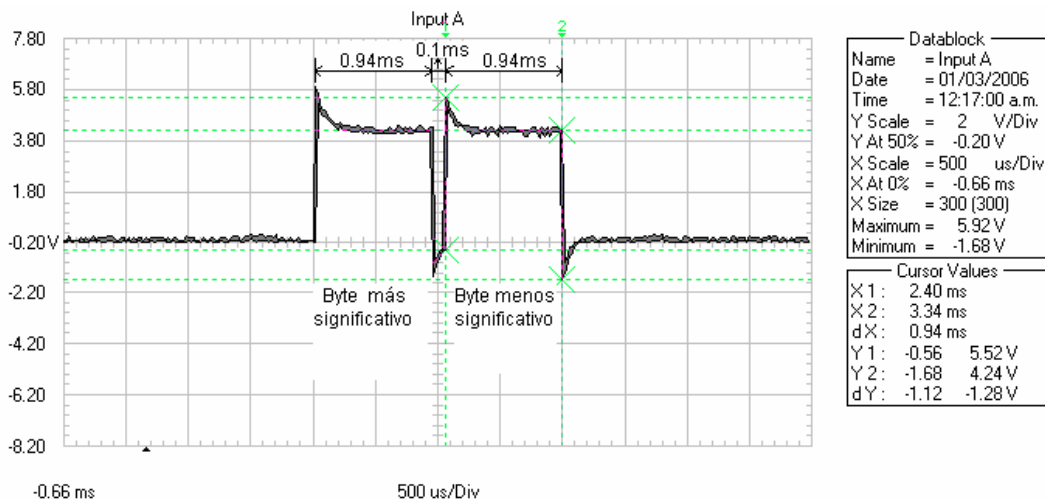


Figura 6.12 Salida del Geiger de un CPS, tiempos de duración y orden de bytes.

La figura 6.12 corresponde a la salida del contador Geiger de un dato CPS. Corresponde al pin 3 del conector DB9 de la salida serial del Geiger. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.12 se tiene la salida del contador Geiger de un dato CPS, se puede observar que el tiempo que tarda en enviar el byte menos significativo tarda en enviar el más significativo, 0.94ms, además hay un lapso de tiempo entre los dos bytes de aproximadamente 0.1ms. Con esta gráfica se determinó cómo iba a operar la interrupción RDA del PIC secundario, ya que primeramente se conocía por medio de las hojas de datos el ciclo de escritura, aproximadamente 5ms, de la eeprom serial 24FC512. Por esto se determinó que primero se debe de captar los dos bytes y luego almacenarlos en la eeprom, ya que si por cada interrupción se almacena no le da tiempo al PIC de recibir el otro byte y se pierde. Debido a la velocidad de operación del PIC, cristal de 20MHz = 50ns, es lo suficientemente rápida para por medio de la interrupción RDA captar los dos bytes.

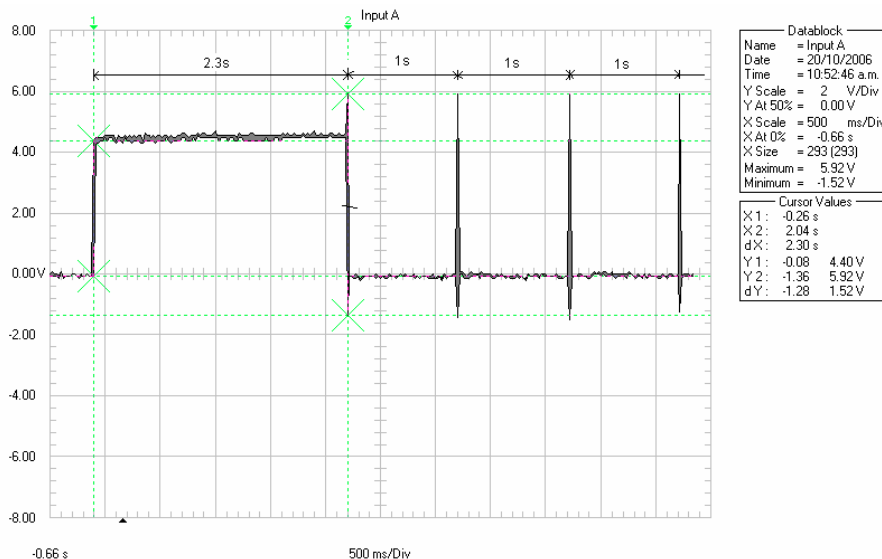


Figura 6.13 Encendido del Geiger y datos de CPS.

La figura 6.13 corresponde a la salida del contador Geiger del encendido y envío de datos CPS al PIC secundario. Corresponde al pin 3 del conector DB9 de la salida serial del Geiger. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.13 se muestra la salida del Geiger para el envío de datos CPS por el puerto serie. Al inicio cuando el Geiger es encendido tarda 2.3s para la inicialización interna del circuito y envío del primer CPS, y partir de acá envía cada segundo un dato CPS como puede ser observado, esta misma señal es enviada al PIC principal para la cuenta de segundos. Es importante mencionar que el tiempo de duración del recorrido como se había mencionado el recorrido dura aproximadamente 11 min. y 15 segundos, lo cual se almacenan $11 \cdot 60 + 15 = 675$ datos CPS. Este recorrido se hizo de esta forma por dos razones, primero por lo curva del motor de velocidad versus torque, ya que el motor debe tener el torque suficiente para mover la estructura y segundo que los datos de radiación sean los verdaderos recibidos en el Geiger, como se conoce todo equipo de medición tiene un tiempo de estabilidad de los datos y si la estructura se mueve muy rápidamente puede ser que el Geiger que capte la radiación adecuadamente, entonces lo que se hace es darle pasos al motor para que tanto el recorrido no sea muy lento y no muy despacio manteniendo una velocidad que dé la fuerza para mover la estructura. Se encontró que dando pasos cada 4ms se dan las dos condiciones es de buena manera.

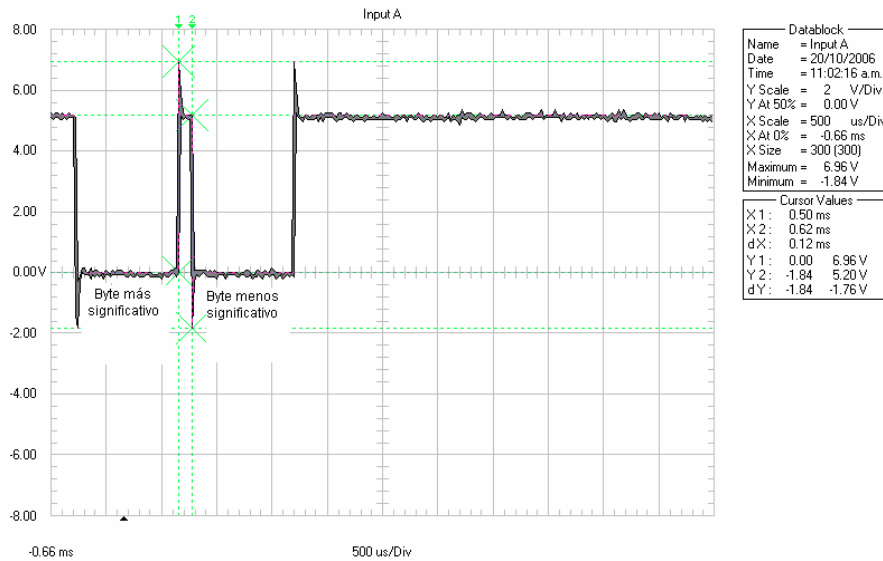


Figura 6.14 Salida del Max232 de un dato de CPS enviado al PIC.

La figura 6.14 corresponde a un dato CPS enviado al PIC secundario pasados por el Max232. Corresponde al pin 12 del Max232 interfase con el contador Geiger. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

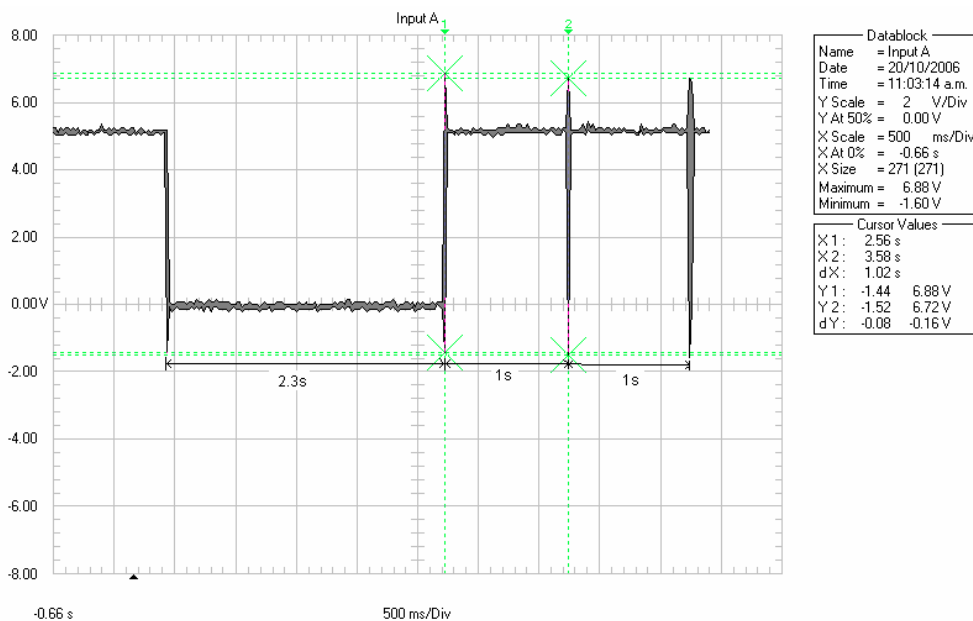


Figura 6.15 Salida del Max232 del encendido del Geiger y datos de CPS.

La figura 6.15 corresponde al encendido del contador y envío de datos CPS al PIC secundario pasados por el. Corresponde al pin 12 del Max232 interfase con el contador Geiger. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.14 se tiene la salida del Max232 de un dato CPS enviado al PIC secundario, este puede serial tiene la característica antes mencionada de que las señales están invertidas, si la señal no se pasa por el Max232 el PIC va a reconocer como un 1 siendo un 0 y un 0 siendo un 1, así que por medio del Max232 se cambia para que los valores sean recibidos correctamente por el PIC, en la figura se observa de igual forma los tiempos entre bytes y la duración de cada byte. De igual forma en la figura 6.15 se tiene la señal hacia el PIC pasada por el Max232.

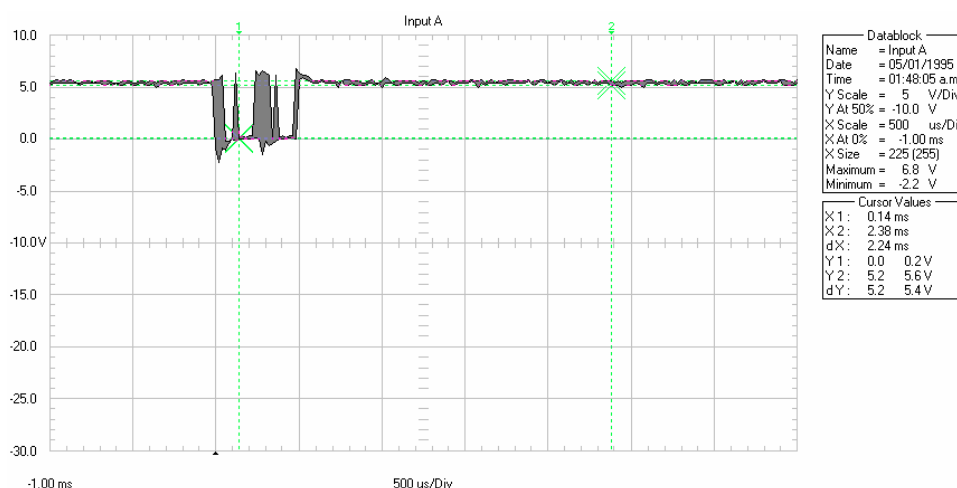


Figura 6.16 Salida del pin SDA del PIC secundario para la EEPROM.

La figura 6.16 corresponde al envío de un byte del PIC secundario a la memoria eeprom externa vía serie. Corresponde al pin C4 del PIC secundario. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

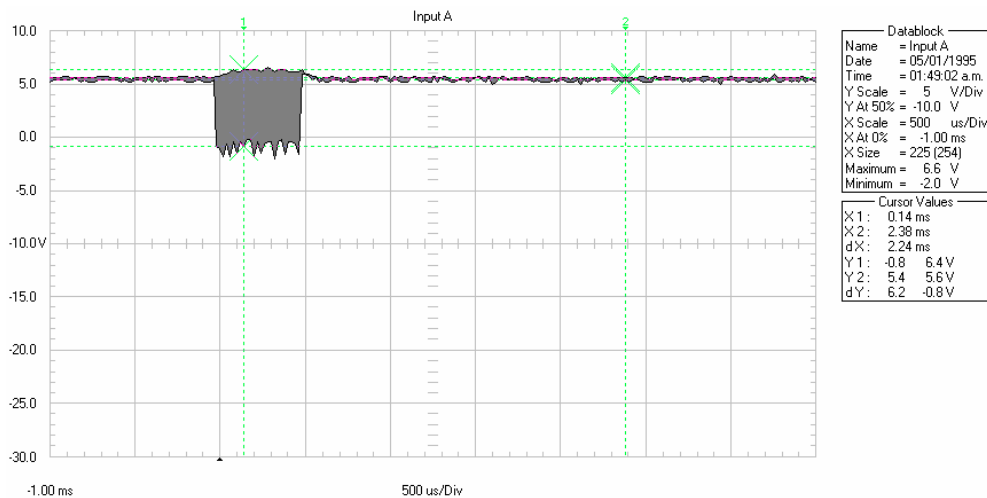


Figura 6.17 Salida del pin SCL del PIC para la EEPROM

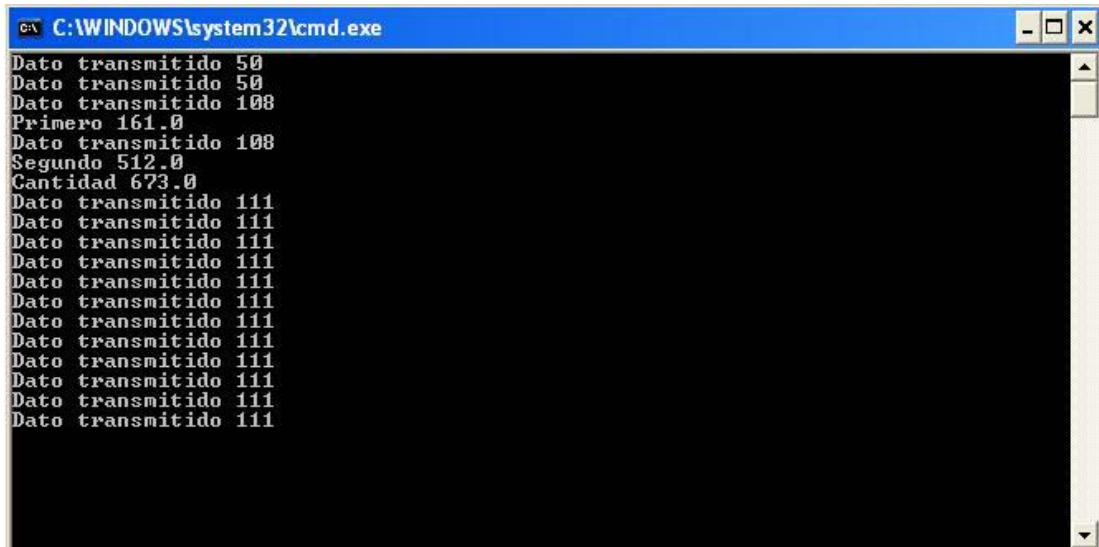
La figura 6.17 corresponde al envío del reloj PIC secundario a la memoria eeprom externa vía serie. Corresponde al pin C3 del PIC secundario. Esta medición se realizó por medio de un osciloscopio con salida a la PC.

En la figura 6.16 se tiene un byte enviado del PIC a la memoria eeprom externa. Esta debe cumplir como especifica el protocolo I2C. Por medio de esta patilla se envía primero el byte alto de la dirección, después el byte bajo de la dirección y finalmente el byte para ser almacenado. De la misma forma en la figura 6.17 se presenta la pin SCL de de sincronización de los datos. El PIC secundario trabaja como maestro en la comunicación y es por ello que para que la eeprom serial detecte bien los bytes se envía la señal de reloj para que esta trabaje adecuadamente.



Figura 6.18 Solicitud de gráfica y verificación de datos.

En la figura 6.18 se muestra el programa en Java cuando se solicita la petición de datos de una gráfica. En este caso se envía el carácter 2 para la petición de la gráfica 2. En la pantalla en negro se puede observar que se envía dos veces el número decimal 50 correspondiente al carácter 2 para que el PIC principal envíe los dos bytes de cantidad de datos de esta gráfica. En este caso el programa en Java recibe un número mayor a cero, lo que indica que la gráfica tiene datos y es a partir de esto que solicita todos los datos, en el caso de que sea cero se le indica que no hay datos y pone a este botón en gris para no volverlo a oprimir.



```
C:\WINDOWS\system32\cmd.exe
Dato transmitido 50
Dato transmitido 50
Dato transmitido 108
Primero 161.0
Dato transmitido 108
Segundo 512.0
Cantidad 673.0
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
Dato transmitido 111
```

Figura 6.19 Datos enviados y recibidos en Java por el puerto serial

En la figura 6.19 se tiene la pantalla de control en donde se muestran los datos enviados y recibidos por el puerto serial en Java. Primeramente se envía dos veces el carácter 2 para conocer si la gráfica tiene datos. Seguidamente se envía el carácter / y se recibe el byte menos significativo de cantidad=161, luego se envía de nuevo el carácter / y se recibe el byte más significativo de cantidad=512, se suman y en total la gráfica tiene 673 datos CPS, inmediatamente el programa envía el carácter o y se van recibiendo estos datos y van siendo almacenados en una lista para ser graficados. Se envía un carácter o por cada dato CPS.

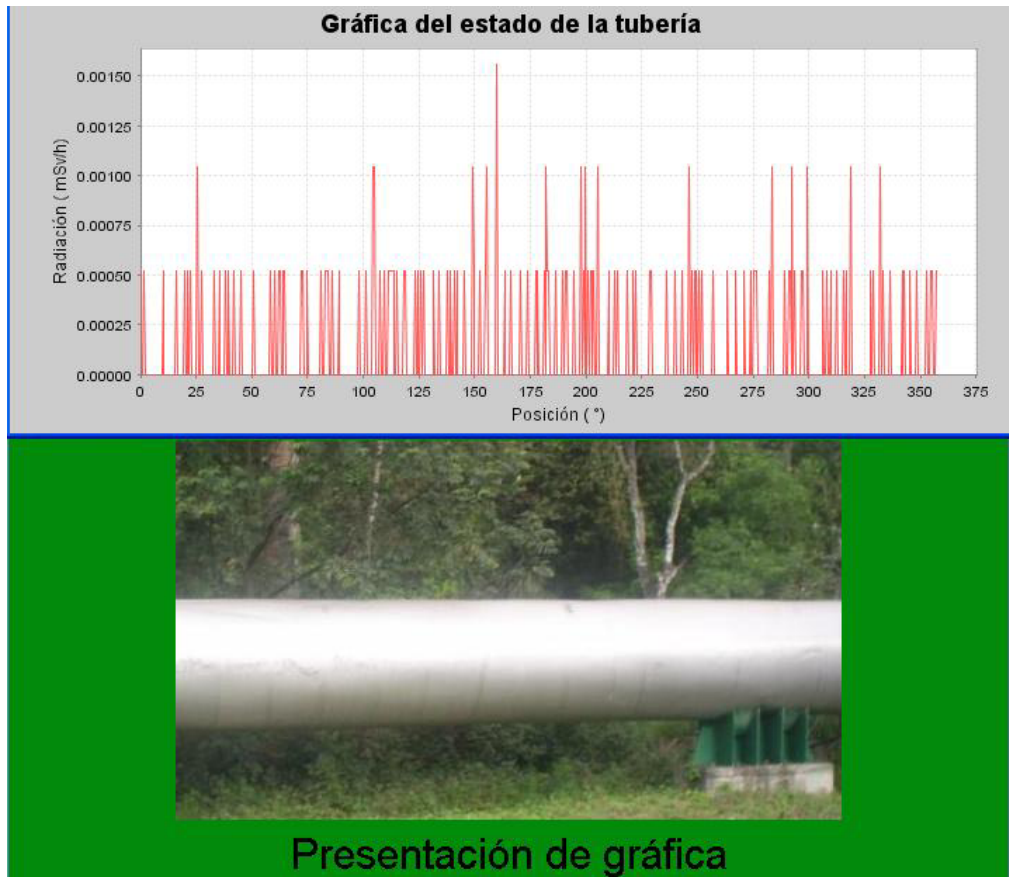


Figura 6.20 Gráfica de Prueba de Posición versus Radiación

En la figura 6.20 se tiene una gráfica de prueba de radiación versus posición, se tomó el Geiger sin radiación con lo cual los niveles son casi cero, máximo de un CPS anda por 3, de esta forma se comprobó el buen funcionamiento del programa en Java.

Es importante mencionar que el valor de 1 CPS es equivalente de en radiación a 0.00052 mSV/h de esta forma. Estos datos son lineales y de esta forma la cantidad sumada en un CPS se multiplica por este valor para tener en las unidades del SI de la dosis de radiación.



Figura 6.21 Solicitud de borrar datos de gráfica recién leída.

En la figura 6.21 se tiene la pregunta del programa al usuario en donde si quiere borrar los datos de la gráfica en la memoria del PIC recién leída. De esta forma si el operario quiere de nuevo bajar los datos los puede hacer. Cabe destacar que con sólo leer una gráfica automáticamente en el PIC se manda un dato que le indica que si se van a hacer otras mediciones empiece de nuevo en la primera gráfica. Es en el caso de que después de hacer el estudio de la gráfica y detecten alguna anomalía, el operario puede de nuevo bajar los datos para corroborar esta medición.

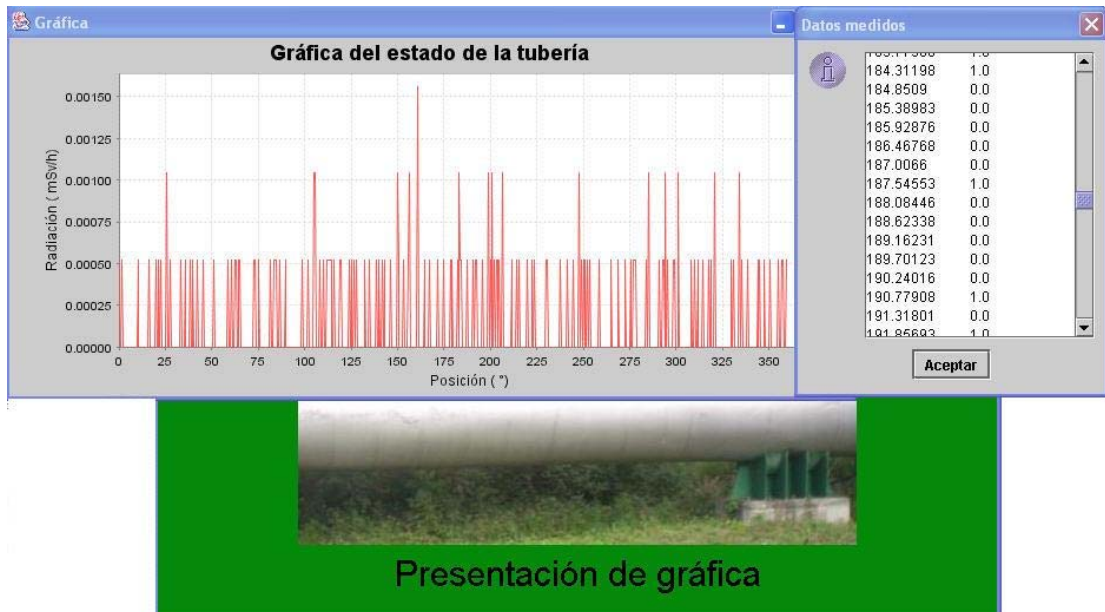


Figura 6.22 Gráfica de Prueba de Posición versus Radiación

En la figura 6.22 se presenta una gráfica de prueba en donde además se presenta la lista de los datos. Del lado derecho de esta lista se encuentran los CPS, de esta forma el que analiza la gráfica y detecta una anomalía puede recorrer esta lista y ver exactamente en qué posición se encuentra la anomalía y además qué nivel de radiación se presenta en ese punto. Como se puede observar estas gráficas de pruebas tienen valores de CPS muy bajos casi, máximo de $3 = 3 \cdot 0.00052 = 0.00156$ mSV/h.

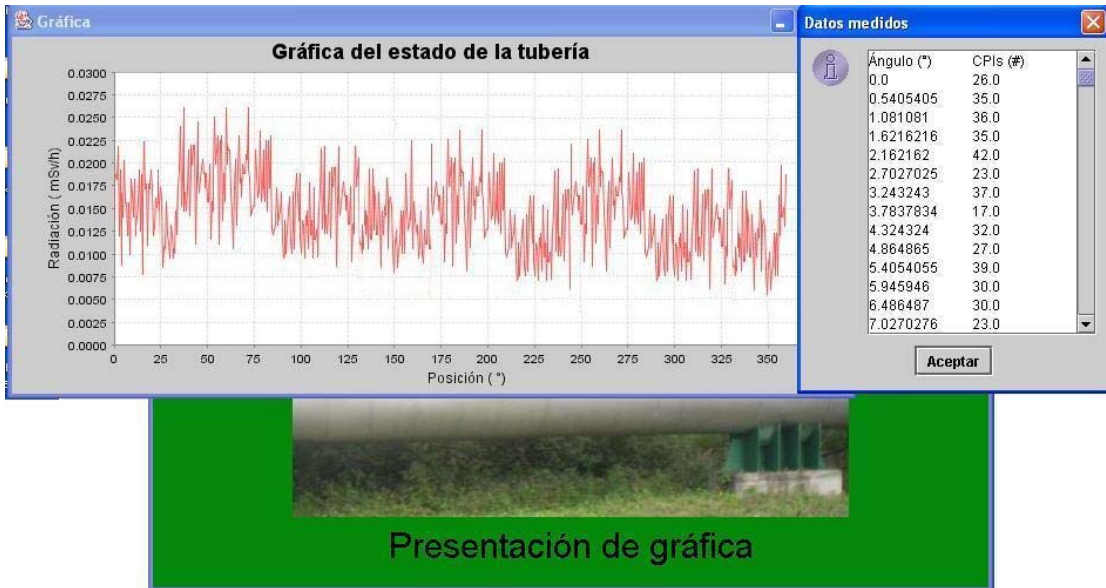


Figura 6.23 Gráfica real de Posición versus Radiación.

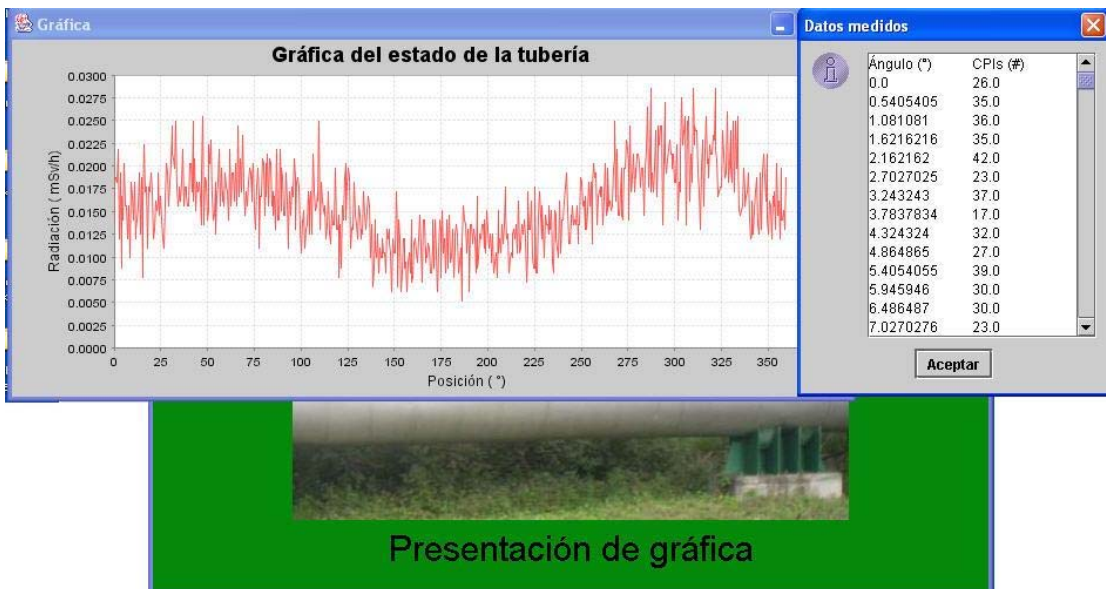


Figura 6.24 Gráfica real de Posición versus Radiación.



Figura 6.25 Gráfica real de Posición versus Radiación.

En la figura 6.23, 6.24 y 6.25 se tienen gráficas reales en la tubería ya con la fuente de radiación. En estas se pueden observar diferentes gráficas para diferentes posiciones de la tubería de prueba, se presentan los niveles de radiación y el ángulo para cada dato. En el eje y se presenta la radiación en mSV/h y en el eje x se presenta la posición para cada dato. Primeramente como se había mencionado en Java al inicio se pregunta la cantidad de datos de la gráfica, y entra al ciclo en donde se solicita dato a dato hasta llegar a la cantidad, entonces cada dato es guardado en una lista en x y y, y son graficados. El tiempo que le toma al programa en Java para tener todos los datos y graficarlos es de aproximadamente 4 minutos, esto porque como se dijo son aproximadamente 675 datos y en la memoria los ciclos de lecturas acarrearán delays. [1]

Es importante mencionar que estas primeras mediciones se están llevando a cabo con una tubería de prueba, la cual presenta todos los estados del acero de un vapor ducto: buen estado, con diferentes niveles de corrosión y con grietas. Estas se realizaron en un cuarto y no con las condiciones ambientales que se presentan en el campo. Sin embargo, se pudieron detectar variaciones en los niveles de radiación para los diferentes estados del acero antes mencionados. Se midieron picos de

radiación cercanos a los 250uSv/h para cuando se tenían grietas, así como niveles entre 30uSv/h y 50uSv/h para cuando el material se encontraba en buen estado y niveles que rondaban los 150uSv/h para cuando el nivel de corrosión no era tan avanzado. Por esto es de esperar que al realizar un análisis detallado de la gráfica se pueda determinar el estado del acero.

Capítulo 7: Conclusiones y Recomendaciones

7.1 Conclusiones

- Se obtuvo una gráfica de radiación (mSV/h) versus posición (°). Cada dato de radiación se monitorea y se almacena en una memoria cada 0.533° de la estructura, lo cual genera datos muy precisos del estado del acero de la tubería.
- Para este equipo de radiación se encontró que el operario con el control debe estar a una distancia de 3 metros para recibir niveles de radiación, mientras la fuente está abierta, que no perjudica su salud.
- Si la velocidad de la estructura es muy rápida el contador Geiger enviará datos de radiación erróneos, ya que le toma cierto tiempo en captar la señal y enviarla por el puerto serial, así estará enviando un dato de radiación de una posición atrás desconocida.
- Para un periodo de paso del motor de 4ms el torque es suficiente para mover la estructura y además la velocidad de rotación es la adecuada para que el contador Geiger reciba y envíe datos estables y reales de radiación.
- Por medio de un control preciso, en el primer dato CPS recibido, se almacena siempre el byte menos significativo en la primera posición y el byte más significativo en la segunda. Esto porque el inicio se puede dar entre los dos bytes de un dato con lo cual erróneamente se almacenaría de primero el byte más significativo.
- Debido a la exactitud de los pulsos y la casi constante velocidad de la estructura el tiempo es directamente proporcional a la variable posición de la fuente.

- Debido a la atención de la interrupción externa del PIC principal cada segundo se da un paso al motor de 8ms, lo que genera un atraso en la velocidad del motor.
- Debido al atraso generado por parte del PIC principal de comunicarle al PIC secundario y al atraso en el acceso la memoria externa, el tiempo entre la petición del programa en Java de enviar un dato hasta que el PIC secundario lo envía es de 64,4ms lo cual genera tiempos de 4 minutos en hacer llegar todos los datos a la gráfica.
- Debido a que el contador Geiger envía un dato en 2 bytes y entre ellos hay un tiempo de 0,1ms no se puede almacenar en la memoria externa byte por byte, ya que el periodo de escritura en la memoria de 5ms, con lo que esta escritura se realiza cada dos interrupciones RDA del PIC secundario.
- El sistema opera satisfactoriamente controlado desde la PC así como controlado externamente por los pulsadores.
- El objetivo de implementar del diseño e implementación de un sistema controlado desde una distancia segura para obtener una gráfica de radiación versus posición se ha logrado. Sin embargo, la meta de detectar a tiempo problemas de corrosión aún no se ha logrado ya a pesar de tener primeras mediciones satisfactoriamente faltan realizar las suficientes pruebas para conocer bien en una gráfica la existencia de corrosión en el acero.

7.2 Recomendación

- Se debe buscar la velocidad de rotación óptima de la estructura para que el contador Geiger detecte y envíe niveles de radiaciones estables y reales. Esto para tratar de hacer un recorrido más rápido pero a la vez viendo que el Geiger responda bien ante la velocidad, además de ello la velocidad es inversamente proporcional con el torque y la velocidad debe ser tal que el torque del motor sea el suficiente para mover la estructura.

Bibliografía

- [1] Curso avanzado de protección radiológica para radiografía industrial y medidores nucleares, febrero de 2005.
- [2] García Cueto, Alfonso R. "Radiografía". México, 1990.
- [3] Scopelliti, José. "Apuntes sobre el curso de radiografía industrial nivel II". Comisión de energía atómica argentina. Departamento de Metalurgia, ITCR, 24 de octubre al 4 de noviembre 1994.
- [4] Sitio Web [en línea] <[http:// www.automotriz.net/tecnica/torque.html](http://www.automotriz.net/tecnica/torque.html)> [Consulta: 5 febrero 2006].
- [5] Sitio Web [En línea]. <<http://www.dccia.ua.es/dccia/inf/asignaturas/ROB/optativos/Sensores/internos.html#inicio>> [Consulta: 9 febrero 2006].
- [6] Sitio Web [En línea]. <<http://www.itapizaco.edu.mx/paginas/JavaTut/parte19/cap19-2.html>> [Consulta: 10 febrero 2006].
- [7] Sitio Web [En línea]. <[http:// www.tuotromedico.com/temas/0#0](http://www.tuotromedico.com/temas/0#0)> [Consulta: 8 febrero 2006].
- [8] Sitio Web *del Instituto Costarricense de Electricidad*. San José, Costa Rica. <<http://www.grupoice.com/esp/cencon/gral/proyeccion/museo/docs/miravalles.pdf>> [Consulta: 13 enero 2006].
- [9] Sitio Web de *Microtroniks* [En línea] <<http://www.microtroniks.com>>. [Consulta: 14 enero 2006].
- [10] Sitio Web de Real Academia Española. [En línea] <[http:// www.rae.com](http://www.rae.com)> [Consulta: 15 junio 2006].
- [11] Sitio Web [En línea]. <http://robots-argentina.com.ar/MotorCC_PuenteH.htm> [Consulta: 22 setiembre 2006].
- [12] Sitio Web [En línea]. <<http://www.todorobot.com.ar/información/tutorial%20stepper/stepper-tutorial.htm>> [Consulta: 22 setiembre 2006].
- [13] Sitio Web [En línea]. <<http://es.wikipedia.org/wiki/RS-232>> [Consulta: 15 mayo 2006].

- [14] Sitio Web [En línea]. <<http://es.wikipedia.org/wiki/I%C2%B2C>> [Consulta: 23 agosto 2006].
- [15] Sitio Web [En línea]. <http://es.wikipedia.org/wiki/Contador_Geiger> [Consulta: 13 abril 2006].
- [16] Sitio Web [En línea]. <<http://www.itapizaco.edu.mx/paginas/JavaTut/froufe/parte19/cap19-2.html>> [Consulta: 9 marzo 2006].
- [17] Sitio Web [En línea]. <http://www.jovenclub.cu/libros/Libros_2/ciencia3/094/htm/sec_9.htm> [Consulta: 29 enero 2006].

Apéndices

A.1 Glosario, abreviaturas y simbología

Corrosión: Destrucción paulatina de los cuerpos metálicos por acción de agentes externos, persista o no su forma.

Cremallera: Es un dispositivo dentado perteneciente a los diversos tipos de engranajes.

Diodo de rodamiento libre: Diodo que se pone en paralelo a una bobina inversamente polarizado para proteger el elemento de conmutación de la sobretensión que causa la bobina cuando es conmutada.

Dosímetro: Aparato que mide dosis, especialmente de radiactividad.

Estator: Parte fija de una máquina rotativa.

Medidor Geiger: Instrumento de medición preciso para radiación alfa, beta y gamma.

Memoria Flash: Forma evolucionada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación mediante impulsos eléctricos.

Motor paso a paso: El motor eléctrico paso a paso es un actuador conversor de tren de impulsos en movimiento angular giratorio

Microcontrolador: Circuito integrado o chip que incluye en su interior las tres unidades funcionales de un ordenador: CPU, Memoria y Unidades de E/S.

Programa alto nivel: Son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos lo que facilita la escritura y la fácil comprensión por el programador.

Puente H: Configuración es una de las más utilizadas en el control de motores de CC. Nombre surge de la posición de los transistores.

Radiación: Flujo de partículas o fotones con suficiente energía para producir ionizaciones en las moléculas que atraviesa.

Rectificador de onda completa: Es el elemento o circuito que permite convertir la corriente alterna en corriente continua.

Resistencias de pull-up: Son resistencias que en general se conectan entre una señal lógica y el positivo y su función es asegurar que esa señal no quede en un estado flotante.

Rotor: La parte giratoria de una máquina.

Torque: Es la fuerza que producen los cuerpos en rotación

Vaporducto: Tuberías superficiales que transporta ya sea agua, vapor o una mezcla de ambos entre las diferentes etapas del proceso de generación de electricidad.

API: Application Programming Interface.

CPS: Cuentas Por Segundo.

EEPROM: Electrical Erasable Programmable Read Only Memory.

I2C: Inter Integrated Circuit.

PIC: Peripheral Interface Controller.

RAM: Random Access Memory.

RISC: Reduced Instruction Set Computer.

TTL: Transistor Transistor Logic.

USART: Universal Synchronous Asynchronous Receiver Transmitter.

A.2 Manuales de usuario

A.2.1 Manual de usuario del programa en Java

Al abrir el programa aparece una pantalla principal, en donde se tienen tres botones: “*Con control*”, “*Sólo graficar*” y “*Salir*”. El botón de control indica que se puede dar control al movimiento de la estructura desde la computadora, con lo cual se puede desde darle pasos al motor, detenerlo así como recibir los datos de radiación para ser graficados. El botón de graficar le indica que puede descargar datos de los recorridos realizados con el control externo.

Al dar clic en el botón *Con control* aparece una sub-pantalla con seis botones: “Fuente en posición inicial”, “Iniciar mediciones”, “Detener”, “Continuar”, “Graficar datos” y “Salir”. Primeramente los únicos botones que pueden ser seleccionados son *Salir* y *Fuente en posición inicial* esto para darle pasos al motor para que se posicione correctamente en el punto de inicio. Al dar clic en esta opción aparece en pantalla un mensaje que le indica que se esta posicionando la fuente, después de esto el botón *Fuente en posición inicial* se pone en gris y el botón *Iniciar mediciones* puede ser oprimido. Es responsabilidad del usuario antes de dar clic en iniciar verificar que el motor se ha detenido y la fuente se encuentra en la posición inicial.

Al dar clic en *Iniciar mediciones* el motor iniciará el recorrido hacia delante, empezando a almacenar los datos de radiación. En este momento este botón se pone en gris y el botón *Detener* puede ser oprimido. En alguna circunstancia si la estructura quiere ser detenida lo que se hace es dar clic en este botón y con esto el motor dejará de dar pasos. Si da clic en este botón se pone en gris y los botones *Continuar* y *Graficar datos* se pueden oprimir. Continuar le indica al motor seguir desde la posición en donde se encuentra y graficar es para hacer llegar al programa los datos de radiación para luego ser graficados. Al dar clic en Continuar este se pone en gris y el botón *Detener* puede ser de nuevo oprimido.

El motor por sí sólo detecta el final del recorrido y detiene sus pasos, a partir de acá se pueden también graficar los datos, para esto al igual que se explicó anteriormente se debe de oprimir *Detener* y luego *Graficar datos*. Una vez se haya dado clic en graficar aparece en pantalla una sub-pantalla que se encarga de pedirle al control que envía todos los datos almacenados para esta medición, esto tarda aproximadamente 4 minutos. Una vez todos los datos son recibidos en la pantalla aparece tanto la lista como la grafica para ser analizada. Des esta forma se concluye esta medición y la cerrar las grafica aparece de nuevo la sub-pantalla de control para

de nuevo hacer otra medición, desde que *Fuente en posición inicial* y *Salir* pueden ser oprimidos.

Es importante mencionar que el botón *Salir* siempre puede ser oprimido, en el caso que la estructura esté en movimiento lo que se genera es un paro de emergencia, con lo cual el motor deja de dar pasos y la fuente de radiación se cierra, además de volver al programa principal. En el caso de que se haya presentado la gráfica nada más vuelve al programa principal.

Desde el programa principal también se pueden bajar los datos para las diferentes gráficas obtenidas con el control externo. Al dar clic en esta aparece una sub-pantalla en donde aparecen seis botones: “Gráfico 1”, “Gráfico 2”, “Gráfico 3”, “Gráfico 4”, “Gráfico 5” y “Salir”. Cada botón indica a que grafico corresponde. Al inicio aparecen que todos pueden ser oprimidos. Al darle clic se cualquiera se envía una señal al control para ver esta gráfica tiene datos, si no es así se le informa con un mensaje y el botón se pone en gris y puede volver a ser oprimido; en cambio si tiene datos se le informa y aparece en pantalla una sub-pantalla que se encarga de pedirle al control que envía todos los datos almacenados para esta medición, esto tarda aproximadamente 4 minutos. Una vez todos los datos son recibidos en la pantalla aparece tanto la lista como la grafica para ser analizada. Al cerrarla vuelve a la misma sub-pantalla para poder pedir los datos de otra gráfica.

A.3 Protocolos de medición

A.3.1 Protocolo de medición para el control externo

Primeramente se debe de instalar el equipo y corroborar que todo esté bien conectado. A su vez el contador Geiger debe estar encendido antes de iniciar las mediciones. Después de esto se alimenta el circuito de control.

Este circuito cuenta con tres botones: “Fuente a cero”, “Iniciar-Continuar” y “Detener-Paro de emergencia”. Primeramente de debe de oprimir el botón de fuente

para que se posicione la fuente en el lugar correcta. Cabe resaltar que si se oprime cualquiera de los otros dos botones el control no va a hacer nada. Al pulsar el botón de fuente el motor va a girar en reversa para posicionarse, y es hasta cuando llega al lugar correcto que el control acepta únicamente el botón de inicio.

Al dar inicio el motor inicia el recorrido hacia delante, el botón de fuente e iniciar no son aceptados y el botón de detener puede ser oprimido en cualquier momento. Si se da detener el motor se detiene, el botón de iniciar puede volver a hacer oprimido para continuar con el recorrido desde donde se encontraba la fuente. En el caso de que se haya dado detener y el motor este detenido, y al volver pulsar detener el control entiende que es un paro de emergencia e inmediatamente cierra la fuente de radiación. Según la emergencia puede ser que se desconecte la alimentación al control o puede iniciar de nuevo otra medición, pero de nuevo posicionando la fuente a cero como en el inicio de toda operación.

A.4 Información sobre la institución

A.4.1 Descripción de la institución

Este proyecto tuvo su desarrollo en el Instituto Costarricense de Electricidad, Sector Energía (ICELEC); es una institución autónoma del Estado que tiene como fin el desarrollo racional de las fuentes productoras de energía en Costa Rica, especialmente los recursos hidrológicos y estimular el uso de la electricidad para el desarrollo industrial y mejora de la calidad de vida de sus habitantes. De hecho, una de las funciones del ICE es la construcción de proyectos de generación de energía, operación y mantenimiento de los mismos.

Debido al amplio campo de trabajo de la institución, ésta cuenta con un departamento llamado MET (Maquinas, Talleres y Equipos), el que se dedica a brindar servicios de apoyo en la parte de mantenimiento y diseños mecánicos de equipos, a los proyectos del ICE en el Sector Electricidad.

A.4.2 Descripción del departamento o sección en la que se realizó el proyecto

El MET partió de la idea de pequeños talleres realizados en la ubicación de cada proyecto en construcción, llamados SETEC (Servicios Técnicos). El objetivo del MET es realizar una centralización de los antiguos SETEC en un punto geográfico fijo, en este caso se realizó en Turrúcares de Alajuela, continuo a la Planta Hidroeléctrica Garita. Todavía se cuenta con pequeños SETEC, en condiciones muy reducidas a las originales, con las que se trabajo durante muchos años, puesto que el MET lleva aproximadamente tres años de haberse fundado, y todavía se encuentra en desarrollo. Por tanto los SETEC existentes pertenecen al MET.

Producción

La función principal del MET es darle un soporte de producción y mantenimiento a los proyectos realizados y los que están en proceso.

En el caso de este departamento se realiza proyectos de desarrollo de equipos los que tendrán como fin satisfacer un objetivo en específico. Por lo general la producción realizada en este deparadamente es muy variada, por tanto se dan proyectos pilotos de desarrollo ya sea mecánico, eléctrico, etc.

Mercado

El mercado de este departamento es totalmente interno de la empresa, dado que solamente se enfoca en la producción, investigación y mantenimiento a proyectos pertenecientes al Instituto Costarricense de Electricidad, esto debido a la política de la Institución.

A.5 Secuencia de acontecimientos:

Control externo

Al conectar el equipo:

- Se energiza el motor y queda enclavado en una posición.
- Se inicia a recibir la interrupción externa pero como sistema no está habitado no se realiza el conteo.
- Se inician a recibir CPS pero como sistema no está habilitado no se almacenan.
- El único pulsador aceptado por el control es el de posicionar la fuente a cero. Ya que es el primer paso que se debe de realizar. Los pulsadores Iniciar y Detener están deshabilitados por el momento.

Al dar fuente a cero:

- Inmediatamente se dan pasos al motor para un recorrido reversa. Este recorrido se detendrá hasta que el sensor final de carrera (sensor magnético) se ponga en alto para indicar que la fuente se encuentra en el punto correcto de inicio.
- Igualmente en este momento se están recibiendo interrupción externa y CPS sin ser tomadas en cuenta.
- Cuando el sistema se posicionó correctamente la fuente se abre y se habilita para que únicamente se botón que se puede pulsar es el de Inicio.
- El motor queda enclavado.

Al pulsar Inicio:

- Se habilita el sistema, aceptándose CPS (se almacenan) y conteo de segundos.
- Se dan pasos al motor pero para hacer el recorrido hacia delante.

- El único pulsador habilitado es el Detener.

Al pulsar Detener:

- Se deshabilita nuevamente el sistema y no se aceptan CPS y conteo de segundos.
- Motor queda enclavado.
- Se puede pulsar de nuevo Detener o Iniciar (Continuar).

Al pulsar Detener de nuevo:

- Se cierra se fuente de radiación (paro de emergencia).
- El único pulsador habilitado es el de posicionar la fuente a cero.
- Reset de toda la información.

Al finalizar el recorrido normalmente:

- Se cierra la fuente de radiación.
- Sensor final de carrera (sensor magnético) pasa a estado alto y se detecta el final.
- El motor queda enclavado.
- Sistema queda deshabilitado y no se aceptan más CPS y conteos.
- El único pulsador habilitado es el de posicionar la fuente a cero.

Control desde PC

Al abrir el programa y encender el equipo:

- Se energiza el motor y queda enclavado en una posición.
- Se inicia a recibir la interrupción externa pero como sistema no está habilitado no se realiza el conteo.

- Se inician a recibir CPS pero como sistema no está habilitado no se almacenan.
- El único botón que se puede oprimir es Fuente en posición inicial. Los demás botones están deshabilitados.

Al dar Fuente en posición inicial:

- Inmediatamente se dan pasos al motor para un recorrido reversa. Este recorrido se detendrá hasta que el sensor final de carrera (sensor magnético) se ponga en alto para indicar que la fuente se encuentra en el punto correcto de inicio.
- Igualmente en este momento se están recibiendo interrupción externa y CPS sin ser tomadas en cuenta.
- Se habilita únicamente el botón Iniciar mediciones, botón Fuente en posición inicial se deshabilita.
- El motor queda enclavado.

Al dar Iniciar mediciones:

- Se habilita el sistema, aceptándose CPS (se almacenan) y conteo de segundos.
- Se dan pasos al motor pero para hacer el recorrido hacia delante.
- El único botón habilitado es Detener.

Al dar Detener:

- Se deshabilita nuevamente el sistema y no se aceptan CPS y conteo de segundos.
- Motor queda enclavado.
- Se puede oprimir Fuente de posición inicial, Continuar, Graficar datos.

Al dar Continuar:

- Se habilita el sistema, aceptándose CPS (se almacenan) y conteo de segundos.
- Se dan pasos al motor pero para hacer el recorrido hacia delante.
- El único botón habilitado es Detener.

Al dar Graficar datos:

- Se recibe señal para enviar hacia la PC todos los datos de CPS almacenados en memoria para el recorrido recién finalizado.

A.6 Programa PIC principal

```
//A0...A3  A0...A3 salidas de codificación para PIC2
//B0      B0   entrada pulsos de 1.seg
//B5...B7  B5...B7 entradas pulsadores de control
//C7      C7   puerto serial, RV de la PC
//D0...D3  D0...D3 salidas del motor stepper
//E0      E0   entrada final de carrera
//E1      E1   led siempre encendido
//E2      E2   abre fuente radiación

#include <16F877A.h>           //Tipo de PIC
#include <delay.h>            //Cristal de 20MHz
#include <fuses.h>            //Fusibles
#include <rs232.h>            //Uso de RS232,
TX:C6 RV:C7

#define puerto_a 0x05 //Dirección de memoria para puerto A
#define puerto_b 0x06 //Dirección de memoria para puerto B
#define puerto_c 0x07 //Dirección de memoria para puerto C
#define puerto_d 0x08 //Dirección de memoria para puerto D
#define puerto_e 0x09 //Dirección de memoria para puerto E
#define priority rb,ext,rda //Prioridad de las interrupciones

char c; //Para recibir caracteres de la PC
```

```

int1 reversa=0; //Movimiento del motor para atrás
int1 habilitar=0; //Mover motor y aceptar segundos de la interrupción externa
long pasos=0; //Cuenta el número de segundos o pasos del motor
int1 termino=0; //Se completó la vuelta
int1 detener=0; //Se dio detener en el control
int1 fuente=0; //Se encuentra la fuente posicionada en el inicio
int1 detencion=0; //El sistema está detenido
int1 cero=0; //Control de botones, no acepta iniciar sin antes la fuente no esta
en inicio
int contador=0; //Control de pasos del motor
int1 solo=0; //Control de pulsador de fuente a cero
int1 poder=0; //Control de interrupción externa

#int_RB //Interrupción del puerto B, cambio en alguna patilla B4...B7
RB_isr()
{
printf(" RB ");
if(bit_test(puerto_b,7)){ //Pregunta si es B7
if(habilitar==0){ //No esta en movimiento?
if(cero==1){ //Además si antes apreté colocar fuente a cero?
if(fuente==1){ //La fuente se encuentra en inicio?
puerto_a=4; //Manda control a Pic2
detencion=0; //Sistema no detenido
habilitar=1; //Acepta segundos, y pasos del motor
printf(" Iniciar");
}
else{}
}
else{}
}
else{}
}
}

else{}

if(bit_test(puerto_b,6)){ //Pregunta si es B6
if(habilitar==1){ //Esta en movimiento?
puerto_a=5; //Manda control a Pic2
detencion=1; //Sistema detenido
habilitar=0; //No acepta segundos, ni pasos del motor
printf(" Detener");
}
else{
if(detencion==1){ //Esta el sistema detenido?
detencion=0; //Reseteo
}
}
}
}

```

```

    puerto_a=15;      //Manda control a Pic2. Detener todo
    detener=1;       //Paro de emergencia, todo detenido
    cero=0;          //Reset se cero, para no dar iniciar hasta volver a colocar la
fuente
    Output_Low(Pin_E2); //Cierre de fuente de radicación
    habilitar=0;      //No acepta segundos, ni pasos del motor
    solo=0;           //Reset de solo
    fuente=0;         //La fuente no esta en posición inicial
    printf(" Detener todo");
    }
else{}

}

}

else{}

if(bit_test(puerto_b,5)){ //Pregunta si es B5
    if(solo==0){ //Fuente no en cero?
        puerto_a=6; //Manda control a Pic2
        solo=1; //Aceptar pulsador una vez
        pasos=0; //Reset del contador de pasos y segundos
        reversa=1; //Se indica mover el motor para atrás
        detencion=0; //Reset de sistema detenido
        termino=0; //Reset de se completó la vuelta
        detener=0; //Reset de sistema detenido
        cero=1; //Se oprimió botón de posicionar la fuente
        printf(" Otra grafica,fuente a cero");
    }
else{}
}

else{}

delay_ms(400); //Tiempo de espera para aceptar un botonazo y no dos a la vez

}

#int_EXT //Interrupción externa, cambio de bajo a alto en patilla B0
EXT_isr()
{
printf(" EXT ");
if(habilitar==1){ //Está habilitado el sistema?
    pasos++; //Aumento en uno pasos, cada segundo se aumenta en 1
    if(pasos>=25){ //Se puede detectar final de carrera
        poder=1;
    }
}
}

```

```

    }
else{}

if(poder==1){
    if(bit_test(puerto_e,0)){ //Final de recorrido
        pasos=0; //Reset de contador
        fuente=0; //La fuente no esta en posición inicial
        Output_Low(Pin_E2); //Se abre fuente de radiación
        habilitar=0; //No se aceptan mas pasos, motor se frena
        cero=0; //Reset se cero, para no dar iniciar hasta volver a colocar la fuente
        termino=1; //Se completó la vuelta
        puerto_a=14; //Manda control a Pic2
        delay_ms(2);
        puerto_a=7;
    }
    else{}
}
else{}

}

else{}
}
}

#int_RDA
rda_isr() { //Interrupción de dato recibido en Rv
    c = getchar(); //El caracter se almacena en c
    printf(" RDA ");

switch (c){ //Se preguntan por diferentes posibilidades

    case 'c' : //Programa java con control
        puerto_a=1; //Manda control a Pic2.
        detener=0; //Reset de detener
        pasos=0; //Reset de numero de pasos
        disable_interrupts(INT_RB); //NO se aceptan botones externos de control
        break;

    case 'l' : //Leer una posición de memoria en PIC2.Cuantos datos son?
        puerto_a=14; //Manda control a Pic2
        delay_ms(2);
        puerto_a=2;break;

    case 'o' : //Leer dato de radiación de EEPROM.Mandar dato a PC
        puerto_a=14; //Manda control a Pic2
        delay_ms(2);
        puerto_a=3;break;

```

```

case 'm' :           //Iniciar
    puerto_a=4;     //Manda control a Pic2
    habilitar=1;break; //Acepta segundos, y pasos del motor

case 'j' :           //Continuar
    puerto_a=4;     //Manda control a Pic2
    habilitar=1;break; //Acepta segundos, y pasos del motor

case 'k' :           //Detener
    puerto_a=5;     //Manda control a Pic2
    habilitar=0;break; //No acepta segundos, ni pasos del motor

case '1' :           //Se va a leer gráfica 1
    puerto_a=14;    //Manda control a Pic2
    reversa=0;      //Reset de reversa
    habilitar=0;    //No acepta segundos, ni pasos del motor
    delay_ms(2);
    puerto_a=8;break;

case '2' :           //Se va a leer gráfica 2
    puerto_a=14;    //Manda control a Pic2
    reversa=0;      //Reset de reversa
    habilitar=0;    //No acepta segundos, ni pasos del motor
    delay_ms(2);
    puerto_a=9;break;

case '3' :           //Se va a leer gráfica 3
    puerto_a=14;    //Manda control a Pic2
    reversa=0;      //Reset de reversa
    habilitar=0;    //No acepta segundos, ni pasos del motor
    delay_ms(2);
    puerto_a=10;break;

case '4' :           //Se va a leer gráfica 4
    puerto_a=14;    //Manda control a Pic2
    reversa=0;      //Reset de reversa
    habilitar=0;    //No acepta segundos, ni pasos del motor
    delay_ms(2);
    puerto_a=11;break;

case '5' :           //Se va a leer gráfica 5
    puerto_a=14;    //Manda control a Pic2
    reversa=0;      //Reset de reversa
    habilitar=0;    //No acepta segundos, ni pasos del motor

```

```

        delay_ms(2);
        puerto_a=12;break;

    case 'b' :           //Borrar datos en memoria de gráfica recién leída
        puerto_a=14;    //Manda control a Pic2
        delay_ms(2);
        puerto_a=13;break;

    case 'p' :           //Se dio un paro de emergencia, detener todo
        puerto_a=15;    //Manda control a Pic2
        detener=1;      //Sistema detenido
        Output_Low(Pin_E2); //Cierra la fuente de radiación
        habilitar=0;    //No acepta segundos, ni pasos del motor
        printf(" Detener todo");
        break;

    case 's' :           //Posicionar la fuente a cero
        puerto_a=14;    //Manda control a Pic2
        detener=0;      //Reset de detener
        pasos=0;        //Reset de pasos
        delay_ms(2);
        termino=0;      //Reset de vuelta completada
        puerto_a=1;     //Manda control a Pic2
        reversa=1;      //Dar movimiento al motor para atrás
        break;

    default : break;

    }
}

void main() {           //Menú principal

    setup_adc_ports(NO_ANALOGS); //No canales analógicos
    setup_adc(ADC_OFF);         //No conversión ADC
    setup_spi(FALSE);
    setup_psp(PSP_DISABLED);
    setup_counters(RTCC_INTERNAL,RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    enable_interrupts(INT_RB);   //Aceptar interrupción RB
    enable_interrupts(INT_EXT);  //Aceptar interrupción EXT
    enable_interrupts(INT_RDA);  //Aceptar interrupción RDA
    enable_interrupts(GLOBAL);   //Aceptar interrupciones

```

```

set_tris_a( 0x00 );           //En puerto A todas salidas
set_tris_b( 0xFF );         //En puerto B todas entradas
set_tris_d( 0x00 );         //En puerto D todas salidas
set_tris_e( 0x01 );         //En puerto E, E0 como entradas y demás salidas
puerto_e=0;
puerto_a=0;
Output_Low(Pin_D0);         //Energiza el motor
Output_Low(Pin_D3);         //Energiza el motor
Output_High(Pin_D1);        //Energiza el motor
Output_High(Pin_D2);        //Energiza el motor
Output_High(Pin_E1);        //E1 siempre alto
reversa=0;
pasos=0;
habilitar=0;
detener=0;

do{

/////
if(reversa==1){             //Se dio dar reversa?
printf(" Entre ");
reversa=0;
printf("%3U",reversa);      //Hacerlo una vez
while(!bit_test(puerto_e,0)){ //Mientras E0 este en bajo
if(detener==0){             //Sistema no detenido?
if(contador==3){           //Da primer paso
puerto_d=0b00001001;
delay_ms(4);
--contador;
}

else if(contador==2){      //Da segundo paso
puerto_d=0b00001010;
delay_ms(4);
--contador;
}

else if(contador==1){      //Da tercer paso
puerto_d=0b00000110;
delay_ms(4);
--contador;
}

else{                       //Da cuarto paso y vuelve al ciclo
puerto_d=0b00000101;

```

```

        delay_ms(4);
        contador=3;
    }

    }

    else{break;} //Si hubo pedido de detención se para el movimiento
}

switch (detener){ //Al salir del while pregunta por el estado de detener
case 0: //Se llego al inicio se haber detener
    Output_High(Pin_E2); //Abrir la fuente de radiación
    printf(" Llego atras ");
    printf("%3U",reversa);
    habilitar=0;
    fuente=1;break; //Fuente esta posicionada en el inicio
case 1: break; //Se dio detener, no hacer nada
}
}

else{}
/////

/////

if(habilitar==1){ //Esta habilitado el sistema
    printf("Adelante");
    while(termino==0){ //Mientras no se haya completado la vuelta
        if(habilitar==0){break;} //Si se da detener sale
        else{}
        if(detener==1){break;} //Si se da paro de emergencia sale
        else{}
        if(contador==0){ //Da primer paso
            puerto_d=0b00001010;
            delay_ms(4);
            ++contador;
        }

        else if(contador==1){ //Da segundo paso
            puerto_d=0b00001001;
            delay_ms(4);
            ++contador;
        }

        else if(contador==2){ //Da tercer paso
            puerto_d=0b00000101;
            delay_ms(4);
            ++contador;
        }
    }
}

```



```

    }

    else{
        //Da cuarto paso y vuelve al ciclo
        puerto_d=0b00000110;
        delay_ms(4);
        contador=0;
    }
}

else{}
/////

} while(TRUE); //Hacerlo siempre

}

```

A.7 Programa PIC secundario

```

#include <16F877A.h> //Tipo de PIC
#include <delay.h> //Cristal de 20MHz
#include <fuses.h> //Fusibles
#include <rs232.h> //Uso de RS232,
TX:C6 RV:C7

#define puerto_a 0x05 //Dirección de memoria para puerto A
#define puerto_b 0x06 //Dirección de memoria para puerto B
#define puerto_c 0x07 //Dirección de memoria para puerto C
#define puerto_d 0x08 //Dirección de memoria para puerto D
#define puerto_e 0x09 //Dirección de memoria para puerto E
#include <24fc512.c> //Incluye driver para uso de memoria EEPROM
24fc512

#define prioridad rb,rda,rtcc //Prioridad en las interrupciones
#define rom 0x2100={0,0,0,0,0,0,0,0,0,0} //Para guardar cantidad de datos en cada gráfica

int port=0; //Para lectura de puerto_b
int i=0; //Para for de corrimiento de puerto_b
char c; //Para recibir caracteres de la PC
int16 numero=0; //Cantidad de datos que llevamos, para dirección de memoria
int1 habilitar=0; //Da paso a recibir datos
int1 primera=0; //Para primer dato recibido

```

```

int1 ir=0;           //Para segundo dato recibido
BYTE cont=0;        //Cuenta el tiempo entre primer y segundo dato recibido
int1 grabar=0;      //Ir a grabar datos
BYTE dato1;         //BYTE 1 de dato
BYTE dato2;         //BYTE 2 de dato
BYTE valor;         //BYTE leído de memoria
BYTE num=0;         //Control es posición 0 de EEPROM del número de gráficas
int menos=0;        //Para cantidad de datos, Mem. del PIC
int mas=0;          //Para cantidad de datos, Mem. del PIC
int grafica=0;      //Llevar un control sobre en cual gráfica estamos
int16 leer=0;       //Posicionarse en una dirección de la memoria para ser leída
int1 datos=0;       //Para leer en memoria del PIC, pos 1 ó pos 2 de cada gráfica
int1 control=0;     //Dice si es Java con control
int1 reinicio=0;    //Al iniciar dar el valor correcto a grafica

#int_RB
RB_isr()             //Interrupción del puerto B, cambio en alguna patilla B4...B7
{
port=puerto_b;     //El valor que tiene el puerto lo pasa a port

for(i=1; i<=4; ++i){ //Corre cuatro bits de port a la derecha
    SHIFT_RIGHT(&port,1,0);
}
switch (port){      //Pregunta por el valor obtenido

case 1 : //Para c
    numero=0;       //Reset de numero
    menos=0;        //Reset de menos
    control=1;      //Programa Java con control
    mas=0;          //Reset de mas
    grafica=1;      //Guardar datos en posición de grafica 1
    break;

case 2 : //Para l
    write_eeprom(0,0); //Poner un 0 en la dirección 0 de EEPROM interna del
PIC
    if(datos==0){   //Leer posición 1 de la grafica que estamos trabajando
        datos=1;    //Lo pone para leer el siguiente
        valor=read_eeprom(grafica); //Lee el valor
        delay_ms(50);
        printf("%3U",valor);} //Lo manda
    else{
        datos=0;    //Reset de datos
        valor=read_eeprom(grafica+1); //Lee posición 2 de la grafica que estamos
trabajando

```

```

        delay_ms(50);
        printf("%3U",valor);}    //Lo manda
    break;

    case 3 : //Para o
        valor=read_ext_eeprom(leer);    //Lee direccion de memoria de Byte 1 del
dato
        delay_ms(8);
        leer++;    //Aumenta la dirección
        delay_ms(50);
        printf("%3U",valor);    //Lo manda
        valor=read_ext_eeprom(leer);    //Lee direccion de memoria de Byte 2 del
dato
        delay_ms(8);
        leer++;    //Aumenta la dirección
        delay_ms(50);
        printf("%3U",valor);    //Lo manda
        break;

    case 4 : //Para m
        habilitar=1;    //Se dio iniciar, acepte datos del Geiger
        primera=0;    //Reset de primera para primer byte recibido
        ir=0;    //Reset de ir para primer byte recibido
        break;

    case 5 : //Para k
        habilitar=0;    //Se dio detener, no acepte datos del Geiger
        break;

    case 6 : //Otra gráfica
        menos=0;    //Reset de menos
        mas=0;    //Reset de mas
        num=read_eeprom(0);    //Lee posición 0 de la EEPROM del PIC
        delay_ms(1);
        switch (num){    //Pregunta por su valor
            case 0 : grafica=1;    //Se esta en gráfica 1
                numero=0;    //Se direcciona el puntero de EEPROM externa a 0
                break;

            case 1 : grafica=3;    //Se esta en gráfica 2
                numero=1024;    //Se direcciona el puntero de EEPROM externa a
1024
                break;

            case 2 : grafica=5;    //Se esta en gráfica 3

```

```

                numero=2048; //Se direcciona el puntero de EEPROM externa a
2048                break;

                case 3 : grafica=7; //Se esta en gráfica 4
                    numero=3072; //Se direcciona el puntero de EEPROM externa a
3072                break;

                case 4 : grafica=9; //Se esta en gráfica 5
                    numero=4096; //Se direcciona el puntero de EEPROM externa a
4096                break;

                default: break;
            }
        break;

        case 7 : //Para terminó vuelta
            if(control==0){ //Es Java sin control?
                num=read_eeprom(0); //Lee posición 0 de la EEPROM del PIC
                delay_ms(1);
                num=num+1; //La aumenta en 1
                write_eeprom(0,num); //La reescribe
            }
            else{
                habilitar=0; //No acepta datos del Geiger
                break;

                case 8 : //Tiene datos grafica 1?
                    write_eeprom(0,0); //Poner un 0 en la dirección 0 de EEPROM
                    interna del PIC
                    grafica=1; //Es para grafica 1
                    leer=0; //Puntero de memoria se pone en 0
                    if(datos==0){ //Leer posición 1 de la grafica que estamos
                    trabajando
                        datos=1; //Lo pone para leer el siguiente
                        valor=read_eeprom(1); //Lee el valor
                        delay_ms(50);
                        printf("%3U",valor);} //Lo manda
                    else{
                        datos=0; //Reset de datos
                        valor=read_eeprom(2); //Leer posición 2 de la grafica que estamos
                    trabajando
                        delay_ms(50);

```

```

        printf("%3U",valor);}      //Lo manda
        break;

    case 9 : //Tiene datos grafica 2?
        grafica=3;                //Es para grafica 2
        leer=1024;                //Puntero de memoria se pone en 1024
        if(datos==0){            //Leer posición 1 de la grafica que estamos
trabajando
            datos=1;              //Lo pone para leer el siguiente
            valor=read_eeprom(3); //Lee el valor
            delay_ms(50);
            printf("%3U",valor);} //Lo manda
        else{
            datos=0;              //Reset de datos
            valor=read_eeprom(4); //Leer posición 2 de la grafica que estamos
trabajando
            delay_ms(50);
            printf("%3U",valor);} //Lo manda
        break;

    case 10 : //Tiene datos grafica 3?
        grafica=5;                //Es para grafica 3
        leer=2048;                //Puntero de memoria se pone en 2048
        if(datos==0){            //Leer posición 1 de la grafica que estamos
trabajando
            datos=1;              //Lo pone para leer el siguiente
            valor=read_eeprom(5); //Lee el valor
            delay_ms(50);
            printf("%3U",valor);} //Lo manda
        else{
            datos=0;              //Reset de datos
            valor=read_eeprom(6); //Leer posición 2 de la grafica que estamos
trabajando
            delay_ms(50);
            printf("%3U",valor);} //Lo manda
        break;

    case 11 : //Tiene datos grafica 4?
        grafica=7;                //Es para grafica 4
        leer=3072;                //Puntero de memoria se pone en 3072
        if(datos==0){            //Leer posición 1 de la grafica que estamos
trabajando
            datos=1;              //Lo pone para leer el siguiente
            valor=read_eeprom(7); //Lee el valor
            delay_ms(50);

```

```

        printf("%3U",valor);} //Lo manda
    else{
        datos=0; //Reset de datos
        valor=read_eeprom(8); //Leer posición 2 de la grafica que estamos
trabajando
        delay_ms(50);
        printf("%3U",valor);} //Lo manda
        break;

    case 12 : //Tiene datos grafica 5?
        grafica=9; //Es para grafica 5
        leer=4096; //Puntero de memoria se pone en 4096
        if(datos==0){ //Leer posición 1 de la grafica que estamos
trabajando
            datos=1; //Lo pone para leer el siguiente
            valor=read_eeprom(9); //Lee el valor
            delay_ms(50);
            printf("%3U",valor);} //Lo manda
        else{
            datos=0; //Reset de datos
            valor=read_eeprom(10); //Leer posición 2 de la grafica que estamos
trabajando
            delay_ms(50);
            printf("%3U",valor);} //Lo manda
            break;

    case 13 : //Para b
        write_eeprom(grafica,0); //Pone en el valor de grafica que estamos un 0
        delay_ms(1);
        write_eeprom(grafica+1,0); //Pone en el valor de grafica+1 que estamos
un 0
        break;

    case 15 : //Para p
        habilitar=0; //No acepta datos del Geiger
        break;

    default : break;
        }
}

#int_rtcc //Interrupción del timer 0
void clock_isr() {

```

```

++cont;      //Aumenta en 1 cont cada vez que se da la interrupción
}

#int_RDA
RDA_isr()    //Interrupción de dato recibido en Rv
{
c=getc();    //El caracter se almacena en c

switch(habilitar){    //Se preguntan por diferentes posibilidades
case 0:break;        //Si habilitar(Iniciar) no esta en 1, no guarda
datos

case 1:            //El sistema esta iniciado
switch(primera){
case 0: if(ir==0){    //Primer byte
enable_interrupts(INT_RTCC); //Se acepta interrupción RTCC
dato1=c;            //Byte se guarda en dato1
cont=0;            //Reset de cont
ir=1;            //Para segundo byte
}
else{
disable_interrupts(INT_RTCC); //Se deshabilita interrupción RTCC
dato2=c;            //Byte se guarda en dato2
ir=0;            //Para primer byte
primera=1;        //Llego primer dato
if(cont>50){    //Distancia grande, dato incorrecto
dato1=dato2;    //Cambio de valores
ir=1;            //Para segundo byte
}
else{grabar=1;}    //Dato correcto, grabarlo
}break;

case 1: if(ir==0){    //Para primer byte
dato1=c;            //Byte se guarda en dato1
ir=1;            //Para segundo byte
}
else{
dato2=c;            //Byte se guarda en dato2
ir=0;            //Para primer byte
grabar=1;        //Grabar dato
}break;
}break;
}

if(grabar==1){    //Grabar dato?

```

```

    grabar=0; //Hacerlo una vez
    write_ext_eeprom(numero,dato1); //En la dirección apuntada por el
    puntero numero, guardar dato1
    delay_ms(10);
    numero++; //Aumenta dirección
    write_ext_eeprom(numero,dato2); //En la dirección apuntada por el
    puntero numero, guardar dato1
    numero++; //Aumenta dirección
    menos++; //Aumenta entero menos
    delay_ms(10);
    printf("%3U",grafica);
    printf(" ");
    printf("%3U",menos);
    write_eeprom(grafica,menos); //Control de datos escritos
    if(menos==0){ //Entero se desbordó
        mas++; //Aumenta en 1 a mas
        write_eeprom(grafica+1,mas); //Control de datos escritos
        printf("%3U",grafica);
        printf(" ");
        printf("%3U",mas);
    }
    else{}
else{}
}

void main() //Menú principal
{
    setup_adc_ports(NO_ANALOGS); //No canales analógicos
    setup_adc(ADC_OFF); //No conversión ADC
    setup_psp(PSP_DISABLED);
    setup_spi(FALSE);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    set_timer0(0); //Tomar timer0
    setup_counters( RTCC_INTERNAL, RTCC_DIV_256 | RTCC_8_BIT);//
    20000000/(256*256*4) = 76 int por seg
    enable_interrupts(INT_RB); //Aceptar interrupción RB
    enable_interrupts(INT_RDA); //Aceptar interrupciónRDA
    enable_interrupts(GLOBAL); //Aceptar interrupciones
    set_tris_a( 0x00 ); //En puerto A todas salidas

    puerto_a = 0; //Puerto A=0
    puerto_b = 0; //Puerto B=0
}

```



```

puerto_c = 0; //Puerto C=0
puerto_d = 0; //Puerto D=0
puerto_e = 0; //Puerto E=0

do{
  Output_High(Pin_A0);      //Siempre A0 en alto

  if(reinicio==0){          //Se reinicio el sistema?
    reinicio=1;             //se pone en uno para hacerlo a una vez
    menos=0;                //Reset de menos
    mas=0;                  //Reset de mas
    num=read_eeprom(0);
    delay_ms(1);
    switch (num){           //Pregunta por su valor
      case 0 : grafica=1;   //Se esta en gráfica 1
        numero=0;          //Se direcciona el puntero de EEPROM externa a 0
        break;

      case 1 : grafica=3;   //Se esta en gráfica 2
        numero=1024;       //Se direcciona el puntero de EEPROM externa a 1024
        break;

      case 2 : grafica=5;   //Se esta en gráfica 3
        numero=2048;       //Se direcciona el puntero de EEPROM externa a 2048
        break;

      case 3 : grafica=7;   //Se esta en gráfica 4
        numero=3072;       //Se direcciona el puntero de EEPROM externa a 3072
        break;

      case 4 : grafica=9;   //Se esta en gráfica 5
        numero=4096;       //Se direcciona el puntero de EEPROM externa a 4096
        break;

      default: break;
    }
  }
  else{}

} while( TRUE ); // Repetir siempre
}

```

A.8 Programa driver de memoria

```
#ifndef EEPROM_SDA

#define EEPROM_SDA PIN_C4      //Pin C4 como SDA
#define EEPROM_SCL PIN_C3      //Pin C3 como SCL

#endif

#include <i2c> //Características de la
comunicación, PIC como máster

#define EEPROM_ADDRESS long int //16 bits para la dirección
#define EEPROM_SIZE 65536 //Tamaño de memoria

void init_ext_eeprom() //Inicialización de la memoria
{
    output_float(EEPROM_SCL); //Al inicio patillas en estao flotante
    output_float(EEPROM_SDA);
}

void write_ext_eeprom(long int address, char data) //Escritura
{
    i2c_start(); //Inicio
    i2c_write(0xa0); //Byte de control
    i2c_write(address>>8); //Manda 8 bits más significativos de la dirección
    i2c_write(address); //Manda 8 bits menos significativos de la dirección
    i2c_write(data); //Manda dato
    i2c_stop(); //Paro
    delay_ms(11); //Tiempo para lectura sea válida
}

BYTE read_ext_eeprom(long int address) { //Lectura
    char data; //Dato a almacenar
    i2c_start(); //Inicio
    i2c_write(0xa0); //Byte de control
    i2c_write(address>>8); //Manda 8 bits más significativos de la dirección
    i2c_write(address); //Manda 8 bits menos significativos de la dirección
    i2c_start(); //Inicio
    i2c_write(0xa1); //Byte de control
```

```
data=i2c_read(0);    //Obtiene el dato
i2c_stop();         //Paro
return(data);       //Retorna el dato leído
```

Anexos

B.1 Especificaciones para el uso de la EEPROM 24FC512:

Direccionamiento el dispositivo:

Una vez la condición de start dado por el maestro, el byte de control debe ser el primer byte recibido. Este byte de control consiste de 4 bits de código de control; para la serie 24XXX512 estos bits son 1010 para lecturas y escrituras. Los siguientes 3 bits son para la selección del dispositivo (A2, A1y A0). Estos son usados para cuando hasta 8 memorias están conectadas al mismo bus y para la selección de cada una se usa estos 3 bits. Corresponden al valor lógico de las patillas 3,2 y 1.

El último bit de control define la operación que se va a realizar, si es lectura o escritura. Cuando este bit se pone en alto indica que una operación de lectura se va a llevar a cabo y cuando se pone en bajo es una escritura la operación a realizar.

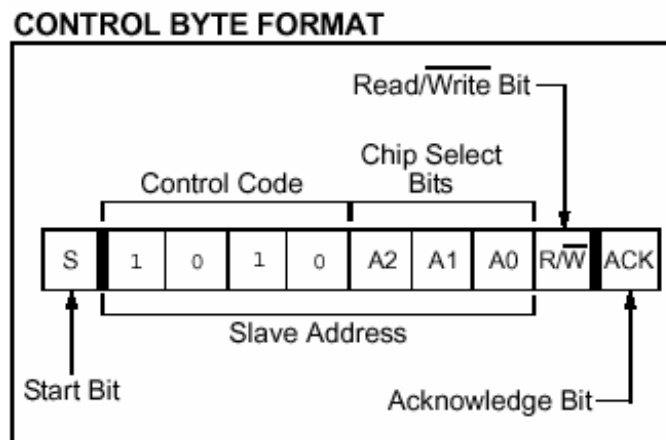


Figura B.1.1 Formato del byte de control

Los siguientes dos bytes recibidos por la memoria son la dirección del primer byte de datos, ya sea a leer o escribir.

ADDRESS SEQUENCE BIT ASSIGNMENTS

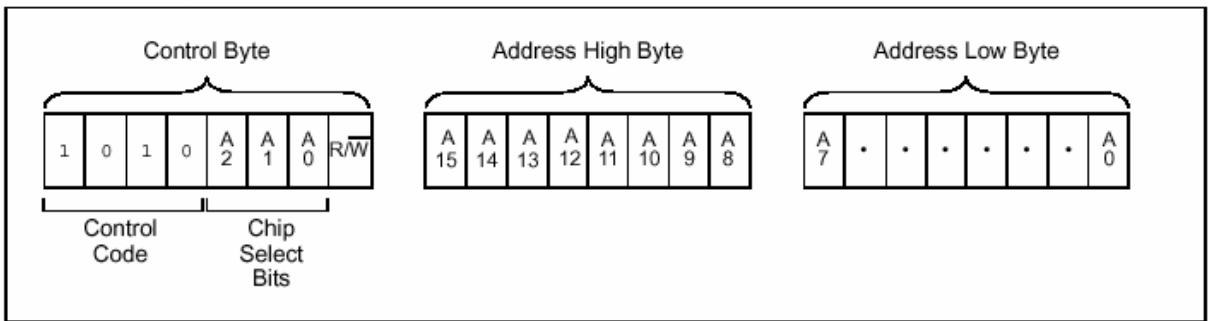


Figura B.1.2 Asignamientos de bits para la secuencia de dirección

Operaciones de escritura:

BYTE WRITE

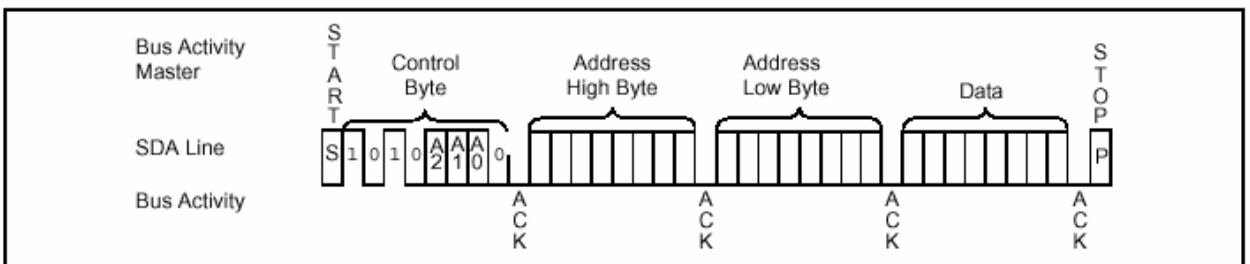


Figura B.1.3 Escritura de 1 bytes

Después del comando de escritura, el contador de dirección interno es aumentado automáticamente en uno y de esta forma se puede dar una escritura por páginas hasta 128 bytes en donde sólo es necesaria la dirección del primer byte.

PAGE WRITE

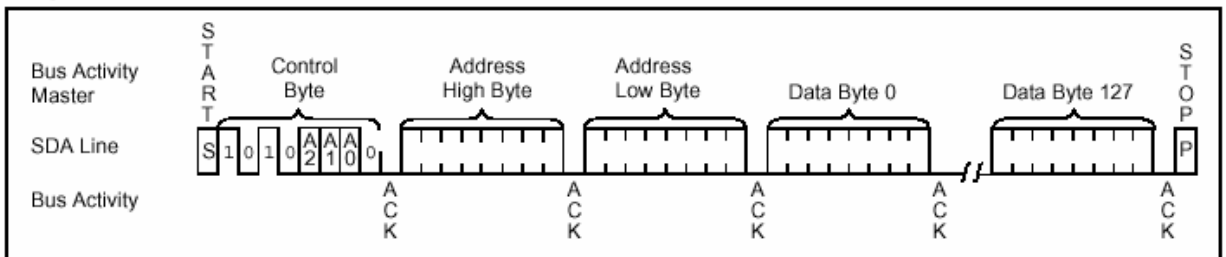


Figura B.1.4 Escritura de una página de bytes

Operación de lectura:

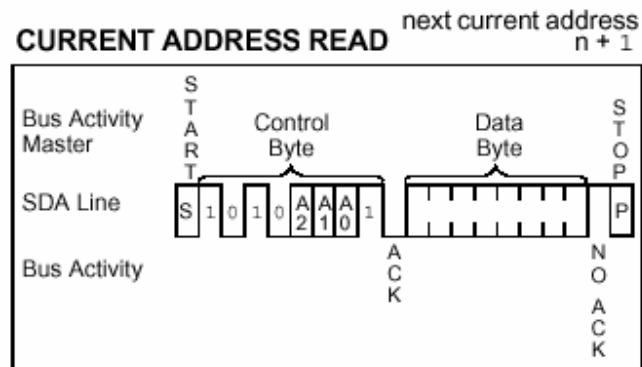


Figura B.1.5 Secuencia de lectura normal de un byte

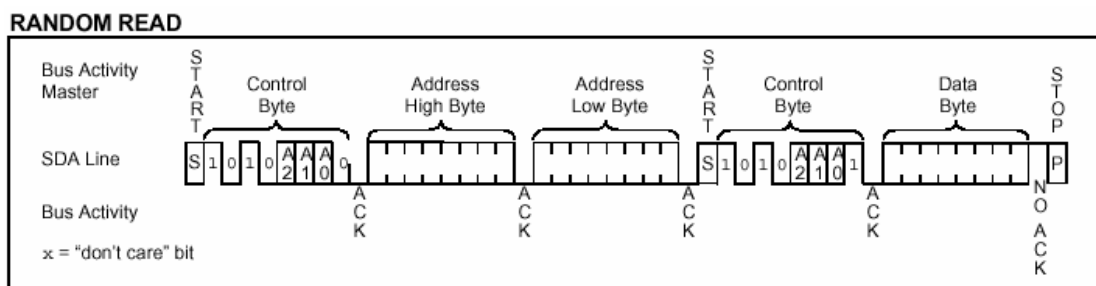


Figura B.1.6 Secuencia de lectura de un byte en una dirección al azar

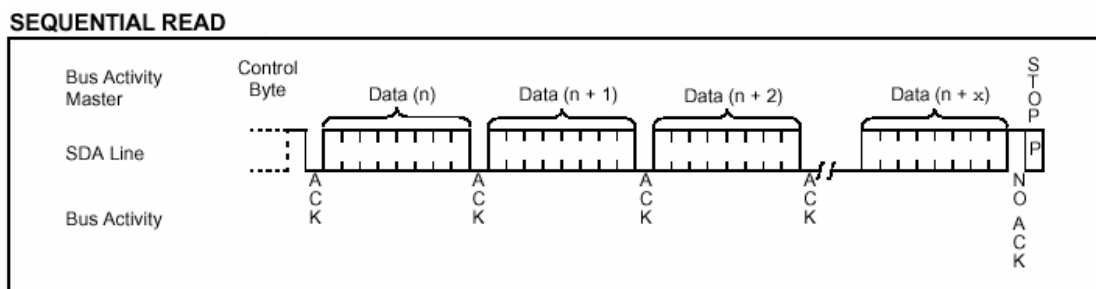


Figura B.1.7 Secuencia de lectura de varios bytes seguidos

B.2 Sobre radiación y cesio-137 [1], [2], [3]

Radiografía

En la actualidad, dentro del campo de la industria existen dos técnicas comúnmente empleadas para la inspección radiográfica:

- Radiografía con rayos X
- Radiografía con rayos gamma.

La principal diferencia entre estas dos técnicas es el origen de la radiación electromagnética; ya que, mientras los rayos X son generados por un alto potencial eléctrico, los rayos gamma se producen por desintegración atómica espontánea de un radioisótopo.

Radioisótopos

Existen alrededor de 100 elementos diferentes con 300 isótopos estables y más de 500 que son inestables. La inestabilidad surge porque algunos isótopos tienen un desequilibrio de neutrones; por lo que intentan alcanzar una relación más estable entre éstos y los protones, por medio de la emisión de radiaciones alfa, beta y gamma. Es por ello que a los isótopos inestables se les conoce como isótopos radiactivos o radioisótopos.

Existen radioisótopos naturales como el Radio-226 y el Polonio-216, o bien, como subproductos de las reacciones de fisión nuclear como es el caso del Cesio-137 o el Plutonio-239. Pero la mayoría de los isótopos radiactivos conocidos actualmente son producidos artificialmente a partir de átomos estables que son transformados, al bombardearlos durante un cierto tiempo con neutrones en un reactor nuclear, donde el proceso de fisión produce una gran cantidad de neutrones libres. A este proceso de transformación de un elemento estable radioactivo, se le conoce como “activación”.

Tipos de radiación

Los isótopos radioactivos, se desintegran emitiendo espontáneamente radiación y partículas subatómicas que se conocen como:

- Partículas Alfa.
- Partículas Beta.
- Partículas Gamma.

Las partículas alfa y beta, son emitidas desde el núcleo del radioisótopo. Estos 2 tipos de partículas no son emitidos simultáneamente por todos los radioisótopos, ya que algunos emiten sólo partículas y otros emiten partículas beta; en tanto que los rayos gamma acompañan unas veces a la emisión de partículas alfa y otras veces a la emisión de partículas beta. Sin embargo, cada sustancia radioactiva emitirá siempre el mismo tipo de radiación y de partículas, con lo que se formará un espectro característico.

Ningún proceso microscópico, físico o químico, como puede ser la elevación o disminución de la temperatura, o la combinación química con otras sustancias estables o radioactivas, puede modificar o alterar la actividad de un radioisótopo.

Partículas Alfa (α)

Después de una serie de experimentos se llegó a la conclusión y comprobación de que las partículas alfa son núcleos Helio. Dichas partículas tienen las siguientes características:

- Son partículas con masa ($6.62 \cdot 10^{-24}$ g).
- Son altamente ionizantes (10 veces más que los rayos gamma).
- Son desviadas en un campo magnético hacia el polo negativo.
- Tienen bajo poder de penetración.
- Su velocidad es aproximadamente de 1/10 veces la velocidad de la luz.

- Pueden ser detenidas fácil y efectivamente por material sólido como las hojas de papel.

Partículas Beta (β)

Las partículas beta se deben a la emisión de un electrón desde el núcleo. Pueden ser consideradas como el producto de la desintegración de un neutrón para formar un protón y un electrón, que entonces se convierte en la emisión de una partícula beta; estas partículas tienen las siguientes características:

- Son partículas con carga negativa.
- Tienen misma masa que los electrones.
- Son emitidas desde el núcleo.
- Su velocidad es de 9/10 de la velocidad de la luz.
- Son desviadas por un campo magnético hacia el polo positivo.
- Tienen bajo poder ionizante.
- Tienen poder de penetración (no muy alto).
- Pueden ser absorbidas totalmente por algunos materiales plásticos con un espesor de 6.2mm (1/4”).

Radiación Gamma (γ)

La radiación gamma es la de mayor poder de penetración, por lo que es la más útil en la inspección radiográfica. A diferencia de las partículas alfa o beta, la radiación gamma no es corpuscular y tiene las siguientes características:

- Se comporta como una onda electromagnética y como fotón.
- Es emitida por el núcleo.
- No posee masa ni carga eléctrica.
- No es desviada por un campo magnético.
- Tiene relativamente bajo poder ionizante.
- Presenta un elevado poder de penetración en materiales sólidos.
- Viaja en línea recta a la velocidad de la luz.

- Presenta una corta longitud de onda.

Unidades de medición

Se describen las unidades con las que se cuantifican las distintas manifestaciones del fenómeno radiológico más importantes para el trabajar en el proyecto.

Actividad:

Es la expresión de la velocidad de desintegración de un radioisótopo. En términos generales, se dice que es el número de desintegraciones que ocurren en un material radioactivo por unidad de tiempo.

Unidad de medición de la actividad:

A partir de 1975 la unidad de actividad es el Becquerelio (Bq), el cual está definido como una desintegración por segundo. La unidad anteriormente utilizada era el Curie (Ci); que se define como la cantidad de radioisótopo que produce $3.7 \cdot 10^9$ desintegraciones/seg.

La equivalencia de ambas unidades es:

$1 \text{ Ci} = 3.7 \cdot 10^9 \text{ Bq.}$ $1 \text{ Ci} = 37 \text{ GBq.}$

Actividad específica:

La actividad específica de cualquier radioisótopo o fuente radioactiva, se define como la actividad en Bq que presenta un gramo del radioisótopo, esto es Bq/g.

Vida media de los elementos radioactivos:

La vida media de un radio isótopo, es el tiempo necesario para que una actividad inicial quede reducida a la mitad. Este tiempo puede variar desde miles de años para algunos radioisótopos, hasta fracciones de segundo para otros.

La actividad de un radioisótopo depende de la vida media del mismo, por lo que es posible calcular la actividad de los radioisótopos por medio de la siguiente ecuación:

$$A = A_0 * 0.5 \text{ Exp } (t / \text{vm})$$

A = Actividad del radioisótopo en un tiempo t.
 A₀ = Actividad inicial del radioisótopo.
 t = Tiempo transcurrido desde la determinación de la actividad inicial.
 vm = Vida media del radioisótopo.

Electrón-Volt

La energía de los rayos X y gamma es medida en miles de electrón-volts (Kev) y millones de electrón-volts (Mev). Siendo un electrón-volt, la energía ganada por un electrón cuando éste es acelerado por un volt.

Emisividad o constante específica de los rayos gamma (γ):

Es también una medida de la energía de radiación que considera al tiempo, al tipo de radioisótopo y la distancia a partir de la fuente. La emisividad se expresa como C/ KhBq a un metro de distancia.

En la tabla B.1 se proporcionan algunas de las constantes de emisividad más comúnmente empleadas en radiografía.

Tabla B.2.1 Constantes de emisividad más comúnmente utilizados en radiografía

Radioisótopo	Constante específica (γ)
Co-60	9.06 * 10E-15
Cs-137	2.23 * 10E-15
Ir-192	3.83 * 10E-15

Rad:

EL Rad establece la dosis de radiación absorbida. Técnicamente, se define como la cantidad de cualquier tipo de energía ionizante, que da como resultado una

absorción de 100 Ergs de energía por cada gramo de materia. Otra unidad para la dosis de radiación absorbida es el Gray $1 \text{ Rad} = 0.1 \text{ Gray}$.

Factor de calidad (QF):

Es una medida de los efectos de los diferentes tipos de radiación en el tejido humano. Es una nueva forma de expresar al RBE y es la relación entre el tipo de radiación absorbida y su efecto biológico.

RBE:

Se define como el Efecto Biológico Relativo, no es precisamente una unidad técnica, más bien es un factor de ponderación necesario para registrar los efectos de la radiación

En la tabla B.2 se presentan los valores del factor RBE para los diferentes tipos de radiación; éstos son iguales para el factor de calidad (QF).

Tabla B.2.2 Valores del factor RBE para diferentes los tipos de radiación

Tipo de radiación	RBE (Factor de calidad)
Gamma (γ)	1
Rayos X	1
Beta (β)	1
Neutrones, protones	10
Alfa (α)	20

REM:

En protección radiológica, es necesario contar con una relación numérica bien definida entre la dosis absorbida y el efecto biológico que produce, ya que la dosis absorbida no es suficiente para predecir la severidad o probabilidad del efecto bajo condiciones no específicas.

El REM es una unidad que da una idea de los efectos biológicos producidos en los individuos por cualquier tipo de radiación. La dosis biológica (REM) para cualquier tipo de radiación se puede obtener multiplicando la dosis absorbida (RAD) por el efecto biológico relativo (RBE), esto es: $\text{REM} = \text{RAD} * \text{RBE}$

La dosis absorbida por el tejido humano de cualquiera de los dos tipos de radiación, es medida y registrada en REM o en mSv. El Siervert (Sv) es una nueva unidad para medir la exposición: $1 \text{ REM} = 0.010 \text{ Sv}$.

Cesio-137

“El Cesio es un metal de color gris, tiene un punto de fusión de 29°C y una densidad de 1.9 gr/cm^3 . Pertenece al grupo de los metales alcalinos y es el más reactivo de ellos, por esta razón no se emplea en forma metálica sino como Cloruro de Cesio y Cesio 137. Se considera que el Cesio 137 es uno de los productos más probables de las reacciones de fisión nuclear (produciéndose también Cesio 133 y Cesio 135), motivo por el cual es un producto abundante en los reactores nucleares. El problema de su obtención consiste en separarlo del uranio y de otros productos de la reacción nuclear, lo cual se logra al obtenerlo como Cloruro de Cesio, que es altamente soluble en agua; por lo que se deben tomar precauciones al encapsularlo. El Cesio 137 tiene una vida media de 30.1 años y su mecanismo de desintegración es muy sencillo, ya que el 92% se desintegra mientras emite una partícula beta y produce Bario-137, que se estabiliza al emitir un solo rayo gamma de 0.66 MeV de energía. La actividad específica de la cápsula dependerá de la cantidad de los otros dos isótopos de Cesio que estén presentes. La actividad específica máxima puede ser de 1000 GBq/gr (25 Ci/gr). Las cápsulas para contener al Cloruro de Cesio en forma de sinterizado, deben ser del tipo de doble pared para evitar posibles fugas.

Por características, el Cesio 137 se emplea actualmente como un isótopo de calibración, ya que emite un solo rayo gamma de energía constante.

Existen otros radioisótopos como el Tulio-170 y el Radio 225 que han caído en desuso por problemas técnicos y de seguridad. Debido a lo anterior, no se describen sus características.

En la siguiente tabla se encuentran resumidas las características de las fuentes de rayos gamma más comunes

Tabla B.2.3 Características de fuentes de rayos gamma más comunes

	Cobalto Co-60	Cesio Cs-137	Iridio Ir-192
Nivel de radiación RhM/Ci RhF/Ci	1.30 14.50	0.32 4.20	0.55 5.90
Energía promedio (MeV)	1.250	0.660	0.355
Equivalente en rayos X	2 a 3	0.20 a 1.50	0.30 a 0.80
Vida media	5.3 años	30 años	75 días
Capa hemirreductora (plomo)	12.7mm (0.5")	7.6mm (0.3")	5.1mm (0.2")
Penetración (Acero)	1 a 7"	1 a 3"	¼ a 3"
Tamaño de la fuente	Pequeña	Relativamente grande	Muy pequeña
Vida media	Relativamente larga	Larga	Muy corta
Actividad específica	Media	Baja	Muy alta
Costo	Bajo	Requiere doble encapsulación	Relativamente bajo
Otros	Disponibilidad rápidamente	Disponibilidad rápidamente	No disponible rápidamente

B.3 Corrosión

La corrosión puede producirse en cualquier punto de la superficie metálica. La velocidad a la que tiene lugar la oxidación de un determinado metal no es constante, sino que depende, entre otros factores, de la temperatura y la presión del oxígeno en la atmósfera oxidante. A mayor temperatura, la oxidación se produce más rápidamente, ya que la difusión de los iones metálicos o de oxígeno se ve favorecida a temperaturas elevadas. Y cuanto mayor sea la presión del oxígeno tanto mayor será la velocidad a la que verifica la oxidación, puesto que en este caso la difusión de los iones O_2^- se realiza con mayor facilidad.

Formación de una película de óxido (Pasos)

1. Absorción

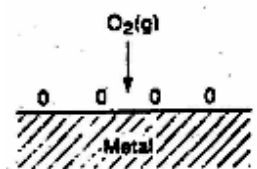


Figura B.3.1 Absorción de película de óxido

2. Nucleación y crecimiento del óxido. Disociación del oxígeno

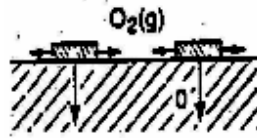


Figura B.3.2 Nucleación y crecimiento de óxido

3. Crecimiento de película. Oxidación interna

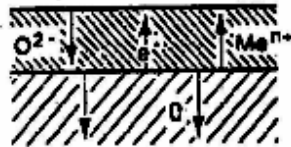


Figura B.3.3 Oxidación interna

4. Formación de Cavidades, porosidades, microgrietas.

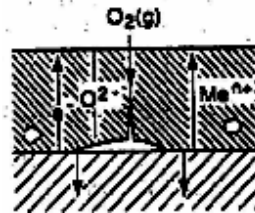


Figura B.3.4 Formación de Cavidades, porosidades, microgrietas

5. Formación de macrogrietas. Posible fase de fusión o evaporación de óxido

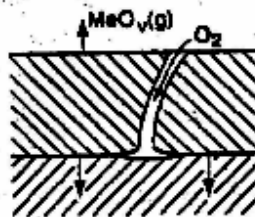


Figura B.3.5 Formación de macrogrietas

B.4 Descripción del contador Geiger GCA-04

Digital Geiger Counter with Wand & PC (serial) Output



Digital Geiger Counter Technical Data

Radiation Detected: Alpha, Beta, Gamma and X-Rays.

Detector: Geiger-Muller tube Ne + Halogen filled with a .38" effective diameter 1.5-2.0mg/cm² mica end window.

Detector: Alpha above 3.0 MeV

Sensitivity: Beta above 50 KeV

Gamma above 7 KeV

Countable Pulse Resolution & Range: 1 Count Per Minute (CPM) - 9,360 Counts Per Second (CPS)

Radiation Resolution & Range: 1.0 uR/hr - 655 mR/hr

(Metric) .01 uSv/hr - 6.5 mSv/hr

LCD Display : 16 character by 2 line LCD (liquid crystal display) with back light provides easy to read output. Secondary indicators; audio (clicks) and visual (LED) included. Also includes a headphone jack.

RS-232: Standard RS-232 output for PC available via DB-9 serial.

Power: 9 volt battery Or AC Adaptor for extended operation

Product Description

Images SI Inc's Digital Geiger Counters are useful for detecting and measuring radioactivity. The model GCA-04W uses an external wand that houses the geiger mueller tube. It's applications are: Education - Classroom demonstrations and experiments. Emergency Services and Domestic Preparedness; HAZMAT and Compliance Verification. Dirty Bomb Screening and EMT's. It detects the following types of radiation: Alpha above 3.0 MeV Beta above 50 KeV Gamma above 7 KeV. The Geiger Mueller (GM) Tube detector is Ne + Halogen filled. The GM tube's end window is made of Mica, with a .38 effective diameter and a density of 1.5-2.0 mg/cm². The Liquid Crystal Display (LCD) is 16 character by 2 line that provides an easy to read output. Switch selection allows Geiger Counter to measure and convert radiation (counts per second) into in mR/hr or mSv/hr. Factory calibration insures accurate measurement. Secondary indicators; audio (clicks) and visual (LED) included. Also includes a headphone jack and a power jack for external power. The GCA-04W outputs serially the counts per second. Serial cable RS-232 adapter is an available accessory.

B.5 Hojas de datos del PIC16F877A



PIC16F87X

28/40-pin 8-Bit CMOS FLASH Microcontrollers

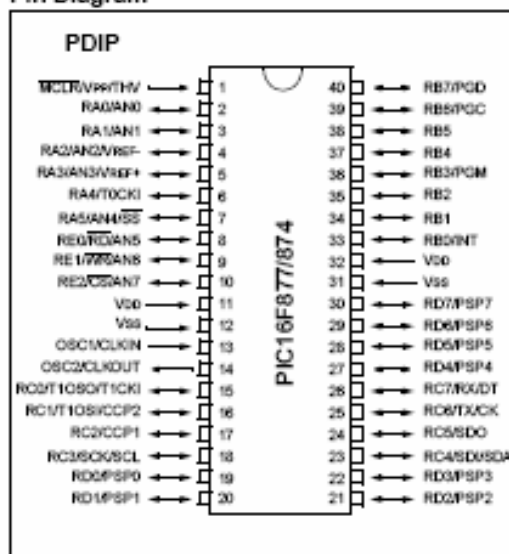
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory.
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM data memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 8-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

B.6 Hojas de datos del Max232

19-4323; Rev 11; 2/03

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

Features

Superior to Bipolar

- ♦ Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- ♦ Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- ♦ Meet All EIA/TIA-232E and V.28 Specifications
- ♦ Multiple Drivers and Receivers
- ♦ 3-State Driver and Receiver Outputs
- ♦ Open-Line Detection (MAX243)

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220CD	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering information continued at end of data sheet.
*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μ F)	SHDN & Three-State	Rx Active In SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0(0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0(0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0(0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0(0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0(0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0(0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0(0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0(0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0(0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0(0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/0	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

MAX220-MAX249

B.7 Hojas de datos de la memoria EEPROM 24FC512



MICROCHIP 24AA512/24LC512/24FC512

512K I²C™ CMOS Serial EEPROM

Device Selection Table

Part Number	V _{CC} Range	Max. Clock Frequency	Temp. Ranges
24AA512	1.8-5.5V	400 kHz ⁽¹⁾	I
24LC512	2.5-5.5V	400 kHz	I, E
24FC512	2.5-5.5V	1 MHz	I

Note 1: 100 kHz for V_{CC} < 2.5V

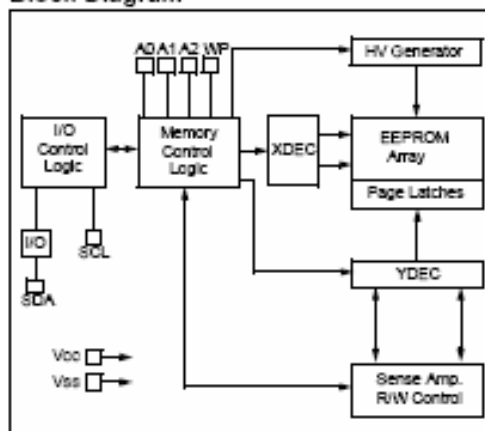
Features:

- Low-power CMOS technology:
 - Maximum write current 5 mA at 5.5V
 - Maximum read current 400 µA at 5.5V
 - Standby current 100 nA, typical at 5.5V
- 2-wire serial interface bus, I²C™ compatible
- Cascadable for up to eight devices
- Self-timed erase/write cycle
- 128-byte Page Write mode available
- 5 ms max. write cycle time
- Hardware write-protect for entire array
- Schmitt Trigger inputs for noise suppression
- 1,000,000 erase/write cycles
- Electrostatic discharge protection > 4000V
- Data retention > 200 years
- 8-pin PDIP, SOIC (208 mil), and DFN packages
- 14-lead TSSOP package
- Pb-free finishes available
- Temperature ranges:
 - Industrial (I): -40°C to +85°C
 - Automotive (E): -40°C to +125°C

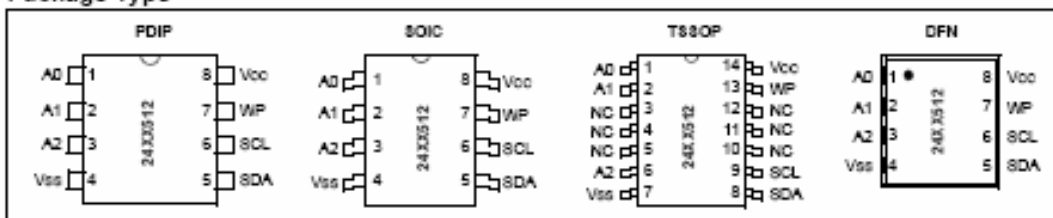
Description:

The Microchip Technology Inc. 24AA512/24LC512/24FC512 (24XX512*) is a 64K x 8 (512 Kbit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low-power applications such as personal communications and data acquisition. This device also has a page write capability of up to 128 bytes of data. This device is capable of both random and sequential reads up to the 512K boundary. Functional address lines allow up to eight devices on the same bus, for up to 4 Mbit address space. This device is available in the standard 8-pin plastic DIP, SOIC, DFN and 14-lead TSSOP packages.

Block Diagram



Package Type



B.8 Hojas de datos del regulador de voltaje LM317



LM117/217
LM317

1.2V TO 37V VOLTAGE REGULATOR

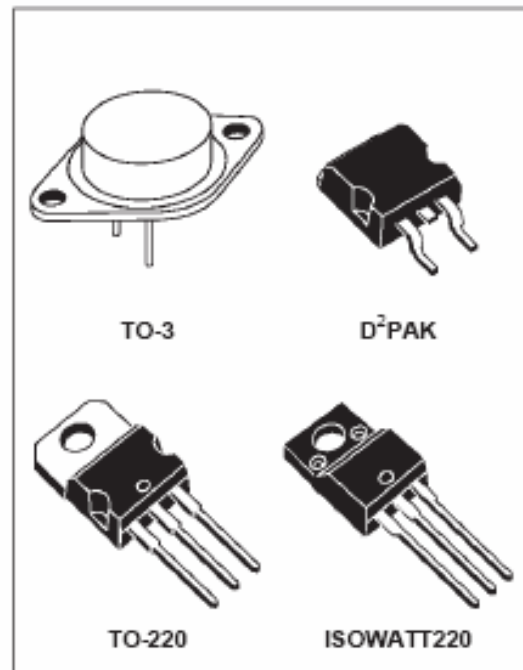
- OUTPUT VOLTAGE RANGE : 1.2 TO 37V
- OUTPUT CURRENT IN EXCESS OF 1.5A
- 0.1% LINE AND LOAD REGULATION
- FLOATING OPERATION FOR HIGH VOLTAGES
- COMPLETE SERIES OF PROTECTIONS :
CURRENT LIMITING, THERMAL
SHUTDOWN AND SOA CONTROL

DESCRIPTION

The LM117/LM217/LM317 are monolithic integrated circuit in TO-220, ISOWATT220, TO-3 and D²PAK packages intended for use as positive adjustable voltage regulators.

They are designed to supply more than 1.5A of load current with an output voltage adjustable over a 1.2 to 37V range.

The nominal output voltage is selected by means of only a resistive divider, making the device exceptionally easy to use and eliminating the stocking of many fixed regulators.



ABSOLUTE MAXIMUM RATING

Symbol	Parameter	Value	Unit
V_{I-O}	Input-output Differential Voltage	40	V
I_O	Output Current	Internally Limited	
T_{op}	Operating Junction Temperature for: LM117 LM217 LM317	-55 to 150 -25 to 150 0 to 125	°C °C °C
P_{tot}	Power Dissipation	Internally Limited	
T_{stg}	Storage Temperature	- 65 to 150	°C

THERMAL DATA

Symbol	Parameter	TO-3	TO-220	ISOWATT220	D ² PAK	Unit
$R_{thj-case}$	Thermal Resistance Junction-case	4	3	4	3	°C/W
$R_{thj-amb}$	Thermal Resistance Junction-ambient	35	50	60	62.5	°C/W

B.9 Hojas de datos del motor paso a paso



MAE® Stepper Motors



Stepper Motors

PennEngineering Motion Technologies offers a wide range of MAE brand stepper motor solutions. The HY series hybrid stepper motors feature low rotor inertia for maximum possible acceleration. The HN series hybrid stepper motors offer a calculated balance between low rotor inertia and high torque. The HS series hybrid stepper motors are optimized for superior torque characteristics. Additionally, both the HN and HS series feature low detent torque to holding torque ratios to provide smooth operation as well as the fine positioning capability required for microstep operation.

Motors may be customized with value added features including, but not limited to: gearboxes, encoders, shaft details, leadwire-connector assemblies, and more.

All specifications shown are typical at 20° C unless otherwise noted.

Shaft extensions

All motors can be supplied with single or double ended shaft.

Rotation

The motor rotation can run clockwise or counterclockwise, depending on the commutation.

Operating temperature

Ambient operating temperature: -20°C to +40°C.

Number of leads

Refer to specifications of individual models for standard lead wire configuration. Motors can be supplied with 4, 6, or 8 leads upon request; however, rated current and torque may be reduced.

Angular accuracy

Standard angular accuracy is $\pm 5\%$. Angular accuracy is defined as the deviation from a theoretical position, in percentage of one step, after any number of steps.

Holding torque

The typical values of holding torque of the different models are indicated in the data charts. Holding torque is measured with two phases each supplied at the rated current.

Specifications and approvals

Motors are manufactured according to EN 60034-1:1995-02. Motors with drive voltage higher than or equal to 120 V are suitable to be fitted on machines equipped with additional insulation or when the motor itself has the grounding through its damping screws.

Due to thermal considerations, stepper motors cannot always be operated continuously in dynamic conditions at the level of their static rated phase current.

Stepper Motors

- Accurate open loop control for high performance positioning applications
- Excellent low speed torque
- Simple, rugged construction for high reliability and long service life
- Smooth, quiet operation
- Standard NEMA frame sizes
- Precision honed stators and ground rotors for tight air gap and maximum performance
- CE approved



Get same day shipment of sample motors for models listed in this bulletin.

PennEngineering Motion Technologies offers a complete line of PITTMAN® and MAE® brand brush, brushless, and stepper motors which can be customized to meet your exact requirements.

B.10 Hojas de datos del transistor de potencia NTE331y NTE332



NTE331 (NPN) & NTE332 (PNP) Silicon Complementary Transistors Audio Power Amp, Switch

Description:

The NTE331 (NPN) and NTE332 (PNP) are silicon epitaxial–base complementary power transistors in a TO–220 plastic package intended for use in power linear and switching applications.

Absolute Maximum Ratings:

Collector–Base Voltage ($I_E = 0$), V_{CB0}	100V
Collector–Emitter Voltage ($I_B = 0$), V_{CE0}	100V
Emitter–Base Voltage ($I_C = 0$), V_{EB0}	5V
Emitter Current, I_E	15A
Collector Current, I_C	15A
Base Current, I_B	5A
Total Power Dissipation ($T_C \leq +25^\circ\text{C}$), P_D	90W
Operating Junction Temperature, T_J	+150°C
Storage Temperature Range, T_{stg}	–65° to +150°C
Thermal Resistance Junction–to–Case, R_{thJC}	1.4°C/W Max

Electrical Characteristics: ($T_C = +25^\circ\text{C}$ unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Collector Cutoff Current	I_{CB0}	$I_E = 0, V_{CB} = 100V$	–	–	500	μA
		$I_E = 0, V_{CB} = 100V, T_C = +150^\circ\text{C}$	–	–	5	mA
Collector Cutoff Current	I_{CE0}	$I_B = 0, V_{CE} = 50V$	–	–	1	mA
Emitter Cutoff Current	I_{EB0}	$I_C = 0, V_{EB} = 5V$	–	–	1	mA
Collector–Emitter Sustaining Voltage	$V_{CE0(sus)}$	$I_B = 0, I_C = 100\text{mA}$, Note 1	100	–	–	V
Collector–Emitter Saturation Voltage	$V_{CE(sat)}$	$I_C = 5A, I_B = 0.5A$, Note 1	–	–	1	V
		$I_C = 10A, I_B = 2.5A$, Note 1	–	–	3	V