



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

**SISTEMA DE AUTOMATIZACIÓN  
DEL PROCESO DE CALIBRACIÓN  
DEL ÁREA DE METROLOGÍA  
DIMENSIONAL DEL LAPEM DE LA  
CFE**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO ELÉCTRICO-ELECTRÓNICO

PRESENTA

**LÁZARO VELÁZQUEZ GERARDO**

DIRECTOR DE TESIS: M.I. BENJAMÍN VALERA  
OROZCO

CIUDAD UNIVERSITARIA, MÉXICO, D.F. ABRIL 2008



*Eternamente agradecido y con toda mi admiración y profundo respeto a:*

Piedad Pillado Arias †

*En agradecimiento a ese amor de madre que siempre tuviste hacia mis hermanos y a mí, fuiste una guía que siempre me cuidaste y orientaste por el camino del bien. Te dedico este pequeño logro a tu recuerdo, nunca comparado con los cuidados y atenciones que tuviste hasta el día de tu partida.*

Pedro Velázquez Trejo †

*En agradecimiento a la disciplina y personalidad que mostrabas siempre y en todo momento, a veces mal comprendida, pero que fue, parte de mi formación como persona. Siempre habrá un gran recuerdo de admiración y cariño.*

*Gracias a Dios por permitirme realizar una de mis metas en esta vida, pero sobretodo por contar con una familia maravillosa que me inculco los valores morales para sobresalir.*

## Agradecimientos

A mis padres Gerónimo Lázaro Martínez y Susana Velázquez Pillado por el esfuerzo, educación y apoyo incondicional realizado para ofrecerme algo tan valioso como lo es una preparación profesional y por alentarme a crecer constantemente como persona.

A mis hermanos Oscar y David por los buenos y malos momentos familiares que hemos pasado.

A la Universidad Nacional Autónoma de México, mi *alma mater* por la excelente preparación profesional y personal que me brindo durante mi estancia en ella.

Al Centro de Ciencias Aplicadas y Desarrollo Tecnológico por la confianza que tiene a los estudiantes para hacerlos partícipes en proyectos de gran relevancia y por ofrecerme un espacio de trabajo durante este tiempo.

A mi director de tesis M.I. Benjamín Valera Orozco, por ser parte fundamental de este logro, por permitirme participar en el presente proyecto de tesis y darme la confianza de aplicar mis conocimientos.

Al M.I. Eric Vilorio López por su gran colaboración y por brindarme el conocimiento necesario para enfrentar este tipo de retos.

A mis compañeros de trabajo en el CCADET: América, Ismael, Enrique y Josué por el gran grupo de trabajo que compartimos.

A mis amigos de toda la vida, por todos los momentos compartidos y en especial por la motivación que me brindaron en los momentos difíciles, porque una mínima acción puede ser una gran ayuda.

## **Reconocimientos**

Al ingeniero Sergio Ochoa Márquez responsable del proyecto en el LAPEM, Irapuato, por todas las facilidades prestadas durante el desarrollo del presente trabajo en las instalaciones del laboratorio de metrología dimensional, así como permitir el uso de los equipos de instrumentación y brindar el tiempo de colaboración con los operadores del área respectiva.

Al Fondo Sectorial de Investigación y Desarrollo Tecnológico en Energía CFE-CONACYT por la beca otorgada durante la realización del presente proyecto.

# Índice temático

<b>Introducción</b>	<b>1</b>
<b>Objetivo</b>	<b>1</b>
<b>Definición del problema</b>	<b>1</b>
Entorno actual	2
Descripción del problema a resolver	2
Relevancia y justificación	2
Alcance y limitaciones	2
<b>Método</b>	<b>3</b>
Esquema general del sistema propuesto	3
<b>Resultados esperados</b>	<b>4</b>
<b>Resumen de la tesis</b>	<b>5</b>
<b>1. Conceptos básicos</b>	<b>7</b>
<b>1.1. Metrología dimensional</b>	<b>7</b>
<b>1.2. Automatización de procesos de calibración</b>	<b>11</b>
1.2.1. Técnicas de calibración	11
1.2.2. LabVIEW	15
1.2.3. Métodos de programación de software	21
<b>1.3. Instrumentación en la calibración de bloques patrón</b>	<b>23</b>
<b>1.4. Instrumentación para la medición de longitud</b>	<b>25</b>
1.4.1. Estación de medición unidimensional	26

---

1.4.2. Codificadores ópticos	27
1.4.3. Sistemas electrónicos para la medición de longitud	29
<b>1.5. Instrumentación complementaria</b>	<b>32</b>
<b>2. Automatización del área de metrología dimensional</b>	<b>35</b>
<b>2.1. Implementación hardware para métodos de calibración dimensional</b>	<b>35</b>
2.1.1. Proceso de calibración de bloques patrón	36
2.1.2. Proceso de calibración para micrómetros e indicadores de carátula	40
2.1.3. Desarrollo de la interfaz electrónica USB	42
<b>2.2. Requisitos de software para métodos de calibración dimensional</b>	<b>50</b>
2.2.1. Requisitos para el procedimiento de calibración de bloques patrón	52
2.2.2. Requisitos para el procedimiento de calibración de micrómetros e indicadores de carátula	56
<b>2.3. Desarrollo de la programación en LabVIEW</b>	<b>61</b>
2.3.1. Programación para el PCMIC	62
2.3.2. Programación para el PCBP	83
<b>3. Resultados y conclusiones</b>	<b>93</b>
<b>3.1. Resultados por etapa</b>	<b>93</b>
3.1.1 Etapa 1	95
3.1.2 Etapa 2	95
3.1.3 Etapa 3	98
3.1.4 Etapa 4	99
<b>3.2. Resultados generales</b>	<b>100</b>

---

<b>3.3. Procedimiento típico para realizar una calibración</b>	<b>102</b>
3.3.1. Software del procedimiento de calibración de micrómetros e indicadores de carátula	102
3.3.2. Software del procedimiento de calibración de bloques patrón	104
<b>3.4. Conclusiones</b>	<b>104</b>
<b>3.5. Trabajo a futuro</b>	<b>105</b>
<b>A. Anexo: Interfaces de comunicación con la computadora</b>	<b>107</b>
<b>A.1. General Purpose Interface Bus (GPIB)</b>	<b>107</b>
A.1.1. Historia	107
A.1.2. Características generales de un cable GPIB	109
A.1.3. Hardware disponible para sistemas GPIB	111
A.1.4. Características de la tarjeta NI PCMCIA-GPIB	111
A.1.5. Software de comunicación GPIB	112
A.1.6. Estructura de sistemas con GPIB	112
<b>A.2. Norma RS-232-C</b>	<b>113</b>
A.2.1. Cable conector	114
A.2.2. Niveles lógicos del RS-232-C	116
A.2.3. Tipo de conexiones	116
A.2.4. Características del convertidor RS-232 a USB de NI	118
<b>A.3. Programación del PIC16C765</b>	<b>119</b>
A.3.1. Configuración del LS7266R1	119
A.3.2. Comunicación USB	123
<b>Bibliografía</b>	<b>125</b>





# Introducción

## Objetivo

Desarrollar el software y hardware para la automatización del proceso de calibración dimensional que se realiza en el Laboratorio de Metrología del LAPEM-CFE<sup>1</sup>. El proyecto contempla la integración de elementos de hardware y software con estándares de comunicación IEEE-488, RS232 o USB que permitan la ejecución del proceso en forma asistida por una computadora personal.

## Definición del problema

Uno de los servicios más solicitados por la industria a los laboratorios de metrología es el de calibración dimensional. La infraestructura que posee el Laboratorio de Metrología del LAPEM-CFE, con elementos como el comparador de bloques patrón y la estación de medición unidimensional, permite realizar dicha tarea. Debido a la importancia del servicio de calibración dimensional, constantemente se ha buscado mejorar su desempeño al introducir elementos que agilicen los procesos de medición.

Se plantea un sistema con base en una PC para automatizar los procesos de captura, procesamiento de lecturas y elaboración de reportes que se realizan como parte de la calibración dimensional usando la infraestructura mencionada.

---

<sup>1</sup> Laboratorio de Pruebas de Equipo y Materiales de la Comisión Federal de Electricidad

## **Entorno actual**

La CFE cuenta con un sistema de laboratorios secundarios de calibración en donde se realizan los servicios de calibración que requieren de una amplia gama de equipos, materiales e instrumentos propiedad de la CFE. En el sistema de laboratorios de la CFE, el Laboratorio de Metrología del LAPEM es el laboratorio primario y encargado, a su vez, de calibrar el equipo patrón de la red secundaria. En este esquema, el Laboratorio de Metrología del LAPEM mantiene los patrones de mayor jerarquía mientras que la red secundaria se encarga de calibrar equipo de uso general.

## **Descripción del problema a resolver**

Debido a la carga de trabajo y al elevado compromiso con la red secundaria de laboratorios, el Laboratorio de Metrología del LAPEM se ve en la imperiosa necesidad de mejorar sustancialmente el tiempo de respuesta y la productividad al realizar calibraciones de los patrones, equipos e instrumentos que llegan a sus instalaciones.

## **Relevancia y justificación**

La relevancia del presente proyecto de tesis radica en la colaboración que se establece entre nuestra institución de educación, la UNAM, y una de las empresas de mayor importancia en el país, la CFE.

Por otra parte, la experiencia y formación al participar en el proyecto se ve incrementada ya que la solución al problema planteado implica la aplicación de una capacidad técnica elevada y un nivel compromiso humano que sólo se ve exigida en proyectos de gran envergadura como el presente.

## **Alcance y limitaciones**

Los procesos de calibración que actualmente se desempeñan en el Laboratorio de Metrología del LAPEM han sido desarrollados por expertos del más alto nivel y su desempeño ha sido probado a lo largo de varios años bajo diferentes circunstancias. Lo anterior limita el presente proyecto a solamente automatizar las componentes de los procesos de calibración que se desempeñan de forma casi manual y que son susceptibles de controlar con una computadora. Lo anterior imposibilita la opción de proponer nuevos procesos de calibración que probablemente impliquen mejoras pero que en contraparte conlleven cierto riesgo y consuman recursos económicos que superen el presupuesto asignado.

Debido a los requisitos especificados por la CFE, se requiere que los procesos automatizados sean programados en el lenguaje LabVIEW, por lo que no existe alternativa alguna de elección. Sin embargo, en situaciones que rebasen las capacidades del software LabVIEW, se contempla la elaboración de código en lenguajes de programación alternativos como Visual C o Visual Basic. Tales situaciones pueden contemplar la implementación de manejadores que no estén disponibles como parte de LabVIEW.

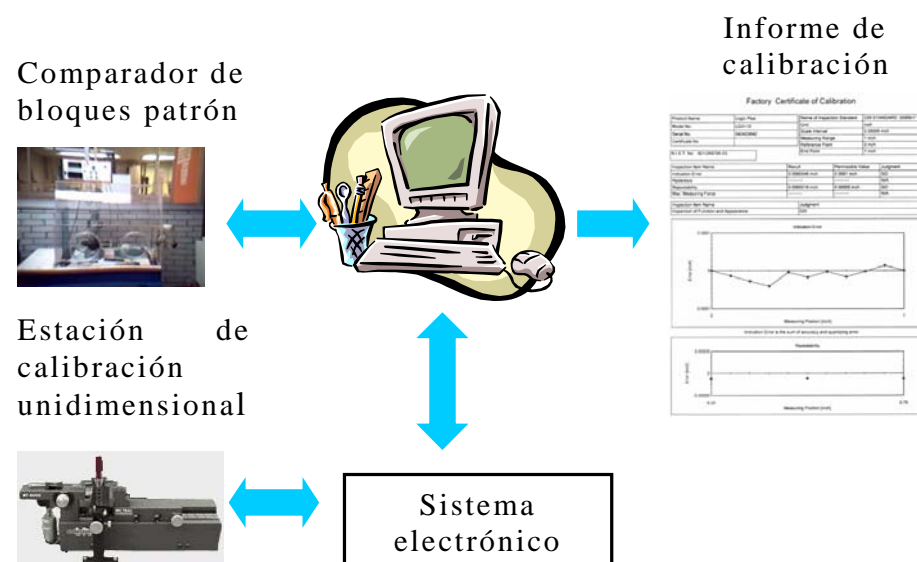
El contenido de tesis también será limitado conforme a los derechos de propiedad intelectual que posee la CFE.

## Método

El presente proyecto de tesis consiste en integrar un sistema automatizado para el control del proceso de calibración dimensional. El sistema deberá incluir una computadora personal, software y un medio estándar de comunicación con el equipo involucrado en la calibración.

### Esquema general del sistema propuesto

En la figura 1, una computadora implementa aquellos procesos susceptibles de automatización como son la adquisición de lecturas, configuración de los instrumentos de medición, cálculos estadísticos y elaboración de informes de calibración.



**Figura 1.** Automatización del proceso de calibración dimensional.

En este esquema se requiere de un medio de comunicación estándar entre los equipos de medición y la computadora personal como pueden ser IEEE-488, USB o RS232. La solución al problema deberá especificar el medio de comunicación idóneo para su compra, instalación y configuración o diseño específico. Por otra parte, en la automatización de los procesos se utilizará el lenguaje de programación LabView debido a que posee capacidades para ejecutar algoritmos de procesamiento de datos, control de procesos y comunicación con equipos de medición.

Adicionalmente, se deberá desarrollar un sistema electrónico para establecer la comunicación entre la estación de calibración unidimensional y una PC, preferentemente por puerto USB. Lo anterior debido a que dicha estación no posee un medio estándar de comunicación que se adapte a los recursos modernos de computación.

## **Resultados esperados**

Implementar un sistema, que incluye hardware y software, para la automatización del proceso de calibración dimensional del Laboratorio de Metrología del LAPEM-CFE, cubriendo las características definidas por sus operadores y buscando mejorar los tiempos involucrados en la captura y transferencia de la información, producto del proceso de calibración.

Las especificaciones del sistema son:

- Reducir el tiempo de calibración manual empleado en ambos procesos del área dimensional.
- Los datos presentados en la pantalla del software deben ser legibles y coincidir con la información generada en ese instante por los instrumentos de calibración.
- Enviar la información de calibración a una hoja de cálculo sin errores de transferencia.
- Facilidad para crear y administrar ordenadamente los reportes de calibración.
- Contar con una interfaz de usuario práctica, eficiente y de fácil manejo para el operador.

## **Resumen de la tesis**

El presente trabajo de tesis consiste en desarrollar un software utilizando LabVIEW 8 para lograr la automatización de los métodos de captura, procesamiento de lecturas de medición y elaboración de reportes que se necesitan en los procedimientos de calibración con el comparador de bloques patrón y la estación de medición unidimensional que se realizan en el Laboratorio de Metrología del LAPEM-CFE.

El capítulo 1 presenta la importancia de los procedimientos e instrumentos de calibración dimensional. Engloba conocimientos básicos en materia de Metrología, Ingeniería de Programación y Electrónica, con el fin de establecer una base teórica orientada a la solución de las necesidades del LAPEM. Se presenta una breve descripción de la instrumentación virtual con LabVIEW así como sus principales características en tareas de automatización.

En el capítulo 2 se presentan los requisitos de hardware y software para cada estación de calibración, así mismo se desarrolla la solución de automatización en los procedimientos de calibración utilizando sistemas de comunicación GPIB (IEEE-488), RS-232 o USB que permiten la transferencia de información de instrumentos de calibración y medición hacia una computadora utilizando software de diseño específico en LabVIEW.

El capítulo 3 presenta los alcances y logros de los procedimientos automatizados con base en la fiabilidad de las mediciones, captura de datos y generación de reportes, así como mostrar el grado de optimización que se obtuvo con el uso de todos los recursos disponibles, que se ve reflejado en una disminución de tiempo en el servicio de calibración.



# Capítulo 1

## Conceptos básicos

El presente capítulo presenta las bases teóricas utilizadas en el desarrollo del presente trabajo de tesis. Engloba conocimientos de áreas como la Metrología, Ingeniería de Programación y Electrónica. Describe adecuadamente la instrumentación utilizada en los procesos de calibración dimensional. Justifica la implementación del software LabVIEW 8 como herramienta de programación para la ejecución de algoritmos en procesamiento de datos, control de procesos y comunicación con equipos de medición.

### 1.1. Metrología dimensional

Es importante hacer una breve presentación de los términos y conceptos que se manejan en el área de metrología para analizar adecuadamente las actividades de medición y calibración en laboratorios dedicados a prestar este tipo de servicio [2].

Medir es la comparación de una determinada magnitud con otra de su misma especie en un determinado sistema de unidades; se compone de un valor numérico, una unidad y una incertidumbre.

Una magnitud se define como el atributo de un cuerpo susceptible de ser diferenciado cualitativamente y determinado cuantitativamente. La magnitud que se quiere medir es conocida como mensurado.

La unidad de medida para la longitud en el Sistema Internacional de Unidades es el metro.

La incertidumbre hace referencia a la repetibilidad de los resultados entre mediciones sucesivas del mensurado bajo algunas condiciones, denominadas de reproducibilidad. Estima un intervalo de valores en el que se sitúa el valor verdadero de una magnitud con un alto grado de probabilidad.

Las condiciones de reproducibilidad establecen que la medición debe hacerse al mismo mensurado; por el mismo operador, instrumento y procedimiento en el mismo lugar, en igualdad de condiciones y con pequeños intervalos de tiempo entre mediciones sucesivas [1].

El valor verdadero en un proceso de medición es aquel que se determinaría en un proceso de medición perfecto, el cual no existe y que por convenio se le asocia a una magnitud con un grado de incertidumbre apropiado para su uso. La incertidumbre es parte importante en la medición, el no indicarla no permite la comparación con otras mediciones de la misma magnitud y mucho menos con el valor verdadero establecido por convención [6].

El término metrología dimensional designa al campo de conocimientos para la medición de todas aquellas propiedades mecánicas en un cuerpo, determinadas por la magnitud de longitud. Forma parte de la metrología mecánica y utiliza patrones nacionales e internacionales de magnitud.

La metrología dimensional tiene una gran importancia dentro de la industria de manufactura en los procesos de medición dimensionales donde se determina las magnitudes lineales y angulares de piezas industriales. La exactitud y las tolerancias de las magnitudes dimensionales determinarán la efectividad y competitividad de cada empresa en su producto final.

Toda empresa tiene equipo de medición de longitud, el cual es parte indispensable dentro de un proceso de producción. Las mediciones dimensionales aseguran y verifican el cumplimiento del producto con respecto a las especificaciones de diseño. Alcanzar los estándares de calidad implica contar con instrumentos de medición calibrados por instituciones especializadas en la materia de metrología que avalen la fiabilidad de la medición.

La calibración es un conjunto de operaciones que tiene por finalidad determinar los errores de un instrumento de medición, el cual puede fungir como patrón. Se deben detectar las posibles fuentes de error para:



- Determinar las variables que afecten la incertidumbre del instrumento.
- Medición del error aportado por cada una de las variables.
- Determinar el efecto total de todas las fuentes de error y evaluar la capacidad de medición del instrumento.

Dentro de la metrología se define como patrón al instrumento de medición destinado a definir, conservar o reproducir la unidad de medida de una magnitud para transmitirla por comparación a otros instrumentos de medición.

La trazabilidad es una propiedad de interrelación entre patrones de referencia, patrones de trabajo e instrumental de medición común, los cuales tienen diferentes órdenes de precisión en la medición de una magnitud determinada. Las incertidumbres correspondientes entre estos determinan el grado de la sucesión. Para determinar la trazabilidad de una calibración es importante tomar en cuenta:

- El patrón utilizado de calibración a su vez debe de estar calibrado por un patrón de mayor exactitud, hasta llegar al mayor grado de exactitud del patrón nacional aceptado.
- La frecuencia de una calibración debe establecer una confianza razonable en el que se garantice que el valor del patrón no se moverá fuera de los límites permitidos entre calibraciones sucesivas.
- La calibración de un instrumento es válida por su desempeño y sólo en el tiempo especificado de calibración, extendido por el prestador del servicio.

Los patrones son definidos por su trazabilidad en:

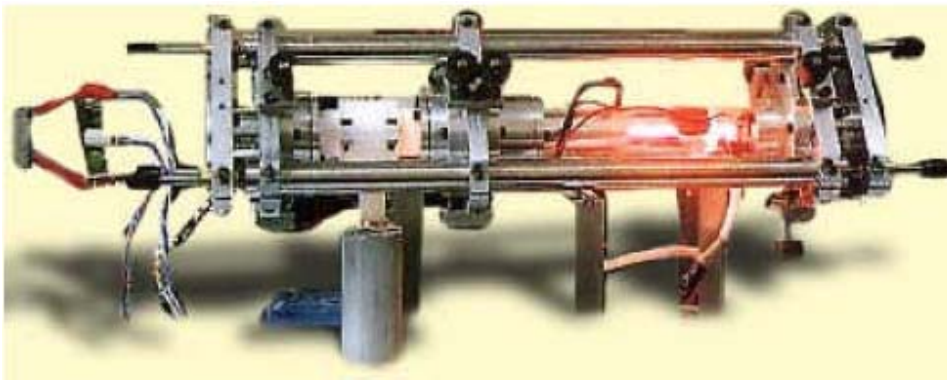
- Patrón primario.
- Patrón secundario.

El patrón primario es reconocido como el valor aceptado de una magnitud sin la necesidad de referirse a otros patrones. Un patrón secundario se establece por la comparación entre un patrón primario de la misma unidad.

Existen laboratorios certificados donde el personal, equipo, procedimientos y la validez de los patrones de determinada unidad cumplen con la trazabilidad, los cuales prestan el servicio de expedir certificados de calibración a sus usuarios, indicando la incertidumbre y el nivel de confianza [7].

Las actividades en México sobre metrología están reguladas en la Ley Federal Sobre Metrología y Normalización, en esta se establece la creación del Centro Nacional de Metrología (CENAM) como el organismo de mayor nivel técnico en la materia y como regulador de las demás materias relativas a la metrología. El CENAM establece como patrón nacional de longitud el metro, cuya materialización esta realizada con un experimento de láser estabilizado en frecuencia, ilustrado en la figura 1.1.

El funcionamiento del laser consiste en una espectroscopia saturada de la molécula de yodo  $^{127}\text{I}_2$ , operando a una longitud de onda de  $\lambda = 632, 991\ 398\ 22\ \text{nm}$  [3].



**Figura 1.1.** Láser estabilizado en frecuencia propiedad del CENAM.

El CENAM cuenta con tres instrumentos de este tipo para mantener entre ellos la frecuencia de referencia y la longitud de onda, mediante calibraciones periódicas se conforma el patrón nacional de longitud.

Como órgano superior en materia de metrología es su deber diseminar la unidad de longitud. En lo que se denomina cadena de trazabilidad. A partir del patrón nacional se realizan calibraciones con láseres He-Ne a 633 nm, de menor estabilidad, pero permitiendo una alta exactitud en su valor. A su vez estos láseres realizan mediciones interferométricas de bloques patrón de referencia, con alta exactitud, que sucesivamente diseminan la unidad de longitud a bloques patrón de calibración de menor exactitud, los cuales por comparación se encargan de calibrar equipo cotidiano de medición industrial para desarrollar actividades de diseño en un producto [4].

La cadena de trazabilidad y diseminación de la unidad de longitud posee diferentes grados de incertidumbre, teniendo el menor valor el láser estabilizado en Yodo, y aumentando en valor al diseminar la unidad de longitud como lo muestra la figura 1.2.

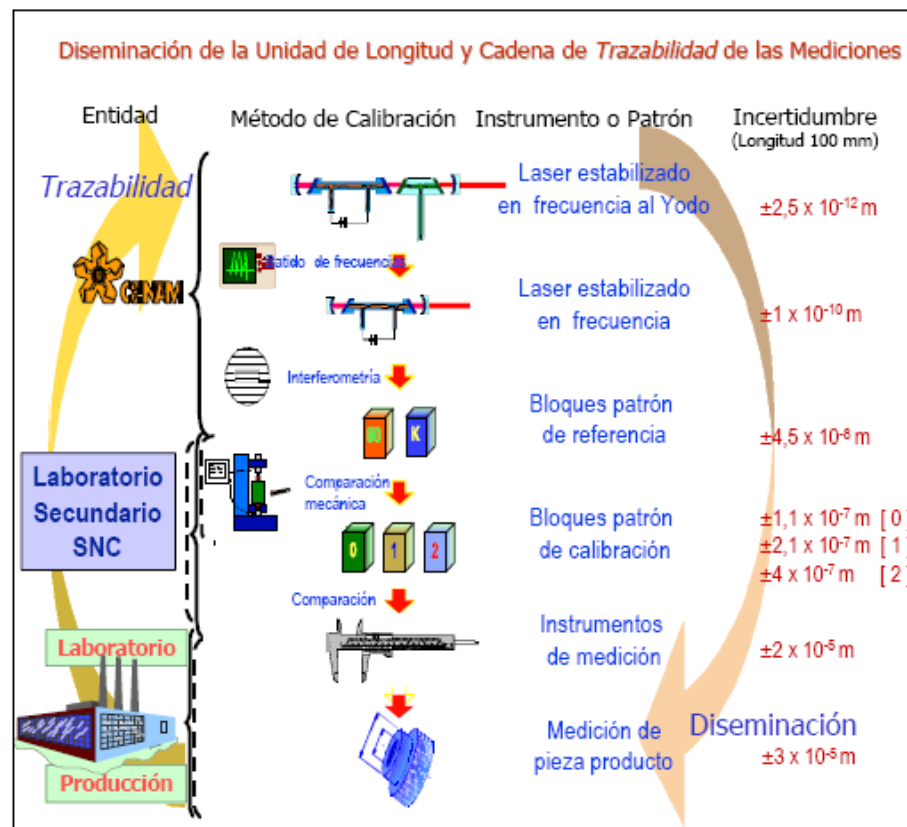


Figura 1.2. Trazabilidad y diseminación de la unidad de longitud.

## 1.2. Automatización de procesos de calibración

Una vez sentadas las bases en materia de metrología, se profundiza en los métodos de calibración aplicados en los laboratorios de calibración dimensional, el planteamiento de la estructura de desarrollo para la automatización, así como una descripción del software LabVIEW 8 y su funcionalidad.

### 1.2.1. Técnicas de calibración

Las actividades de calibración más comunes en el área de calibración de longitud son las siguientes:

- 1) Calibración de bloques patrón.
- 2) Calibración de micrómetros para medición de exteriores.
- 3) Calibración de micrómetros para medición de interiores.
- 4) Calibración de micrómetros para medición de interiores con extensiones.

- 5) Calibración de micrómetros para medición de profundidades.
- 6) Calibración de micrómetros para medición de profundidades con extensiones.
- 7) Calibración de indicadores de carátula.

Un procedimiento es un conjunto de operaciones descritas específicamente para realizar mediciones particulares de un instrumento de acuerdo a un método dado, el cual proporciona suficientes detalles para que un operador pueda realizar una medición sin necesitar más información.

Los métodos de medición dentro de un procedimiento pueden ser clasificados en:

- Método de Substitución.
- Método diferencial.
- Método nulo o cero.

De los métodos citados anteriormente será objeto de interés el método diferencial.

El método diferencial consiste en aquellas mediciones que se realizan por comparación contra un patrón o instrumento de jerarquía metrológica superior, este término implica una exactitud mayor y una incertidumbre menor. Para el caso de la calibración de bloques patrón e instrumentos como micrómetros e indicadores de carátula se aplica este método [5].

Dentro del método diferencial se considera a la temperatura como una variable de influencia considerable. La temperatura de referencia normalizada es de 20°C, y en el laboratorio se deberá garantizar una temperatura de  $\pm 1^\circ\text{C}$  en la zona de trabajo durante todo el día. Se deberá contar con instrumentación adicional con el objetivo de sensar la temperatura con al menos una resolución de 0.5°C y elaborar un registro de varias lecturas por día. En caso de deshabilitar la climatización del lugar de trabajo en días no laborables se deberá especificar el tiempo de estabilización necesario para iniciar una calibración, este será no menor a dos horas [14].

Se requiere una resolución de 0.01°C en el control de temperatura del lugar y contar con termómetros de resolución de 0.1°C para registrar la temperatura del patrón y del equipo bajo calibración.

La humedad debe de mantenerse dentro de un intervalo de 45% a 65% de humedad relativa y contar con instrumentación encargada de monitorear esta variable. El sistema regulador de temperatura debe contar con medios para filtrar el aire, para eliminar partículas de polvo que influyan en la exactitud de las mediciones y calibraciones.

Se debe contar con balances de incertidumbre documentados en cada uno de los servicios de calibración. En caso de patrones e instrumentos de alta exactitud se debe contar con un modelo matemático para el cálculo de incertidumbre.

Los micrómetros e indicadores de carátula son instrumentos de media y baja exactitud; para el caso de bloques patrón se consideran como de alta exactitud.

Cabe destacar que los métodos del cálculo de incertidumbre son modelos matemáticos con derechos intelectuales y por razones de confidencialidad no son ampliamente difundidos.

Para hacer un manejo más acorde en la solución que se presenta en este trabajo de tesis, se decidió abordar dos de los procedimientos más representativos y susceptibles de automatización:

- Procedimiento para la calibración de bloques patrón.
- Procedimiento para la calibración de micrómetros e indicadores de carátula.

Ambos procedimientos de calibración se realizan en el Laboratorio de Metrología del LAPEM.

### **Procedimiento para la calibración de bloques patrón**

La figura 1.3 muestra un juego de bloques patrón, construidos en acero, cada bloque determina un valor nominal de longitud. La utilidad de este elemento es para la calibración de instrumentos de menor exactitud como son vernieres y micrómetros. Estos, son utilizados cotidianamente en la industria de manufactura y laboratorios de calibración [8].



**Figura 1.3.** Juego de bloques patrón.

La calibración de bloques patrón establece una relación entre dos valores. Una magnitud indicada por un instrumento de medición denominado comparador de bloques patrón, adquirida mediante un sistema de comparación mecánica sobre un bloque patrón. La otra magnitud es el valor de longitud de otro bloque patrón que actúa como referencia a cero y cuya desviación en la longitud central es conocida mediante certificado de calibración.

De esta forma se pueden realizar comparaciones sobre distintos puntos en la superficie de la cara lateral del bloque bajo calibración. Se emite un criterio sobre el bloque bajo calibración en el que se le considera apto o no para seguir calibrando instrumentos de medición cotidianos, o en su defecto el rango de error sobre el valor nominal de longitud especificado por el fabricante.

En la sección 1.3 se describe a fondo las características principales del comparador de bloques patrón.

### **Procedimiento para la calibración de micrómetros e indicadores de carátula**

Para el caso del segundo procedimiento, se debe tener desarrollado un método de calibración para cada tipo de micrómetro. En la práctica el procedimiento implica comparar longitud de instrumentos cotidianos con la estación de medición unidimensional [10, 11].

La máquina de medición cuenta con un desplazamiento horizontal de aproximadamente 34 pulg. Sobre esta se colocan accesorios mecánicos que permiten el montaje de diversos tipos de micrómetros e indicadores de carátula. Es necesario mover el punto de referencia cero sobre el eje horizontal a un valor determinado en la escala del instrumento, como el procedimiento de calibración lo establece [9].

La calibración consiste en comparar el valor desplegado en la computadora de la estación de medición de desplazamiento unidimensional, con el valor que presenta el instrumento bajo calibración en su escala de lectura; de esta manera se tiene un criterio de comparación y se determina si el instrumento esta en las condiciones apropiadas para mediciones de carácter industrial.

Comúnmente, para realizar desplazamientos inferiores a una pulgada se cuenta con un ajuste fino, mientras que para desplazamientos superiores existe un control burdo.

En la sección 1.4.1 se describe a fondo las características principales de la estación de medición unidimensional.

### **1.2.2. LabVIEW**

Se plantea la justificación del uso de este software para la solución del problema con base al objetivo de diseñar sistemas de instrumentación utilizando los recursos de una computadora.

Un instrumento virtual es una representación de un sistema de instrumentación tradicional sobre la base del aprovechamiento del hardware de una computadora, en el cual se centra un software que pueda aprovechar los recursos de la misma, tales como hojas de cálculo, productividad, exhibición de datos y capacidad de conexión. Permitiendo la construcción de instrumentos de medición y automatización con características definidas por el usuario, lo cual rompe las limitantes de los instrumentos tradicionales con funciones predeterminadas o que requieren la necesidad de acoplar accesorios adicionales de comunicación. Requiere además del procesador de la computadora, puertos de comunicación y módulos de adquisición de datos.

Un software con el cual se puede desarrollar un instrumento virtual es LabVIEW (Laboratory Virtual Engeneering Workbench) que a diferencia de los programas tradicionales con líneas de código, emplea la programación gráfica o lenguaje G para la creación de programas en diagramas de bloques.

LabVIEW es parte integral de la instrumentación virtual dado que puede crear eficientemente sus propias aplicaciones, diseñando e integrando las rutinas que requiere un proceso en particular. También desarrolla la interfaz de usuario que mejor satisfaga el objetivo de la aplicación y del personal que va a interactuar con ellas. Puede definir la forma y el tiempo en que se tenga que adquirir datos desde

cualquier dispositivo, la manera como los procesa, manipula y almacena, hasta como presentar los resultados finales al usuario [22].

Las aplicaciones desarrolladas en LabVIEW están dirigidas a tareas orientadas en:

- Adquisición de datos.
- Control de instrumentos.
- Automatización industrial.

Presenta facilidades para el manejo de:

- Interfaz de comunicaciones como puerto serie, puerto paralelo, GPIB, USB, entre otras.
- Herramientas para el procesamiento digital de señales.
- Visualización y manejo de gráficas con datos dinámicos.
- Tiempo Real.
- Transporte de datos a programas de terceros.

Su entorno es compatible con lenguajes tradicionales de programación. Permite procesar lenguajes de programación como C o Basic y utilizar código heredado como DLLs e integrar software usando controles ActiveX.

Tiene la capacidad de integrar cualquier hardware por lo que se puede configurar y usar rápidamente cualquier dispositivo de medición autónomo. Cuenta con más de 1000 librerías compatibles con instrumentos de diversos fabricantes.

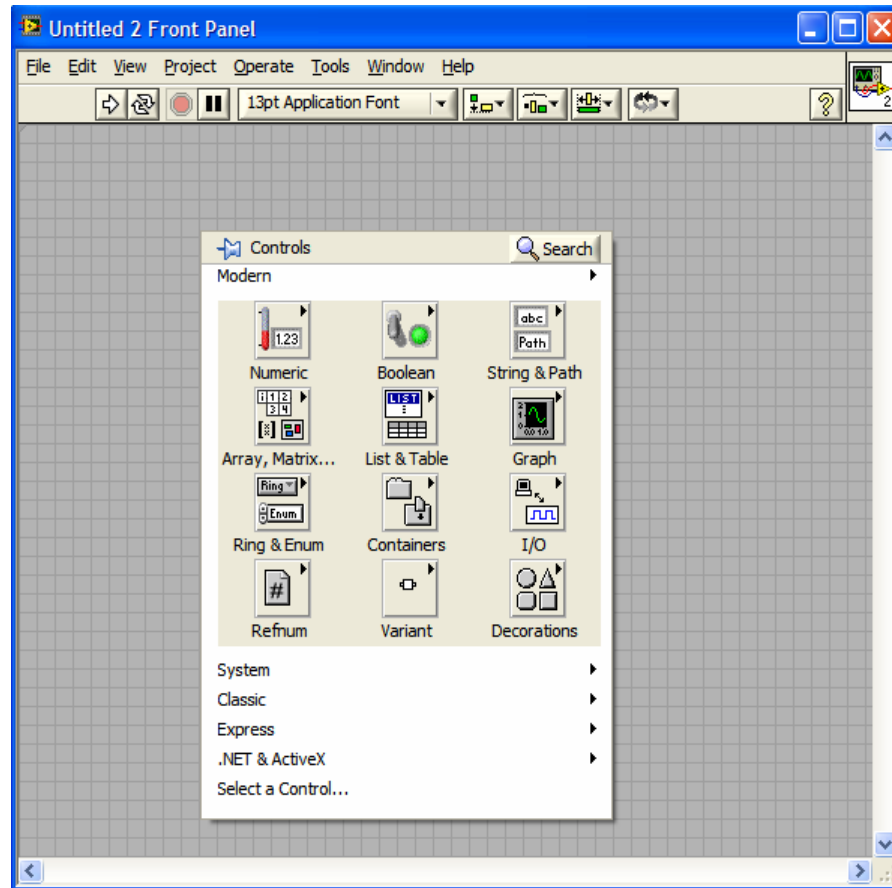
Las principales ventajas que se obtienen al utilizar LabVIEW son las siguientes:

- El tiempo de desarrollo en las aplicaciones se reduce sustancialmente debido a la programación gráfica.
- El programa final puede ser actualizado constantemente, tanto en software como en hardware.
- Permite la creación de sistemas que integran las funciones de adquisición, análisis y presentación de datos.
- Cuenta con un compilador gráfico que permite la ejecución de una aplicación en el menor tiempo posible.

Un programa en LabVIEW recibe el nombre de Instrumento Virtual (VI) por sus siglas en inglés. Un VI tiene la apariencia y funcionamiento de un instrumento real con el cual se puede interactuar

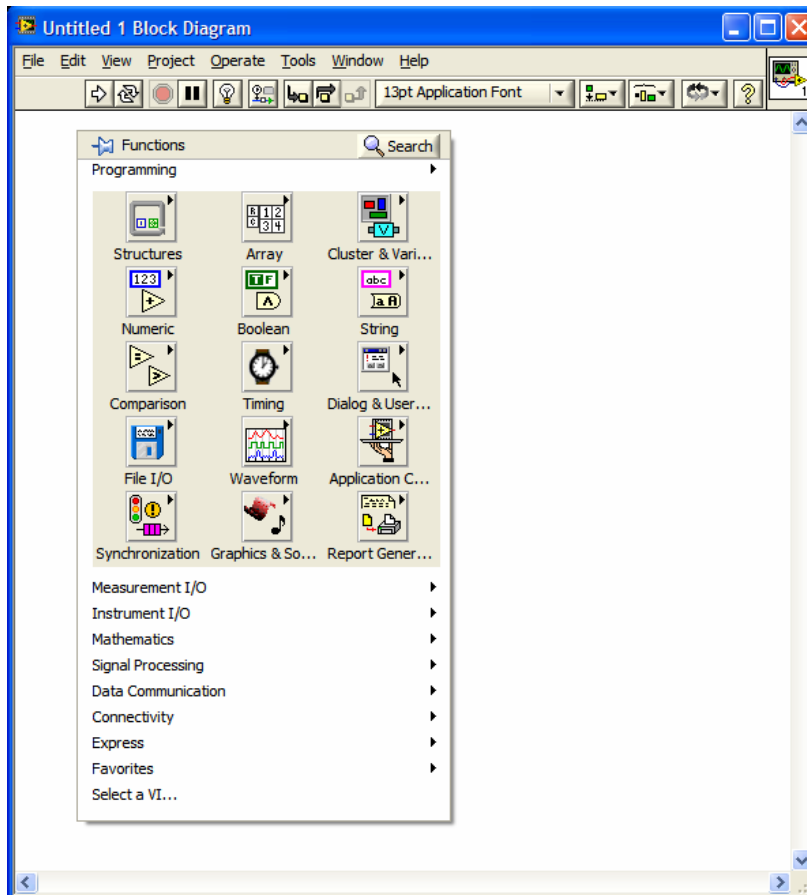


por algún medio de comunicación de una PC. La plataforma de diseño en LabVIEW muestra dos pantallas: *Panel frontal* ilustrado en la figura 1.4 y *Diagrama de bloques* en la figura 1.5.



**Figura 1.4.** Panel frontal en LabVIEW.

El panel frontal es la interfaz gráfica del VI con el programador. Se encuentran numerosas herramientas de presentación como botones, interruptores, gráficas y desplazadores como se puede ver en figura 1.6. Cada una de estas herramientas están dirigidas a capturar las entradas procedentes del usuario o realizar la función de mostrar las salidas o resultados de la ejecución del programa. Para el desarrollo de estas tareas se dispone de una biblioteca de controles e indicadores de diversos tipos que actúan como entradas y salidas respectivamente.



**Figura 1.5.** Pantalla del diagrama de bloques.

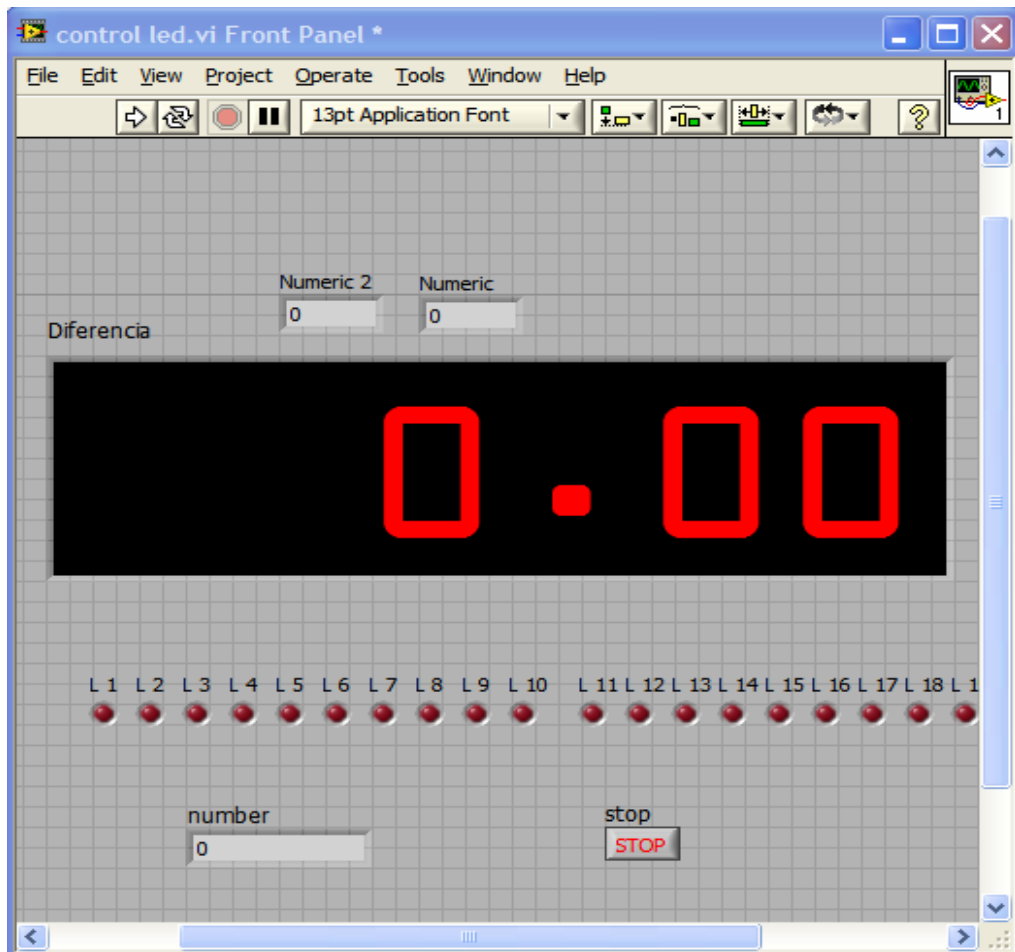
Cuando se agrega un control desde la biblioteca al panel frontal, se crea una variable cuyos valores son ajustados por el usuario.

Retomando la característica de programación gráfica, esta consiste en bloques gráficos programados con una tarea determinada, donde el usuario hace uso de estos elementos para crear sistemas completos del grado de complejidad que se requiera.

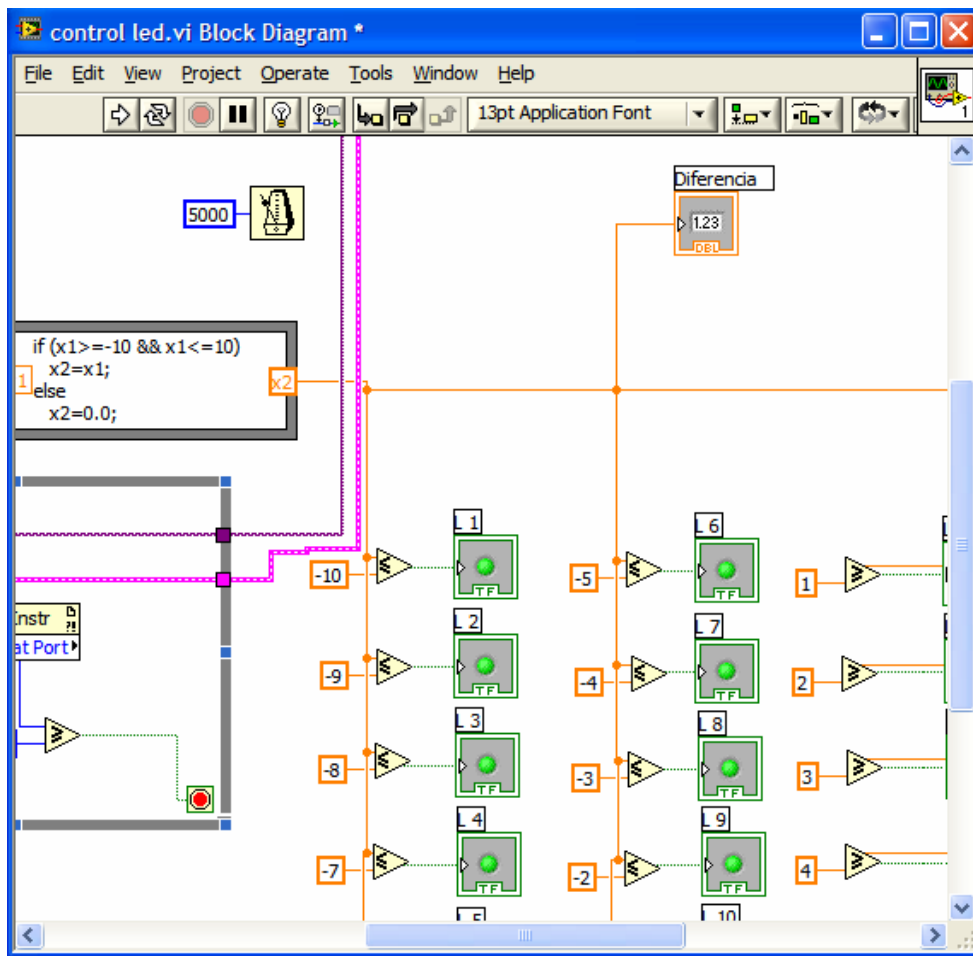
La pantalla de diagrama de bloques es el código fuente construido con lenguaje G donde se define su funcionalidad utilizando los bloques disponibles. Estos son instrumentos de bajo nivel, funciones, constantes y estructuras de control. Los controles e indicadores colocados en el panel frontal se materializan en el diagrama de bloques mediante terminales de entrada o salida.

Para construir un programa se deben unir todos los bloques por medio de cables para hacer fluir los datos coherentemente hacia una función, estructura, indicador o cualquier tarea implementada por el diseñador del instrumento virtual como se muestra en la figura 1.7.

Las conexiones en LabVIEW solo se pueden dar entre bloques que representan el mismo tipo de dato específico que se maneja como entrada al VI. El sentido de flujo del programa es de izquierda a derecha en el diagrama de bloques y se determina por las operaciones o funciones que se realizan.



**Figura 1.6.** Interfaz gráfica desarrollada en LabVIEW.



**Figura 1.7.** Elementos del diagrama de bloques en un VI.

Cuenta con estructuras de declaraciones causales que ejecutan cierta parte del código en forma condicional o repetitiva; bucles for, while, case, etc.

Una cómoda herramienta de programación es la creación de SubVI, que implementa subrutinas con la propiedad de ser heredadas en otra parte de la aplicación dentro de un VI general. El usuario define el número de entradas y salidas del SubVI, puede editar un nuevo bloque funcional y editarlo con una figura reconocible por él.

Para el control de instrumentación mediante la computadora, LabVIEW incorpora en sus bibliotecas muchos bloques de aplicaciones con protocolos de comunicación IEEE 488 referentes a la interfaz GPIB (General Purpose Interface Bus). Debido al avance en materia de procesadores de 16 y 32 bits, se ha podido incorporar equipos por medio de GPIB consiguiendo altas velocidades y grandes capacidades de almacenamiento.

Así mismo tiene un importante apartado de funciones para puerto serie RS-232 y USB, lo que abre las posibilidades de comunicación y desarrollo de hardware libre con microcontroladores que cuenten con alguno de estos recursos de comunicación [17].

### 1.2.3. Métodos de programación de software

Es necesario introducir una base teórica que sustente la creación del software, conocer los métodos de su generación, planificación, documentación y la forma correcta de desarrollo, con el objetivo de establecer el orden y estimar los tiempos de trabajo. Considerando que se necesita recopilar información sobre los procedimientos que se realizan en el LAPEM, además de promover reuniones con el personal técnico y depurar con ellos las funciones requerida por el sistema.

El software es la suma total de los programas de computadora, procedimientos, reglas, documentación asociada y los datos que pertenecen a un sistema de cómputo. La *Ingeniería de Software* es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro; en el que se establecen las bases para la generación de nuevo software.

Un software puede ser un producto comercial o de desarrollo específico. Debe contar con las siguientes características.

- **Mantenibles.** Capacidad de actualizarse a las nuevas innovaciones tecnológicas.
- **Confiabilidad.** No debe ser fuente de daños físicos o económicos en caso de fallos.
- **Eficiencia.** No desperdiciar los recursos del sistema.
- **Utilización adecuada.** Debe contar con una interfaz de usuario que le permita acceder fácilmente a las aplicaciones.

El proceso de ingeniería de software se define como un conjunto de etapas parcialmente ordenadas con la intención de lograr un software de calidad. Abarca un conjunto de tres elementos clave: métodos, herramientas y procedimientos. Facilitando el control del proceso de desarrollo de software.

- **Métodos.** Indican la construcción del software abarcando las etapas de planificación y estimación de proyectos, análisis de los requerimientos del sistema, arquitectura del programa, algoritmos, codificación, pruebas y actualizaciones.
- **Herramientas.** Sobre las cuales se suministre soporte automático a los métodos a realizar.

- Procedimientos. Definen la secuencia de aplicación de métodos requeridos como son documentos, informes o formas que aseguren la calidad y coordinen los cambios y las directrices del proyecto en creación.

El objetivo es analizar las necesidades del usuario y traducirlas en requerimientos de software, estos requerimientos transformarlos en diseño y a su vez implementarlo en código, el cual será evaluado, documentado y certificado para uso operativo.

Es necesario seguir un modelo de desarrollo de software que más se adecue a los requerimientos del programador. Se eligió seguir el método de construcción de prototipos por ser el más adecuado a los requerimientos y formas del equipo de trabajo.

La construcción de prototipos es un proceso que facilita al programador la creación de un modelo del software que puede tomar algunas de estas formas:

- 1) Un modelo con base en PC que describa la interacción hombre maquina, de manera que facilite al usuario la comprensión de cómo se producirá tal interacción.
- 2) Un prototipo que implemente algunos subconjuntos de la función total requerida por el usuario.
- 3) Un programa existente que ejecute parte o toda la función deseada, pero que tenga la característica para soportar actualizaciones.

Esta metodología comienza con el diseño a partir de especificaciones claramente planteadas por el usuario, iniciando un intercambio de ideas y propuestas hasta evolucionar a un sistema final.

LabVIEW se acopla adecuadamente a este modelo de desarrollo debido a que se pueden crear rápidamente SubVI, que realicen determinadas funciones para después acoplarse y crear un sistema completo [20].

La construcción de prototipos comienza con la recolección de los requisitos. El programador y el usuario se reúnen y definen los objetivos globales para el software, identifican todos los requisitos conocidos y perfilan las áreas en donde será necesaria una mayor definición.

Se fija un plazo corto donde tendrá lugar un diseño rápido. Tal diseño se enfocará sobre la representación de los aspectos del software visibles al usuario y conducirán a la construcción del prototipo. El

prototipo es evaluado por el cliente usuario y se utiliza para refinar los requisitos del software a desarrollar.

Se produce entonces un proceso interactivo en el prototipo de afinación para satisfacer las demandas del cliente, al mismo tiempo que facilita al desarrollador la comprensión de lo que se solicita.

Idealmente el prototipo sirve como mecanismo para identificar los requisitos del software. Pero a su vez es susceptible de ciertas omisiones o eventos propiciados por la interacción con el usuario. Se debe de tomar en cuenta aspectos de mantenimiento a largo plazo y mejorar los métodos de programación entre cada prototipo.

Sustentada en la Ingeniería de Software se ha presentado el modelo a seguir en el diseño del software, es muy adecuado considerando que tanto los procedimientos de calibración como los equipos adicionales de instrumentación no eran conocidos en un inicio. El modelo de construcción de prototipos permite reuniones con los usuarios para conocer los requerimientos, presentar una propuesta a las primeras demandas y posteriormente agregar mejoras para llegar a un software que abarque todos los requerimientos que se esperan por parte del solicitante [25].

### **1.3. Instrumentación en la calibración de bloques patrón**

El comparador de bloques patrón es el instrumento central en el procedimiento, gran parte del método depende de su correcta operación.

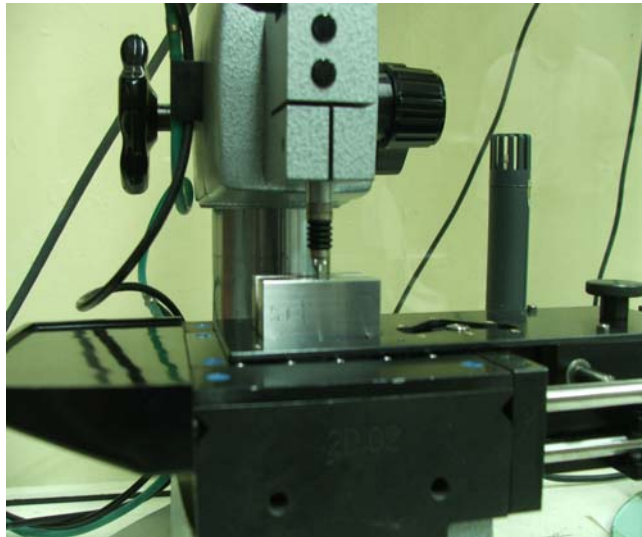
Las mediciones realizadas con este equipo determinan el grado de calibración de un bloque patrón.

En la sección 1.5 se hablará de la instrumentación complementaria destinada a la medición de temperatura y humedad, así como las características de los equipos.

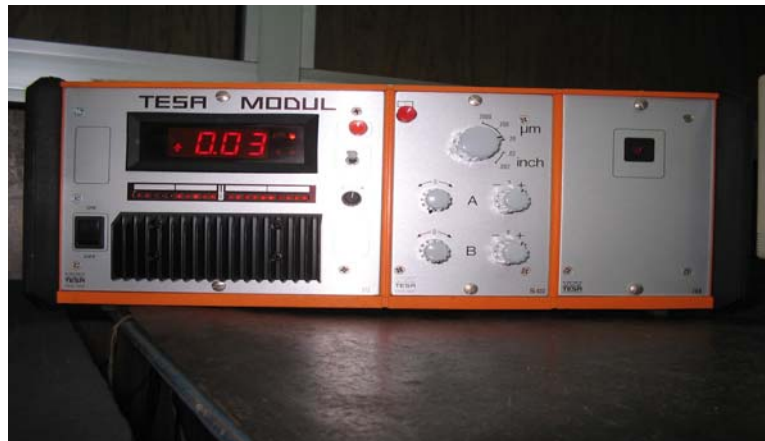
#### **Comparador de bloques patrón**

Generalmente un sistema comparador de bloques patrón opera en bloques de sección rectangular en un rango de longitud nominal de 0.5 a 100 mm. Consta de un comparador mecánico, mostrado en la figura 1.8; y una unidad central de procesamiento ilustrada en la figura 1.9. La unidad central de procesamiento visualiza la medición del comparador mecánico.

El comparador mecánico se conforma por dos palpadores, inferior y superior, colocados sobre un poste metálico de desplazamiento vertical fino o burdo. El palpador inferior se encuentra fijo, mientras que el palpador superior puede desplazarse verticalmente sobre el poste para aproximarse lo más posible a la superficie del bloque patrón.



**Figura 1.8.** Comparador mecánico.



**Figura 1.9.** Unidad central de procesamiento.

Sujeto a la base del poste se encuentra una plantilla donde se coloca el bloque patrón y el bloque bajo calibración. La plantilla es deslizable sobre la base, y permite el contacto superior e inferior de los palpadores sobre las caras del bloque.

Tiene seis puntos de movilidad. El primer punto de movilidad es sobre el centro del bloque patrón. Los demás puntos movibles actúan



sobre el bloque bajo calibración, e incluyen el centro y las esquinas de la superficie.

La unidad central de procesamiento transduce la señal de desplazamiento mecánico, ocurrido en los palpadores, en un dato de longitud observable en un despliegue electrónico con diferentes escalas en  $\mu\text{m}$ .

Cada señal en los palpadores es adquirida por medio de los canales A y B. Físicamente se tienen perillas de control para ajustar la referencia a cero del canal A y B. El equipo se complementa con una bomba de vacío que es controlada por un botón de accionamiento. Al accionar el botón, el palpador desciende hacia el bloque, despliega una lectura en pantalla, un segundo accionamiento ocasiona que el palpador se levante; además de que el puerto de salida serie envíe un dato de salida.

Los palpadores y el botón de accionamiento están conectados a la unidad central. También éste cuenta con puerto serie RS-232 de comunicación y un puerto adicional como lo muestra la figura 1.10.



**Figura 1.10.** Puertos de comunicación de la unidad central de procesamiento.

#### **1.4. Instrumentación para la medición de longitud**

El instrumento principal es la estación de medición unidimensional. Esta orientada a calibrar diferentes instrumentos que comúnmente se ocupan en la industria. No importando el tipo de micrómetro o indicador bajo calibración, su función será la medición de longitud. Presenta un problema de incompatibilidad con las nuevas tendencias de comunicación. Se describirá a fondo su modo de comunicación actual para establecer la viabilidad de una nueva interfaz USB.

### 1.4.1. Estación de medición unidimensional

La figura 1.11 muestra la estación de medición unidimensional, mide desplazamiento horizontal en mm o en pulgadas. El instrumento es montado sobre el banco deslizable de la estación, la cual tiene sensores en cuadratura que transducen el desplazamiento en un valor numérico de longitud.



**Figura 1.11.** Estación de medición unidimensional.

La señal de los sensores llega a una tarjeta de adquisición ilustrada en figura 1.12, la cual ya no es compatible con los equipos de cómputo actuales.



**Figura 1.12.** Tarjeta de adquisición de datos.

La figura 1.13 muestra la tarjeta de adquisición montada en un puerto ISA de la PC, la cual tiene interfaz de comunicación DB-50 y opera como punto de conexión para concentrar la información del

desplazamiento de los ejes de la estación de medición unidimensional por medio de otros conectores; DB9 para el control burdo de desplazamientos mayores a una pulgada y cable DIN 7 para el control fino de desplazamientos inferiores a una pulgada.



**Figura 1.13.** Interfaz de comunicación para tarjeta de adquisición.

El fabricante de la estación de medición unidimensional, proporciona al comprador, software adicional para la visualización numérica de las mediciones y funciones especiales que demanda el procedimiento de calibración. La toma de resultados es 100% manual a través de la pantalla como se puede ver en la figura 1.14.



**Figura 1.14.** Visualización numérica de desplazamiento en la estación unidimensional.

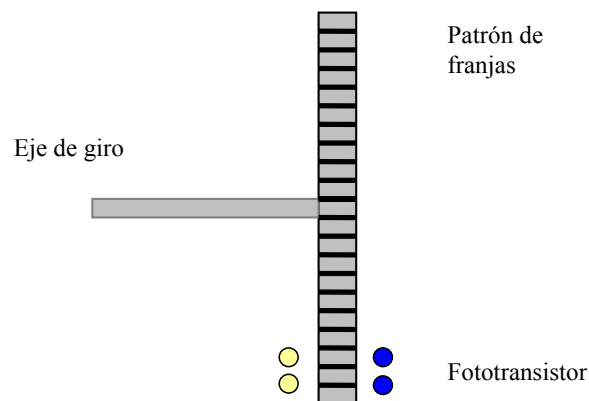
#### **1.4.2. Codificadores ópticos**

La estación de medición unidimensional incluye una interfaz que hace posible mostrar el desplazamiento longitudinal de sus ejes mecánicos y ser observados numéricamente por medio de un software de desarrollo industrial.

Parte del desarrollo del presente proyecto de tesis se centra en transducir el desplazamiento de la máquina de medición unidimensional, para lo cual utiliza sensores en cuadratura posicionados en los ejes de desplazamiento de la máquina.

### Principio de operación

La figura 1.15 representa un codificador óptico. Este consiste en un cilindro giratorio con una distribución de series de franjas oscuras y claras sobre la superficie, al girar se interpondrá entre los emisores y los fototransistores generando una señal en cuadratura.

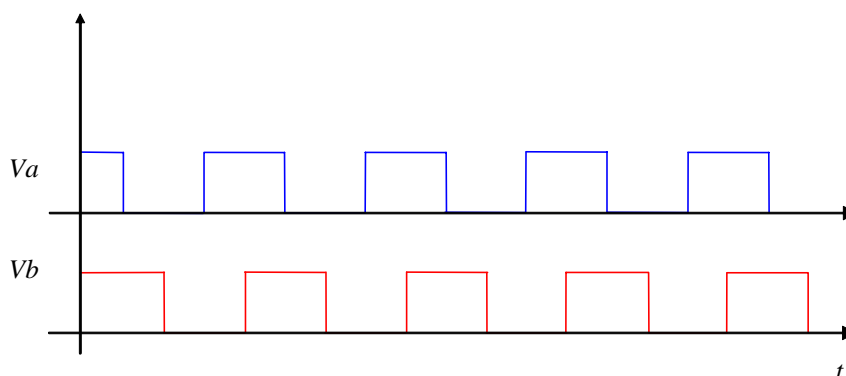


**Figura 1.15.** Codificador en cuadratura.

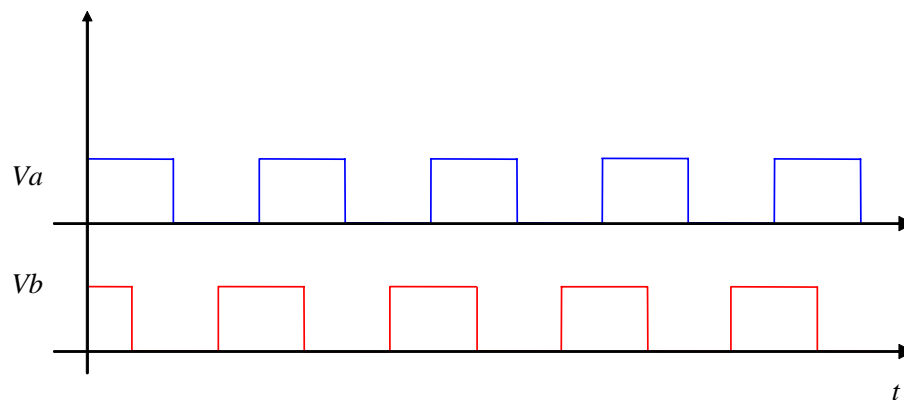
### Decodificación

La posición de cada fototransistor es un cuarto de la distancia existente entre dos franjas. Esta disposición provocará dos señales;  $V_a$  y  $V_b$ , entre ambas señales se provocará un desfase de  $90^\circ$  o de un cuarto de ciclo. La resolución será de  $360^\circ/N$ , donde  $N$  es el número de franjas presente en la superficie del cilindro.

Las señales en cuadratura  $V_a$  y  $V_b$  girando en sentido horario y antihorario se muestran en figura 1.16 y 1.17 respectivamente.



**Figura 1.16.**  $V_a$  y  $V_b$  rotando en sentido horario.



**Figura 1.17.**  $V_a$  y  $V_b$  rotando en sentido antihorario.

Las señales de salida de los codificadores en cuadratura contienen la información y el sentido del desplazamiento realizado.

### 1.4.3. Sistemas electrónicos para la medición de longitud

Para resolver el problema de adquisición de datos es necesario desarrollar una tarjeta electrónica que transfiera la información de los sensores en cuadratura de la estación de medición unidimensional a la computadora por puerto USB. Para esto se incorpora un contador en cuadratura y un microcontrolador que cuente con puerto USB de comunicación.

#### Contadores en cuadratura

El problema consiste en decodificar la señal de cuadratura proveniente de los sensores de desplazamiento de la estación de medición unidimensional. Se emplea un contador en cuadratura, se trata del LS7266.

Construido en tecnología CMOS cuenta con dos contadores independientes de 24 bits X y Y, ambos programables por medio de registros internos de 8 bits, así como un control de habilitación o deshabilitación. Cuenta con un bus de comunicación de 8 bits que le permite programar el conteo en tiempo real de señales en cuadratura, el cual funciona como canal de datos y canal de control mediante el control de habilitación del circuito.

Para escribir al canal de datos se cuenta con registros *preset latch* y para la salida están habilitados los registros *output latch*, ambos escriben o leen información de 3 bytes en un ciclo de reloj. Un *byte pointer* se encarga de incrementar automáticamente el direccionamiento del siguiente byte para no crear conflictos de comunicación.

Por medio del canal de control se escriben los comandos internos para controlar la transferencia de datos del *preset latch* hacia el contador y la comunicación de salida del contador hacia los *output latch*.

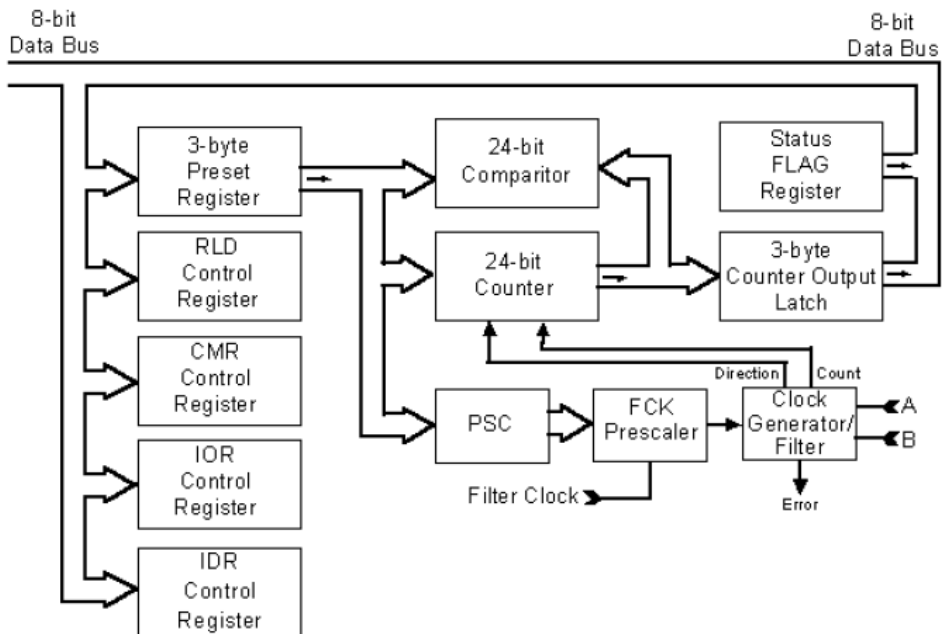
El canal de control cuando lee regresa 6 bits de estado de los registros de bandera. Cuando se escribe los datos son enviados a uno de los cuatro diferentes registros de control.

- RLD. Registro de decodificación para *preset* y carga.
- CMR. Registro del modo de conteo.
- IOR. Registros de control para entrada y salida.
- IDR. Registro de control de índice.

La figura 1.18 muestra un diagrama a bloques de la estructura interna del contador en cuadratura y el acceso a sus registros de control por el bus de datos.

De los ocho bits que se envía al canal de control solo son configurables D0-D4, D5 y D6 que determinan el tipo de registro de control y D7 identifica cual de los dos contadores de 24 bits esta habilitado.

Es necesario programar un microcontrolador que administre los registros de control del contador en cuadratura y adquiera los datos del contador de 24 bits, así como la habilitación de cada eje X o Y [12].



**Figura 1.18.** Diagrama de bloques del contador en cuadratura.

## **Microcontroladores**

Un microcontrolador incorpora en su estructura interna procesador, memoria, reloj, puertos de entrada y salida, temporizadores, puertos de comunicación y control de interrupciones; la forma de programación es sencilla y su construcción esta encaminada a un uso en específico. Con lo cual resulta ser una elección correcta para el desarrollo de la tarjeta electrónica en el presente proyecto por su funcionalidad y tamaño reducido.

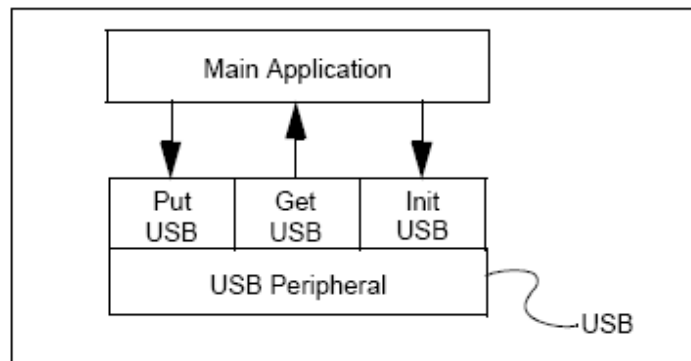
Principales características de microcontroladores de la gama media:

- Manejo de 8 bits.
- Frecuencia de operación de 6MHz hasta 24MHz.
- Set reducido de 35 instrucciones.
- Memoria de programación de 8k.
- 33 puertos de entrada y salida.
- 3 Temporizadores.
- 256 bytes en RAM.
- Puerto Universal Serial Bus (USB 1.1).

Dentro de la gran diversidad en el mercado de los microcontroladores, se ha elegido el PIC16C765 de la gama media de Microchip, el cual cumple con los elementos necesarios para llevar a cabo el control del contador en cuadratura y sobretodo cuenta con puerto de comunicación USB.

Al conectar un dispositivo USB el proceso de comunicación requiere la identificación del dispositivo, el inicio y la forma de la transmisión por USB. Para la configuración del puerto USB, Microchip pone a disposición un firmware con el código en lenguaje ensamblador de la configuración del puerto USB. Aísla la configuración del puerto USB y evita al usuario el manejo del protocolo de USB y el manejo complicado de interrupciones ocasionadas por la recepción o envío de datos.

La figura 1.19 es una representación de bloques de las funciones que realiza el firmware al agregarlo en el código principal, tales como iniciar correctamente el puerto, recibir y comunicar por puerto USB [13].



**Figura 1.19.** Estructura del firmware proporcionado por Microchip.

## 1.5. Instrumentación complementaria

Fue necesario revisar los recursos encontrados en el LAPEM para llevar a cabo el monitoreo de temperatura y humedad. Analizar las posibilidades de comunicación de los equipos de instrumentación y proponer una estructura adecuada que cumpla cabalmente con el procedimiento.

En el procedimiento para la calibración de bloques patrón el LAPEM cuenta a su disposición de un termómetro de módulos apilables con resistencia de platino. El procedimiento requiere mediciones de temperatura en diferentes puntos de interés; dicho instrumento visto en la figura 1.20, tiene la característica de poder agregar hasta 96 módulos adicionales.

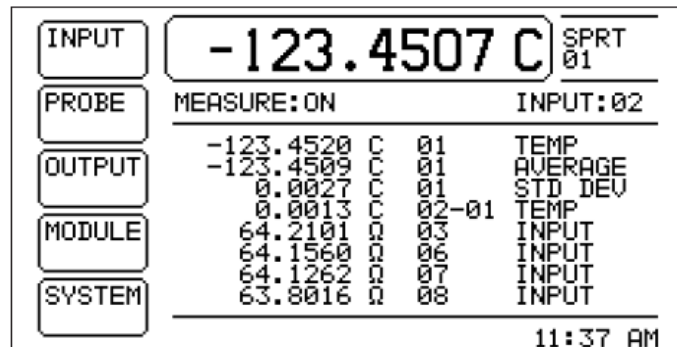


**Figura 1.20.** Termómetro de módulos apilables con resistencia de platino

Cuenta con un modulo central en el que se conectan los canales adicionales de medición. El modulo central convierte la medición de resistencia o voltaje a temperatura con uso de un algoritmo estándar programado en la memoria del microprocesador.



Su versatilidad le permite sensar los canales adicionales en operación y mostrarlos en pantalla. La figura 1.21 ilustra la pantalla del instrumento de la cual se toman en forma manual los valores de temperatura.



**Figura 1.21** Esquema de la pantalla de usuario.

Su interfaz incorpora comunicación RS-232; adicionalmente un modulo de comunicación IEEE-488 GPIB puede anexarse al modulo central y permitir el flujo de comunicación hacia la PC.



## Capítulo 2

# Automatización del área de metrología dimensional

Se presenta el desarrollo de automatización de los procedimientos de calibración de bloques patrón y de micrómetros e indicadores de carátula del LAPEM. Se planteo la necesidad de conseguir un sistema integral hardware-software independiente para cada procedimiento. El sistema debe contar con una arquitectura hardware de comunicación entre los instrumentos de medición, lo cual requiere desarrollar nuevo software que optimice todos los recursos de instrumentación y procesamiento de información, con el firme objetivo de reducir los tiempos de calibración, cumplir con las demandas de los procedimientos de calibración y con los requisitos establecidos por el personal técnico. Mención aparte merece el desarrollo de la tarjeta decodificadora, necesaria para resolver el problema de comunicación de la estación de medición unidimensional.

### 2.1. Implementación hardware para métodos de calibración dimensional

El objetivo es automatizar los procedimientos referentes a la calibración de bloques patrón y micrómetros e indicadores de carátula empleando un lenguaje de alto nivel como LabVIEW, aprovechando al máximo los recursos existentes o en su caso, proponer instrumentación adicional que sea adaptable a la propuesta de automatización. Durante el trabajo en el LAPEM se realizaron observaciones para dictaminar el grado de automatización para ambos procedimientos, se evaluaron los

recursos de comunicación de los instrumentos y se propuso equipo adicional para diseñar una estructura dirigida a optimizar los tiempos de trabajo.

### 2.1.1. Proceso de calibración de bloques patrón

De aquí en adelante las siglas PCBP harán referencia al Proceso de Calibración de Bloques Patrón. El objetivo de dicho procedimiento establece los pasos a seguir en la calibración de bloques patrón por el método diferencial. Actualmente la configuración de algunos instrumentos y la toma de datos se realizan en forma manual; lo que da como resultado el incremento del tiempo de calibración y un aumento de error al transferir la información recabada. Otro factor de incremento es el número de elementos bajo calibración, el cual puede variar de acuerdo al modelo de juego de bloques patrón. La figura 2.1 ilustra un juego de bloques patrón de nueve elementos, pero existen juegos de hasta 112 bloques.



**Figura 2.1.** Juego de bloques patrón.

El termómetro con resistencia de platino se emplea para medir la temperatura sobre puntos estratégicos dentro del procedimiento. En la figura 2.2 se observa la colocación de las puntas de medición. El objetivo es tomar la diferencia de temperaturas entre un bloque patrón y uno bajo calibración después de un determinado tiempo de estabilización. El termómetro procesa la medición asignando un canal primario a cada una de las resistencias de platino. Se configura en modo “scan channel”, y alternadamente cambia el canal primario y despliega en pantalla la lectura correspondiente.



**Figura 2.2.** Medición de temperatura con el termómetro de resistencia de platino.

El PCBP tiene como requisito medir la temperatura y la humedad del lugar en el momento que ocurre la calibración. Debido a que los equipos encontrados en el LAPEM no contaban con ningún tipo de comunicación hacia la PC y solo permitían la toma de datos en forma manual a través de la pantalla, se dictamino adquirir equipo adicional.

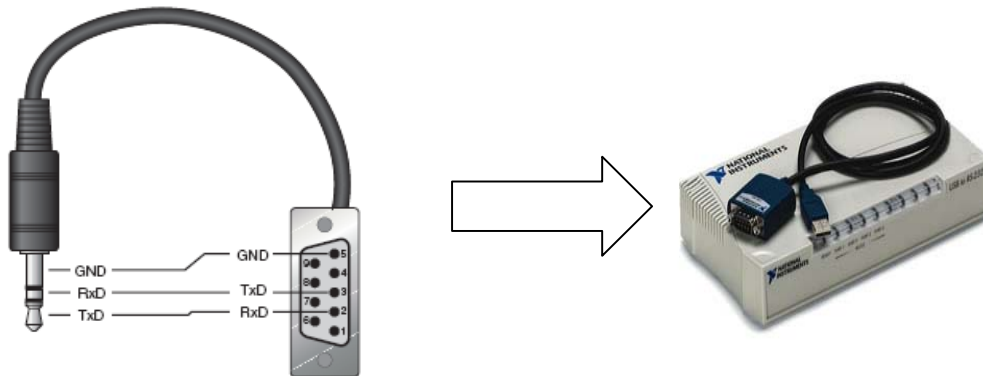
La figura 2.3 muestra un termohigrómetro digital que cuenta con un sensor para la medición de temperatura y humedad relativa.



**Figura 2.3.** Termohigrómetro digital.

La resolución de este equipo es  $0.001^{\circ}\text{C}$  mientras que para humedad es de  $0.01\% \text{ RH}$  y cumple con los requerimientos establecidos en los procesos del LAPEM.

Como medio de comunicación cuenta con una interfaz RS-232, sin embargo se tiene contemplado la conversión a comunicación USB en vista de que muchos equipos actuales de computación portables ya no incorporan esta interfaz en sus equipos. Para esta tarea se utilizará un convertidor RS-232 a USB, como lo muestra la figura 2.4, ver anexo A.2, [23].



**Figura 2.4.** Conversión de cable serie RS-232 a USB.

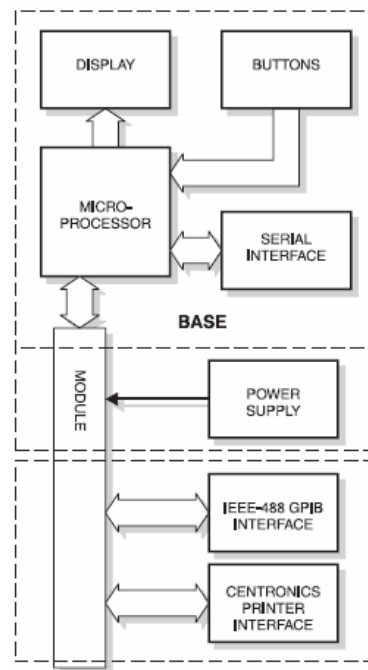
El esquema interno del termómetro de resistencia de platino mostrado en figura 2.5 muestra la capacidad de insertar un accesorio de comunicación GPIB, el cual es adicionado al módulo central habilitando el intercambio de comunicación por este puerto. Se incorporará el uso de protocolos GPIB como medio de comunicación. Para tal labor es necesario adquirir una tarjeta de comunicación GPIB-PCMCIA compatible con los protocolos IEEE-488 como lo muestra la figura 2.6, ver anexo A.1 [19].

Para la comunicación con el instrumento comparador de bloques patrón se establecieron los siguientes parámetros de comunicación serie:

- 1200 Bauds.
- Bits de transmisión: 7.
- Bit de inicio: 1.
- Bits de paro: 2.
- Paridad: No.

Se denominará al comparador de bloques patrón como Instrumento Patrón (IP) y el bloque a calibrar Instrumento Bajo Prueba (IBP). Las bases del sistema propuesto se dirigen a adquirir datos de equipos de

instrumentación sin ocasionar fuentes de errores o causar retardos en la transmisión de datos.



**Figura 2.5.** Diagrama de bloques de la estructura interna del termómetro de resistencia de platino.



**Figura 2.6.** Tarjeta GPIB-PCMCIA de intercambio de protocolos IEEE-488.

El sistema PCBP desarrollado se muestra en la figura 2.7; tiene una arquitectura con base en una computadora portátil a la cual se conecta el IP a través de una interfaz RS232-USB para obtener la información de la medición, el IBP será calibrado por el IP. Así mismo, el equipo adicional de instrumentación se comunica por interfaz RS232-USB y una tarjeta GPIB-PCMCIA, el control y la comunicación entre los elementos del sistema se realizará a través de rutinas de programación.

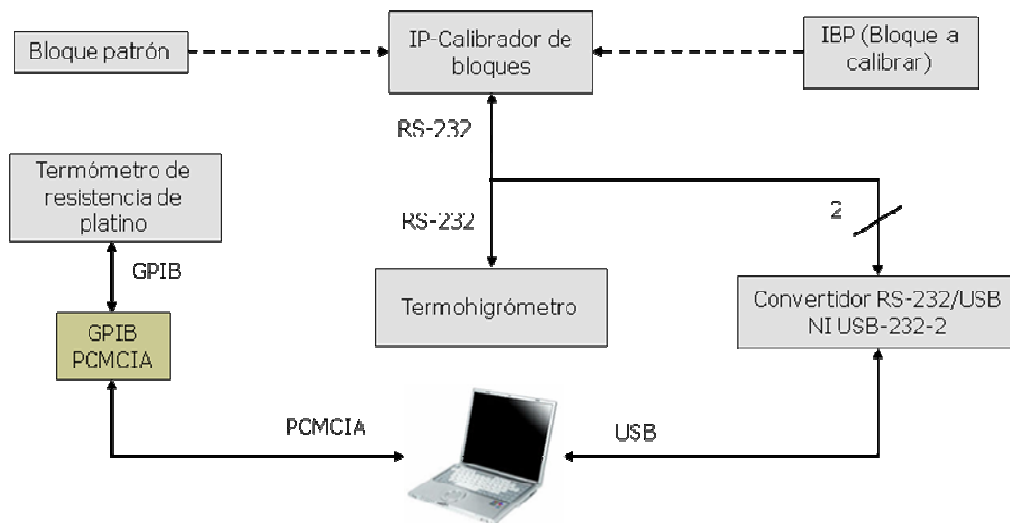


Figura 2.7. Diagrama general de comunicación para el PCBP.

### 2.1.2. Proceso de calibración para micrómetros e indicadores de carátula

El sistema propuesto tiene por objetivo la automatización de la transferencia de información, producto de las lecturas hechas con la estación de medición unidimensional, en los procesos de calibración para micrómetros e indicadores de carátula. De ahora en adelante este procedimiento será referido a las siglas PCMIC.

Actualmente el trabajo de calibración utilizando la estación de medición unidimensional implica trabajo manual en lo referente a la transferencia de información a una hoja de cálculo. Surge así la necesidad de automatizar la transferencia de la información recabada, buscando de esta manera el ahorro de tiempo y la minimización de errores en el proceso mencionado.

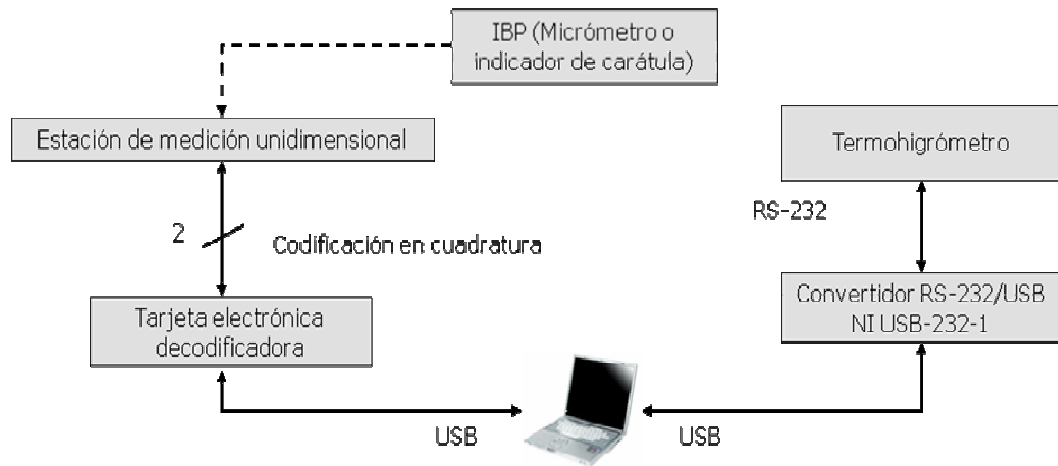
Para este procedimiento se necesita la medición de temperatura ambiente y de humedad relativa, por lo que se propone el mismo termohigrómetro que se utilizó en el PCBP, el cual cumple con los requisitos establecidos en cuanto a resolución por el PCMIC.

Se incorpora al esquema de comunicación la tarjeta electrónica decodificadora, la cual utilizará la interfaz USB como puerto de comunicación.

El sistema PCMIC propuesto se muestra en la figura 2.8; tiene una arquitectura con base en una computadora portátil a la cual se conecta el IP vía la tarjeta electrónica con interfaz USB para obtener la información de medición, el IBP será calibrado por el IP. Así mismo,



el equipo adicional de instrumentación se comunica por interfaz RS232-USB, el control y la comunicación entre los elementos conformantes se realizará con rutinas de programación.



**Figura 2.8.** Diagrama general de comunicación para el PCMIC.

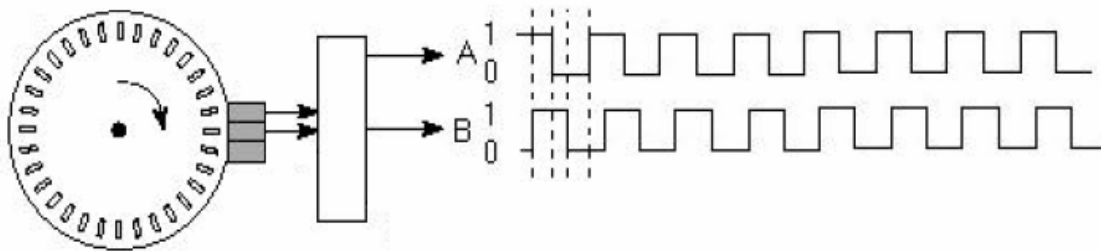
La figura 2.9 muestra en operación al IP del PCMIC, la información en pantalla es un desplazamiento que corresponde a un intervalo de medición de un instrumento. Con esta medición se establece un criterio de calibración sobre el IBP y es de suma importancia procesar correctamente el conteo de los codificadores en cuadratura a unidades de longitud.



**Figura 2.9.** Estación de medición unidimensional en operación.

### 2.1.3. Desarrollo de la interfaz electrónica USB

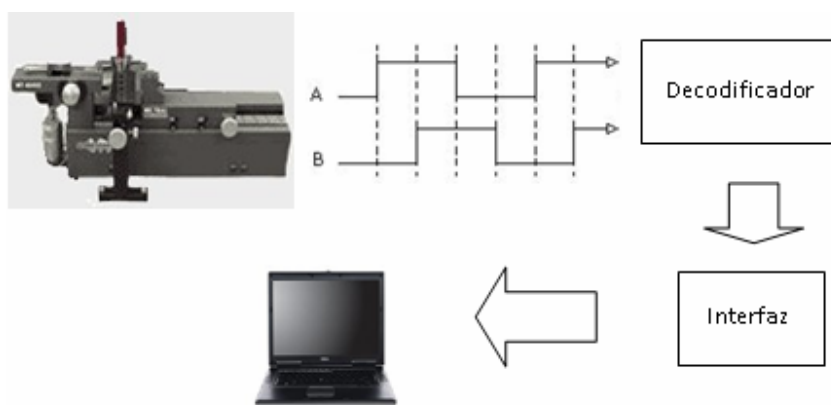
El codificador en cuadratura se localiza en cada uno de los ejes de desplazamiento de la estación de medición. La figura 2.10 muestra la composición de un codificador en cuadratura, generalmente consta de un disco unido a un eje giratorio, con sectores opacos y transparentes; lo cual le permite generar patrones de rotación en dos señales, A y B, de pulsos desfasados un cuarto de ciclo. El conteo de los pulsos determina la posición de la estación de medición unidimensional, mientras que la diferencia de fase determina la dirección de rotación.



**Figura 2.10.** Codificador en cuadratura.

Un decodificador de cuadratura toma las señales de los codificadores y realiza un conteo de los flancos ascendentes o descendentes de los pulsos recibidos. Este conteo puede determinar posición o distancia.

La solución consiste en desarrollar un sistema que realice la decodificación de las señales en cuadratura y que sea capaz de transmitirla por puerto USB a una computadora personal, como se ilustra en la figura 2.11



**Figura 2.11.** Objetivo del sistema electrónico.

Para la decodificación se eligió un circuito integrado de fin específico, se trata del decodificador LS7266R1, con capacidad para

decodificar dos ejes, dos contadores de 24 bits independientes y un bus de datos de 8 bits. Las instrucciones de control son de 5 bits y el modo de conteo puede ser configurado en X1, X2 y X4 según se requiera la resolución. Opera a una frecuencia máxima de 17 MHz para conteo en cuadratura.

En lo que respecta a la interfaz de comunicación se eligió el microcontrolador PIC16C765 con puerto USB, además de que implementa fácilmente elementos de hardware y firmware, ver anexo A.3.

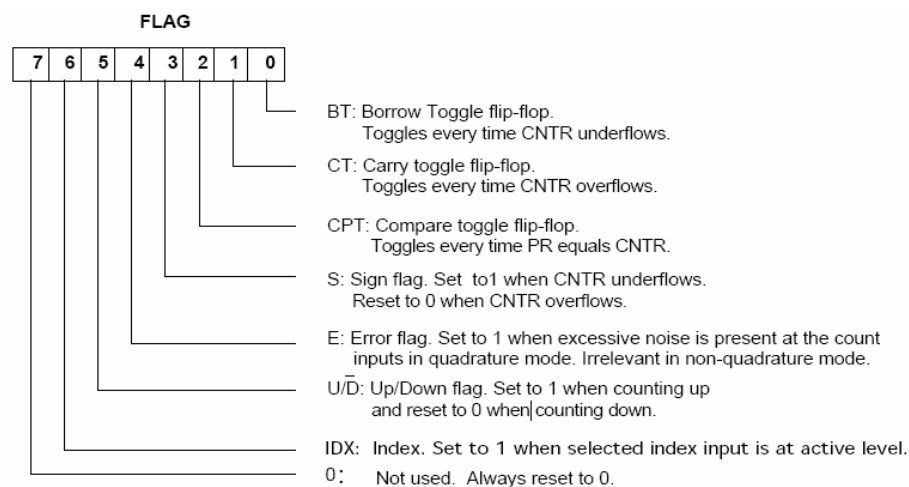
El microcontrolador realizará básicamente las siguientes funciones:

- Envío de comandos de control para el decodificador.
- Recepción de información proveniente del decodificador.
- Conteo de los pulsos de la señal en cuadratura recibidos.
- Envío de información por puerto USB hacia la PC.

El control sobre el decodificador en cuadratura LS7266R1 se elaboro con base en la configuración de sus registros de control.

La figura 2.12 muestra las opciones disponibles para los registros de bandera, indica la información del estado actual de los contadores “CNTR’s” para leer la salida del bus de datos. Destaca el bit 4 que en estado activo indica una salida excesiva en ruido.

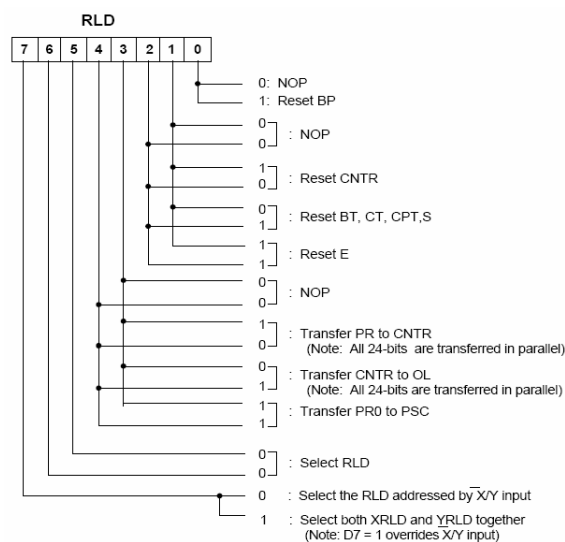
Los registros de reinicio y carga ó “RLD” se muestran en la figura 2.13; controlan la transferencia de información de los registros “PRESET” a los contadores “CNTR”, y de estos a los registros de salida “output latch” u “OL”, además pueden configurar el reinicio del contador, bandera y el byte indicador de secuencia BP.



**Figura 2.12.** Registros de bandera.

La configuración del conteo se realizará en modo de cuadratura X4 para obtener la resolución presente en la estación de medición unidimensional. Las características del sistema son las siguientes:

- Una revolución en el ajuste de desplazamiento burdo corresponde a un intervalo de 4 pulgadas, el cual presenta un conteo de 400,000 pulsos en cuadratura.
- El recorrido horizontal de la estación de medición unidimensional es de 24 pulgadas, se tiene un conteo de 2,400,000 pulsos en cuadratura.
- El ajuste fino es para desplazamientos inferiores a una pulgada correspondiente a un conteo de 100,000 pulsos en cuadratura.
- El conteo máximo permitido con 24 bits es de 16,777,216 y nuestra aplicación requiere como máximo 2,400,000.



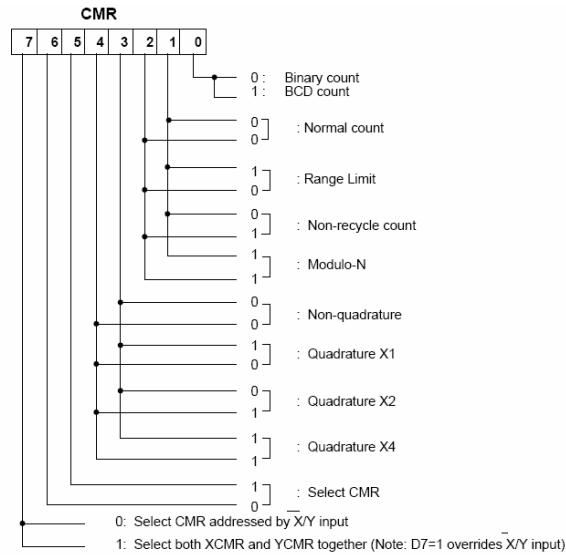
**Figura 2.13.** Registros de reinicio y carga.

De esta forma en el registro de configuración de conteo ó CMR, figura 2.14, se determina configurar en cuadratura X4 y a utilizar el modo de conteo Modulo-N.

El microcontrolador será programado para realizar las siguientes tareas sobre el decodificador:

- Escribir a los registros *preset*, de cada eje X o Y en el decodificador.
- El flujo de entrada o salida de información habilitando el eje correspondiente.

- Configuración del modo de conteo.
- Leer la información en los registros de salida *output latch*.
- Lectura de las banderas de control.



**Figura 2.14.** Registros de configuración de conteo.

El modo de direccionamiento para algunas de las instrucciones que se implementaron en el programa, se ilustran en la figura 2.15.

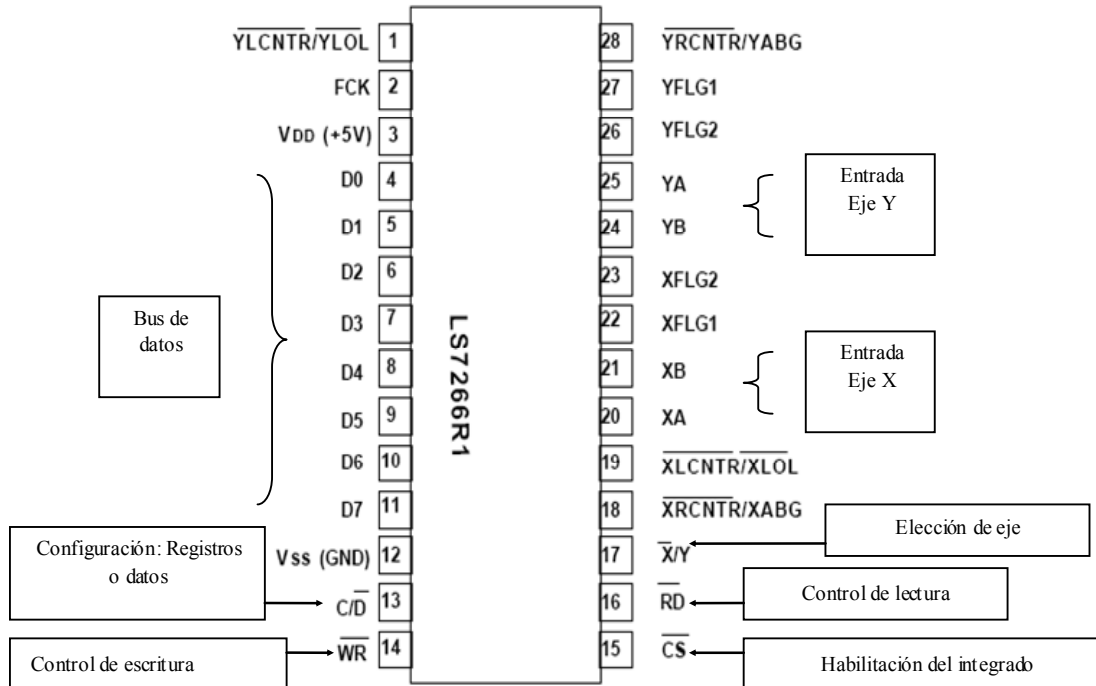
REGISTER ADDRESSING MODES								
D7	D6	D5	C/D	RD	WR	X/Y	CS	FUNCTION
X	X	X	X	X	X	X	1	Disable both axes for Read/Write
X	X	X	0	1		0	0	Write to XPR byte segment addressed by XBP (Note 3)
X	X	X	0	1		1	0	Write to YPR byte segment addressed by YBP (Note 3)
0	0	0	1	1		0	0	Write to XRLD
0	0	0	1	1		1	0	Write to YRLD
1	0	0	1	1		X	0	Write to both XRLD and YRLD
0	0	1	1	1		0	0	Write to XCMR
0	0	1	1	1		1	0	Write to YCMR
1	0	1	1	1		X	0	Write to both XCMR and YCMR
0	1	0	1	1		0	0	Write to XIOR
0	1	0	1	1		1	0	Write to YIOR
1	1	0	1	1		X	0	Write to both XIOR and YIOR
0	1	1	1	1		0	0	Write to XIDR
0	1	1	1	1		1	0	Write to YIDR
1	1	1	1	1		X	0	Write to both XIDR and YIDR
X	X	X	0	0	1	0	0	Read XOL byte segment addressed by XBP (Note 3)
X	X	X	0	0	1	1	0	Read YOL byte segment addressed by YBP (Note 3)
X	X	X	1	0	1	0	0	Read XFLAG
X	X	X	1	0	1	1	0	Read YFLAG

X = Don't Care

**Figura 2.15.** Modos de direccionamiento para el decodificador LS7266R1.

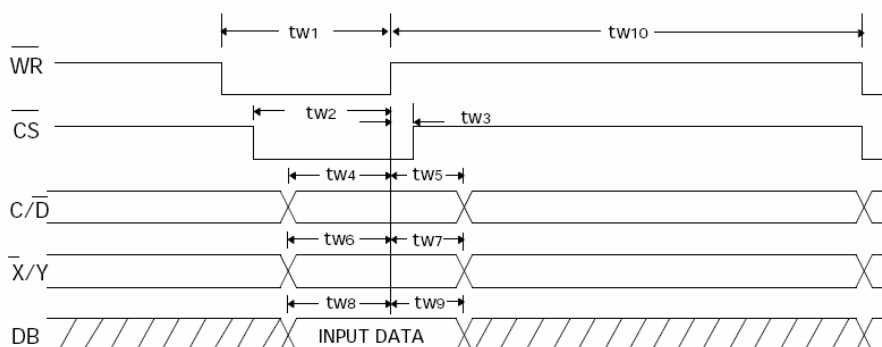
Básicamente la programación se desarrolló tomando en cuenta los ciclos de lectura y escritura de datos o registros de control, así como

la habilitación y deshabilitación del integrado. La figura 2.16 muestra el patigrama del circuito integrado LS7266R1, en esta podemos ver el bus de datos, la entrada de las señales en cuadratura y las entradas para configurar ciertos parámetros de interés, como lo son las terminales 13-17 [12].



**Figura 2.16.** Patigrama del decodificador LS7266R1.

La secuencia de lectura se ilustra en la figura 2.17, en la cual RD (patilla 16) en nivel bajo configura al integrado para leer información de los "OL's" proveniente de los registros PRESET.

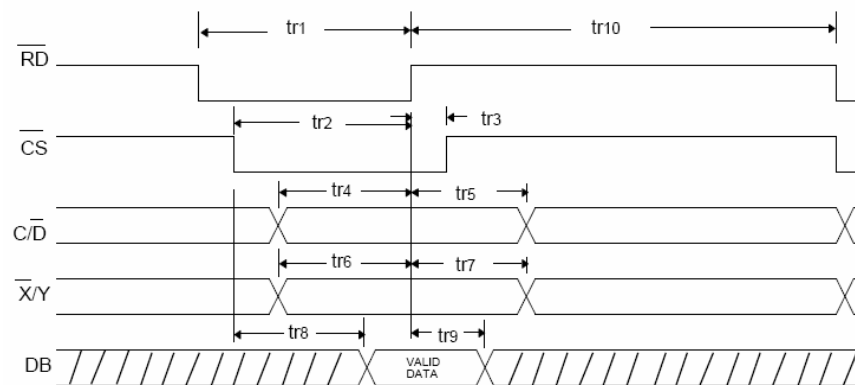


**Figura 2.17.** Ciclo de lectura.

$C/\bar{D}$  identifica para estado alto si la lectura es un registro de control o datos en nivel bajo,  $\bar{X}/Y$  es la selección entre cualquiera de los dos ejes X o Y.

Para el ciclo de escritura, mostrado en la figura 2.18, la señal  $\overline{RD}$  pasa a nivel bajo y configura al integrado para escribir, espera el paso a nivel bajo de  $\overline{CS}$  para habilitar, el tiempo mínimo de operación  $tw1$  y  $tw2$  es de 45 ns para ambas señales. Entonces empieza el envío de información, y esta se deshabilita cuando  $\overline{RD}$  cambia a estado alto, lo que provoca que sólo se tenga un tiempo mínimo de transmisión correspondiente a  $tw9$  de 10 ns antes de quedar deshabilitada la función escribir.

El tiempo de espera mínimo para iniciar un nuevo ciclo de escritura es de  $tw10$  de 90 ns. De igual forma  $C/\overline{D}$  identifica entre datos o registros de control y  $\overline{X}/Y$  el eje en uso.



**Figura 2.18.** Ciclo de escritura.

Como herramienta de programación se utilizó lenguaje ensamblador. El funcionamiento se encuentra representado por el diagrama de flujo de la figura 2.19.

La idea de programación consiste en configurar los puertos del microcontrolador para intercambiar información con el decodificador, iniciarlo y configurarlo en modo de cuadratura X4 e incorporar el firmware de comunicación USB al programa principal. Entonces inicia el protocolo de comunicación de registros de control y datos entre microcontrolador y decodificador, con lo que se realizará continuamente el conteo de las señales en cuadratura para cada uno de los ejes X y Y. El microcontrolador puede ser manejado por software proveniente de la PC, en caso de recibir un dato 1, 2 o 3 reiniciará ambos ejes, individualmente el eje X o Y, respectivamente. Por default se enviará el valor cero y continuará con el envío del conteo correspondiente al desplazamiento realizado por la estación de medición unidimensional, utilizando el puerto USB del microcontrolador.

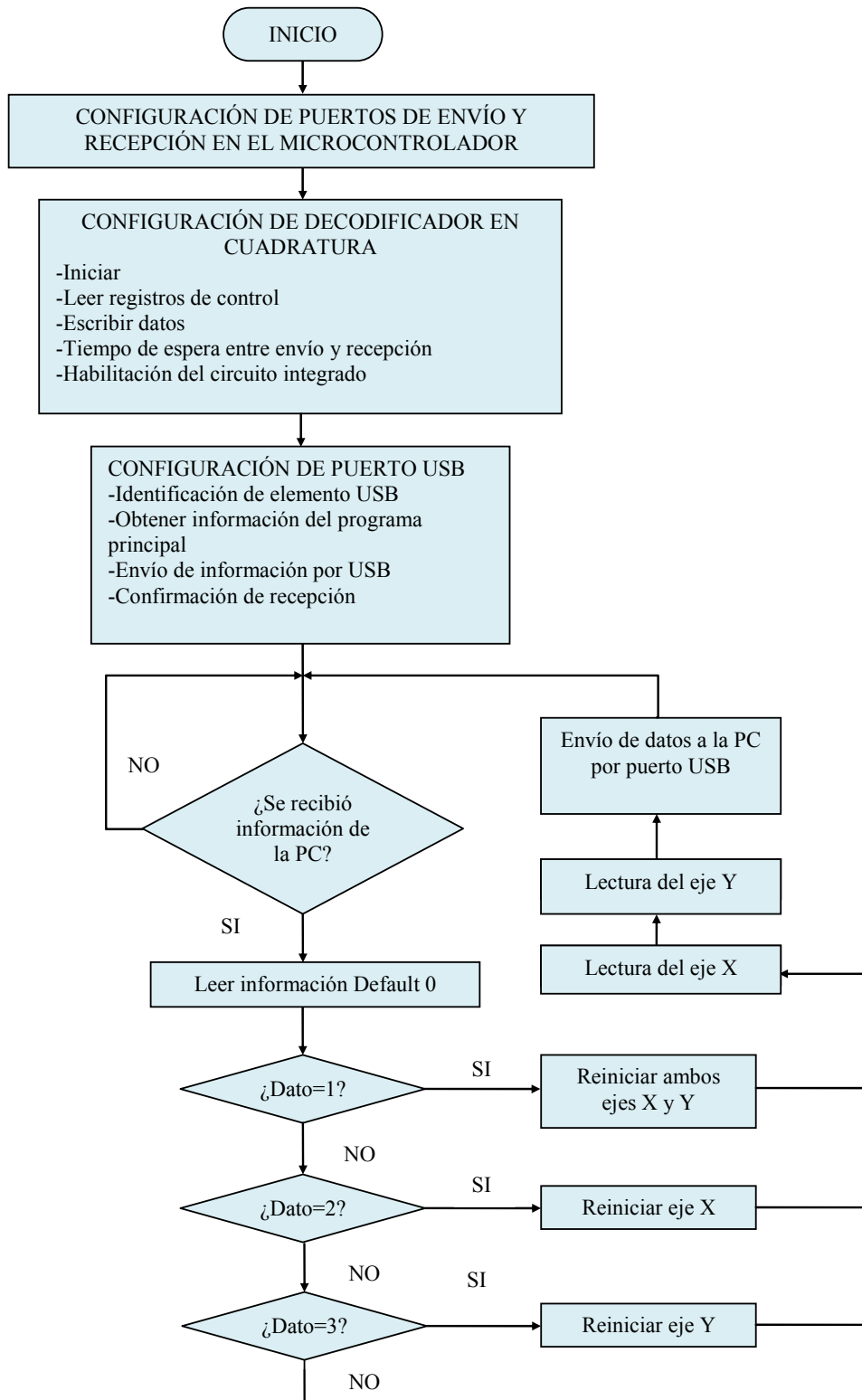
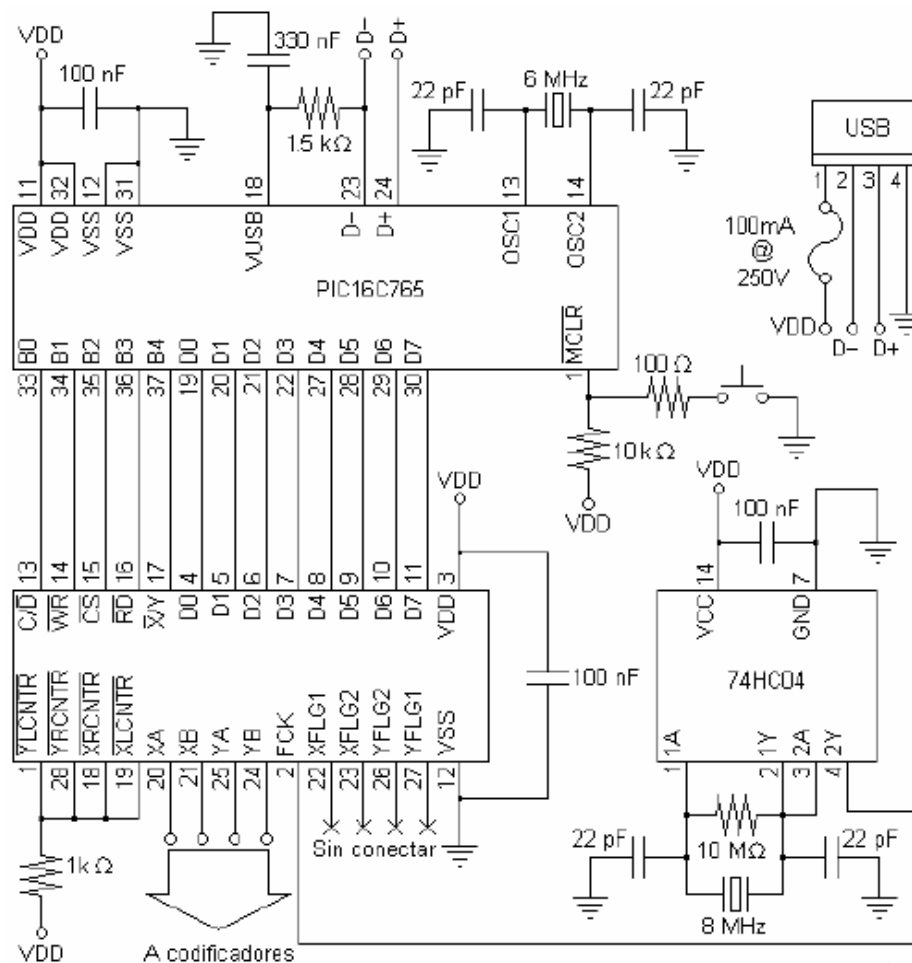


Figura 2.19. Diagrama de flujo del microcontrolador.



La figura 2.20 muestra el diagrama de implementación para la tarjeta decodificadora de señales en cuadratura. El puerto B del microcontrolador es utilizado para habilitar al decodificador, así como para la elección del eje y el modo de operación a ejecutar: leer o escribir. El puerto D se acondiciona para el intercambio de datos y de registros de control. Obsérvese que los pines 1,18,19 y 28 están en estado alto, lo cual configura al decodificador en carga directa, es decir, se pasa la información de los CNTR's hacia los registros de salida sin verificar el índice de las señales en cuadratura.



**Figura 2.20.** Diagrama de la tarjeta decodificadora.

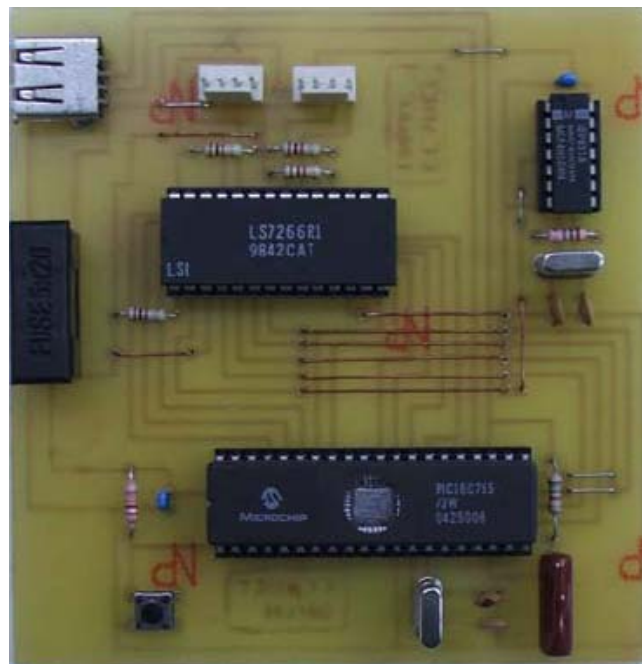
Se utiliza un circuito de reloj utilizando el inversor 74HC04 y un cristal de cuarzo de 4 MHz, para generar la señal de reloj para el decodificador. Para el microcontrolador se utiliza un cristal de cuarzo de 6 MHz. Por especificación del fabricante, la señal FCK en el decodificador para conteo en cuadratura debe cumplir con la siguiente condición:  $f_{FCK} \geq 8f_{QA}$  ó  $8f_{QB}$ .

Las señales de los codificadores estaban en el rango de 100 kHz a 300 kHz, al utilizar 8MHz el límite era 1 MHz para los codificadores. Por lo tanto estaba en un rango adecuado para la operación.

El consumo de la tarjeta en operación es de 60mA, por lo que se decide ocupar la alimentación del puerto USB de la computadora. Las señales  $D^-$  y  $D^+$  se encuentran en las terminales 23 y 24 del microcontrolador.

Para los conectores de la estación de medición unidimensional se utiliza interfaz DB-50.

Finalmente en la figura 2.21 se muestra físicamente la tarjeta decodificadora con interfaz USB desarrollada. Posteriormente se describe el software en LabVIEW para adquirir dicha información y convertirla a una magnitud de longitud en unidades de pulgadas o milímetros.



**Figura 2.21.** Tarjeta decodificadora con interfaz USB.

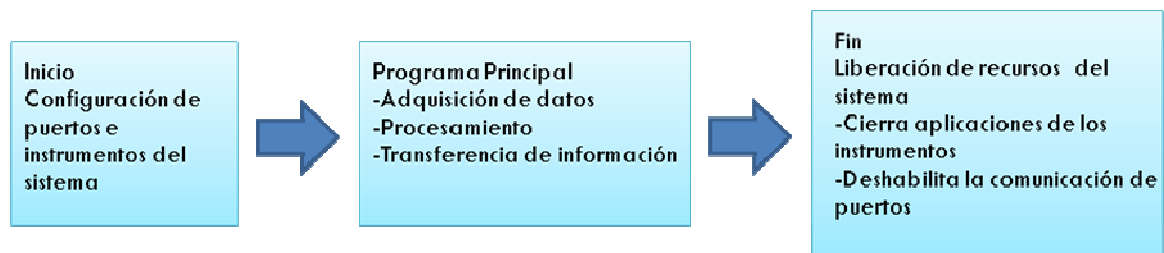
## **2.2. Requisitos de software para métodos de calibración dimensional**

Durante el trabajo realizado en las instalaciones del LAPEM, se trazó el objetivo de entrevistar a los operadores a cargo de las calibraciones en el área de metrología dimensional. Uno de los motivos era observar el correcto desarrollo de ambos procedimientos

de calibración y determinar las tareas susceptibles a automatizar e identificar las fuentes de retraso para el operador.

Se ideó un concepto de software que realizará automáticamente dichas labores y donde el trabajo del operador solo estaría destinado a la colocación del instrumento bajo prueba, aportar su experiencia en el manejo de los instrumentos patrones y a emitir un juicio final sobre la calibración.

La estructura básica del software debe evitarle al operador consumir tiempo en labores de configuración de instrumentos, adquisición de datos, transferencia de información y elaboración de reportes, además de brindarle seguridad a los equipos mientras se adquiere la información de medición. La figura 2.22 ilustra la organización del software con sus respectivas tareas, orientadas a la solución del problema.



**Figura 2.22.** Diagrama de bloques de la estructura básica de software.

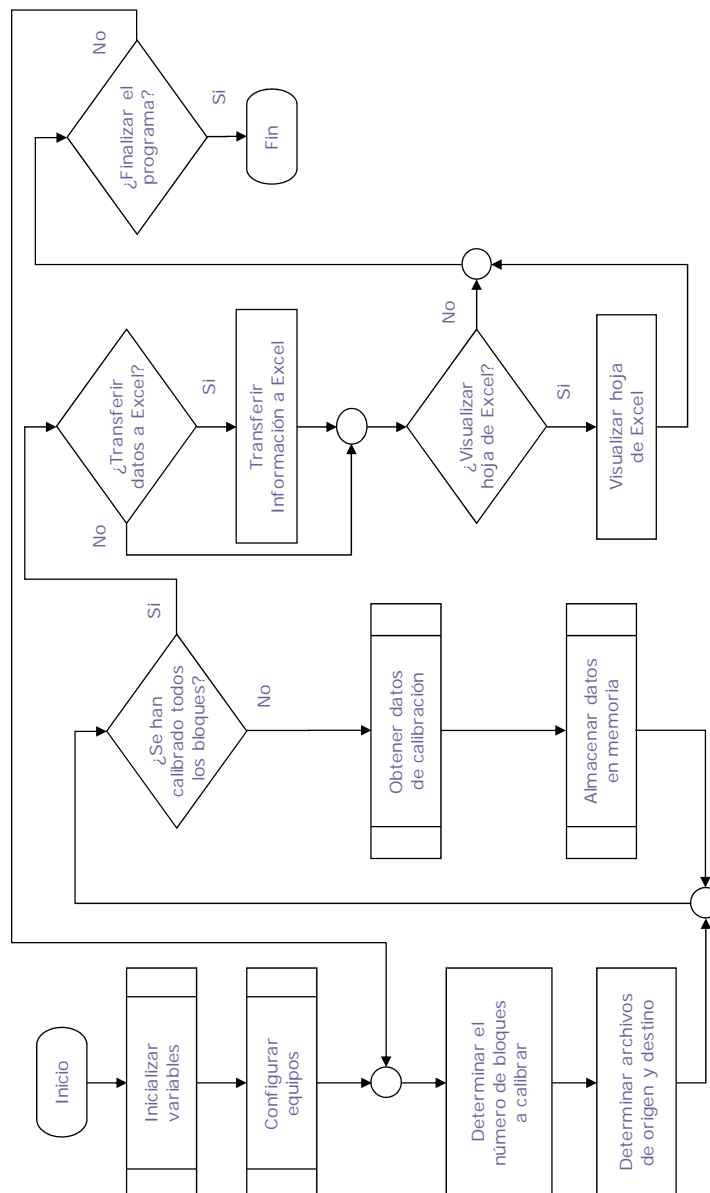
Cada proceso de calibración sigue una metodología, indica el correcto manejo de los equipos patrones, número de series y repeticiones de lecturas en determinados intervalos; así como el momento y las condiciones en el cual una medición será aprobatoria. Entonces el software deberá de realizar la adquisición de datos conservando el orden coherente que demanda el procedimiento.

Con base a los requisitos establecidos por el operador, se dispuso a realizar una interfaz de manejo simple que le permita controlar el flujo de información.

El objetivo principal es transmitir, en indicadores visuales, la información de todos los instrumentos de medición en el mismo tiempo en que se generan, sin retrasos ni variaciones de dígitos. La razón corresponde a la técnica con que se desenvuelve el operador, es decir, la estabilización del instrumento, la variación del último dígito, la repetibilidad del valor y la falta de sincronización tendrán como resultado desconfianza en el uso del software final. Los indicadores visuales son sobre fondo negro con números en color rojo para permitir al operador trabajar a una distancia considerable.

### 2.2.1. Requisitos para el procedimiento de calibración de bloques patrón

Al observar el trabajo desarrollado por el operador en el momento de la calibración de bloques patrón, se elaboró un diagrama de flujo, ilustrado en la figura 2.23, el cual describe las actividades que realiza el operador con el comparador de bloques patrón incorporando a su vez las opciones del manejo de datos a una hoja de cálculo.



**Figura 2.23.** Diagrama de flujo del PCBP.

Se tiene contemplado la operación para el termómetro de resistencia de platino y el termohigrómetro digital. Cuando el operador decida adquirir la información de los instrumentos involucrados en el proceso le bastará con oprimir un botón.

La figura 2.24 muestra la propuesta de interfaz de usuario para el PCBP desarrollada en LabVIEW. Se le agregaron las siguientes implementaciones:

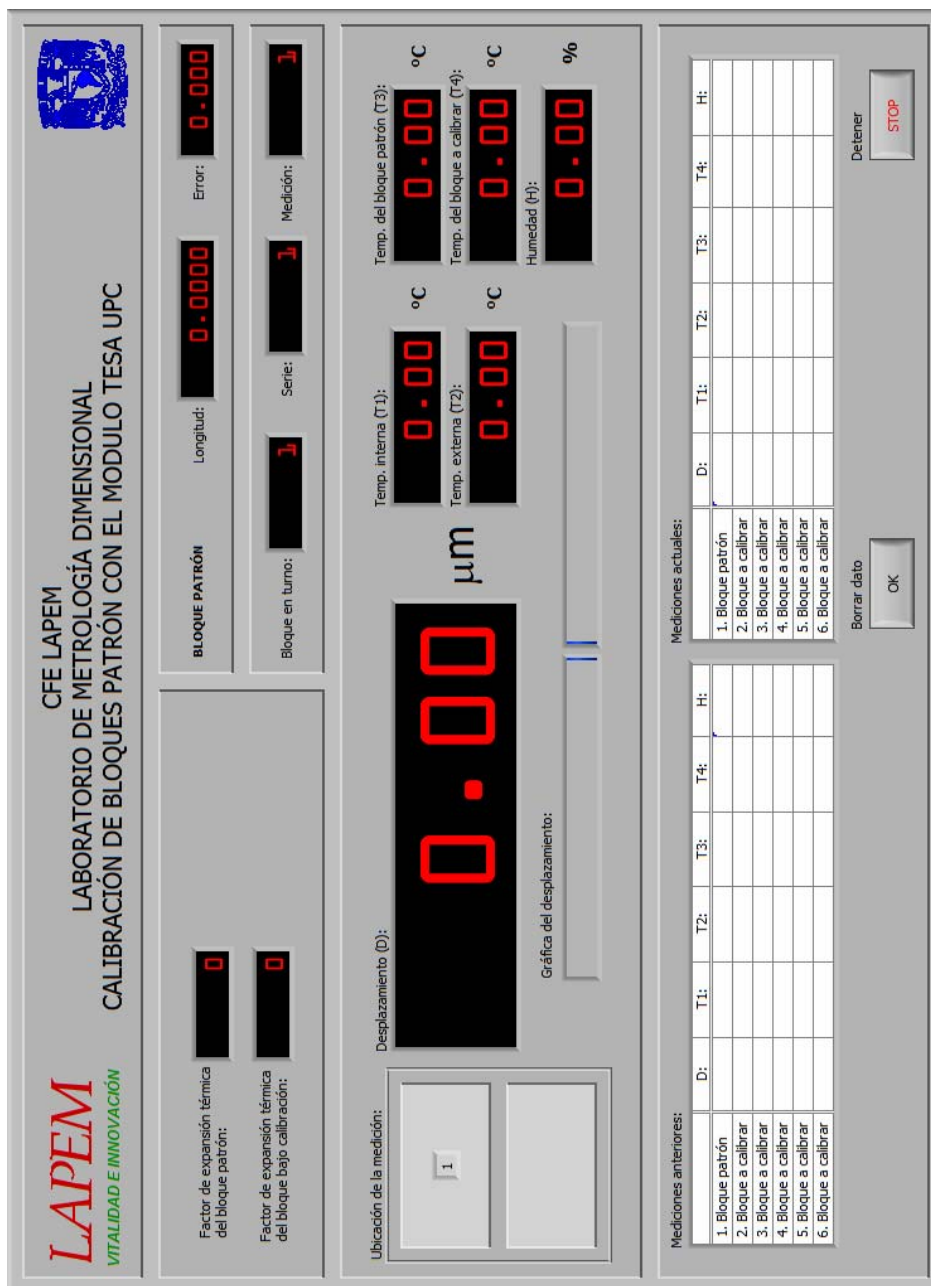


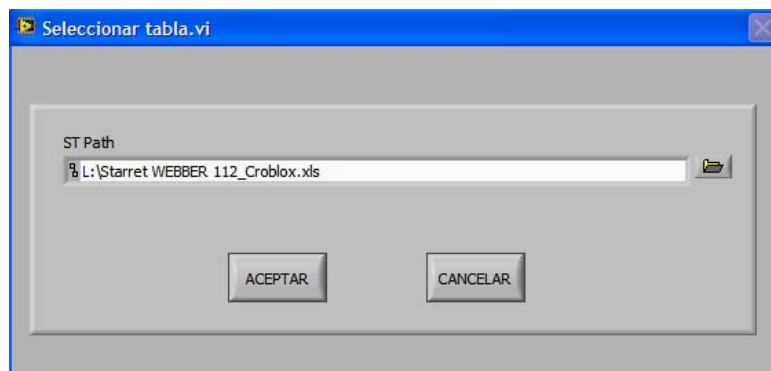
Figura 2.24. Interfaz gráfica de usuario para el PCBP.

- La parte superior izquierda contiene un control numérico donde el operador ingresa el valor del coeficiente de expansión lineal correspondiente al tipo de material del bloque patrón y bajo calibración.
- La parte superior derecha contiene indicadores numéricos que le permiten al usuario observar la secuencia de calibración. Muestran el número de bloque, la serie correspondiente y el punto sobre el bloque donde se realiza la medición. Así como la longitud nominal del bloque patrón con su correspondiente error.
- En la parte central se visualizan los valores de temperaturas externa, interna, del bloque patrón y bajo calibración, así como la humedad y el valor de desplazamiento enviado por el comparador de bloques patrón, en forma numérica y gráfica.
- Adicionalmente se implemento un visualizador gráfico de un bloque con un indicador numérico, en el cual se despliega el orden de los puntos de medición sobre las esquinas de la superficie del bloque bajo calibración en turno y de acuerdo a la secuencia en que es realizada por el operador con el comparador de bloques patrón.
- En la parte inferior se implementaron dos tablas donde el operador puede ver la información de la calibración: a la derecha se muestran los datos actuales y a la izquierda las mediciones anteriores que se han adquirido. De este modo el operador puede ver la forma en que transcurre el proceso de calibración.
- También se localizan el botón de borrar dato y de paro de ejecución de programa. Si no hay más puntos de medición por realizar, el operador termina su calibración y todos los recursos utilizados por el software son liberados.
- El control de captura de las mediciones de todos los instrumentos, esta sincronizado con el botón accionador de la bomba de vacío que se encuentra localizado en la unidad central de procesamiento del comparador de bloques patrón.

El PCBP se realiza utilizando un juego de bloques patrón de referencia de 112 elementos, el operador comúnmente lo utiliza para calibraciones de juegos con menor número de elementos. Entonces se planteo la posibilidad de abrir un archivo de Excel que incluía la información de los bloques patrón de referencia.

Esta tarea se incluyo en un menú desplegado durante la ejecución del programa principal; la figura 2.25 muestra la pantalla para tal

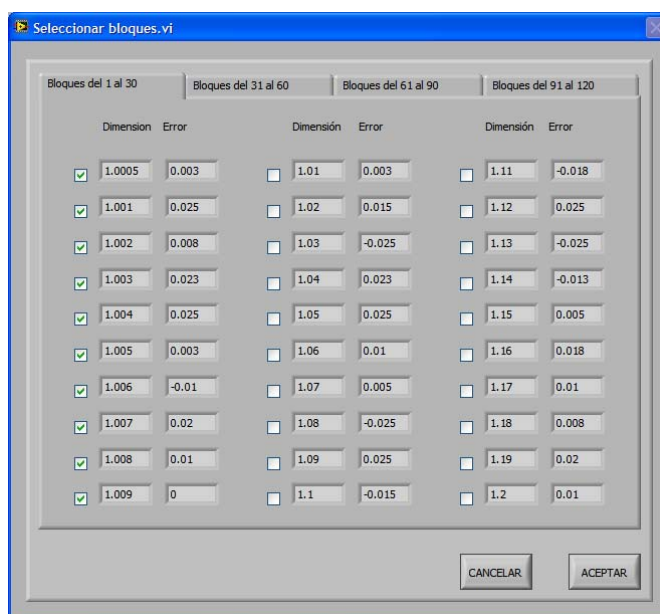
labor y consiste en ingresar la ruta del archivo con extensión xls y aceptar. De esta forma el operador puede ingresar diferentes juegos de bloques patrón.



**Figura 2.25.** Pantalla de selección de tabla.

Mediante otra opción se pueden elegir ciertos elementos de todo el catálogo de longitudes nominales, los puntos seleccionados corresponden a los valores bajo calibración y entonces el programa se sujeta a calibrar sobre la elección en forma coherente a los puntos establecidos por el procedimiento.

La figura 2.26 muestra una pantalla de usuario donde se observa un folder con cuatro contenedores con información de bloques patrón y con la posibilidad de seleccionar determinada información.

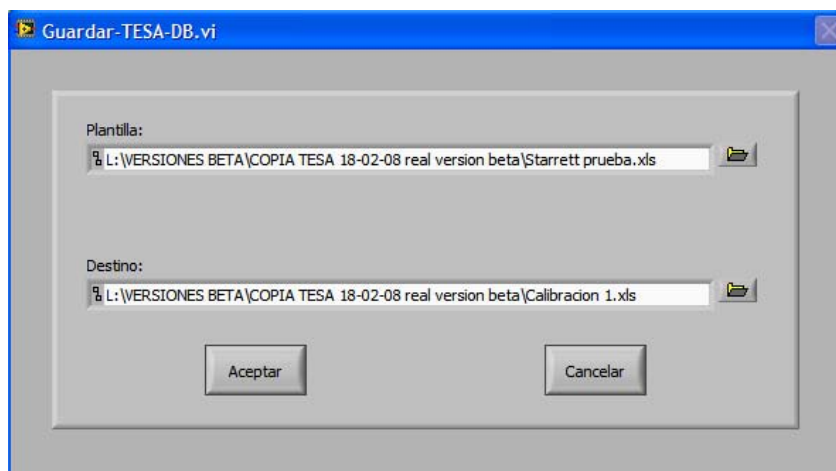


**Figura 2.26.** Pantalla de selección de bloques.

Una vez concluida la elección de calibración, el usuario puede transferir los datos capturados a una hoja de cálculo o elaborar su

correspondiente reporte de calibración. La figura 2.27 muestra la pantalla para guardar los datos de calibración, se ingresa la ruta correspondiente a un archivo plantilla predeterminado por el laboratorio para generar sus reportes y se guarda en un archivo destino con la información, sin alterar la plantilla.

En una pantalla muy similar a la de la figura 2.27 se creó la opción de recuperar archivos que hayan quedado pendientes por calibrar, esto debido a que el procedimiento incrementa en tiempo de acuerdo al número de elementos bajo calibración. Se agrega la ruta del archivo con la calibración inconclusa y el programa reinicia en el número de bloque, serie y punto de medición donde se suspendió anteriormente.



**Figura 2.27.** Pantalla para guardar información.

### **2.2.2. Requisitos para el procedimiento de calibración de micrómetros e indicadores de carátula**

De igual forma se desarrollo un diagrama de flujo que describiera la solución a la automatización del PCMIC. Cabe destacar que existen diferentes tipos de micrómetros, los cuales exigen diferentes series de medición en diversos intervalos. Pero el objetivo primario de la estación de medición unidimensional no cambia, por lo que no importa el tipo de instrumento bajo calibración, la solución de automatización se ajusta adecuadamente a cada caso. En la figura 2.28 se muestra el primer diagrama de flujo principal. El operador calibra sus instrumentos en forma normal, pero existen extensiones que se pueden agregar en determinado punto del procedimiento, los cuales involucran otros intervalos de medición y series de repetición. Para esta parte del procedimiento se desarrollo el diagrama de flujo de la figura 2.29.



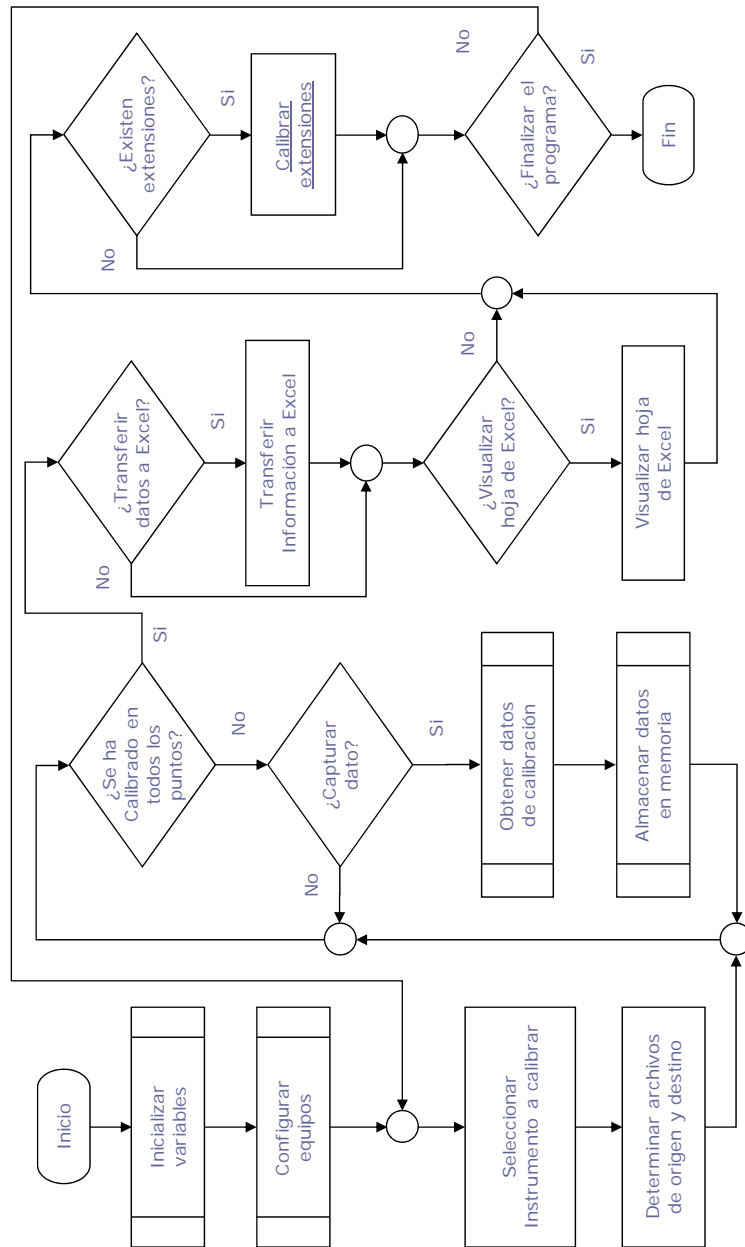
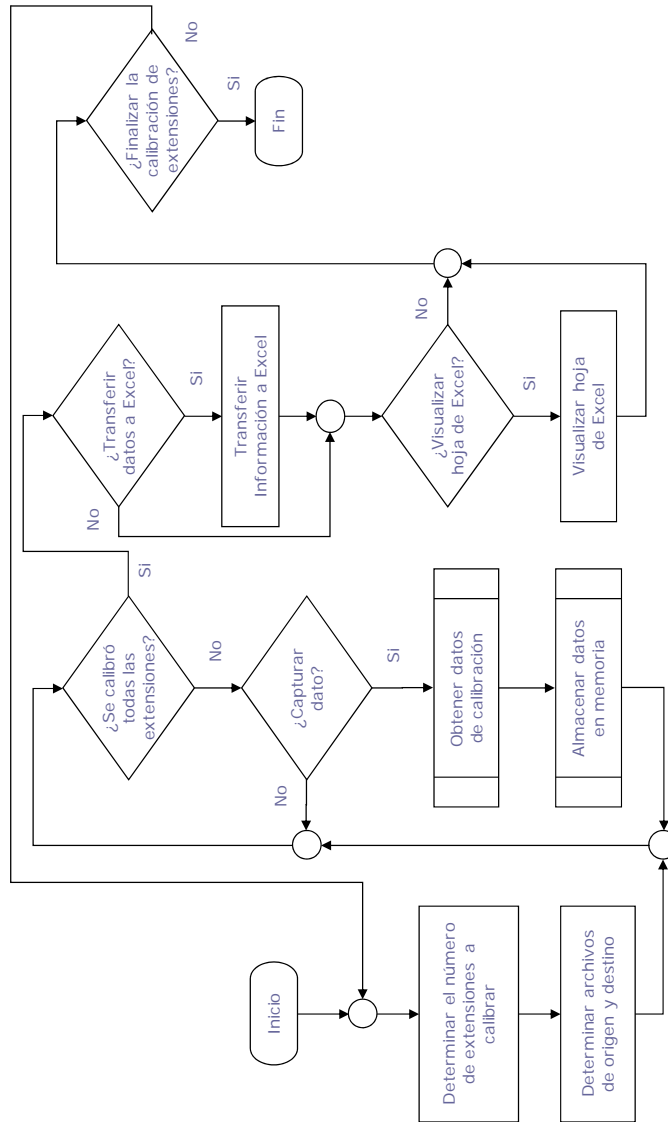


Figura 2.28. Diagrama de flujo general para el PCMIC.



**Figura 2.29.** Diagrama de flujo particular para micrómetros con extensiones dentro del PCMIC.

El funcionamiento sobre los instrumentos complementarios consiste en adquirir información de temperatura y humedad en el momento que se aprueba la lectura con la estación de medición unidimensional. El PCMIC involucra gran parte de la programación al procesamiento de datos que se hace sobre la tarjeta decodificadora.

Al igual que en el PCBP, se incorpora en el menú de ejecución las opciones de guardar y recuperar una calibración inconclusa. Una vez

que se hayan hecho las lecturas de todos los puntos de calibración, la información se puede transferir a una hoja de cálculo utilizando el mismo SubVI y respetando el mismo funcionamiento de operación.

Para este caso se implementa un botón de captura que controla la adquisición de longitud, temperatura y humedad; así como las opciones de borrar información y alto de ejecución del programa.


En cuanto a la secuencia de ejecución, se toman ciertos intervalos predefinidos por el PCMIC donde se realizan tres mediciones y se crean indicadores que guían al operador durante la utilización del software.

Se le añadió un selector para tener la posibilidad de elegir el tipo de instrumento que estará bajo calibración, de esta parte destaca la opción de extensiones; la cual no presenta un intervalo fijo de mediciones debido a que el valor límite es determinado por el modelo del instrumento bajo calibración.


Algunas funciones del software del PCMIC implementadas en la interfaz de usuario de la figura 2.30 se mencionan a continuación:

- En la parte superior se visualizan los intervalos de medición predefinidos normalmente por el PCMIC. Cuenta con una opción para ingresar valores definidos por el propio usuario.
- En la parte central se localizan los indicadores visuales de los instrumentos que conforman la estación de trabajo. Se muestran las lecturas emitidas por el termohigrómetro digital así como el desplazamiento de los dos ejes de medición fino y burdo de la estación de medición unidimensional y la correspondiente suma de valores.
- A la izquierda de los indicadores de cada eje se habilitan controles de reinicio a cero para mover la referencia de origen a un determinado punto de la estación de medición unidimensional. También están localizados unos selectores para cambiar el sentido de desplazamiento a positivo o negativo. Ambos controles fueron establecidos por el operador con base a su experiencia en el desarrollo del PCMIC.
- A la derecha de los indicadores numéricos de longitud se encuentran unos indicadores de barras que corresponden al desplazamiento en forma gráfica.
- En la parte central derecha se muestra un selector de unidades, el software puede desplegar mediciones en milímetros o pulgadas. Al habilitar cualquier opción el software cambiará

automáticamente en todas sus aplicaciones al sistema de medición elegido.



**LABORATORIO DE METROLOGIA DIMENSIONAL**  
**MÓDULO PARA CALIBRACIONES CON LA MAQUINA DE MEDICIÓN MIC-TRAC MT-4000**



**Puntos de Medición:**

0.000	2.500	5.100	7.700	10.300	12.900	15.500	18.100	20.700	23.300	25.000
-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------

**Modo de las lecturas:**

Predefinidas  Usuario

**Unidades:**  mm  pulgadas

**Instrumentos:**

Micrómetro de exteriores  
 Micrómetro de interiores  
 Micrómetro de profundidades  
 Indicadores de carátula  
 Extensiones

**Serie de lectura:**   

**Próxima lectura:**

**Próxima lectura evaluada:** 0.000

**Temperatura:** 20.13 °C

**Humedad:** 45.83 %

**Eje X:** 72.5213 mm

**Eje Y:** -47.8068 mm

**Eje X + Eje Y:** 24.7145 mm

**OK Button**  (-x)  (-y)

**OK Button 2**  x = 0  y = 0

**OK Button 3**  x = 0, y = 0

**Datos capturados:**

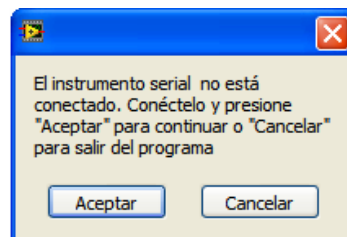
	Eje X	Eje Y	X + Y	Temperatura	Humedad
Medición 1					
Medición 2					
Medición 3					

**Capturar** **Borrar**

DETERNER

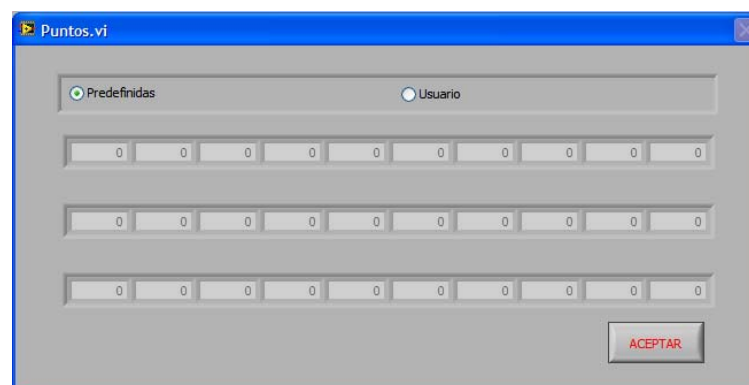
**Figura 2.30.** Interfaz gráfica de usuario para el PCBP.

- En la parte inferior se despliega una tabla con las mediciones capturadas en el intervalo de medición del correspondiente instrumento. Contiene el recorrido independiente de ambos ejes, la suma, la temperatura y la humedad.
- El programa manda mensajes de error como el de la figura 2.31 en caso de que el operador haya omitido alguna conexión de instrumento, y no iniciará su ejecución hasta que se encuentren todos los elementos de la estación de trabajo.



**Figura 2.31.** Interfaz gráfica de usuario para el PCMIC.

- Referente a la calibración con extensiones, en la figura 2.32 se muestra una pantalla de ingreso de valores, para aquellas extensiones que propasen los puntos predefinidos. Es una acción previsoras, ya que usualmente no se han calibrado extensiones tan grandes siguiendo el PCMIC, pero en el mercado existen instrumentos con tales características.



**Figura 2.32.** Pantalla de ingreso de puntos particulares de medición.

### 2.3. Desarrollo de la programación en LabVIEW

Como herramienta de programación para la solución se utilizo LabVIEW en su versión 8.0, siguiendo las bases de la instrumentación virtual se desarrollará un software que incorpora todos los recursos de hardware previamente establecidos en la sección 2.1, así como el aprovechamiento de la PC para los requerimientos de la sección 2.2.

Existen tareas de programación que son afines para ambos procedimientos de calibración y que pueden ser aplicadas a cada software a través de la creación de un SubVI.

Por razones de confidencialidad las figuras que ilustran código del diagrama de bloques de LabVIEW, son representativas de la solución y solo tratan de ilustrar la idea de programación, con el objetivo de no mostrar el código original, propiedad de CFE.

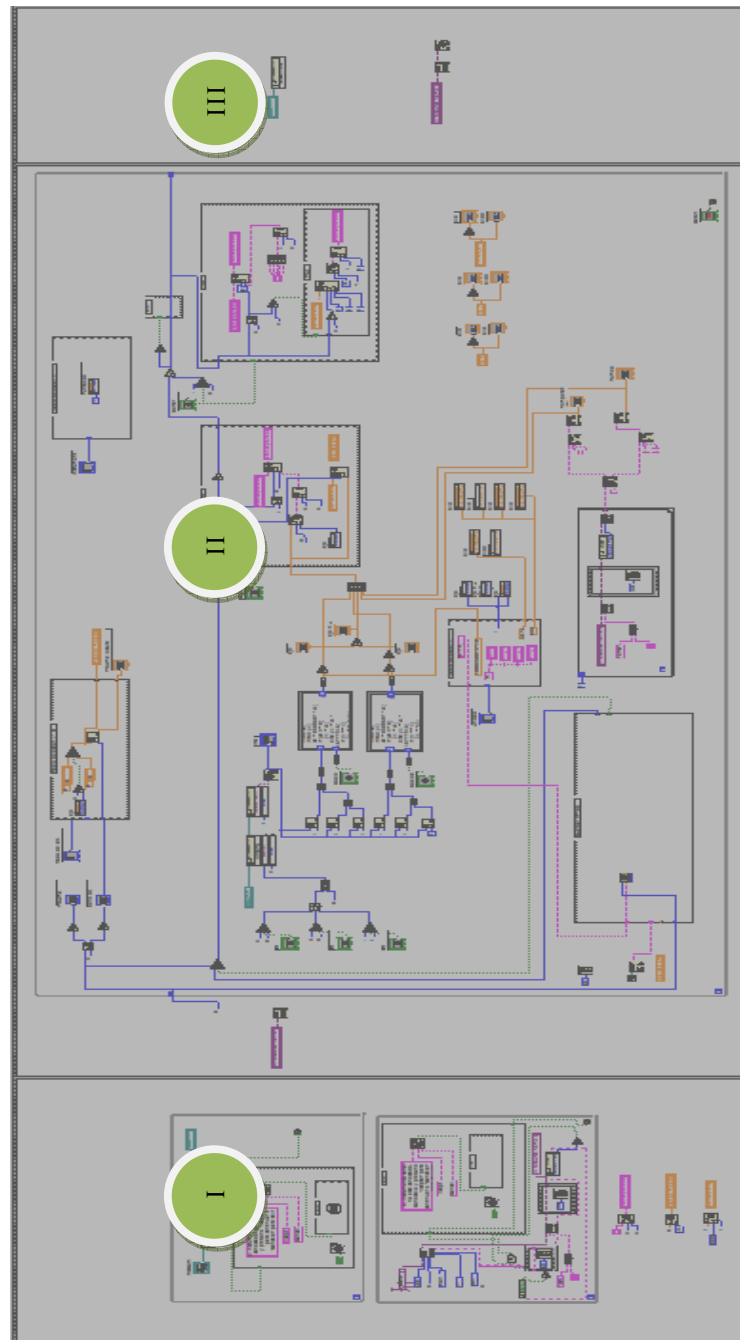
### 2.3.1. Programación para el PCMIC

El procedimiento de calibración incorpora dos instrumentos de medición, el termohigrómetro digital con interfaz RS-232-USB y la estación de medición unidimensional con USB. El objetivo principal es lograr la comunicación con ambos dispositivos de instrumentación y obtener sus mediciones en el instante que especifica el método de calibración.

La figura 2.33 muestra la estructura general de ejecución del software, se trata de una secuencia de programación denominada “*flat sequence structure*”, conformada por tres subdiagramas de programación, los cuales se ejecutan secuencialmente de izquierda a derecha, en caso de no terminar cierta tarea de programación adecuadamente, el programa no continuará su ejecución. La figura ilustra tres bloques de programación dentro de las cuales se pueden implementar diferentes estructuras de programación encaminadas a una tarea específica.

- Zona I. Inicializar instrumentos (habilitación de puerto serie) y dispositivo de interfaz humana (HID), así como la declaración de variables.
- Zona II. Programa principal.
- Zona III. Liberación de puerto serie y desconexión de dispositivo de interfaz humana.

Empezaremos con la descripción de la programación en la zona I.



**Figura 2.33.** Secuencia general de ejecución del software para PCMIC.

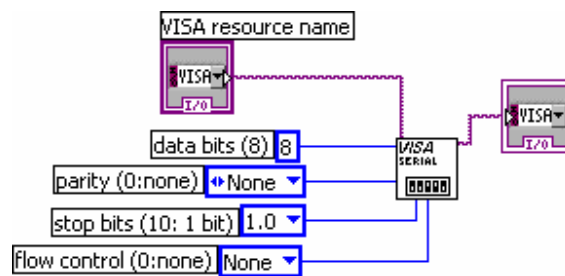
### Configuración de puerto serie

La comunicación por interfaz RS-232 es un protocolo comúnmente utilizado por dispositivos de instrumentación. El ciclo de comunicación se conforma por el envío y recepción de bytes de información, un bit a la vez. Las características más importantes de la comunicación serial son la velocidad de transmisión, paridad, número de bits de datos y de alto. Para que dos puertos se puedan comunicar,

es necesario que los parámetros de transmisión sean iguales entre el puerto serial del instrumento y la PC.

Se debe programar la secuencia de código adecuadamente, incluyendo los caracteres de término, para asegurarse que la comunicación se logre con éxito. Existe un nuevo estándar para comunicarse con instrumentos en LabVIEW, conocido como Arquitectura de Software para Instrumentos Virtuales (VISA). El estándar VISA unifica todos los protocolos de comunicación utilizados en la industria de la instrumentación, para hacer software que pueda ser interpretado y reutilizado, sin importar la operación final del instrumento.

La figura 2.34 muestra la configuración del puerto serie utilizando funciones VISA y configurando acorde a las características de comunicación del termohigrómetro digital. Cada vez que se conecta un dispositivo serie utilizando un convertidor a USB, el administrador de hardware le asigna un número de identificación correspondiente a la ranura USB. Al configurar la función VISA de inicio de puerto serie se debe indicar la ubicación del instrumento en el parámetro “VISA resource name” [21].



**Figura 2.34.** Configuración de puerto serie.

### **Comunicación con dispositivos de interfaz humana**

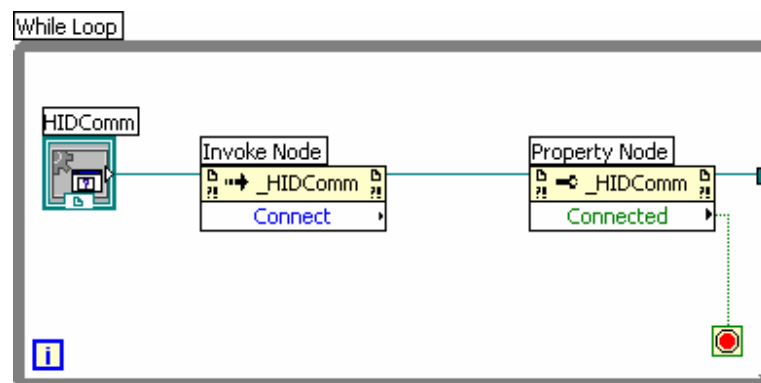
Con base en la característica que LabVIEW tiene para incorporar controles ActiveX, se implementó un control de Dispositivo de Interfaz Humana (HID) para recibir datos provenientes de un dispositivo USB.

De este modo al insertar la tarjeta se activa el control ActiveX de HID quien identifica al nuevo dispositivo en el administrador de hardware.

Por medio de éste la tarjeta se identifica en el ambiente de programación LabVIEW creando un bloque funcional asociado con el control ActiveX, el cual puede ser utilizado como una referencia o nodo en la estructura de programación.



La figura 2.35 muestra el desarrollo descrito anteriormente dentro de un ciclo *while*. El primer elemento es el control ActiveX, después se utiliza un bloque funcional denominado *Invoke Node*, el cual asocia un determinado método o función a una referencia o nodo establecida por el programador, en este caso se trata del HIDComm. El método asociado se identifica como *Connect* y consiste en identificar la presencia de un dispositivo HID, en caso de no encontrarse se genera un error en la secuencia de programación. Con el bloque funcional *Property Node* se puede leer o escribir sobre la referencia, en este caso se lee el estado en que se encuentra el control ActiveX, en caso de estar desconectada la tarjeta, no se dará por concluido el ciclo *while* y la secuencia del programa no continuará [24].



**Figura 2.35.** Implementación de ActiveX en LabVIEW.

Esta es la manera en la que una conexión física con USB puede ser visible en nuestro ambiente de programación, con esta sencilla implementación se puede programar diversas aplicaciones entorno a la tarjeta decodificadora invocando la referencia ActiveX y asociándole distintos métodos.

Para ilustrar la idea de secuencia en esta zona, se describirá el código de la figura 2.36, el cual realiza la labor de identificar la tarjeta decodificadora, en caso de estar desconectada se abre un cuadro de diálogo que incluye un mensaje que reporta el estado de conexión y que además solicita una acción de respuesta. Si la respuesta no es satisfactoria no se puede acceder al programa principal.

Profundizando en la figura 2.36, se utiliza una estructura de programación *case structure*, cuenta con paneles programables para establecer una elección entre falso o verdadero sobre el estado actual de la referencia ActiveX. Para el caso verdadero, la tarjeta está conectada, entonces no se programa ninguna tarea y por consiguiente el ciclo *while* termina su ejecución.

Para el caso falso, se programa un cuadro de diálogo que indica el estado de conexión de la tarjeta, si se opta por cancelar llevará al bloque de *stop* que terminará la ejecución del programa. Si la opción elegida es aceptar se volverá a leer el estado del control ActiveX y no dará paso al programa principal hasta que la tarjeta decodificadora haya sido conectada.

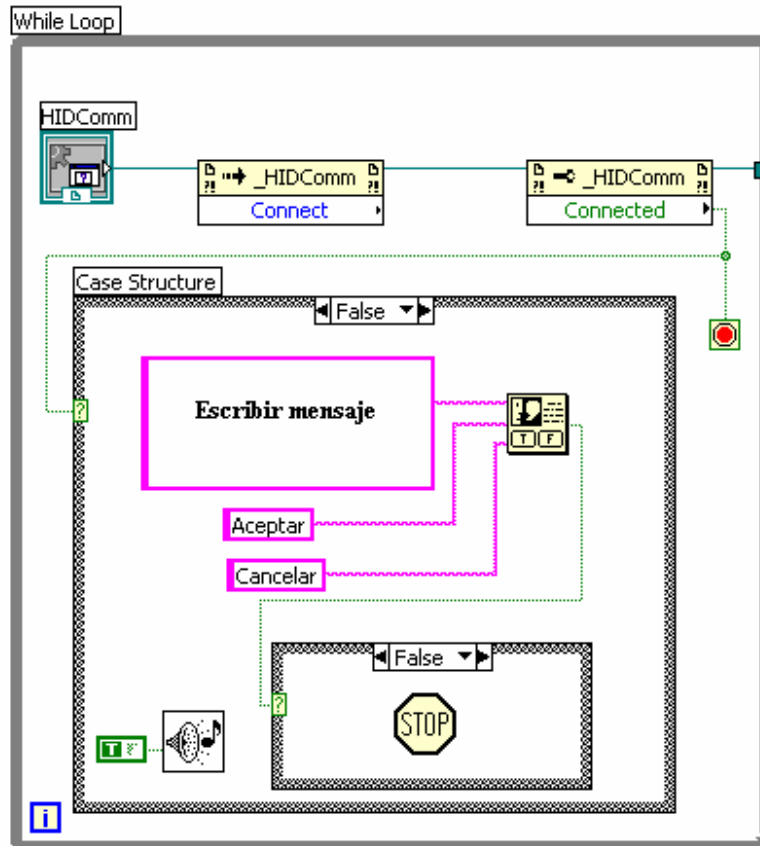
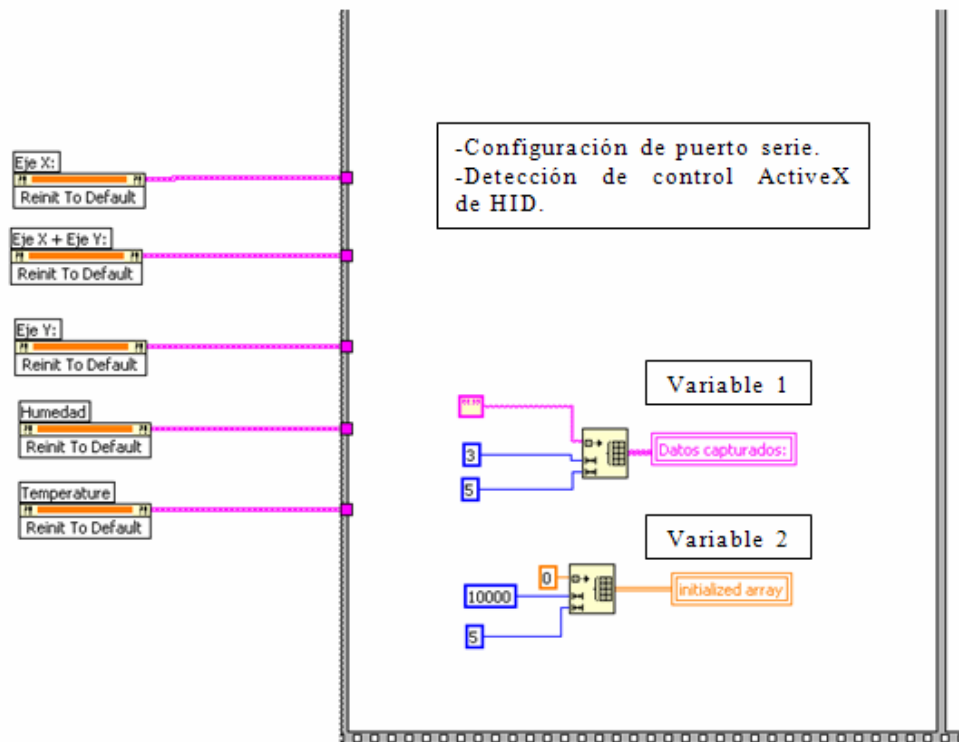


Figura 2.36. Detección de control ActiveX.

### Declaración de variables

La figura 2.37 muestra la declaración de dos variables en la zona I, ambas son llamadas por diversas rutinas en el programa principal con el objetivo de almacenar información. LabVIEW cuenta con tablas o arreglos para este fin, y además se puede declarar como variables locales. La variable 1 es una tabla donde en forma permanente se desplegarán las mediciones realizadas por los instrumentos, maneja datos tipo *string*. La variable 2 es un arreglo de datos tipo *double* con la información de la instrumentación. Los bloques conectados a la izquierda del bloque restablecen a cero el valor de cualquier indicador numérico que se encuentre en el programa principal.



**Figura 2.37.** Declaración de variables.

El programa principal en la zona II se centra en la adquisición y procesamiento de datos de los instrumentos correspondientes al PCMIC; el diagrama de bloques de la figura 2.38 muestra la programación para establecer un circuito de comunicación entre instrumentos, obsérvese que la ejecución de envío y recepción se realiza constantemente y sin retrasos de información.

Así mismo se coloca el botón de control que permite detener la ejecución del programa principal.

Sobre los datos adquiridos se establecen diversas tareas de procesamiento y requerimientos del PCMIC, figura 2.39. La adquisición de datos es continua, mientras que las demás funciones están condicionadas por el estado activo de un botón de control o condición dentro de una estructura de programación independiente.

### **Comunicación con control ActiveX HID**

El objetivo a conseguir es el envío de información de control hacia el microcontrolador, así como la recepción del conteo de pulsos de la señal en cuadratura. La figura 2.40 muestra la implementación en LabVIEW.

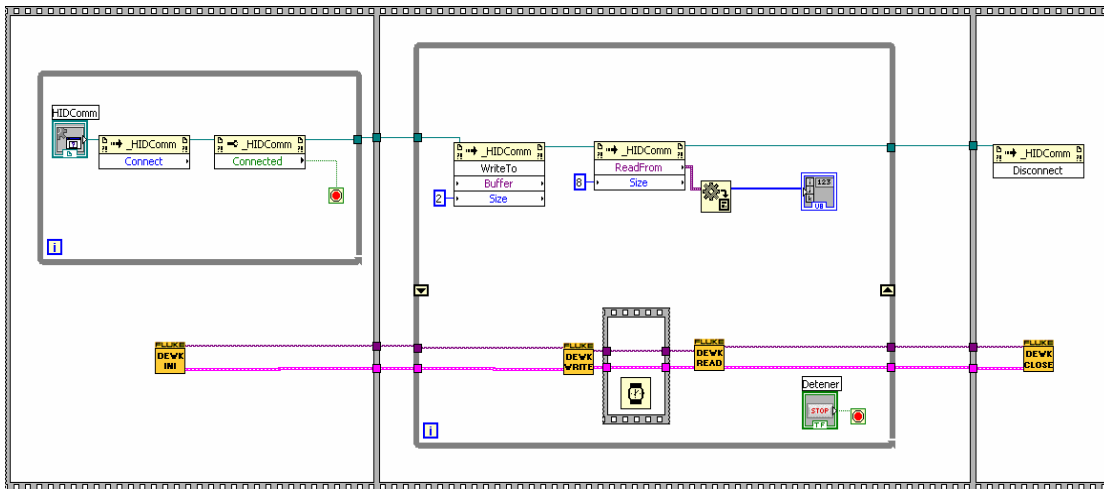


Figura 2.38. Adquisición de datos.

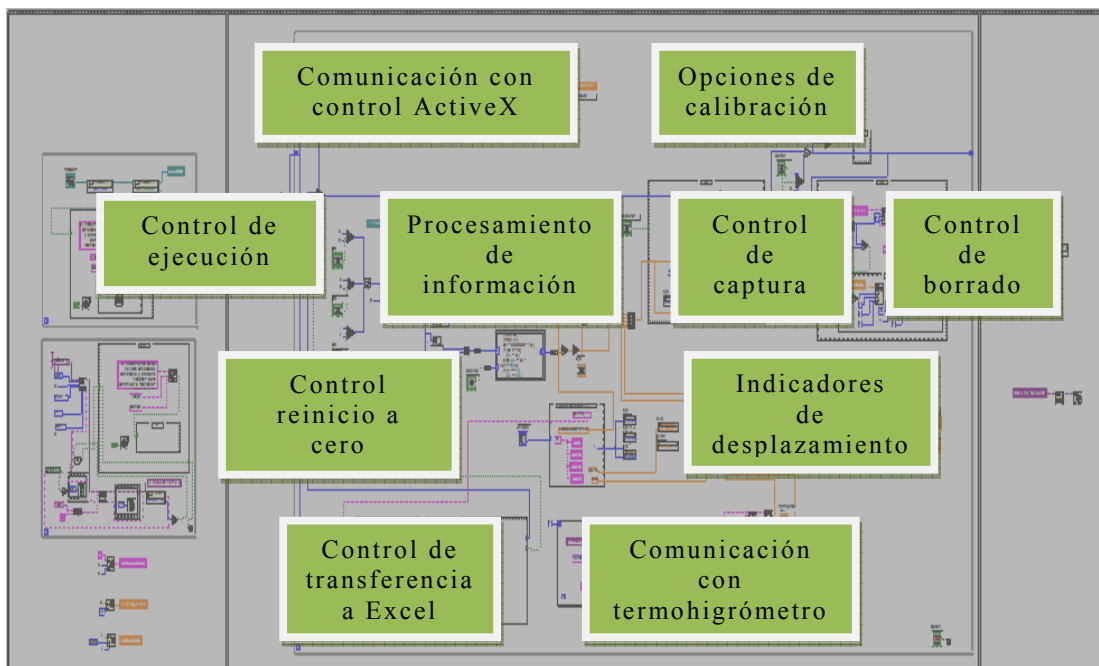
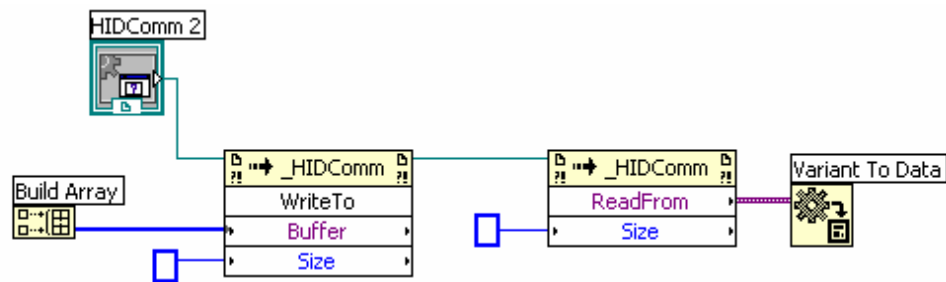


Figura 2.39. Tareas de realizadas por el programa principal.

Anteriormente se comentó el uso de *invoke node*, pues nuevamente utilizando la referencia ActiveX se programan nuevos métodos, *Write To* y *ReadFrom*, ambos deben ser configurados en *Size* con el número de bits que se van a transmitir y recibir respectivamente.



**Figura 2.40.** Envío y recepción de información a través del control ActiveX.

Para escribir información, previamente se guarda en un arreglo y posteriormente se transfiere al *buffer* de salida de la computadora. Para recibir datos adecuadamente se tiene que hacer una conversión utilizando el bloque funcional *Variant To Data* para lograr la compatibilidad con LabVIEW.

En el diagrama de flujo para el microcontrolador de la figura 2.19, se estableció el envío de información en forma continua siempre y cuando recibiera un cero el microcontrolador, por lo tanto no se implementa ninguna secuencia o condición.

### Control de reinicio a cero

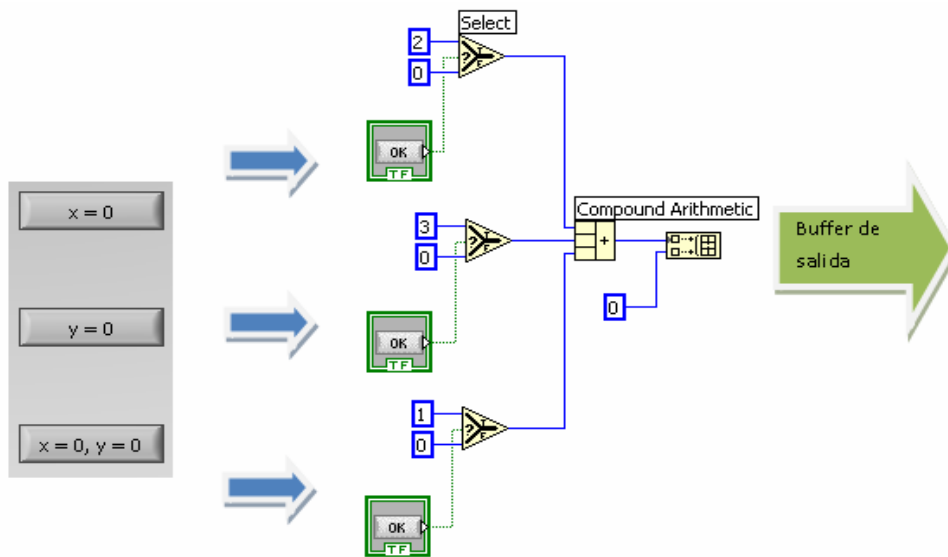
El operador necesitaba reiniciar los ejes a cero en determinado punto del procedimiento de calibración, en otras palabras tener el control desde LabVIEW para reiniciar el conteo del microcontrolador.

La figura 2.41 muestra en el lado izquierdo los botones que se generan en el panel de control, cada botón genera determinada información al ser presionado, y la envía al buffer de salida para transferirla al microcontrolador.

Se emplea un expresión de comparación denominada *select*, realiza una elección entre los valores escritos en sus dos terminales, mientras el botón no se presione enviará constantemente el valor cero y el microcontrolador mandará el conteo de pulsos en cuadratura.

En caso de presionar el botón, el estado de la selección cambiará al otro valor. Puede ser 1, 2 o 3 y en consecuencia el conteo reinicia para el eje X, Y o ambos, respectivamente.

La información de los selectores es sumada y agrupada en un arreglo, por default se agrega el cero para mantener al microcontrolador activo en todo momento, el otro valor agregado reiniciará la cuenta o se tratará de otro cero, finalmente la información es transmitida.



**Figura 2.41.** Control de reinicio de ejes de desplazamiento.

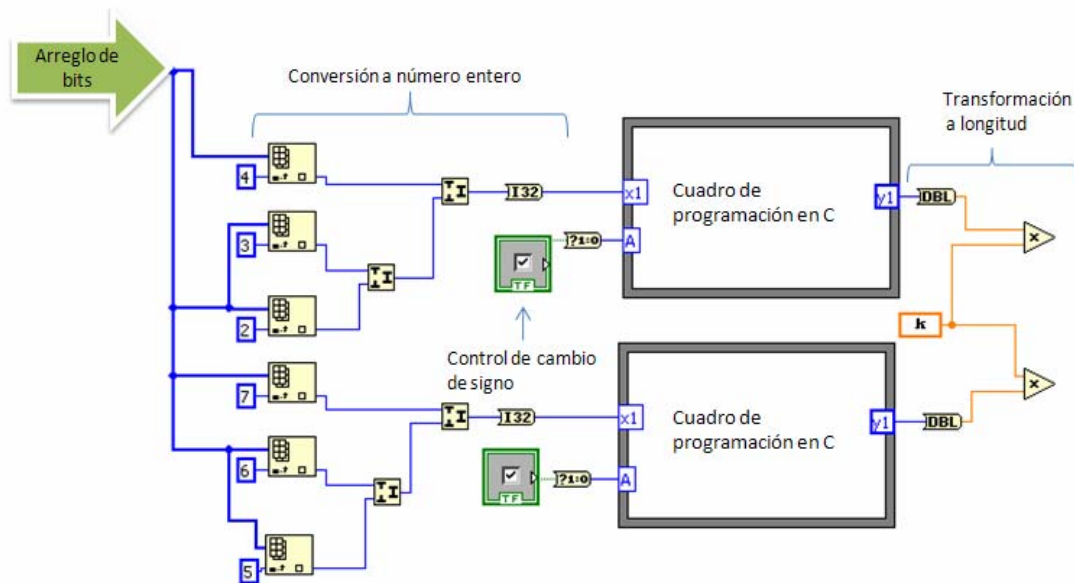
### Procesamiento de información

La figura 2.42 muestra el código que se encarga de transformar los bytes en una magnitud de longitud. La recepción de datos consiste en agrupar en arreglos de bytes la información proveniente del control ActiveX, de los cuales 2 bytes son utilizados para escribir información de control al microcontrolador y los 6 restantes son para recibir los 3 bytes del conteo de cada eje de desplazamiento.

La parte de conversión a número entero concatena 3 bytes de cada eje para formar una palabra de 24 bits, la cual es transformada en un entero de 32 bits en el rango de  $[-(2^{31})$  a  $(2^{31})-1]$ . Se utiliza 32 bits, ya que LabVIEW no presenta una conversión exclusiva para 24 bits.

Después de la conversión, la información entra a un cuadro de programación en C. Donde se tiene como entradas el dato del conteo almacenado en la variable x1 y el estado booleano del control de cambio de signo en la variable A.

El cuadro de programación en C tiene la función de cambiar el signo de la magnitud de desplazamiento utilizando un control en la interfaz gráfica de usuario.



**Figura 2.42.** Conversión de bits a longitud.

El diagrama de flujo de la figura 2.43 ilustra la secuencia a seguir en el cuadro de programación C. Consiste en leer el valor entero  $x1$ , se compara con el límite 1,  $[-(2^{23}) a (2^{23})-1]$ , correspondiente al máximo entero positivo que se puede formar con 24 bits. Si entra en rango, el dato se almacena en  $y1$  y esta a la espera de leer el estado booleano de la variable A. En caso contrario se trata de un número negativo y se procede a tomar el discriminante surgido entre la diferencia del valor recibido y el valor total de  $2^{24}$ . Después se lee el estado booleano del control de signo, si esta activo se multiplicará por uno negativo, en caso contrario  $y1$  se establece como resultado final.

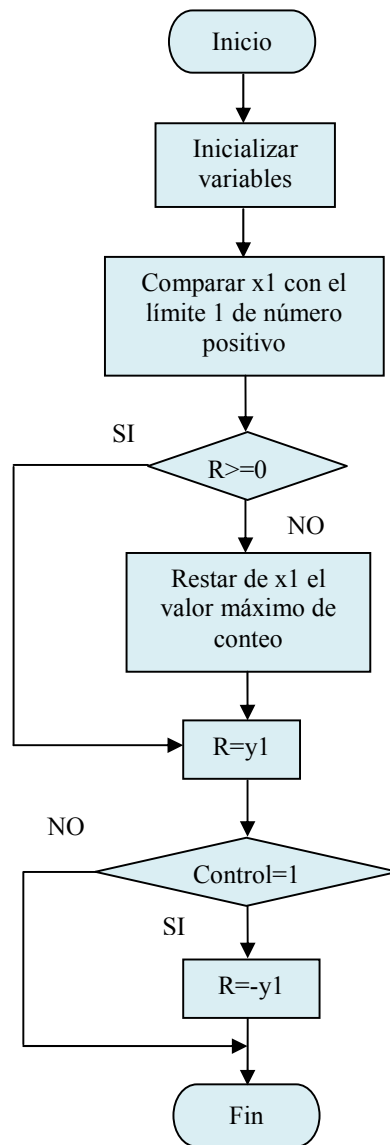
A la salida del cuadro de programación C,  $y1$  es un dato del tipo *double*, el cual es multiplicado por una constante de cambio, determinada por la relación total de conteos entre el desplazamiento total de la estación de medición unidimensional.

### Comunicación con el termohigrómetro

El proceso de comunicación consiste en dos sencillas tareas de aplicación de funciones VISA, *Write* y *Read*. La figura 2.44 utiliza ambas funciones para establecer comunicación con el termohigrómetro. Entre ambas tareas se debe establecer un tiempo de espera, para este instrumento se estableció un tiempo de 250 milisegundos. La función *read* necesita recibir el número de bytes adecuado del puerto serie de la computadora.

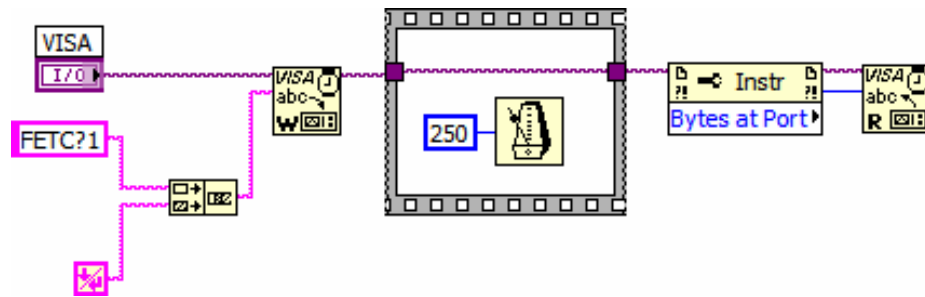
La palabra FETC?1 es una instrucción del instrumento que solicita la lectura más actual para el canal 1 del termohigrómetro. De esta

manera es posible recibir continuamente el valor de temperatura y humedad.



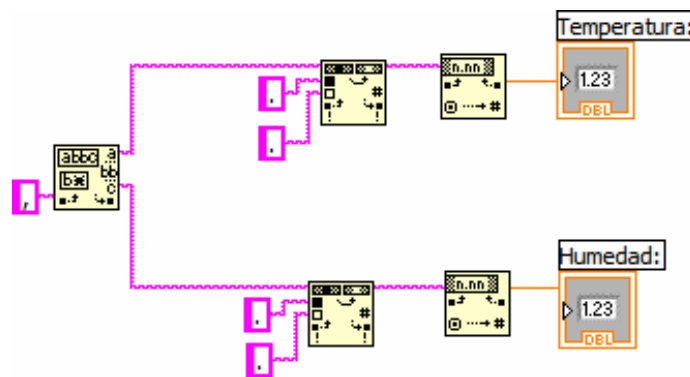
**Figura 2.43.** Diagrama de flujo para control de cambio de signo.





**Figura 2.44.** Ciclo de comunicación por puerto serie.

La instrucción FETC?1 es un dato tipo “string”, por lo que LabVIEW recibe también una respuesta en este formato, es necesario determinar la forma de respuesta de cada instrumento y procesarla adecuadamente a otro tipo de dato manejable para operaciones numéricas o simplemente para poder desplegarla en un indicador. La figura 2.45 realiza el tratamiento adecuado para la conversión de un dato tipo *string* a un dato tipo *double*.

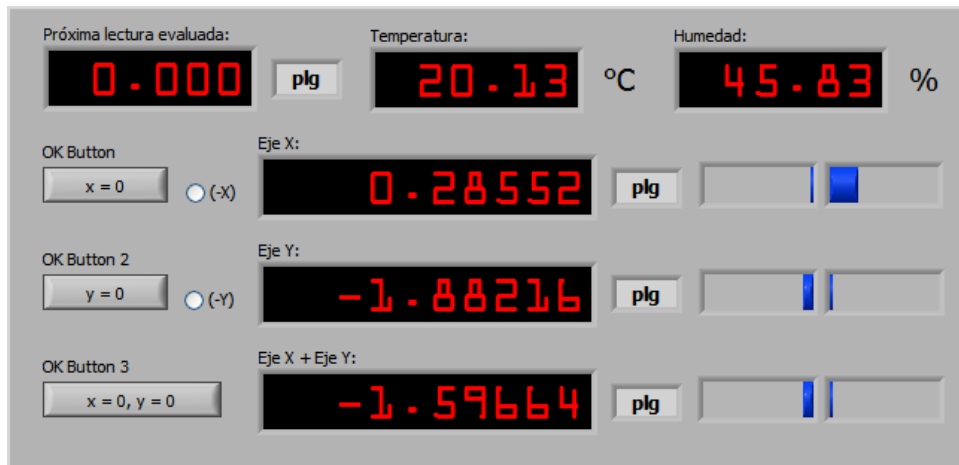


**Figura 2.45.** Conversión de dato tipo *string* a tipo *double*.

De esta forma se puede ver en un indicador numérico en la pantalla de usuario los valores de temperatura y humedad.

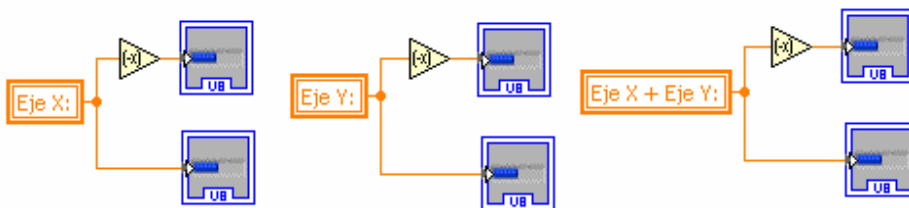
### Indicadores de desplazamiento

Como ayuda visual para el operador, además de los indicadores numéricos para los ejes de la estación de medición unidimensional, se solicitaron indicadores de barras horizontales como se muestra en la figura 2.46.



**Figura 2.46.** Indicadores numéricos y de barras para medición de longitud.

La programación en la figura 2.47 consiste en almacenar los datos obtenidos en el procesamiento de datos en variables que escriban la información a los indicadores de barras y numéricos. Con esto la información es vista en forma gráfica por el usuario.



**Figura 2.47.** Diagrama de bloques para indicadores numéricos y de barras.

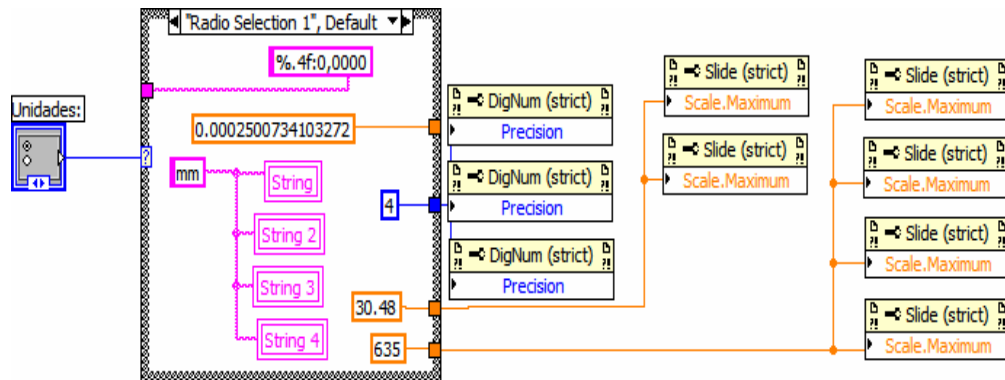
### Selección de unidades de calibración

Obsérvese en la figura 2.46 las unidades en pulgadas para la calibración, si el instrumento viene en milímetros se puede cambiar de unidad mediante un selector.

Se tienen que cambiar varios parámetros dentro del programa:

- La constante que convierte el conteo del microcontrolador.
- La precisión en dígitos es diferente entre las unidades de medición.
- La precisión en el desplazamiento de la barra cambia de acuerdo al desplazamiento máximo en pulgadas o milímetros.
- La unidad en todos los indicadores cambian a la unidad correspondiente.

El diagrama de bloques con la correspondiente programación se muestra en la figura 2.48.



**Figura 2.48.** Diagrama de bloques para el selector de unidad.

### Control de ejecución

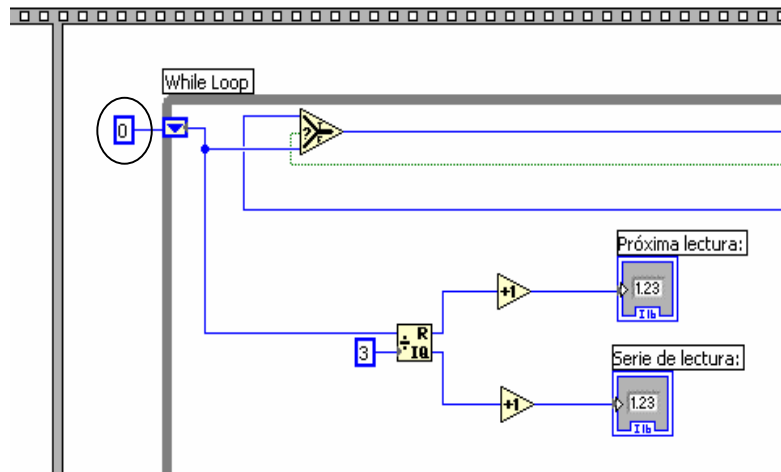
Anteriormente se comentó que todas las tareas de programación se realizan de forma independiente, pero todas están dentro de un *while loop*. El PCMIC es un proceso que requiere tomar determinado número de mediciones y series, dentro de los cuales se transfiere información a Excel, sin embargo el operador puede eliminar y reemplazar una medición que no sea de su aprobación sin que esto implique la pérdida de información tomada con anterioridad.

La interfaz de usuario muestra indicadores del número de serie y medición actual, así como el siguiente punto en el procedimiento de calibración. Para hacer esto posible se lleva un registro sobre el ciclo *while*, el cual genera un índice numérico de ejecución. Cada vez que se realice un ciclo completo se incrementará.

El círculo negro en la figura 2.49 encierra el índice numérico correspondiente al turno de ejecución que se asigna al ciclo *while*, este número es dividido entre tres, corresponde a las repeticiones de mediciones que se realizan en una serie. El bloque funcional entrega el cociente y el residuo de la división. Al superar 3 iteraciones de lectura el cociente es 1 y debido al incremento muestra la segunda serie en el indicador de “Serie de Lectura”, mientras que el residuo es cero y debido al incremento se vuelve a establecer en la primera lectura.

Al realizar la captura de información, la ejecución del programa principal incrementa este índice, dicho incremento se encuentra condicionado por el cambio a estado activo del botón de control de “Borrar Dato”. Si se mantiene en estado inactivo la serie de información actual se coloca en la tabla de visualización. Pero en caso de eliminar la información, provocará que el registro del ciclo *while* no incremente, lo cual permitirá realizar nuevamente las tareas de captura y transferencia de información a Excel para realizar una nueva

medición sin romper la secuencia establecida en el procedimiento de calibración.



**Figura 2.49.** Registro de control en el ciclo *while*.

### Control de captura de información

El siguiente punto es desarrollar un control que adquiriera toda la información en el instante de aprobación del operador, y que sea visible en una tabla en la pantalla de usuario. La figura 2.50 muestra la tabla donde el operador visualizará las mediciones de la estación de medición unidimensional y del termohigrómetro.

Datos capturados:					
	Eje X	Eje Y	X + Y	Temperatura	Humedad
Medición 1					
Medición 2					
Medición 3					

**Figura 2.50.** Tabla de visualización.

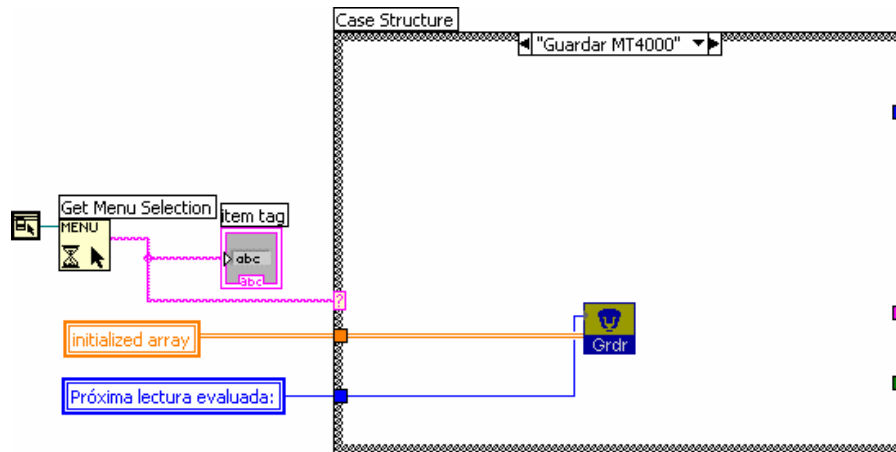
La figura 2.51 muestra el desarrollo para adquirir la información de ambos instrumentos al activarse el botón, lo cual provocará que se ejecute el caso verdadero del *case structure*, para el caso falso no se implementa ninguna tarea, por lo que el ciclo termina, permitiendo la libre ejecución de otras tareas dentro del software. Básicamente los siguientes cinco bloques funcionales conforman el proceso.

- 1) Los datos tipo *double* con la información de los instrumentos son introducidos en un arreglo.
- 2) Realiza una transformación a dato tipo *string* compatible con la tabla de visualización de usuario.



- Guardar la información de calibración en un archivo establecido por el usuario.
- Abrir un archivo de Excel que haya quedado pendiente y poder reiniciar en ese punto la calibración cuando así lo disponga el operador.

Para realizar la operación de guardado se utilizó el código de la figura 2.52 utilizando el subVI “Guardar”.



**Figura 2.52.** Transferencia de información a Excel.

La figura 2.53 muestra la estructura interna del SubVI “Guardar”, una estructura conformada por un ciclo *while*, en su interior se inserta un *flat sequence structure* definido con tres zonas de secuencia.

- Zona I Iniciar arreglo.
- Zona II Generación de archivos nuevos.
- Zona III Guardar información con extensión xls.

Este SubVI recibe como entrada un arreglo numérico, es un arreglo en el que se almacena paulatinamente la información de medición.

La zona II crea un directorio de las unidades de almacenamiento de información disponibles en el sistema y donde el usuario puede elegir o crear un nuevo archivo. Se implementaron dos botones de control para confirmar o cancelar la generación del archivo.

La zona III declara como variables el nombre de ambos archivos y mediante el uso de bloques funcionales para la generación de reportes se genera un archivo de tipo Excel. Se configura la presentación de la información, se construye una tabla con la precisión numérica adecuada a la elección de unidades y se le asigna un nombre a la hoja donde se almacenará el arreglo con los puntos de calibración.



Una vez definido el archivo, se procede a elegir el tipo de datos que serán enviados y se especifica el número de renglón y columna en Excel donde se va a empezar a insertar la información.

La figura 2.54 muestra la rutina con el SubVI “Abrir” que se encarga de recuperar un archivo pendiente a calibrar. No tiene variables de entrada, solo recibe la referencia del archivo a recuperar, el cual es ingresado en ejecución del programa como se aprecia en la figura 2.55

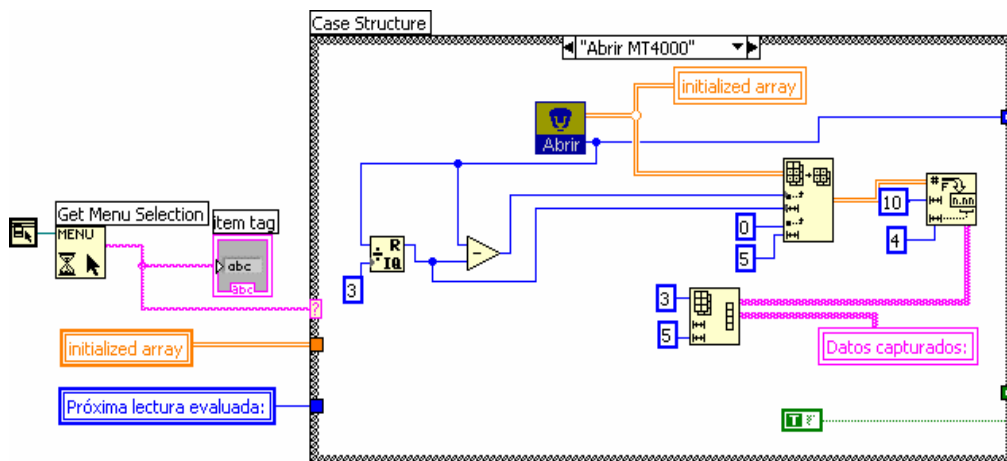


Figura 2.54. Generación de archivo tipo Excel.

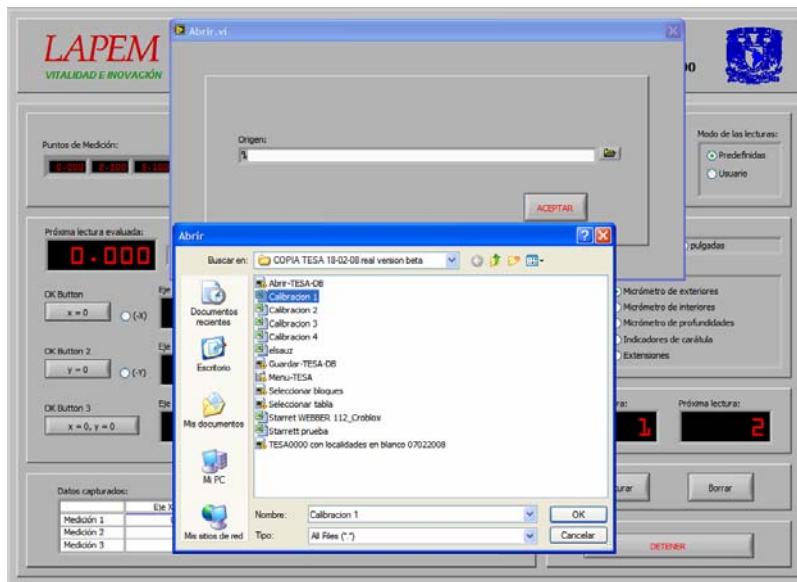
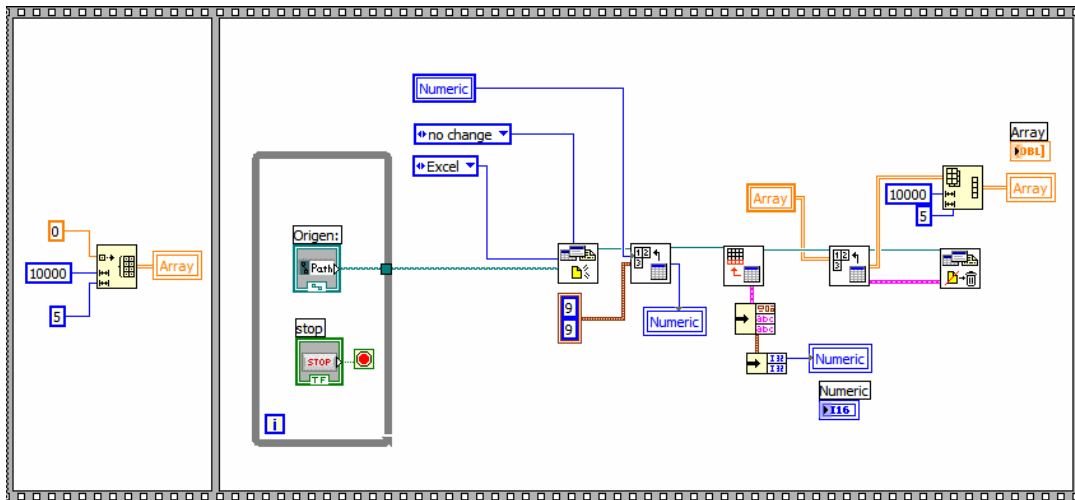


Figura 2.55. Interfaz de usuario para abrir archivo de calibración.

La figura 2.56 muestra el contenido del SubVI “Abrir”. En el ciclo while se inserta un bloque funcional donde se escribe el nombre del nuevo archivo que se quiere generar.



El SubVI está construido con bloques de generación de reportes, conociendo la tarea que realiza cada uno solo se procede a configurar con base a los parámetros deseados y por supuesto siguiendo una secuencia coherente.



**Figura 2.56.** SubVI Abrir.

Este SubVI se le asigna un archivo como referencia, y por medio de una función, lee la información de la hoja de Excel e identifica por celda el último renglón donde se escribió. De esta manera genera dos salidas de información. Un arreglo con la información de calibración pendiente y un valor numérico que sirve para restablecer la secuencia del programa en el punto donde fue interrumpido anteriormente [16].

### Control de borrado

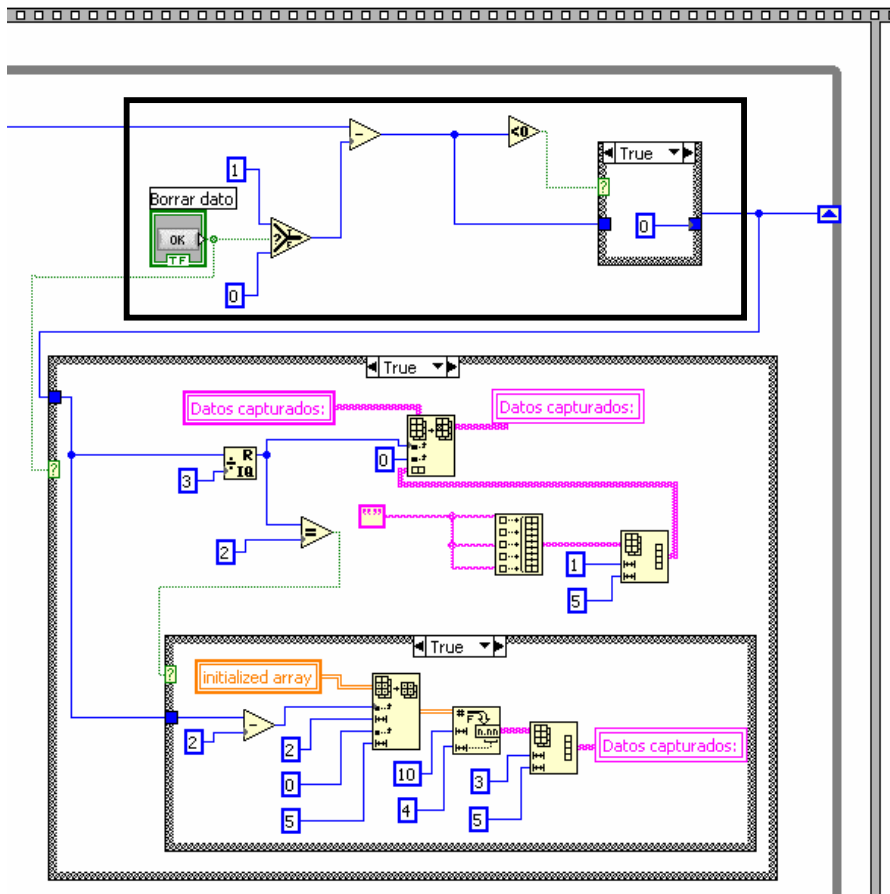
Para explicar la rutina de borrado tomaremos como referencia la figura 2.57. La rutina se conforma de dos *case structure* que toman como referencia el estado lógico del botón de control “Borrar Dato”. La programación correspondiente para el caso activo lee el número de registro de ejecución del ciclo *while*, que externamente tiene una programación lógica complementaria.

Si el control de “Borrar Dato” no es presionado, el estado es cero y no se ejecuta ninguna rutina en la *case structure*, entonces el registro de ejecución pasa sin problemas la estructura lógica implementada en el rectángulo negro, incrementándose normalmente y preparado para capturar otra serie de mediciones.

En caso activo el registro de ejecución es decrementado y no da por terminado el ciclo *while* del programa principal. Este valor entra al *case structure* y reemplaza un renglón de la serie de datos capturados por cadenas de ceros. Entonces la secuencia de programación para

capturar información y enviar a Excel se decrementa una vez, por lo que al realizar una nueva no se perderá la secuencia del procedimiento y remplazará coherentemente con el valor en turno, tanto en la tabla de interfaz de usuario como en el archivo Excel donde se este guardando la información.

En caso de solicitar borrar una serie completa, se ejecuta el otro *case structure*, el cual remplaza los valores por un arreglo de ceros de 3 renglones por 5 columnas.



**Figura 2.57.** Diagrama de bloques para el control de borrado.

Para concluir el software del PCMIC solo falta incluir las tareas de programación para la zona III. Básicamente corresponde en cerrar adecuadamente la referencia del control ActiveX y el puerto serie RS-232. El diagrama de bloques de la figura 2.58 cierra la referencia del control ActiveX utilizando un nuevo método “*disconnect*”, mientras que para cerrar adecuadamente el termohigrómetro digital utiliza el SubVI “Close”.

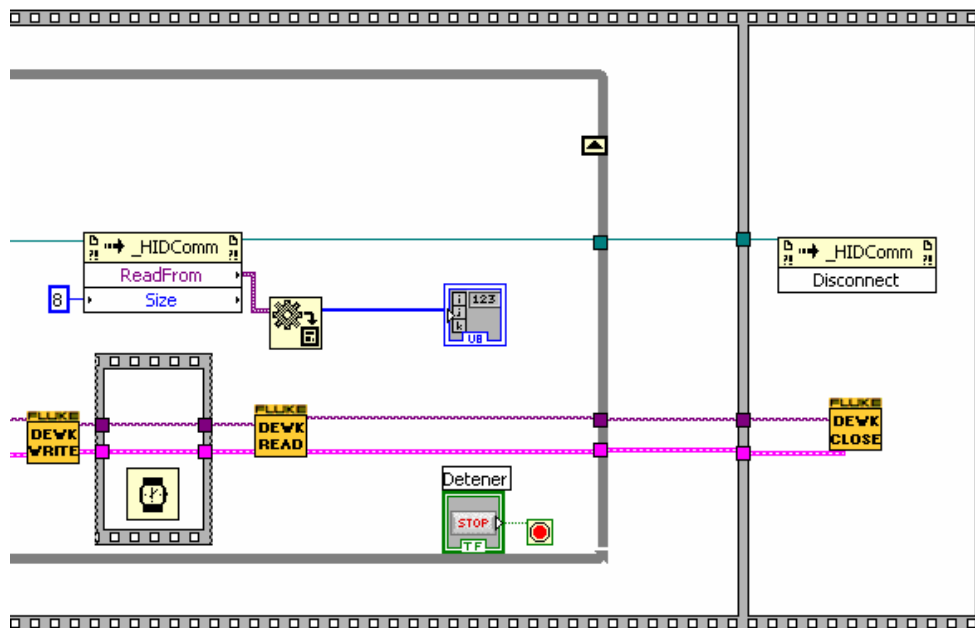


Figura 2.58. Liberación de recursos del sistema.

El SubVI “Close” se forma con las funciones mostradas en la figura 2.59, se vuelve a llamar la referencia del instrumento serie, la primera instrucción verifica el buffer de entrada y en caso de haber algo lo elimina, esto para que al iniciar una nueva serie de calibración conserve valores que sean actuales. Al final se emplea una función VISA que cierra adecuadamente y con toda seguridad el instrumento.



Figura 2.59. Diagrama de bloques para el SubVI close.

### 2.3.2. Programación para el PCBP

El procedimiento de calibración de bloques patrón requiere incorporar información de tres instrumentos de medición; el comparador de bloques patrón, el termohigrómetro digital y el termómetro de resistencia de platino. El objetivo principal es adquirir la información de temperatura y humedad en el instante que el operador de por aprobado una medición correcta con el comparador de bloques patrón.

Una de las ventajas de LabVIEW que fue de gran ayuda, fue la posibilidad de reutilizar código con el uso del SubVI. Muchas de las tareas de programación fueron afines para el PCMIC y fueron utilizadas para el software del PCBP.

La figura 2.60 muestra la estructura general de ejecución del software, se retoma la misma base utilizada para el PCMIC, la secuencia de programación denominada “*flat sequence structure*”.

- Zona I Inicializar instrumentos (habilitación de puerto serie y dispositivo de trabajo con GPIB) y declaración de variables.
- Zona II Programa Principal.
- Zona III Liberación de puerto serie y desconexión de GPIB.

Para la zona I tenemos una actividad ya realizada con anterioridad y es la habilitación del puerto serie, lo nuevo es la apertura de GPIB. Se declaran en igual forma los dos tipos de variables para datos tipo *string* y numérico.

### **Inicio de comunicación serie RS-232**

El objetivo principal se orienta a la operación del comparador de bloques patrón, instrumento principal en el PCBP, el cual cuenta con interfaz RS-232. La figura 2.61 muestra el empleo de la función VISA de configuración de puerto serie, se inicia el puerto de la PC con las mismas características de transmisión de la unidad central de procesamiento del comparador de bloques patrón.

Anteriormente en el PCMIC se describió la programación que inicia adecuadamente el puerto serie y por tratarse del mismo termohigrómetro que utilizó funciones VISA, fácilmente se incorpora la misma secuencia de inicio para PCBP, razón por la cual no se mencionará en este apartado.

### **Inicio de comunicación con GPIB**

Las funciones VISA son compatibles a los protocolos GPIB. La figura 2.62 muestra el inicio de esta interfaz. Se necesita conocer la dirección del instrumento; para este caso se le asignó la dirección 22 al termómetro de resistencia de platino. Además se debe establecer un tiempo de espera razonable antes de iniciar cualquier tarea con el instrumento [18].

El programa principal en la zona II se centra en la adquisición y procesamiento de datos de los instrumentos correspondientes al PCBP; el diagrama de bloques de la figura 2.63 muestra la programación para establecer un circuito de comunicación entre los instrumentos. Como observación, el termómetro de resistencia de platino no hace mediciones inmediatas de cada uno de sus puntas de medición, consume cierto tiempo de espera entre el cambio de cada uno de sus canales primarios.

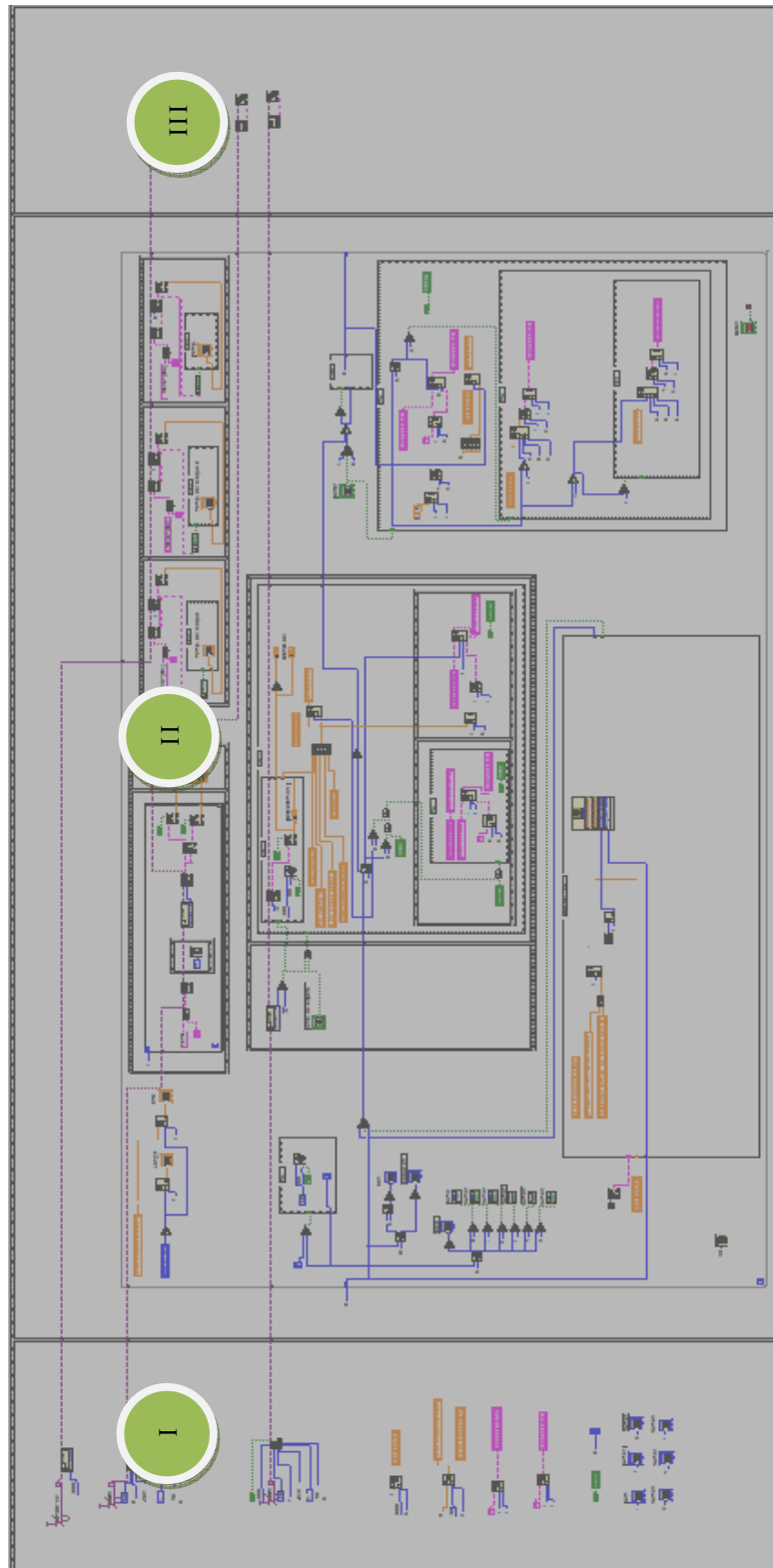
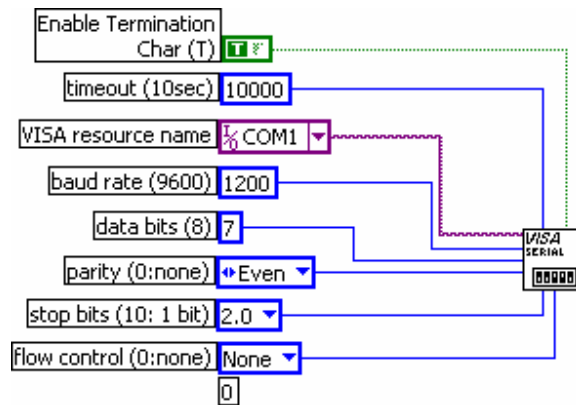
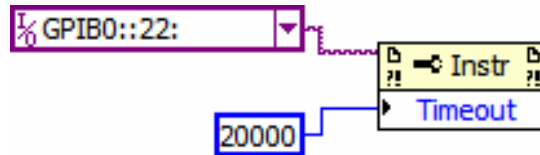


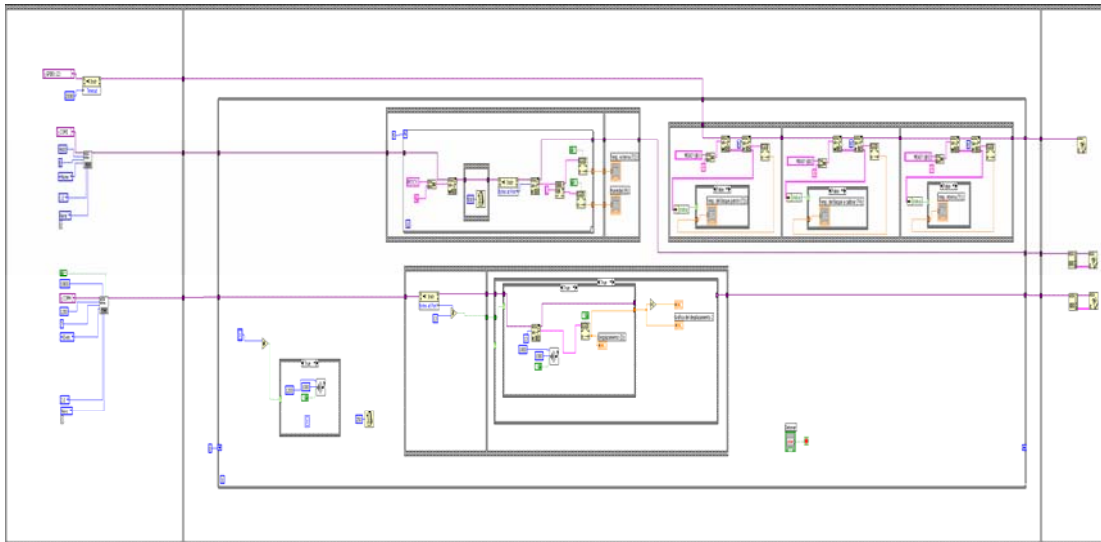
Figura 2.60. Secuencia general de ejecución de software para el PCBP.



**Figura 2.61.** Configuración de puerto serie para la unidad central de procesamiento.



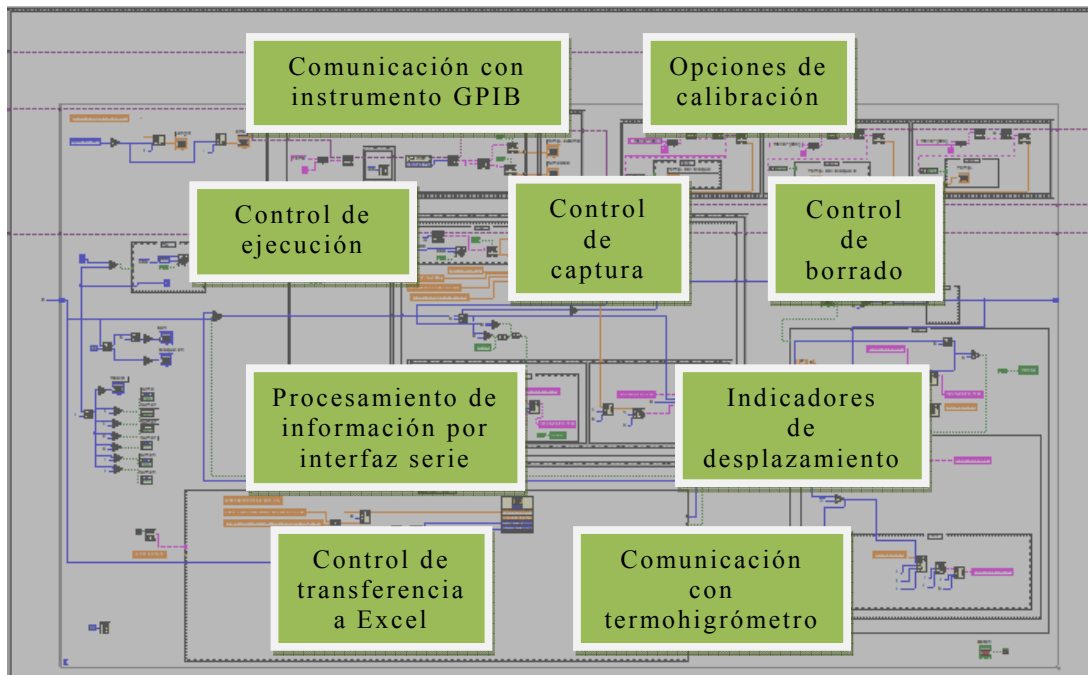
**Figura 2.62.** Diagrama de bloques para iniciar GPIB.



**Figura 2.63.** Diagrama de bloques para la comunicación de instrumentos de medición.

Sobre la información de calibración se establecen varias tareas de programación, figura 2.64, relacionadas con el procesamiento o con requerimientos establecidos por el PCBP. La adquisición de datos

estará disponible al accionar el control del comparador de bloques patrón.



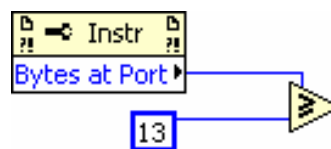
**Figura 2.64.** Tareas de programación para el PCBP.

### **Comunicación serie con termohigrómetro**

Se utiliza el mismo diagrama de bloques que se utilizó en el PCMIC mediante la creación de un SubVI. Por lo tanto no se mencionará nada adicional sobre la programación.

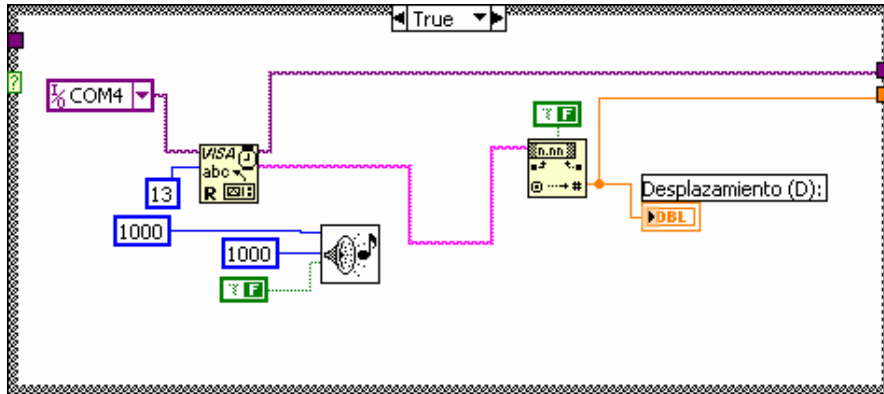
### **Comunicación con el comparador de bloques patrón**

Al pulsar el botón del comparador de bloques patrón se inicia la transmisión de información hacia la computadora. En ese momento se crea la condición para leer el número de bits que recibe el buffer de entrada de la computadora. Como lo muestra la figura 2.65 se compara la variable con 13 bits. Al recibir 13 bits de información los demás instrumentos reciben la orden para capturar información, ya sea temperatura o humedad.



**Figura 2.65.** Condición de inicio de operación de instrumentos de medición.

Una vez cumplida la condición se activa el caso verdadero de la figura 2.66 y se inicia el proceso de leer la información en el buffer y se realiza el correspondiente proceso de conversión a un dato tipo *double*. Como ayuda auxiliar se agrega un sonido de alerta que avisa que la medición ha sido capturada correctamente.



**Figura 2.66.** Diagrama de bloques para el procesamiento de longitud del comparador de bloque patrón.

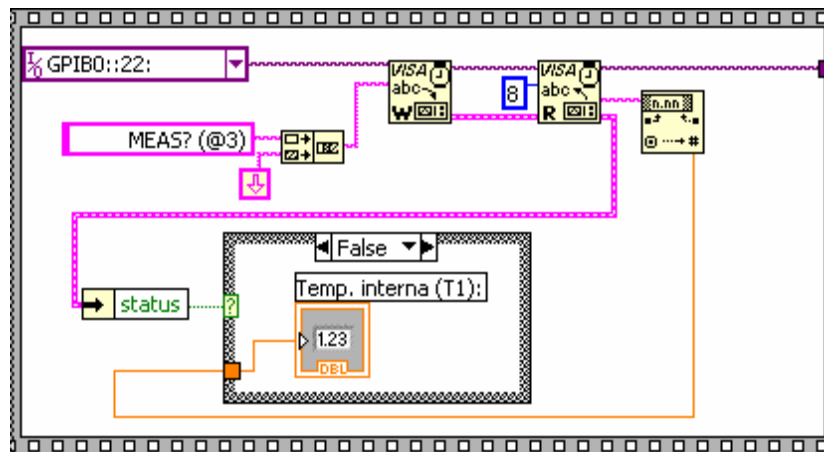
### Comunicación con el termómetro de resistencia de platino

La operación en forma manual de este instrumento presenta un retraso aproximado de cuatro segundos entre cambio de canal de medición. En el set de instrucciones de comandos GPIB no se encontró un comando apropiado para recibir información sin cambiar previamente de canal.

La lectura en modo remoto debe ser independiente para cada canal y hasta que el canal deseado cambie a primario, solo en ese momento se le puede solicitar medición de temperatura utilizando el comando MEAS?(@\_); el parámetro vacío es para indicar el número de canal del cual se requiere la medición.

La figura 2.67 ilustra el diagrama de bloques para leer el canal primario número tres del instrumento con interfaz GPIB localizado en la dirección 22. Sino existe un error de retraso en la transmisión, se muestra el valor correcto en la pantalla de usuario. Así se evita tener retrasos entre el cambio de cada canal de medición.





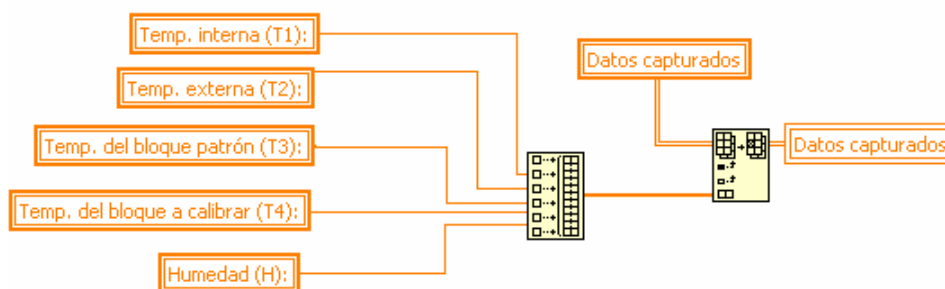
**Figura 2.67.** Diagrama de bloques para comunicación con el termómetro de resistencia de platino.

### Control de captura

El control de captura es un accionador, que por medio de la condición del número de bits transmitidos, ordena a las rutinas de lectura de cada instrumento y almacena su información en una variable. Todos los valores en las variables son agrupadas en un arreglo como lo muestra el diagrama de bloques de la figura 2.68.

### Control de transferencia a Excel

Se le incorporan de manera sencilla los SubVI creados para guardar y abrir un archivo de calibración en el PCMIC. Mostrando una vez más las ventajas de programar con LabVIEW. Al igual que el otro procedimiento se introduce un menú de ejecución para acceder a estas rutinas, y se agrega además la programación para poder adquirir las longitudes nominales de determinado juego de bloques y seleccionar solo los elementos que estarán bajo calibración. A continuación se describirá como seleccionar de una lista ciertos valores, ya que la forma para cargar el archivo es muy parecida a la programación de abrir un archivo.

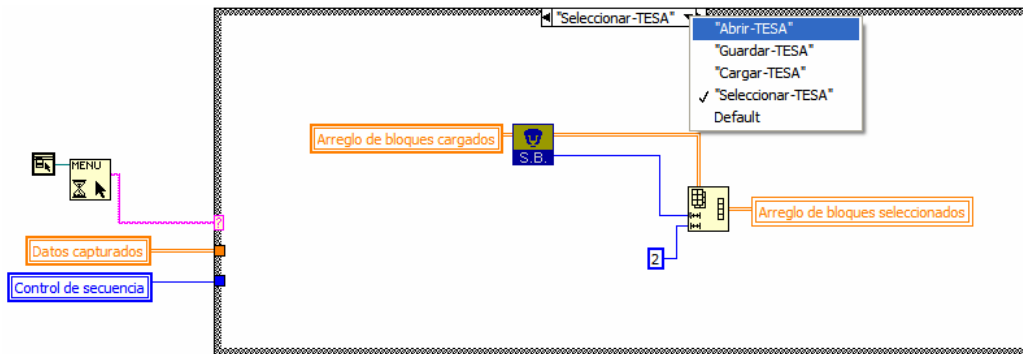


**Figura 2.68.** Construcción de arreglo con información de calibración.

### Control de selección de elementos en un juego de bloques

La figura 2.69 muestra la implementación de un menú en LabVIEW, esta conectado a una estructura "Case". Se encuentran cuatro paneles de programación y en cada uno se encuentra un SubVI a la espera de ser seleccionado, desde la pantalla de usuario el operador puede elegir tareas como abrir, guardar, cargar y seleccionar archivos de Excel.

Se describe la programación relacionada con la pantalla de selección de bloques (figura 2.26) de una lista de valores de bloques patrón. Se recupera la información de Excel con el contenido de longitudes, una vez convertido en un arreglo numérico, se asocia el estado del selector, si esta marcado se relaciona con uno y si no fue elegido se le asigna un cero.



**Figura 2.69.** Diagrama de bloques para la creación de un menú.

De esta manera se crean dos arreglos, el numérico con la información y el de estados booleanos de la selección realizada por el usuario. El diagrama de bloques de la figura 2.70 lee los estados booleanos recorriendo todos los índices del arreglo, si es activo toma el valor relacionado con el arreglo numérico y es trasladado a un nuevo arreglo de valores seleccionados. El cual es reconocido por el software como los elementos bajo calibración y sobre los cuales se desarrollará el procedimiento.





## Capítulo 3

### Resultados y conclusiones

El CCADET, a través del Laboratorio de Metrología, fue parte fundamental en la realización del proyecto “Automatización del Laboratorio de Metrología del LAPEM” de CFE. La organización para desarrollar con éxito el presente proyecto contempló 4 etapas con su correspondiente asignación de recursos económicos y visitas de trabajo en las instalaciones del LAPEM. Dentro del esquema de trabajo se planteó obtener resultados por etapas, así se planearon ciertos objetivos a cumplir en determinado tiempo hasta llegar a un resultado final y adecuado a las necesidades del usuario. De esta forma se establecieron las siguientes etapas con su correspondiente objetivo general.





- Etapa 1. Diseño conceptual y desarrollo de hardware.
- Etapa 2. Diseño y desarrollo de las aplicaciones de automatización.
- Etapa 3. Integración, pruebas y evaluación general.
- Etapa 4. Validación y entrega.

#### 3.1. Resultados por etapa

Se eligió el modelo prototipado para desarrollo de software y se trató de relacionar de cierta manera con el avance de las etapas administrativas del proyecto. De esta manera se pueden presentar adecuadamente los resultados del presente trabajo de tesis. La tabla 3.1 relaciona las etapas del proyecto con las fases del modelo prototipado.

En cada etapa se realizó una verificación y validación de los objetivos planteados, una vez cumplidos se realiza la transición a una nueva etapa. Se realizan labores de planificación (Plan-P), desarrollo (Do-D), verificación (Check-C) y acción (Action-A).

Cabe destacar la participación del personal del LAPEM en la explicación de ambos procedimientos de calibración y en establecer en manera conjunta los requisitos del software hasta su aprobación final.

Etapas del proyecto		Fases del modelo prototipado	
Etapa 1 	Determinación de requisitos	Fase 1	Investigación preliminar
	Diseño preliminar	Fase 2	Análisis
			Diseño
			Evaluación
Modificación			
Etapa 2 	Desarrollo de la programación	Fase 3	Diseño técnico
Etapa 3 	Instalación de las estaciones automatizadas preliminares	Fase 4	Programación y pruebas
Etapa 4 	Instalación de las estaciones automatizadas finales	Fase 5	Operación y mantenimiento
	Capacitación		
	Validación y entrega		

**Tabla 3.1.** Esquema de trabajo del proyecto y fases del modelo prototipado.

En la etapa 1 se planteó el problema, se tuvo que hacer una introducción en conceptos de metrología para entender plenamente la importancia de los procedimientos de calibración. Se realizó el análisis, diseño, evaluación y modificación del PCMIC y PCBP para establecer conjuntamente aspectos de automatización.

En la etapa 2 del proyecto se desarrolló el diseño técnico del prototipo, LabVIEW se adaptó muy bien en esta parte ya que fue posible construir una interfaz de usuario con todos los requerimientos establecidos hasta el momento.

Para la etapa 3 del proyecto se tenía una idea bastante clara de las necesidades del operador del LAPEM y se empezó con la programación, se realizaban pruebas con los equipo de instrumentación y se interactuaba con el operador para afinar detalles.

La última etapa se relaciona con la fase 5 del modelo prototipado, la cual contempla la instalación de la estación de trabajo final con la versión del software de calibración correspondiente. El operador recibirá capacitación para utilizar el software y posteriormente se realizará el mantenimiento al software.

### 3.1.1. Etapa 1

Se analizaron a profundidad ambos procesos de calibración y se plantearon los requisitos para el diseño del software. El CCADET a través de su Laboratorio de Metrología fue responsable de la capacitación en esta área, de esta manera se pudo entender con mayor facilidad los procedimientos de calibración una vez que se entrevistó al operador. En la tabla 3.2 se observan los objetivos, descripción, método de validación, las herramientas utilizadas y el método de revisión para la etapa 1. En lo subsecuente se explicarán los resultados por etapa tomando esta tabla como base.

Objetivo	Descripción	Método de validación	Herramientas de validación	Método de revisión
Propuesta general de conexiones para la estación de calibración	Se identificaron los requisitos de hardware para implementar la estación de calibración	Observación	Presentación por medio de diapositivas	Firma del cliente
Establecer un modelo de la interfaz de usuario	Se identificaron los elementos y/o requisitos necesarios para desarrollar la interfaz de usuario	Entrevista y encuesta al cliente	Presentación por medio de diapositivas	Firma del cliente
Obtener los requisitos para desarrollar la calibración	Se identificaron los requisitos y aspectos técnicos necesarios para realizar el PCMIC y PCBP	Entrevista y encuesta al cliente	Presentación por medio de diapositivas	Firma del cliente

**Tabla 3.2.** Verificación y validación de la etapa 1.

### 3.1.2. Etapa 2

El objetivo era presentar una versión preliminar de la interfaz de usuario, la cual debería contar con funciones que reducirían el tiempo de calibración. La tabla 3.3 enlista en forma general los puntos a realizar para la etapa 2.

Se incorporaron los requisitos establecidos por el procedimiento de calibración así como funciones especiales que el operador ha desarrollado con base en su experiencia.

Objetivo	Descripción	Método de validación	Herramientas de validación	Método de revisión
Mostrar la comunicación con los instrumentos	Se realizan pruebas de adquisición y transferencia de datos de los instrumentos. El momento de adquisición es en el momento que el instrumento lo genera en pantalla. Con la misma resolución y sin atrasos en el sistema.	Validado por el fabricante del equipo  Observación	Software de desarrollo  Pruebas de repetibilidad.	Firma del cliente
Mostrar la manipulación de datos y la transferencia hacia la hoja de cálculo	El dato no sufre alteración alguna en los indicadores numéricos de la pantalla de usuario, posteriormente se transfiere a una hoja de cálculo conservando sus valores decimales	Comparativo	Comparación visual entre resultados capturados por software y en forma manual.	Firma del cliente
Mostrar la funcionalidad parcial de la interfaz de usuario	El cliente utilizó una versión del software para observar características de diseño como ventanas de diálogo, botones, indicadores, colores, diseño, fuentes, punto decimal, etc.	Observación	Software desarrollado y opinión del cliente	Firma del cliente
Presentación del primer prototipo de la tarjeta decodificadora	Conectar las terminales de la estación unidimensional a la tarjeta y ver en pantalla de usuario el conteo generado por el desplazamiento	Calibración  Observación	Software de desarrollo  Calibración mediante un bloque patrón	Firma del cliente

**Tabla 3.3.** Verificación y validación de la etapa 2.

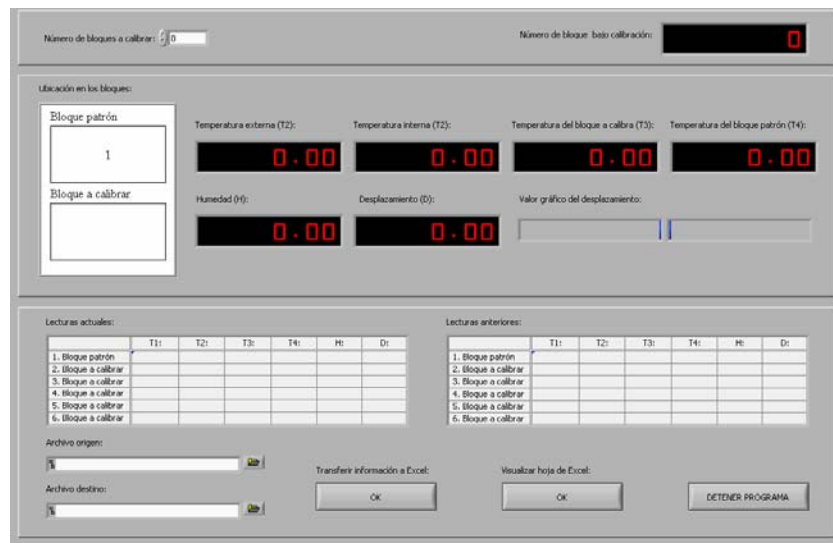
Se presentó la pantalla desarrollada y se describió lo que sería la nueva forma de calibración a los operadores, aquí inicio un trabajo en conjunto con el operador en el que se intercambiaron ideas de trabajo. Tal interacción evaluaba el software paulatinamente o en determinado momento surgían nuevas tareas o la revisión de ciertas funciones a petición del operador.



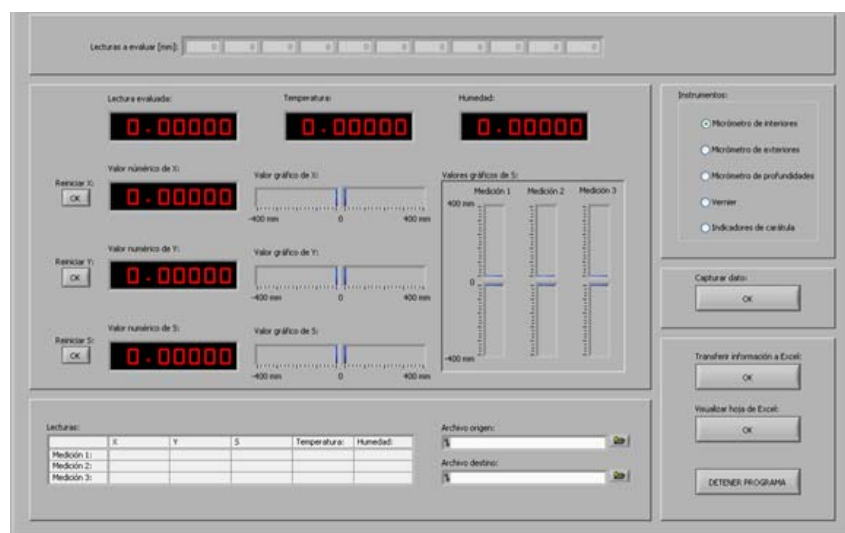
El interactuar directamente con los equipo de calibración permitió entender los procedimientos y formular algoritmos de programación en forma correcta.

La figura 3.1 y 3.2 muestran las pantallas preliminares que se presentaron en esta etapa, ambas evolucionaron hasta llegar a la pantalla de usuario final.

Con el primer prototipo de tarjeta se estimó la constante de transformación del conteo de pulsos en cuadratura a una magnitud longitudinal.



**Figura 3.1.** Interfaz preliminar de usuario para el PCBP.



**Figura 3.2.** Interfaz preliminar de usuario para el PCMIC.

### 3.1.3. Etapa 3

En la tabla 3.4 se muestran los objetivos y la manera en que fueron validados.

Objetivo	Descripción	Método de validación	Herramientas de validación	Método de revisión
Mostrar la funcionalidad de la versión preliminar del software	Se realizaron pruebas al software con comunicación de instrumentos, manipulación de datos y envió de resultados a hojas de cálculo	Observación Comparativo	Comparación entre resultados del software y calibraciones en forma manual	Firma del cliente
Mejorar el tiempo de calibración empleando el software	Se registró el tiempo empleado para realizar las pruebas de la calibración utilizando el software	Comparativo	Comparación entre el tiempo registrado al realizar una calibración manual contra el software	Firma del cliente
Comprobar el correcto funcionamiento de la tarjeta electrónica	Se realizaron pruebas de calibración con base en el PCMIC y utilizando la interfaz creada	Observación	Uso del software de desarrollo Calibración por bloques patrón	Firma del cliente

**Tabla 3.4.** Verificación y validación de la etapa 3.

Esta etapa fue de depuración, se realizaron pruebas para identificar y corregir errores, con el objetivo de crear la versión funcional del software.

Se construyó el prototipo final de la tarjeta electrónica, se realizó una prueba con el software de desarrollo y se verificó que la conversión a longitud fuera de forma adecuada. La figura 3.3 muestra el prototipo final.



**Figura 3.3.** Conexión de la tarjeta electrónica USB con la estación de medición unidimensional.

#### 3.1.4. Etapa 4

Esta etapa no ha sido concluida, en la tabla 3.5 se observan los objetivos y el método de validación propuesto para realizar la aprobación final del software. Sin embargo las partes más importantes se han realizado hasta el momento con éxito.

Objetivo	Descripción	Método de validación	Herramientas de validación	Método de revisión
Implementar la estación de calibración	Se realizó la conexión fija de instrumentos de calibración en conjunto con el hardware	Encuesta al cliente	Realización de pruebas	Firma del cliente
Instalar la versión final del software	Se instalará la versión final del software con las modificaciones o errores detectados en la etapa anterior	Por observación Comparativo	Comparación entre resultados del software y calibraciones en forma manual	Firma del cliente
Elaboración de Manuales de Usuario	Se realizarán los manuales de usuario para el software desarrollado	Manuales	Revisión del cliente	Firma del cliente
Capacitación al personal	Dar capacitación al personal del laboratorio de metrología dimensional para utilizar y dar mantenimiento al software	Encuesta al cliente	Cursos	Firma del cliente

**Tabla 3.5.** Verificación y validación de la etapa 4.

Para esta etapa se ha realizado un 50% de los objetivos establecidos, se ha cumplido con la implementación de las estaciones de trabajo y con la instalación final del software.

### 3.2. Resultados generales

Se obtuvo el software final con todos los requisitos establecidos por ambos procedimientos de calibración, es decir, bloques patrón, micrómetros de cualquier tipo e indicadores de carátula. La interfaz gráfica de programación fue realizada a partir de una interacción constante con el especialista en el LAPEM y en un estudio exhaustivo de los manuales de todos los instrumentos de calibración.

Los resultados son los siguientes:

- 1) La interfaz de usuario es eficiente, práctica y de fácil manejo.
- 2) Se logró enviar de manera automática los datos a una hoja de cálculo y su correspondiente presentación en un reporte de calibración.
- 3) Existe facilidad para crear y manejar de forma ordenada los reportes de calibración. El error por transferencia de datos se elimina, ya que el operador no manipula la información en ningún momento y se garantiza en un 100% que la transferencia de los valores de medición es exactamente la misma que se muestra en las pantallas de los equipos de instrumentación.
- 4) Se logró una reducción en el tiempo de calibración de ambos procedimientos. Para el PCMIC se redujo el tiempo de calibración manual de 45 minutos a solo 15 minutos con el uso del software. Mientras que para el PCBP en forma manual se tenía un tiempo de 50 minutos y se logro realizarlo en 20 minutos con el software.
- 5) Para realizar manualmente el PCBP se debe de tomar en cuenta el número de bloques bajo calibración. Si aumentamos el juego de bloques, aumentamos el tiempo de transferencia al reporte de calibración. Con el software la transferencia es automática y no hay incrementos de tiempo. Un caso similar sucede cuando se agregan extensiones en el PCMIC.
- 6) Se instalaron adecuadamente las estaciones de trabajo en el área de metrología dimensional del LAPEM. Como se observa en la figura 3.4 y 3.5 respectivamente.
- 7) Se cumplió con los puntos establecidos en los *Formatos de evaluación técnica y valoración de usuario* de cada etapa, figura

- 3.6; por lo que se cumplió cabalmente con la solución del problema y se consiguió el objetivo final.
- 8) Se crearon dos estaciones de trabajo que incorporan todos los recursos de hardware y aprovechen al máximo los recursos de procesamiento de una computadora mediante una poderosa herramienta de programación como lo es LabVIEW.



**Figura 3.4.** Instalación de la estación de trabajo para el PCMIC.



**Figura 3.5.** Instalación de la estación de trabajo para el PCBP.

PROYECTO LAPEM			CFE Comisión Federal de Electricidad
Evaluación técnica y Valoración del usuario para el producto entregable de la etapa 2			
Fecha	Lugar	Área	
11/05/2007	LAPEM Irapuato	TIEMPO Y FRECUENCIA	
Procedimiento	Método	Equipos	
K3153F09			
<p>En la Etapa 2 del proyecto se están desarrollando los programas para automatizar los procesos de calibración requeridos.</p> <p>El producto entregable de la etapa 2 consiste en la versión beta de la programación, misma que se probará en el equipo de la UNAM para que los usuarios del LAPEM.CFE emitan sus comentarios y observaciones con respecto a la funcionalidad de las soluciones.</p> <p><b>EVALUACIÓN TÉCNICA AL PRODUCTO ENTREGABLE DE LA ETAPA 2</b></p> <p>¿Del método que se está evaluando, que porcentaje de satisfacción estima que se tiene al momento de la evaluación?</p> <p>¿Se requieren modificaciones al producto entregable de la etapa 2?</p> <p><input type="checkbox"/> SI <input type="checkbox"/> No</p> <p>En caso de ser afirmativa la anterior, en qué áreas de requieren</p> <p><input type="checkbox"/> Diseño de la interfase gráfica <input type="checkbox"/> Comunicación con los instrumentos <input type="checkbox"/> Hardware <input type="checkbox"/> Procedimiento programado</p> <p>¿Qué impacto tendrían las modificaciones en el producto entregable de la etapa 2?</p> <p><input type="checkbox"/> Mayor <input type="checkbox"/> Menor <input type="checkbox"/> Ninguna</p> <p>¿Qué factores fueron determinantes para el buen cumplimiento del producto entregable de la etapa 2?</p> <p>Comentarios</p> <p><b>VALORACIÓN DEL USUARIO AL PRODUCTO ENTREGABLE DE LA ETAPA 2</b></p> <p>¿El producto entregable de la etapa 2 cumple con sus expectativas de automatización del proceso de calibración?</p> <p><input type="checkbox"/> SI <input type="checkbox"/> No</p> <p>¿Considera que el producto entregable de la etapa 2 agiliza las actividades a seguir para la calibración?</p> <p><input type="checkbox"/> SI <input type="checkbox"/> No</p> <p>¿Acepta el producto entregable de la etapa 2?</p> <p><input type="checkbox"/> SI <input type="checkbox"/> No</p> <p>Comentarios</p> <p>_____ Experto técnico LAPEM</p> <p>_____ Experto técnico UNAM</p>			

**Figura 3.6.** Formato de evaluación técnica y valoración de usuario.

### 3.3. Procedimiento típico para realizar una calibración

La estructura de programación y secuencia de operación en ambas estaciones de calibración es muy similar. De esta forma el operador, en una futura capacitación, no tendrá que aprender distintas secuencias de operación y le será familiar el procedimiento de ambos sistemas para realizar una calibración.

#### 3.3.1. Software del procedimiento de calibración de micrómetros e indicadores de carátula

1. Verificar que todos los instrumentos se encuentren conectados de acuerdo al esquema general de automatización de la figura 2.8 para el PCMIC.
2. Ejecutar el software.
3. Verificar en los recursos del sistema la presencia de los equipos de instrumentación participantes en la calibración.
4. Ingresar los datos de calibración: Nombre del reporte, reporte de historial, número de identificación y número de folio.

5. Seleccionar la clase de instrumento bajo calibración; micrómetros de exteriores, interiores, profundidades, con extensiones ó si se trata de un indicador de carátula.
6. Seleccionar el sistema de medición, en milímetros o pulgadas. Todas las funciones dentro de la pantalla cambian bajo el régimen de unidad seleccionada.
7. Seleccionar el “*Modo de lectura*”. Cuenta con dos opciones, si se quiere que el programa realice la calibración del instrumento de acuerdo a los intervalos establecidos por el propio PCMIC, elija la opción “Predefinidas” o en su defecto elegir “*Usuario*” para ingresar desde teclado intervalos distintos a los preestablecidos.
8. Si el instrumento bajo calibración tiene “extensiones”, elija el menú “*Puntos de medición*”, esta acción desplegará una pantalla en la cual se puede ingresar los intervalos de medición que el instrumento requiera.
9. La secuencia de calibración ha sido cargada por el software. Ahora realice el trabajo de calibración con base al PCMIC.
10. Utilizar como referencia los indicadores de “*Secuencia de lectura*” quienes mostraran el número de medición, la serie que se está realizando y el valor del intervalo en ese momento.
11. Los controles de reinicio a cero y cambio de signo se encuentran a izquierda de los indicadores de longitud.
12. Para adquirir una medición oprima el botón “*Capturar*”, si a juicio del operador el valor no es aprobatorio del todo, puede eliminar cierta elección con “*Borrar*” y la secuencia de ejecución no se perderá.
13. Una vez concluido todos los intervalos de medición o en caso de postergar la calibración, se deben guardar los datos de calibración. En el menú, seleccione “*Archivo*”, elegir la opción “*Guardar*”. Se desplegará una pantalla donde ingresará la ruta del archivo donde se encuentra el reporte de calibración.
14. Tal archivo contará con la selección de intervalos de calibración, con los valores de calibración elegidos y con el correspondiente informe de calibración.
15. Para continuar con una calibración inconclusa repita los pasos del 1 al 3. Seleccionar en el menú “*Archivo*” y elegir la opción “*Abrir*”. Inserte la ruta y el nombre del archivo que desea recuperar. Continúe el PCMIC en forma normal.
16. Presione el botón “*Detener*” para cerrar la aplicación.

### 3.3.2. Software del procedimiento de calibración de bloques patrón

Los pasos del 1 al 4 son los mismos del PCMIC. La conexión de instrumentos es acorde a la estación de la figura 2.7 del PCBP.

5. Elegir en el menú la opción “*Bloques*”, seleccionar “*Cargar*”; se abrirá la ruta de acceso de un archivo xls donde se encuentran cargados los datos del juego patrón de bloques. Este archivo debe ser creado por el operador tomando en cuenta el juego que utilizará como patrón.
6. Dentro del mismo menú “*Bloques*”, elegir la opción “*Seleccionar*”, para desplegar una carpeta con todos los valores nominales y de error del juego de bloques elegido como patrón, previamente cargado en el paso 5. Seleccionar con una paloma en el recuadro del valor que sea de su elección desplazándose con el *mouse* y dando un *clic*.
7. Ingresar por medio del teclado los valores numéricos del factor de expansión térmica para el bloque patrón y bajo calibración.
8. La secuencia de calibración ha sido cargada por el software. Ahora realice el trabajo de calibración con base al PCBP.
9. Utilizar como referencia los indicadores de “*Bloque en turno*”, “*Serie de lectura*” y “*Medición*” para no perder la secuencia de calibración. El esquema “*Ubicación de la medición*” despliega un recuadro numérico correspondiente a la forma en que se desplaza el comparador mecánico del instrumento patrón.
10. Para adquirir una medición oprima el botón accionador del comparador de bloques patrón. Se escuchará un sonido de validación. Con el botón “*Borrar*” el operador puede eliminar cierta información sin perder la secuencia de ejecución.
11. Seguir los pasos 13, 14 y 15 del PCMIC.
12. Presione el botón “*Detener*” para cerrar la aplicación.

### 3.4. Conclusiones

Se desarrollaron dos sistemas de software-hardware para realizar de manera automática la calibración de micrómetros, indicadores de carátula y bloques patrón del LAPEM-CFE. El control que tiene el operador sobre el software le permite seguir al pie de la letra el procedimiento de calibración correspondiente, y registrar aspectos particulares en cada prueba de calibración, elimina el tiempo necesario



para la configuración de instrumentos y mejora la calidad del reporte de calibración al eliminar los errores por transcripción de números.

El tiempo de calibración se reduce en gran parte para ambos procedimientos, por lo que el laboratorio puede ofrecer sus servicios de calibración en menor tiempo y satisfacer una eventual demanda de servicio, además de haber incorporado una de las últimas tecnologías como lo es la instrumentación virtual.

El software no sufre retrasos en la adquisición de datos por lo que tiene un 100% de confiabilidad en los resultados y registros que presenta.

El diseño de este trabajo de tesis es una parte del proyecto “Automatización del Laboratorio de Metrología del LAPEM-CFE” realizado entre la Comisión Federal de Electricidad (CFE), el Consejo Nacional de Ciencia y Tecnología (CONACyT) y la Universidad Nacional Autónoma de México (UNAM) con número CFE-2005-C03-12.

### **3.5. Trabajo a futuro**

El trabajo a futuro contempla la culminación de la etapa 4 del proyecto y realizar las siguientes actividades:

- Realizar pruebas exhaustivas al software desarrollado.
- Elaboración de los manuales de usuario, administrador y ayuda para el correcto funcionamiento del software desarrollado.

Capacitación al personal del laboratorio de metrología dimensional del LAPEM en la operación y mantenimiento del software desarrollado.



## Anexo

# Interfaces de comunicación con la computadora

El presente anexo describe las interfases de comunicación utilizadas en el proyecto de tesis para conectar instrumentos de medición y computadoras personales. También se incluye un ejemplo de programación utilizado en el desarrollo de la tarjeta decodificadora.

### A.1. General Purpose Interface Bus (GPIB)

GPIB es un bus y protocolo estándar para el control y comunicación con instrumentos de medición, que permite configurar sistemas automatizados en el laboratorio y en la industria con gran flexibilidad y potencia [26].

#### A.1.1. Historia

El primer antecedente fue desarrollado en 1960 por Hewlett-Packard, su nombre original fue HP-IB, utilizaba un simple bus en paralelo y varias líneas de control individual.

La estandarización llegó en 1975 por el *Institute of Electrical and Electronics Engineers* en referencia como *IEEE Standard Digital Interface for Programmable Instrumentation*, IEEE-488-1975 ó 488.1. Se definían los parámetros mecánicos, eléctricos y el protocolo básico de GPIB. Mientras que en 1987 el estándar *IEEE-488.2 Codes, Formats, Protocols and Common Commands for IEEE-488.1* delimitó en forma más concreta la sintaxis de programación, definiendo comandos, formato de mensajes, estructuras de datos, protocolos de error y estado de los instrumentos.

A pesar de los estándares IEEE 488.1 y 488.2, existía libertad para que cada fabricante eligiera los comandos para sus instrumentos. En 1990 un grupo de empresas fabricantes de instrumentos acordaron crear un conjunto de órdenes con una sintaxis común y que respetaba los principios del estándar 488.2, surge así el SCPI (Comandos Estándar para Instrumentos Programables).

Las normas SCPI e IEEE 488.2 tratan las limitaciones y las ambigüedades de la norma original IEEE 488. La norma IEEE 488.2 permite la posibilidad de diseñar sistemas de prueba más compatibles y más productivos. La norma SCPI simplifica la tarea de programación definiendo un único conjunto de comandos comprensibles para los instrumentos programables, sin importar el tipo o el fabricante.

El estándar IEEE 488.1 simplifica enormemente la interconexión de los instrumentos programables, definiendo claramente las especificaciones mecánicas, eléctricas, y el protocolo hardware. Inicialmente, los instrumentos de diversos fabricantes se conectaron por el cable estándar.

En concreto, la norma IEEE 488.1 no direccionaba formatos de datos, informes de estado, protocolos de intercambio de mensajes, comandos de configuración comunes, o comandos específicos de cada instrumento.

La norma IEEE 488.2 realzó y consolidó la IEEE 488.1 estandarizando formatos de datos, informes de estado, gestión de errores, las funciones del controlador y los comandos comunes a los cuales todos los instrumentos debían responder de una manera determinada. Estandarizados estos elementos, los sistemas IEEE 488.2 son mucho más compatibles y fiables. El IEEE 488.2 se enfocó principalmente a los asuntos de protocolo del software, manteniendo así la compatibilidad con el estándar orientado a hardware del IEEE 488.1.

El SCPI se construyó basándose en el IEEE 488.2 definiendo los comandos específicos de cada instrumento y estandarizando la programación de los mismos. Los sistemas de SCPI son mucho más fáciles de programar y de mantener. En muchos casos, se puede intercambiar o actualizar los instrumentos sin tener que cambiar el programa.

Los comandos SCPI se escriben como texto ASCII, tienen una estructura jerárquica por niveles, separados por dos puntos (:). Los caracteres en mayúsculas son necesarios para especificar la orden, mientras que los que están en minúsculas pueden suprimirse, sirviendo sólo para facilitar la lectura al programador.

Los comandos en sí pueden ser escritos indistintamente en mayúsculas o minúsculas como se muestra a continuación.

---

```
HORizontal:MAIn:SCAle <valor>
HOR:MAIN:SCA <valor>
```

---

Cada instrumento posee un set de instrucciones orientado a realizar determinada función en vía remota a través de una PC.

National Instruments desarrolló y patentó el protocolo de alta velocidad handshake GPIB (llamado HS488) para incrementar la velocidad de transferencia de datos en sistemas GPIB. Todos los dispositivos involucrados en la transferencia de datos deben soportar el protocolo HS488, si algún dispositivo no-HS488 se incorpora al sistema, los dispositivos HS488 usan automáticamente el protocolo handshake IEEE 488.1.

Existe además una nueva estandarización, que permite acceder a instrumentos del más alto nivel sin importar el tipo de interfaz de comunicación, GPIB o puerto serie. Un convenio entre Agilent y National Instruments creó el estándar VISA (Virtual Instruments Software Architecture).

### A.1.2. Características generales de un cable GPIB

El bus de transmisión de datos de GPIB es de 8 bits en paralelo, y lógica negativa con niveles TTL. Verdadero si el voltaje es  $\leq 0.8$  V y falso si el voltaje es  $\geq 2.0$  V. Los cables y conectores se ilustran en la figura A.1, consta de 24 pines, repartidos de la siguiente forma:

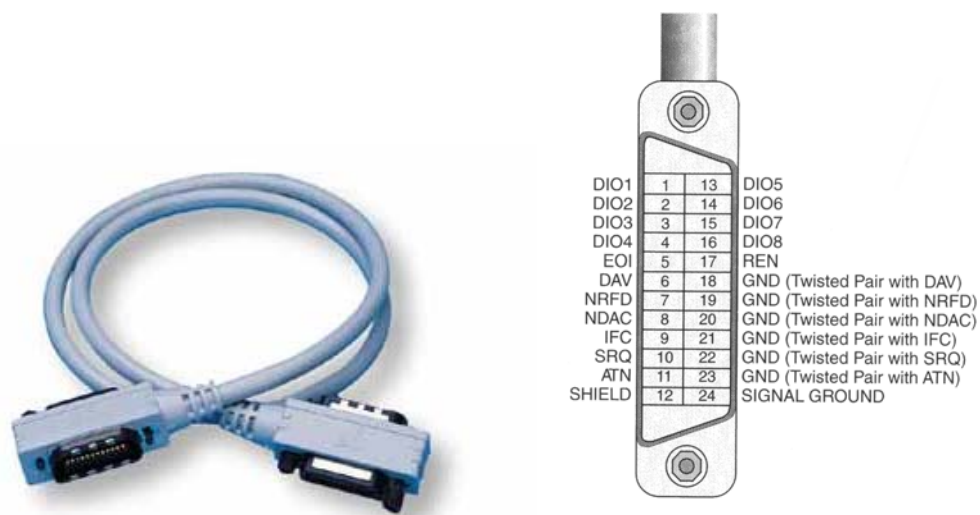


Figura A.1. Cable y diagrama de conexión GPIB.

Las ocho líneas de datos, DIO1 a DIO8, llevan tanto datos como mensajes de comando. El estado de la línea de Atención (ATN) determina si la información son datos o comandos. Todos los comandos y la mayoría de los datos utilizan el código ASCII o ISO de 7-bit, el octavo bit, DIO8 define la paridad o no se utiliza.

Cuenta con 3 líneas para el control asíncrono de la comunicación: NRFD, NDAC y NRDAV. Mediante estas líneas se verifica la correcta transmisión de los datos, que es una de las fortalezas del GPIB.

- NRFD (no preparado para datos, Not Ready For Data): Indica cuando un dispositivo está o no preparado para recibir un byte de mensaje.
- NDAC (no datos aceptados, Not Data Accepted): Indica cuando un dispositivo ha validado o no un byte de mensaje.
- DAV (datos válidos, DAta Valid): Indica cuándo las señales en las líneas de datos son estables y se pueden aceptar con seguridad por los dispositivos.

Se cuenta con 5 líneas que gestionan la transmisión de comandos (ATN, IFC, REN, SRQ y EOI).

- ATN (atención, ATtNtion): El controlador pone ATN en verdadero cuando utiliza las líneas de datos para enviar comandos, y falso para enviar mensajes de datos.
- IFC (borrado de Interfaz, InterFace Clear): El controlador del Sistema inicializa el Bus.
- REN (remoto permitido, Remote ENable): El controlador del sistema habilita al dispositivo GPIB en modo remoto o local.
- SRQ (petición del servicio, Service ReQuest): El dispositivo GPIB solicita atención al controlador.
- EOI (termina o identifica, End Or Identify): La línea de EOI tiene dos propósitos. El instrumento utiliza la línea de EOI para marcar el fin del mensaje, y el controlador solicita información a los dispositivos.

El resto del cable esta compuesto por las tierras de las diferentes líneas.

Para que el bus GPIB alcance la velocidad de transmisión para el que fue diseñado (hasta 8 Mbytes/s) deben cumplirse los siguientes requisitos:

- Debe de haber un máximo de 15 dispositivos conectados al bus, y al menos dos tercios de ellos deben estar encendidos.
- La separación máxima entre dos dispositivos es 4 m, y la separación promedio en toda la red debe ser menor de 2 m.
- La longitud total de la red no debe exceder los 20 m.

### **A.1.3. Hardware disponible para sistemas GPIB**

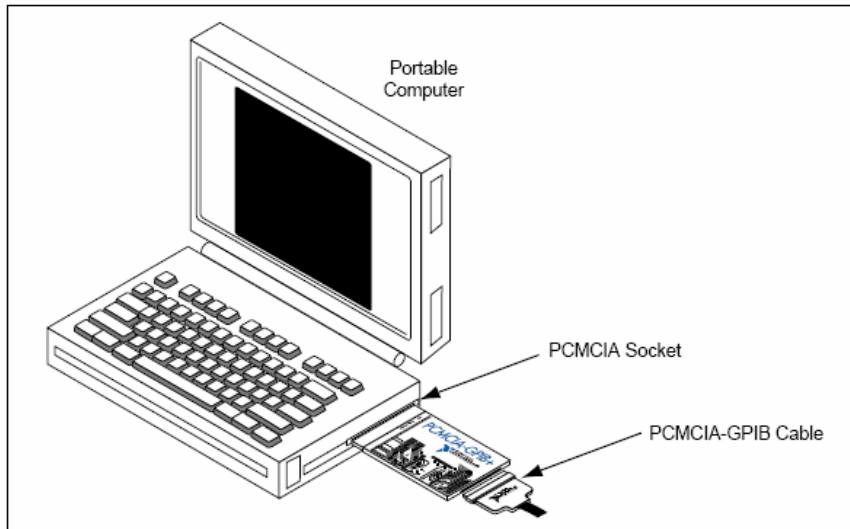
Un sistema típico constará de un ordenador con una tarjeta controladora de protocolos GPIB IEEE 488. Existen tarjetas GPIB para prácticamente todos los sistemas presentes en el mercado como PC, Macintosh, Sun, etc.

En el caso concreto de la PC, la tarjeta GPIB puede conectarse al bus ISA, PCI, PCMCIA (portátiles), USB, Ethernet, Firewire, y los puertos serie y paralelo. De igual forma existen adaptadores para los estándares de comunicación RS-232 y RS-485.

### **A.1.4. Características de la tarjeta NI PCMCIA-GPIB**

La figura A.2 muestra la inserción de una tarjeta controladora de protocolos GPIB para computadoras portátiles. Algunas de sus características son las siguientes:

- Compatible con IEEE 488, 488.1, 488.2 y HS488.
- Rango de transferencia: 1.3 Mbytes/s con IEEE 488.1 y 2.2 Mbytes/s con HS488.
- Sistema operativo: Windows, Mac OS, DOS.
- I/O FIFO buffer size: 32 MB.
- Conexión máxima de 14 dispositivos.
- Longitud máxima del cable: 14 m.
- Software recomendado: LabVIEW, LabWindows/CVI, Measurement Studio.
- Compatibilidad “*plug and play*”.
- Alimentación: 5VDC, 65 mA.



**Figura A.2.** Tarjeta PCMCIA-GPIB de *National Instruments*.

### **A.1.5. Software de comunicación GPIB**

Los fabricantes de equipo de instrumentación cotidianamente integran un software de desarrollo específico para utilizarse con el equipo, este generalmente realiza la función más elemental y no es modificable por el usuario.

El problema surge cuando se desea implementar mayores funciones al equipo de instrumentación o integrar con otros programas, entonces el usuario se ve limitado en desarrollo, o en su defecto, tiene que adquirir accesorios adicionales.

Existe software con el cual se puede programar este tipo de tareas, uno de ellos es LabWindows/CVI, un entorno de desarrollo basado en C. Cuenta con librerías para la comunicación entre dispositivos de instrumentación con interfaz serie, paralelo, GPIB, TCP/IP, etc.

Actualmente National Instruments, creador de LabWindows/CVI, ofrece su software de instrumentación virtual LabVIEW, orientado a programación gráfica y manteniendo la compatibilidad con otros compiladores de propósito general, como lo son Visual Basic, Visual C++, .NET, etc. Resalta la facilidad para el desarrollo de la interfaz gráfica de usuario adaptada a los instrumentos de medición.

### **A.1.6. Estructura de sistemas con GPIB**

Un sistema GPIB consiste en una serie de instrumentos de medida conectados a un bus, y controlados por una PC dotado de una tarjeta GPIB que gestiona el flujo de datos y comandos a los diferentes elementos del sistema. Existe bastante libertad en la configuración



topológica del bus, que, en general, es una combinación de disposiciones lineales y en estrella, como se muestra en la figura A.3.



**Figura A.3.** Configuración de un sistema GPIB.

Permite la conexión de hasta 15 instrumentos en un solo bus paralelo de 8 bits mediante una conexión en cadena con el dispositivo más lento, el cual determina la velocidad de transferencia.

## A.2. Norma RS-232-C

En 1969, la EIA (Asociación de Industrias Electrónicas), los Laboratorios Bell y los fabricantes de equipos de comunicaciones, formularon cooperativamente y emitieron, el EIA RS-232, después de algunas revisiones se convirtió en el RS-232-C [27]. Para su estudio se distinguen dos partes o bloques:

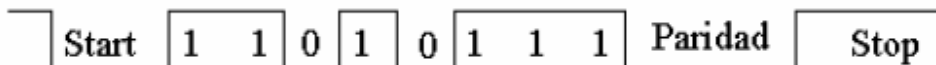
- Bloque 1: Norma asociada a la conexión. Referente a un conjunto de reglas que por convenio adoptan todos los fabricantes de equipos para lograr su compatibilidad.
- Bloque 2: Hardware asociado. Para realizar la conexión definida en el bloque 1 es necesario una circuitería compatible, comúnmente una tarjeta insertada en el ordenador y una dirección signada.

El propósito de la conexión RS-232-C es la conexión entre un equipo terminal de datos (DTE) y un equipo de comunicaciones de datos (DCE) empleando intercambio de datos binarios en serie.

El tiempo necesario para la transmisión de un byte es mayor que el correspondiente a la transmisión de ocho bits individuales. Deben añadirse otros bits suplementarios. El conjunto de bits completo a enviar puede ser el siguiente:

- **Bit de inicio** (*start*).
- **Bits de datos** (7/8).
- **Bit de paridad.** Este bit comprueba si los bits de datos han sido correctamente recibidos. Existen las siguientes variantes:
  - Paridad par. Si la suma de los bits de datos es par, el bit de paridad es 1, si es impar, el bit de paridad es 0.
  - Paridad impar. Si la suma de los bits de datos es impar, el bit de paridad es 1, si es par, el bit de paridad es 0.
  - Sin paridad. No se utiliza.
- **Bit de paro** (*stop*). Pueden ser uno o dos bits.

Es necesario enviar un mínimo de 10 bits por cada byte, figura A.4. Esto provocará una disminución de velocidad respecto a la transmisión en paralelo, pero es aceptable para los dispositivos externos usualmente utilizados. La menor velocidad es compensada por el incremento de seguridad y alcance de la señal.



**Figura A.4.** Secuencia de bits para comunicación serie.

### A.2.1. Cable conector

Consta de 25 patillas (DB-25), aunque en muchos equipos se utilizan conectores de 9 patillas (DB-9). En la figura A.5 se muestran las patillas del RS-232-C. El fabricante de equipos puede utilizar cualquiera de las restantes.

La función de cada una de las patillas la impone la norma:

- **Patilla2:** TRANSMISIÓN DE DATOS (TXD): Transmite datos del DTE (Ordenador) hacia el DCE (Modem).
- **Patilla3:** RECEPCIÓN DE DATOS (RXD): Transmite datos del DCE hacia el DTE.

DTE		DCE
2	TRANSMISIÓN DE DATOS (TXD)	2
3	RECEPCIÓN DE DATOS (RXD)	3
4	PETICIÓN DE ENVÍO (RTS)	4
5	DISPUESTO PARA ENVIAR (CTS)	5
6	DISPOSITIVO DE DATOS LISTO (DSR)	6
7	CIRCUITO COMÚN (SG)	7
8	DETECCIÓN DE PORTADORA DE DATOS (DCD)	8
20	TERMINAL DE DATOS LISTO (DTR)	20
22	INDICADOR DE LLAMADA (RI)	22

**Figura A.5.** Patigrama para el conector RS-232-C.

- **Patilla4:** PETICIÓN DE ENVÍO (RTS): Patilla de control, salida de propósito general del DTE. Sus usos varían ampliamente.
- **Patilla5:** DISPUESTO PARA ENVIAR (CTS): Entrada de propósito general del DTE. Normalmente mediante esta patilla el DCE indica cuando está preparado para aceptar datos para su posterior transmisión.
- **Patilla6:** DISPOSITIVO DE DATOS LISTO (DSR): Entrada de propósito general que indica al DTE que el DCE está encendido y listo para funcionar.
- **Patilla7:** CIRCUITO COMÚN (SG). Punto de referencia de todos los voltajes de la conexión
- **Patilla8:** DETECCIÓN DE PORTADORA (DCD): Se utiliza para desestabilizar la recepción en el DTE.
- **Patilla20:** TERMINAL DE DATOS LISTO (DTR). Indica al DCE que el DTE está encendido y listo para funcionar.
- **Patilla22:** INDICADOR DE LLAMADA (RI). Si el DCE es un modem, sirve para indicar al DTE que se está recibiendo una llamada telefónica.

Los nombres de la señal y su número de patilla son exactamente los mismos para el DTE y para el DCE. Pero opuestos funcionalmente, una salida en el DTE es una entrada en el DCE y viceversa.

Evidentemente para que exista flujo de datos entre dispositivos RS-232 únicamente serán imprescindibles las patillas 2 (TXD), 3 (RXD) y 7 (CIRCUITO COMÚN). En la figura A.6 se muestra la relación entre las patillas de DB-25 y el DB-9.

SEÑAL	DB-9	DB-25
DCD	1	8
RXD	2	3
TXD	3	2
DTR	4	20
SG	5	7
DSR	6	6
RTS	7	4
CTS	8	5
RI	9	22

**Figura A.6.** Relación entre DB-25 y DB-9.

### A.2.2. Niveles lógicos del RS-232-C

Los datos se transmiten con lógica negativa.

#### Definiciones lógicas para las salidas RS-232-C

Para garantizar un 0, una patilla de salida debe mantener un voltaje entre +5 y +15 v. Para 1 debe estar entre -5 v y -15v. Existe una región de transición +5 y -5 donde los niveles lógicos no están definidos.

#### Definiciones lógicas para las entradas RS-232-C

La única diferencia entre esta definición y la de las salidas es el ancho de la región de transición. La zona lógica indefinida de la entrada es únicamente de 6 voltios, mientras el área correspondiente para la salida es de 10v. Esta diferencia entre voltajes se conoce como margen de ruido.

En cuanto a los niveles lógicos de control y acoplamiento:

- Las entradas están habilitadas cuando son positivas y deshabilitadas cuando son negativas.
- Las salidas están afirmadas cuando son positivas e inhibidas cuando son negativas.

### A.2.3. Tipo de conexiones

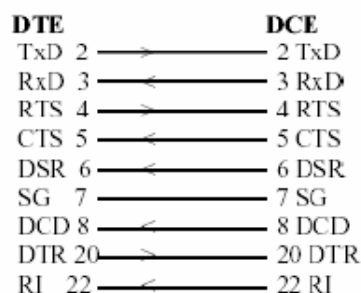
A pesar de la gran difusión de la norma RS-232-C no existe un único modelo estándar de cable que permita la interconexión de dos dispositivos RS-232-C, depende de dos factores:

- a) El género de dispositivo. Si se trata de dos dispositivos de distinto género la conexión es la natural, es decir se conectan entre si las patillas con el mismo número. Sin embargo si el género es el mismo es necesario intercambiar algunas patillas con el fin de mantener las entradas unidas con las salidas.

- b) El tipo de control de flujo de la comunicación. Si este se hace a través de algún protocolo software, como el XON/XOFF, basta con las líneas de transmisión y recepción de datos, puesto que el control se realiza a través de estas. En el caso de control de flujo por hardware son necesarias más líneas para la gestión de la comunicación. Existen, además, algunas formas de control de flujo híbridas a través de software ayudadas por algunas líneas hardware.

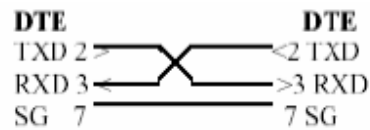
La conexión natural del estándar RS-232-C es entre un DTE, por ejemplo una PC, y un DCE, como un módem. Si el control de flujo es por hardware las señales de control cobran mayor importancia. El módem debe saber si el PC está listo antes de contestar automáticamente a una llamada, y el PC debe saber si el módem está encendido y si está recibiendo la señal de transmisión del sistema remoto. La forma adecuada de unir un DTE con un DCE es conectar los pines cuyos números son iguales, es decir el 1 del DTE con el 1 del DCE, el 2 con el 2 y así hasta la línea 25. En el caso de que la conexión sea sólo asíncrona, basta con conectar las nueve líneas como se muestra en la el diagrama de la figura 7 (2, 3, 4, 5, 6, 7, 8, 20 y 22).

Si además el control de flujo de datos entre el DTE y el DCE se realiza mediante software (utilizando un protocolo como el Hayes para los módems) solo es necesario unir tres líneas como se ilustra en el diagrama de la figura A.7 (2, 3, y 7), ya que las utilizadas para el control de flujo por hardware ya no serían necesarias.



**Figura A.7.** Diagrama de conexión asíncrona.

Uno de los usos no estándar más comunes de la norma RS-232-C es la conexión de dos DTE entre si. La forma más sencilla de conectar dos DTE (por ejemplo dos PC's) es unir la línea 2 de unos de ellos a la 3 del otro, y del mismo modo la 2 del segundo DTE a la 3 del primero, figura A.8. En la versión más básica de este tipo de cable, que se denomina "de módem nulo", no se conecta ninguna de las otras patillas excepto la tierra (7), que conforma el retorno común para los circuitos transmisor y receptor.



**Figura A.8.** Diagrama de conexión utilizando un software de control.

Un cable de tres hilos es suficiente, y funcionará con la mayoría de los programas (los que emplean control de flujo de la comunicación software), pero no con todos.

Algunos programas inspeccionan las líneas CTS, DSR y DCD y no funcionarían a no ser que alguna o todas estas señales sean 0 lógico u ON (control de flujo hardware). No obstante se puede engañar al programa conectando adecuadamente entre sí las líneas de control de los dos DTE. Existen muchos ejemplos de estos tipos de cableados, dependiendo normalmente su configuración del software de comunicaciones empleado.

#### **A.2.4. Características del convertidor RS-232 a USB de NI**

El convertidor de RS232 a USB que comercializa National Instruments, figura A.9, tiene las siguientes características:

- Compatible con puerto serial estándar de la PC.
- Transmisión de 128 B FIFO.
- Maneja hasta 8 bits de datos.
- Rango máximo de transferencia: 230.4 kb/s.
- Sistema operativo: Windows 2000/XP.
- Software recomendado: LabVIEW, LabWindows/CVI y Measurement Studio.
- Alimentación: 5 VDC típica 100 mA, máxima 200 mA.



Figura A.9. Convertidor NI RS232-USB.

### A.3. Programación del PIC16C765

A continuación se presentan los segmentos de programación ensamblador PIC16C765 [13] más representativos desarrollados para el presente proyecto de tesis. Se omiten los programas completos debido a lo extenso del código y los derechos de propiedad intelectual.

#### A.3.1. Configuración del LS7266R1

A continuación se muestra el código ensamblador del PIC16C765 utilizado para configurar el circuito contador LS7266R1 [12].

---

```

;*****
;*
;*   Archivos necesarios:   P16F877.INC
;*
;*****
;*
;*   Programa para el decodificador LS7266R1.
;*   Se usa el puerto B como bus de control y el puerto D como
;*   bus de datos.
;*
;*****

list      p=16f877           ; Directiva para definir procesador
#include <p16f877.inc>       ; Archivo para definir variables del procesador
errorlevel -302 ; Suprime el mensaje "Register not in bank0, check page bits"

        __CONFIG _CP_OFF & _WDT_OFF & _BODEN_ON & _PWRTE_ON & _HS_OSC &
        _WRT_ENABLE_ON & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

```

---

---

```

; '__CONFIG' es la directiva usada para insertar la configuración dentro del
archivo asm.
;***** DEFINICION DE VARIABLES

w_temp      EQU      0x70          ; Variable usada para resguardar W
status_temp EQU      0x71          ; Variable usada para resguardar el registro
de estado
cont1       EQU      0x20
cont2       EQU      0x21

;***** DEFINICION DE COMANDOS
CLOCK_DATA  EQU      d'14'
CLOCK_SETUP EQU      0x98
INPUT_SETUP EQU      0xC1
QUAD_X1     EQU      0xA8
QUAD_X2     EQU      0xB0
QUAD_X4     EQU      0xB8
BP_RESET    EQU      0x01
BP_RESETB   EQU      0x81
CNTR_RESET  EQU      0x02
CNTR_RESETB EQU      0x82
TRSFRRPR_CNTR EQU    0x08
TRSFRCNTR_OL EQU    0x90
EFLAG_RESET EQU      0x86

;***** INICIALIZACION

                ORG      0x000          ; Vector de reset del procesador
ceros          clrfs   PCLATH          ; Se asegura que los bits de pagina están en
                goto    principal      ; Se va al inicio del programa principal

;***** SERVICIO A INTERRUPCION

                ORG      0x004          ; Localidad del vector de interrupción
registro W     movwf   w_temp          ; Se resguarda el contenido actual del
a W            movf    STATUS,w        ; Se pasa el contenido del registro de estado
                movwf  status_temp     ; Se resguarda el registro de estado
                movf   status_temp,w   ; Se recupera copia del registro de estado
                movwf  STATUS          ; Se restaura el registro de estado
                swapf  w_temp,f        ; Se restaura el registro W como estaba
antes de la interrupcion swapf  w_temp,w
                retfie                   ; Regreso de la interrupción

;***** PROGRAMA PRINCIPAL
principal

                BANKSEL TRISA
                movlw  0x06
                movwf  ADCON1          ;Puerto A y E configurado como digital
                movlw  0xff
                movwf  TRISA          ; Puerto A como entrada
                clrfs  TRISB          ; Puertos B, C, D y E como salidas
                clrfs  TRISC
                clrfs  TRISD
                clrfs  TRISE
                BANKSEL PORTA

```

---



---

```
        call    INI_LS7266R1

ciclo
    BANKSEL   TRISD           ; Puerto D como salida
    clrf     TRISD           ; para escribir al 7266
    BANKSEL   PORTD

    movlw    b'10000001'     ; Reset BP, X y Y
    call     ESRECO
    movlw    b'10000000'     ; NOP en RLD
    call     ESRECO

    movlw    b'10010000'     ; Transfiere CNTR a OL
    call     ESRECO
    movlw    b'10000000'     ; NOP en RLD
    call     ESRECO

    BANKSEL   TRISD           ; Puerto D como entrada
    movlw    0xFF            ; para tomar lecturas del 7266
    movwf    TRISD
    BANKSEL   PORTD

;***** Lectura del byte menos significativo

    movLw    b'00000010
    movwf    PORTB
    NOP

    movf     PORTD, W
    movwf    PORTC

    call     retardo

    movLw    b'00001110
    movwf    PORTB

;***** Lectura del byte medio

    movLw    b'00000010
    movwf    PORTB
    NOP

    movf     PORTD, W
    movwf    PORTC

    call     retardo

    movLw    b'00001110
    movwf    PORTB

;***** Lectura del byte mas significativo

    movLw    b'00000010
    movwf    PORTB
    NOP

    movf     PORTD, W
    movwf    PORTC

    call     retardo

    movLw    b'00001110
    movwf    PORTB
```

---

```

        incf     PORTE, F
        goto    ciclo

;***** SUBRUTINAS

;***** Inicializa al LS7266R1
INI_LS7266R1

        movlw   b'10000110'      ; Reset E flag
        call    ESRECO
        movlw   b'10000000'      ; NOP
        call    ESRECO

        movlw   b'10000001'      ; Reset BP, X y Y
        call    ESRECO
        movlw   b'10000000'      ; NOP
        call    ESRECO

n=0, fFCKn=fFCK/(n+1)
        movlw   0x00              ; Preescalador para el filtro de reloj
        call    ESREXPR          ; Escribe en XPR0

        movlw   b'10011000'      ; Transfiere PR0 a PSC, X y Y
        call    ESRECO
        movlw   b'10000000'      ; NOP
        call    ESRECO

        movlw   b'11000001'      ; Habilita entradas A y B, X y Y
        call    ESRECO

        movlw   b'10111000'      ; Cuadratura x4
        call    ESRECO

        movlw   b'10000010'      ; Reset CNTR, X y Y
        call    ESRECO

        return

;***** Escribir en ambos registros de control
;
; -----
; | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
; -----
; | x  | x  | x  | 'X/Y | 'RD | 'CS | 'WR | C/'D |
; -----
;

ESRECO
        movwf   PORTD
        movlw   b'00001001'      ; B7, B6, B5: nada, B4: 'X/Y, B3: 'RD,
B2: 'CS, B1: 'WR, B0: C/'D
        movwf   PORTB          ; Se activan 'CS y 'WR para escritura
        movlw   b'00001111'
        movwf   PORTB          ; Se desactivan 'CS y 'WR
        return

;***** Escribir en registro XPR y YPR
;
; -----
; | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
; -----
; | x  | x  | x  | 'X/Y | 'RD | 'CS | 'WR | C/'D |
; -----
;

```

---

```

ESREXPR
    movwf    PORTD
    movlw   b'00001000'
    movwf   PORTB
    movlw   b'00001110'
    movwf   PORTB
    return

ESREYPR
    movwf    PORTD
    movlw   b'00011000'
    movwf   PORTB
    movlw   b'00011110'
    movwf   PORTB
    return

;***** Retardo
retardo
    movlw   0x55                ; Carga w con 255 decimal
    movwf   cont1              ; Se mueve w a cont1
carga    movlw 0xff            ; Carga w con 255 decimal
    movwf   cont2              ; Se mueve w a cont1
salto2   decfsz cont2, f       ; Si cont2 no es cero se sigue
decrementando, sino se salta la siguiente instruccion
    goto    salto2            ; Salto a salto
    decfsz  cont1, f          ; Si cont1 no es cero se sigue
decrementando, sino se salta la siguiente instruccion
    goto    carga             ; Salto a carga
    return

                                END                                ; Directiva de fin de programa

```

---

### A.3.2. Comunicación USB

Se mostro el código para la operación del decodificador en cuadratura LS7266R1. No se incluye mayor parte del código referente a la comunicación por puerto USB, debido a las limitaciones de espacio.

Para este objetivo Microchip provee a sus usuarios software específicamente para manejar la interfaz USB con el microcontrolador PIC16C745/65. Se trata de una simple interfaz “*Put/Get*” de comunicación. El punto importante es el manejo de interrupciones sin dejar de mencionar la importancia de enumerar procesos y la comunicación de datos.

En forma básica inicializa la interfaz USB para que el *host* enumere al dispositivo que contenga el pic16c745/65, de esta manera nuestra aplicación principal puede iniciar el envío y recepción de datos por USB.

La versión de este software de interfaz USB esta disponible en el sitio web <http://www.microchip.com/>.



## Bibliografía

- [1]. Asociación de empresas de montaje y mantenimiento industrial, “Guía de buenas prácticas en materia de Metrología. Estudio comparativo del uso de aplicaciones metrológicas en el sector industrial”, ADEMI, enero 2008, pp. 1-61.
- [2]. CENAM, “División de metrología dimensional”, CENAM, enero 2008, <http://www.cenam.mx/dimensional/todo.htm>.
- [3]. CENAM, “Laboratorio de bloques patrón por comparación mecánica e interferometría”, CENAM, enero 2008, <http://www.cenam.mx/dimensional/Labs.htm>.
- [4]. CENAM, “Patrón nacional de longitud”, CENAM, enero 2008, pp. 1.
- [5]. Colín Carlos, Díaz Mario, “Guía técnica sobre trazabilidad e incertidumbre en metrología dimensional”, Secretaría de Economía-CENAM-entidad mexicana de acreditación, enero 2008, pp. 1-12.
- [6]. Hernández Gutiérrez Ignacio, “El CENAM y el sistema Metrológico Nacional”, CENAM-Secretaría de Economía, enero 2008, pp. 1-11.
- [7]. Laboratorio de Metrología, Curso de Metrología Dimensional, CCADET-UNAM, abril 2007.
- [8]. LAPEM-CFE. Procedimiento para la calibración de bloques patrón, LAPEM-CFE, 2006, pp. 1-15.
- [9]. LAPEM-CFE. Procedimiento para la calibración de indicadores de carátula mayores a 1 mm, LAPEM-CFE, 2006, pp. 1-19.

- [10]. LAPEM-CFE. Procedimiento para la calibración de micrómetros de profundidades analógicos y digitales, LAPEM-CFE, 2006, pp. 1-21.
- [11]. LAPEM-CFE. Procedimiento para la calibración de micrómetros para medición de exteriores, LAPEM-CFE, 2006, pp. 1-26.
- [12]. LSI/CSI, “LS7266R1 24 bit dual-axis quadrature counter”, LSI Computer Systems Inc, 2004, pp. 1-15.
- [13]. Microchip, “PIC16C745/65 Datasheet”, Microchip Technology, 2000, pp. 1-71.
- [14]. Molina Javier, “Acreditación de la competencia de un laboratorio de metrología dimensional: normas y requisitos”, Depto. de Ingeniería Industrial y Manufactura. Universidad Autónoma de Ciudad Juárez, enero 2008, pp. 1-12.
- [15]. National Instruments, “Adquiera, analice y presente con LabVIEW”, National Instruments, febrero 2008, <http://www.ni.com/swf/presentation/esa/aap/>.
- [16]. National Instruments, “Creating a Report in Microsoft Excel Using the LabVIEW Report Generation Toolkit”, National Instruments, febrero 2008 <http://zone.ni.com/devzone/cda/tut/p/id/3180>.
- [17]. National Instruments, “Getting started with LabVIEW”, National Instruments, febrero 2008, <http://zone.ni.com/devzone/cda/tut/p/id/7001>.
- [18]. National Instruments, “GPIB Instrument control tutorial”, National Instruments, febrero 2008, <http://zone.ni.com/devzone/cda/tut/p/id/2761>.
- [19]. National Instruments, “GPIB (IEEE 488) Solutions”, National Instruments, febrero 2008, <http://www.ni.com/gpib/>.
- [20]. National Instruments, “Prácticas para crear grandes aplicaciones de LabVIEW con un enfoque de desarrollo estructurado”, National Instruments, febrero 2008, <http://zone.ni.com/devzone/cda/tut/p/id/5932>.
- [21]. National Instruments, “Punto de inicio en comunicación serial”, National Instruments, febrero 2008, <http://digital.ni.com/public.nsf/allkb/8C5F1FDC45A30155862570E500711955>.

- 
- [22]. National Instruments, “¿Qué es la instrumentación virtual?”, National Instruments, febrero 2008, <http://digital.ni.com/worldwide/latam.nsf/web/all/01E4BFF8EC93532086256B6000669953>.
- [23]. National Instruments, “Serial Interfaces”, National Instruments, febrero 2008, <http://www.ni.com/serial/>.
- [24]. National Instruments, “Using external code in LabVIEW”, National Instruments, junio 2000, pp 17-50.
- [25]. Pressman Roger, Ingeniería de Software: un enfoque práctico, 3ª Edición Mc Graw-Hill, 1993.
- [26]. Seco Fernando, “Conexión de instrumentos de medida con GPIB”, Consejo Superior de Investigaciones Científicas, febrero 2008, pp. 1-12.
- [27]. Torres León Esteban, “Norma RS-232-C”, Benemérita Universidad Autónoma de Puebla Facultad de Ciencias de la Computación, febrero 2008, pp. 1-10.

