

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA
AUTOMATIZACIÓN Y CONTROL

PROYECTO DE GRADO PARA LA OBTENCIÓN
DEL TÍTULO DE INGENIERÍA

**DESARROLLO DE UN SOFTWARE DE ADQUISICIÓN DE
DATOS EN LABVIEW PARA LA RED DE SENSORES
INALÁMBRICOS CROSSBOW ORIENTADO A LA
REALIZACIÓN DE PRÁCTICAS EN EL LABORATORIO
DE INSTRUMENTACIÓN Y SENSORES DEL
DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

CARLOS ALBERTO VIVANCO ALBUJA

Sangolquí – Ecuador

2008

DESARROLLO DE UN SOFTWARE DE ADQUISICIÓN DE
DATOS EN LABVIEW PARA LA RED DE SENSORES
INALÁMBRICOS CROSSBOW ORIENTADO A LA
REALIZACIÓN DE PRÁCTICAS EN EL LABORATORIO
DE INSTRUMENTACIÓN Y SENSORES DEL
DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CERTIFICACIÓN

Certificamos que el Sr. CARLOS ALBERTO VIVANCO ALBUJA desarrolló y finalizó el proyecto en su totalidad, como requerimiento parcial para la obtención el Título de INGENIERO ELECTRÓNICO, bajo nuestra dirección.

Ing. Pablo Sevilla
DIRECTOR

Ing. Román Lara
CODIRECTOR

RESUMEN

El presente proyecto de tesis contempla el desarrollo de un Software de Adquisición de Datos para la Red de Sensores Inalámbricos Crossbow utilizando el programa Instrumentación Virtual LabVIEW; este software será utilizado didácticamente para realizar prácticas en el Laboratorio de Instrumentación y Sensores del Departamento de Eléctrica y Electrónica de la ESPE.

La estación base MIB520, los nodos MPR2400 y los sensores MTS310, corresponden el hardware que en conjunto forman la Red de Sensores. El tablero MTS310 consta de un termistor, una fotoresistencia, un detector de tono, un acelerómetro, un magnetómetro y un zumbador, de estos sensores únicamente se utilizarán los primeros cuatro elementos, cuya información se encuentra distribuida dentro de una trama de 39 bytes, en la que también se incluye información del nodo y el estado de la batería.

La interfaz permite la visualización amigable de cada una de las magnitudes en ventanas independientes; para las mediciones de temperatura y luz el software cuenta con una simulación de acción de control dependiendo de los niveles de alarmas configurados, un gráfico histórico del proceso de muestreo y la generación de un archivo de los datos adquiridos.

El proyecto finaliza con la elaboración de las Guías de Prácticas de laboratorio para los alumnos de la materia de Instrumentación y Sensores.

DEDICATORIA

A mis padres por el apoyo y el gran sacrificio que hicieron durante todo este tiempo para que pueda lograr este objetivo en mi vida, a mi hermana que creció junto a mí y fue un pilar importante para no darme por vencido cuando los problemas embargaron nuestras vidas, a mis abuelos por la fe que tenían en alcanzar este sueño compartido y a mis amigos con quienes compartí estos maravillosos años llenos de buenos momentos que jamás olvidare...

AGRADECIMIENTO

Al apoyo y confianza depositada en mí por parte del Ing. Pablo Sevilla mentalizador de este proyecto, por sus conocimientos brindados como director de tesis durante el tiempo de elaboración de este documento, a quien pude acudir para despejar mis dudas siempre encontrando las puertas abiertas no solo como profesor sino como amigo; de la misma manera un gran agradecimiento al Ing. Román Lara codirector y a todos los profesores que me guiaron durante la carrera en la Escuela Politécnica del Ejército.

No puedo continuar sin antes expresar mi agradecimiento especial a una persona que me ayudo durante todos estos años, que no hace falta escribir su nombre aquí porque le estaré eternamente agradecido.

PRÓLOGO

Los Sistemas de Adquisición de Datos conforman una parte fundamental en la instrumentación, donde la primera etapa de estos sistemas es la medición o sensamiento de magnitudes físicas.

La rápida evolución de la electrónica en todas sus ramas ha generado nuevas tecnologías y tendencias de sensamiento, dando como partida las Redes de Sensores Inalámbricos, en la actualidad estas redes son muy importantes y aplicativas dentro de distintos campos como son la industria, el medioambiente, la salud, tácticas de guerra, prevención de desastres naturales entre otros.

Otro despunte que ha tenido la Ingeniería Electrónica es la Instrumentación Virtual, debido a que muchas veces la realización de una medida requiere la intervención de varios instrumentos, unos generan estímulos sobre el dispositivo de medición y otros recogen la respuesta a estos estímulos, este conjunto es un sistema de instrumentación y puede ser reemplazado por una computadora personal como si fuera un instrumento real. Un sistema de instrumentación virtual está enfocado a los instrumentos encargados de medir señales, registrar datos y en ciertos casos decidir las acciones de control.

En este proyecto se propone la interacción entre las Redes de Sensores Inalámbricos Crossbow con las que cuenta el Departamento de Eléctrica y Electrónica con la Instrumentación Virtual, para fortalecer el estudio de Sistemas de Adquisición de Datos en la materia de Instrumentación y Sensores que en la actualidad por falta de equipo necesario únicamente se lo hace de manera teórica sin la posibilidad de realizar prácticas de laboratorio que complementen el estudio.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	III
RESUMEN.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
PRÓLOGO	VII
ÍNDICE DE CONTENIDOS	VIII
GLOSARIO	XI
CAPÍTULO 1.....	12
INTRODUCCIÓN.....	12
1.1 ANTECEDENTES.....	12
1.2 JUSTIFICACIÓN	12
1.3 IMPORTANCIA.....	13
1.4 ALCANCE.....	13
CAPÍTULO 2.....	15
GENERALIDADES	15
2.1 CONCEPTOS BÁSICOS.....	15
2.1.1 Errores sistemáticos	15
2.1.2 Errores accidentales.....	15
2.1.3 Error absoluto y relativo	15
2.1.4 Terminología	16
2.1.5 Nomenclatura P&ID.....	17
2.1.6 Acondicionamiento y procesado de señal.....	19
2.2 WIRELESS SENSOR NETWORKS	19
2.2.1 Características	19
2.2.2 Arquitectura.....	19
2.2.3 IEEE 802.15.4	23
2.2.4 Aplicaciones	23
2.3 SISTEMAS DE ADQUISICIÓN DE DATOS	24
2.3.1 Características	24
2.3.2 Arquitectura.....	25
2.3.3 Tipos de Sistemas de Adquisición de Datos.....	26
2.4 INSTRUMENTACIÓN VIRTUAL	28
2.4.1 Definición	28
2.4.2 La instrumentación virtual en la Ingeniería Electrónica.....	28
2.4.3 Diseño de instrumentos virtuales.....	29
CAPÍTULO 3.....	30
SENSORES MTS310, NODOS MPR2400 Y MODULO MIB520	30
3.1 CARACTERÍSTICAS GENERALES	30
3.1.1 TinyOS y NesC.....	30
3.1.2 MoteView.....	31
3.2 MPR2400 Y MTS310.....	31
3.2.1 Descripción.....	32
3.3 MIB520	34
3.3.1 Descripción.....	34
3.4 PROGRAMACIÓN NODOS Y ESTACIÓN BASE.....	35
3.4.1 Programación mediante Cygwin.....	35
3.4.2 Programación mediante MoteConfig 2.0	36
3.5 COMUNICACIÓN SERIAL USB.....	39
3.6 TRAMA DE DATOS	39
3.7 CONVERSIÓN A UNIDADES DE INGENIERÍA	41

CAPÍTULO 4.....	44
DISEÑO DEL SOFTWARE.....	44
4.1 LABVIEW	44
4.1.1 VISA.....	45
4.1.2 Configuración Serial USB.....	45
4.2 DISEÑO DE LA INTERFAZ HMI	46
4.2.1 Características Generales	46
4.2.2 Parámetros	47
4.3 PROGRAMACIÓN EN LABVIEW	55
4.3.1 Diagramas de Flujo.....	56
4.3.2 Descripción de Herramientas Utilizadas	64
CAPÍTULO 5.....	71
PRUEBAS Y RESULTADOS	71
5.1 PRUEBAS DE TRANSMISIÓN.....	71
5.2 PRUEBAS DE ADQUISICIÓN DE DATOS.....	73
5.3 ANÁLISIS DE RESULTADOS.....	77
CAPÍTULO 6.....	78
PRACTICAS DE LABORATORIO.....	78
6.1 PRÁCTICA 1: SONIDO	78
6.1.1 Objetivos.....	78
6.1.2 Marco Teórico.....	78
6.1.3 Equipo Necesario	79
6.1.4 Desarrollo de la práctica.....	79
6.1.5 Cuestionario	81
6.1.6 Gráficas y Análisis de Resultados	82
6.1.7 Conclusiones y Recomendaciones	83
6.1.8 Bibliografía.....	83
6.2 PRÁCTICA 2: ILUMINACIÓN.....	84
6.2.1 Objetivos.....	84
6.2.2 Marco Teórico.....	84
6.2.3 Equipo Necesario	87
6.2.4 Desarrollo de la Práctica	87
6.2.5 Cuestionario	89
6.2.6 Gráficas y Análisis de Resultados	90
6.2.7 Conclusiones y Recomendaciones	91
6.2.8 Bibliografía.....	91
6.3 PRÁCTICA 3: TEMPERATURA.....	92
6.3.1 Objetivos.....	92
6.3.2 Marco Teórico.....	92
6.3.3 Equipo Necesario	93
6.3.4 Desarrollo de la Práctica.....	94
6.3.5 Cuestionario	95
6.3.6 Gráficas y Análisis de resultados	97
6.3.7 Conclusiones y Recomendaciones	97
6.3.8 Bibliografía.....	97
6.4 PRACTICA 4: MOVIMIENTO	98
6.4.1 Objetivos.....	98
6.4.2 Marco Teórico.....	98
6.4.3 Equipo Necesario	99
6.4.4 Desarrollo de la Práctica.....	100
6.4.5 Cuestionario	101
6.4.6 Graficas y Análisis de Resultados	102
6.4.7 Conclusiones y Recomendaciones	102
6.4.8 Bibliografía.....	102
CAPÍTULO 7.....	103
CONCLUSIONES Y RECOMENDACIONES.....	103
7.1 CONCLUSIONES	103
7.2 RECOMENDACIONES	104

REFERENCIAS BIBLIOGRÁFICAS	106
ANEXO 1	108
CÓDIGO INTERFAZ HMI	108
ANEXO 2	129
CÓDIGO DE PROGRAMACIÓN NODOS Y ESTACIÓN BASE.....	129
ANEXO 3	142
MANUAL DE USUARIO	142
ANEXO 4	151
FORMAS PARA DIAGRAMAS DE FLUJO	151
ANEXO 5	152
GUÍAS DE LABORATORIO	152
ÍNDICE DE FIGURAS	177
ÍNDICE DE TABLAS	179

GLOSARIO

ADC: Analog Digital Converter (Conversor Análogo Digital).

Fotoresistencia: Elemento electrónico que varía su resistencia con el cambio de luz incidente sobre la superficie de detección.

HMI: Human Machine Interface.

IEEE 802.15.4: protocolo para la transmisión de datos de forma bidireccional entre pequeños dispositivos como sensores de seguridad, módulos de automatización, mandos a distancia con tecnologías robustas y fiables.

ISP: in-system processor.

LabVIEW: Laboratory Virtual Instrument Engineering Workbench.

MEMS: Micro Electro Mechanic System (Micro sistema electro mecánico).

MIB520: Mote Interface Board.

MICAz: Última generación en Plataformas de hardware para motas de Crossbow Technology.

MPR2400: Mote Processor Radio.

MTS310: Mote Sensors.

NesC: Network Embedded Systems C.

P&ID: Process and Instrumentation Diagram (Diagrama de Proceso e Instrumentación) .

PLC: Programmable Logic Controller (Controlador Lógico Programable).

SAD: Sistema de Adquisición de Datos.

Sistema Embebido: es aquel que integra en su circuitería un sistema digital reprogramable o reconfigurable, tal como un Microcontrolador.

Termistor: Elemento electrónico que varía su resistencia con el cambio de temperatura.

TinyOS: Tiny microthreading Operative System

USB: Universal Serial Bus (Bus Serial Universal).

VISA: Virtual Instrument Software Architecture (Arquitectura de Software para Instrumentos Virtuales).

WSN: Wireless Sensor Network (Red de Sensores Inalámbricos).

CAPÍTULO 1

INTRODUCCIÓN

1.1 ANTECEDENTES

El Laboratorio de Instrumentación y Sensores del Departamento de Eléctrica y Electrónica de la ESPE dispone de varios módulos analógicos para analizar el comportamiento de ciertas magnitudes físicas, que complementan el pensum de estudios mediante prácticas didácticas relacionadas a los distintos temas que se tratan en el aula.

Sin embargo dentro del pensum académico también se encuentra estipulado la digitalización de señales mediante sistemas de adquisición de datos que actualmente se los trata en teoría, debido a que el laboratorio carece de algún modulo para realizar prácticas que refuercen estos contenidos.

El Departamento de Eléctrica y Electrónica adquirió hace ya un tiempo atrás un sistema WSN¹ cuyo software de paquete permite únicamente el monitoreo de datos de las distintas magnitudes físicas, sin poseer la capacidad de realizar algún tipo de análisis matemático o físico con las mismas. Luego de haber sido adquirido este sistema se realizo un proyecto de grado utilizando una red sensorial denominado: **Estudio, diseño e implementación de una Red Sensorial Inalámbrica para el Monitoreo y Control de Temperatura en un Ambiente Industrial Reducido**, en este proyecto se utilizo una red de cinco sensores de temperatura, donde la visualización del ambiente fue realizada en Visual Basic; cuyo objetivo principal fue el monitoreo y control de la Temperatura en un área industrial reducida, que simulará un Cuarto de Telecomunicaciones Avanzadas.

1.2 JUSTIFICACIÓN

Las prácticas de laboratorio perfeccionan el estudio y comprensión de sistemas de adquisición de datos para situaciones no consideradas en la parte teórica ya sean estas

¹ WSN: Siglas en ingles de Red de Sensores Inalámbricos (Wireless Sensor Network)

perturbaciones, ruido, atenuación, retardos en la transmisión, entre otros. Los sistemas de adquisición son de gran importancia ya que representan la primera etapa en el proceso de control y en muchos casos una herramienta que permite aprovechar las bondades de un sistema micro procesado.

Por los antecedentes antes mencionados y de acuerdo a las nuevas tendencias en sensamiento se ha pensado en fortalecer el laboratorio de instrumentación y sensores mediante un software para análisis de adquisición de datos que cubrirá el programa de estudios a cabalidad sin dejar vacíos en el mismo y así contar con alumnos mejor preparados especialmente para quienes se orienten a la especialidad de automatización y control.

1.3 IMPORTANCIA

En la actualidad para llevar un control eficaz sobre ciertos procesos es necesario captar una serie de datos, para posteriormente analizarlos, tratarlos, almacenarlos y llevar a cabo una presentación clara de la evolución de dicho proceso. Generalmente, los datos o variables que se han de captar tienen un carácter analógico, mientras que su tratamiento, almacenamiento y análisis son mucho más eficientes cuando se hace digitalmente.

Esto implica una serie de etapas electrónicas que permitan disponer de una conversión de los datos desde el campo analógico al campo digital, sin que por ello se deban perder aspectos fundamentales para el proceso que se desea controlar.

La automatización no sólo se focaliza en la generación de procesos más rápidos, sino también en que estos sean más exactos y que informen a operarios sobre el funcionamiento o sobre fallos que se puedan presentar.

1.4 ALCANCE

El proyecto pretende elaborar un software para adquisición de datos provenientes de la red de sensores inalámbricos conformada por los dispositivos MPR2400 y MTS310 (sensores) y el módulo MIB-520CA (base de comunicación USB) de la compañía CROSSBOW® adquiridos por el Departamento de Eléctrica y Electrónica el año anterior.

En este software se analizará las magnitudes físicas de sonido, iluminación, temperatura y aceleración; utilizando el sensor acústico, sensor de iluminación, sensor de temperatura y acelerómetro respectivamente.

La primera fase del proyecto constará de un estudio del hardware: estación base MIB520CA y dispositivos de sensado MTS310 y nodo MPR2400; así como también el estudio del software de fábrica MoteView, con sus respectivas configuraciones y programaciones.

Una vez familiarizado con la Red de Sensores Inalámbricos se procederá a determinar la comunicación serial USB entre LabVIEW™ y los dispositivos CROSSBOW®, así como también el análisis e interpretación de las magnitudes físicas en la trama de datos.

Determinada la comunicación HARDWARE-SOFTWARE y descifrada la trama de datos, la tercera fase del proyecto será el desarrollo del software en LabVIEW®.

Finalmente se realizará ejemplos para prácticas de laboratorio que permitan una asimilación completa del comportamiento de estas magnitudes físicas.

CAPÍTULO 2

GENERALIDADES

2.1 CONCEPTOS BÁSICOS

La instrumentación es una de las partes fundamentales en la electrónica, permite medir magnitudes físicas cuyo objetivo es la obtención de datos que se transmiten a dispositivos de representación, registro o elementos de control.

En instrumentación como en otros experimentos nunca se puede medir un valor sin errores debido a varios aspectos, ya sean estos humanos, ambientales o instrumentales. Los errores se los puede clasificar en sistemáticos y accidentales.

2.1.1 Errores sistemáticos

Son errores producidos por métodos inadecuados, instrumentos defectuosos o por la utilización de los mismos en condiciones físicas para las cuales no estaban diseñados. Estos errores no forman parte de la teoría de errores, son equivocaciones que pueden evitarse mediante la utilización de procedimientos e instrumentos de medición adecuados.

2.1.2 Errores accidentales

También conocidos como errores aleatorios, son producto de causas incontrolables e imprevisibles que producen distintos resultados bajo las mismas condiciones de medida. Los errores accidentales no pueden evitarse, pero se puede estimar su magnitud por medio de métodos estadísticos.

2.1.3 Error absoluto y relativo

Por varios motivos no es posible determinar exactamente un error, pero si no se conoce el error en una medición no se sabe qué tan confiable es el resultado, por eso es que un resultado experimental sin un error asociado no representa nada.

- **Error Absoluto**

Error absoluto se lo puede definir como la diferencia que se tiene entre el valor obtenido y el verdadero.

- **Error Relativo**

El error relativo es definido por el cociente entre el error absoluto y el valor real. El error absoluto y el error relativo son aproximadamente iguales para números cercanos a uno. Para números no cercanos a uno puede haber una gran diferencia.

2.1.4 Terminología

A continuación se definirán algunos los términos utilizados en la instrumentación.

- **Rango**

Se conoce como rango a la región comprendida entre los límites superior e inferior en la cual una cantidad es medida.

- **Alcance**

El alcance o Span es la diferencia algebraica entre el valor máximo y mínimo del rango de medición.

- **Exactitud**

El grado de coincidencia entre el resultado anunciado y el valor de referencia reconocido.

- **Precisión**

Es la concordancia entre los resultados obtenidos en mediciones independientes realizadas bajo las mismas condiciones.

- **Sensibilidad**

Es la relación entre la variación observada en la indicación de instrumento respecto al cambio de la variable medida. Suele indicarse como porcentaje del alcance o span.

- **Resolución y Repetibilidad**

La **resolución** es el cambio más pequeño del valor medido al cual el instrumento responde.

Repetibilidad es la repetición de las medidas tomadas por el mismo operador utilizando el mismo equipamiento, bajo idénticas condiciones dentro de intervalos cortos de tiempo.

2.1.5 Nomenclatura P&ID²

Los instrumentos de medición y control se designan y representan, de acuerdo a la Tabla. 2.1. Tabla. 2.2. y Tabla. 2.3. [1]

	PRIMERA LETRA	LETRAS SUCESIVAS
A	Análisis	Alarma
B	Llama	Libre
C	Conductividad	Controlador
D	Densidad o peso específico	Diferencial
E	Voltaje	Elemento primario
F	Flujo	Relación
H	Manual (Arranque Manual)	Alto
I	Corriente eléctrica	Indicador
J	Potencia	Exploración
K	Tiempo	Estación de control
L	Nivel	Bajo o luz piloto
M	Humedad	Medio o intermedio
O	Libre	Orificio
P	Presión o vacío	Punto de prueba
Q	Cantidad	Totalizador
R	Radioactividad	Registrador
S	Velocidad o frecuencia	Interruptor o seguridad
T	Temperatura	Transmisor
V	Viscosidad	Válvula
W	Peso o Fuerza	Depósito
Y	Libre	Relé o computador
Z	Posición	E.f.c sin clasificar

Tabla. 2.1. Significado de las letras de identificación funcional

² P&ID: Siglas en Ingles de Diagrama de Proceso e Instrumentación (Process and Instrumentation Diagram)

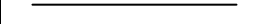
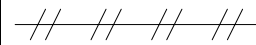

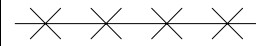

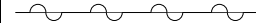
SEÑAL	DESCRIPCIÓN
	Conexión a proceso o enlace mecánico.
	Señal neumática
	Señal eléctrica
	Tubo Capilar
	Señal hidráulica
	Señal electromagnética o sónica

Tabla. 2.2. Símbolos de señales y/o conexiones







SÍMBOLO	DESCRIPCIÓN
	Instrumento ubicado localmente
	Instrumento ubicado en un panel, accesible al operador
	Instrumento ubicado detrás de un panel, no accesible al operador
	Sistema de Control distribuido
	PLC, accesible al operador
	Computador no accesible al operador

Tabla. 2.3. Simbología de instrumentos generales

Cabe señalar que la utilización de esta simbología no es obligatoria, pero si recomendada para una identificación clara de los instrumentos.

2.1.6 Acondicionamiento y procesado de señal

“El termino acondicionamiento de señal se utiliza para el elemento de un sistema de medida que convierte la señal del transductor en un formato adecuado para realizar el posterior procesado y, así, la visualización. La salida un acondicionador de señal es, normalmente, una tensión o corriente en c.c. Típicamente, los acondicionadores de señal son puentes donde un cambio de resistencia, capacitancia o inductancia se convierte en una diferencia de potencial o corriente de desequilibrio.

El término procesado de señal se utiliza, a menudo, para los procesos que se producen en la señal a fin de adecuarla para poder visualizar.” [2]

2.2 WIRELESS SENSOR NETWORKS

Una Red de Sensores Inalámbricos (WSN) es un sistema en el cual se relacionan la detección, procesamiento y transmisión/recepción de información de diversas magnitudes físicas de varios nodos distribuidos en un área, sin interconexión física entre ellos a una estación o nodo base.

2.2.1 Características

En una WSN al no existir interconexión entre nodos y estación base las características principales que presenta este sistema son: facilidad de despliegue al no utilizar una infraestructura de red, es decir el enrutamiento entre nodos es sin visión directa y con comunicaciones multisalto, posee una topología dinámica en la que sus nodos son autoconfigurables con tolerancia a fallos de muy bajo consumo de energía en funcionamiento, larga autonomía, tamaño reducido y operación sin mantenimiento.

La operación de una WSN se puede realizar en dos bandas a 2.4 GHz con una tasa máxima de transferencia de 250 Kbps, y de 868-928 MHz para una tasa de datos comprendida entre 20 y 40 Kbps y el rango de funcionamiento de los nodos oscila entre 10m hasta 100m.

2.2.2 Arquitectura

Una Red de Sensores Inalámbricos está conformada por los siguientes elementos:

- **Sensores**

Toman del medio la información y la convierten en señales eléctricas.



Figura. 2.1. Sensores Crossbow MTS310

- **Nodos**

Conocidos como motas, toman los datos del sensor y envían su información a la estación base.



Figura. 2.2. Nodos o Motas Crossbow MPR2400

- **Gateway**

Elementos para la interconexión entre la red de sensores y un computador o una red TCP/IP.



Figura. 2.3. Gateway

- **Estación Base.**

Recibe los datos enviados por las motas en un ordenador común o un **sistema embebido**³.



Figura. 2.4. Estación Base MIB520

La arquitectura de una WSN se la puede dividir de acuerdo al dinamismo de la red, la distribución de los nodos (Topología), comunicación y habilidades de los nodos, la entrega de los datos.

- **Dinamismo de la Red**

Los sensores son dinámicos lo que provoca mayor complejidad para gestionar la red. Las aplicaciones son recreativas cuya utilidad es la de monitorear eventos, con alta generación de tráfico en el Gateway.

- **Topología**

Puede ser determinista es decir que tiene caminos predeterminados para el routing, o Ad-hoc donde la distribución de los nodos es no uniforme y mejoran la eficiencia energética con mayor rendimiento.

³ **Sistema Embebido:** es aquel que integra en su circuitería un sistema digital reprogramable o reconfigurable, tal como un Microcontrolador.

- **Comunicación de Nodos**

Pueden ser:

Simple Hop o directa funciona si el nodo esta cerca de la estación base.

Multi-Hop mayor complejidad de administración, menor energía consumida, esta comunicación es inevitable para sensores distribuidos no uniformemente.

- **Entrega de datos**

Los datos que en una WSN son enviados periódicamente y al terminar la transmisión los sensores se detienen, se conoce como entrega **continúa**; sin embargo también existe la entrega de datos **Even-Driven** que únicamente envían datos cuando ocurre un evento.

- **Habilidades de los Nodos**

Las habilidades de los nodos en una WSN dependen de las aplicaciones a las cuales ellos se hallan relacionados; si poseen una misma computación, comunicación y consumo la habilidad es **Homogénea**, en cambio es **Heterogénea** si cada uno de los nodos tiene una o varias funciones.

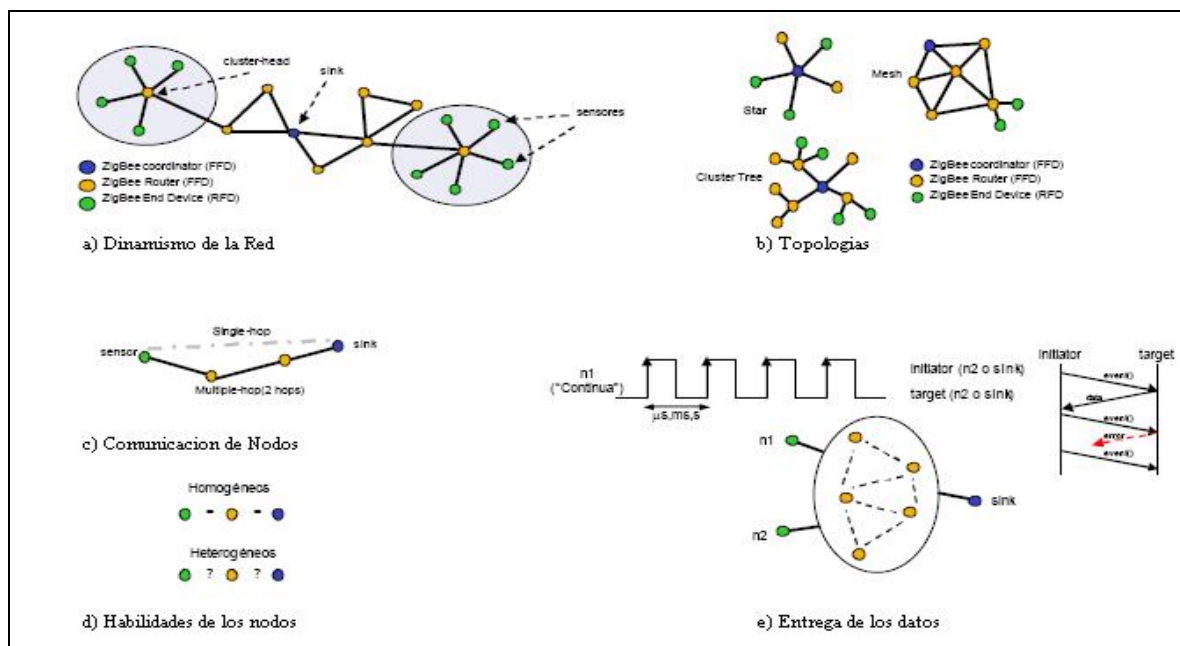


Figura. 2.5. Arquitectura Wireless Sensor Network

2.2.3 IEEE 802.15.4

“El estándar IEEE 802.15.4 es un protocolo sencillo, pero a la vez potente, para la transmisión de datos de forma bidireccional entre pequeños dispositivos como sensores de seguridad, módulos de automatización, mandos a distancia, que usa tecnologías robustas y fiables, como la modulación de espectro ensanchado, mecanismos de prioridad, comprobación de errores. Este estándar especifica el nivel físico (PHY) y el nivel de acceso al medio (MAC) y podrá ser usado en la bandas de 2,4 GHz, 915 MHz (América) y 868 MHz (Europa), lo cual le hace un estándar global.” [3]

- **ZigBee**

ZigBee es una tecnología creada para redes inalámbricas de bajo coste basado sobre la capa física y capa de red de la norma IEEE 802.15.4, nació porque Bluetooth y Wi-Fi no satisfacían ciertos requisitos.

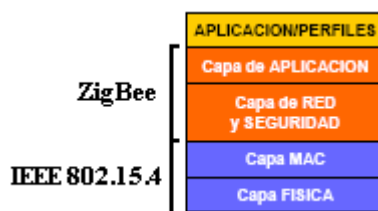


Figura. 2.6. ZigBee vs IEEE 802.15.4

2.2.4 Aplicaciones

Esta nueva tendencia en sensamiento ya consta de múltiples aplicaciones alrededor del mundo en agricultura y monitoreo medioambiental, aplicaciones militares, monitoreo de la salud, e ingeniería en general.

Con respecto a aplicaciones en la agricultura se pueden destacar: la exploración y vigilancia en ambientes inhóspitos; en monitoreo geofísico para la detección de desastres naturales sean estos erupciones volcánicas y actividades sísmicas, incendios forestales, inundaciones, supervisión de la calidad del agua, monitoreo de temperatura o la humedad de las hojas, con el fin de detectar rápidamente situaciones adversas y desencadenar los tratamientos apropiados.

En el aspecto militar las aplicaciones de WSN's están relacionadas a protección, tácticas de guerra, detección de tropas enemigas, los franco tiradores pueden ser localizados mediante sensores acústicos múltiples, detección de armas químicas o biológicas.

En la medicina las redes de sensores inalámbricos actualmente se encuentran monitoreando datos fisiológicos de los pacientes como la temperatura del cuerpo, tensión arterial, sensores diminutos en la circulación sanguínea pueden analizar continuamente la sangre y previene coagulación y trombosis.

En ingeniería general las aplicaciones son muy diversas, en las que van desde teclados musicales virtuales utilizando acelerómetros, pasando por domótica para el control de luz, temperatura, y movimiento, los micrófonos para la activación mediante voz, y sensores de presión en las sillas, regular el paso de aire, hasta la telemática y telemetría en autos de turismo y competencia.

Otra de las aplicaciones si dejar de ser la menos importante, es la integración de WSN en procesos industriales, teniendo alta eficiencia y flexibilidad al contar sensores de múltiples magnitudes físicas en un solo dispositivo. [9]

2.3 SISTEMAS DE ADQUISICIÓN DE DATOS

Un Sistema de Adquisición de Datos (SAD) es una configuración física y lógica que permite realizar captura de información de un experimento.

La configuración física o hardware viene dada por un conjunto de dispositivos electrónicos que permiten la conversión de señales analógicas en datos digitales. Por configuración lógica o software se entiende al sistema que realiza el control del hardware y transporta los datos a un computador, para procesarlos y analizarlos.

2.3.1 Características

La principal función de un sistema de adquisición de datos consiste en transmitir las señales analógicas y/o discretas por medio de sus canales de entrada tomadas a través diferentes sensores de un proceso cualquiera.

2.3.2 Arquitectura

Los Sistemas de Adquisición de Datos en su arquitectura se encuentran las siguientes etapas: Sensamiento, Acondicionamiento, Multiplexación, Amplificación, Muestreo y Retención, Conversor Análogo Digital y Control, todos ellos mostrados en Figura. 2.7.

- **Sensamiento y Acondicionamiento**

Sensores y transductores forman parte de esta etapa, su función es la de medir la variable física y convertirla en señal eléctrica. Sin embargo esta señal eléctrica convertida es de muy bajo nivel de voltaje o corriente, por lo que es necesario realizar un acondicionamiento previo para el resto de los módulos de un SAD.

- **Multiplexación**

Un multiplexor es un circuito que se encarga de seleccionar la señal de entrada a ser tratada, de existir una única señal este circuito no es necesario.

- **Amplificación**

Su circuito principal es un amplificador de instrumentación, cuya función es la amplificar la señal de entrada para conseguir un mismo margen de esta señal a la entrada del ADC⁴ y así conseguir una máxima resolución; si se dispone de un SAD de varios canales de entrada, cada canal tendrá su rango de resolución, por lo que se hace necesario que el amplificador sea programable.

- **Muestreo y Retención**

Toma la muestra de un determinado canal y la mantiene. Es un circuito necesario cuando la señal de entrada varía considerablemente durante la conversión.

- **Conversor Análogo Digital**

Proporciona un código digital de salida de la muestra analógica recibida a cada instante. Es la etapa más importante de un Sistema de Adquisición de Datos, la característica de este circuito es que puede condicionar al resto del sistema.

⁴ ADC: Siglas en ingles de Conversor Análogo Digital (Analog Digital Converter)

- **Control**

Como se menciona anteriormente cuando existen varias señales de entrada, es necesaria la presencia de un circuito de control que permita la selección, la amplificación, el muestreo y rango de conversión.

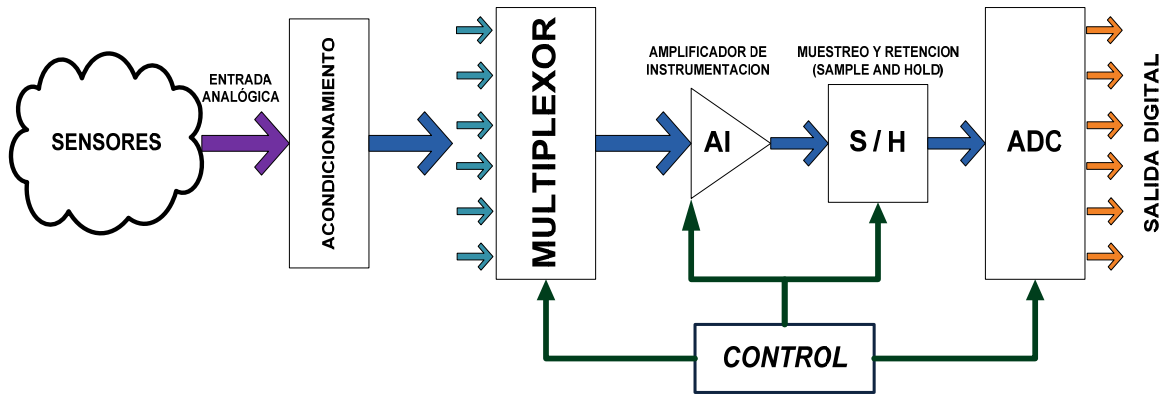


Figura. 2.7. Arquitectura Sistema de Adquisición de Datos

2.3.3 Tipos de Sistemas de Adquisición de Datos

A los SAD se los puede clasificar de acuerdo al número de canales de entrada que poseen, en **Monocanales** y **Multicanales**.

- **Sistemas de Adquisición de Datos Monocanales**

Son sistemas diseñados para la adquisición de una sola señal de entrada.

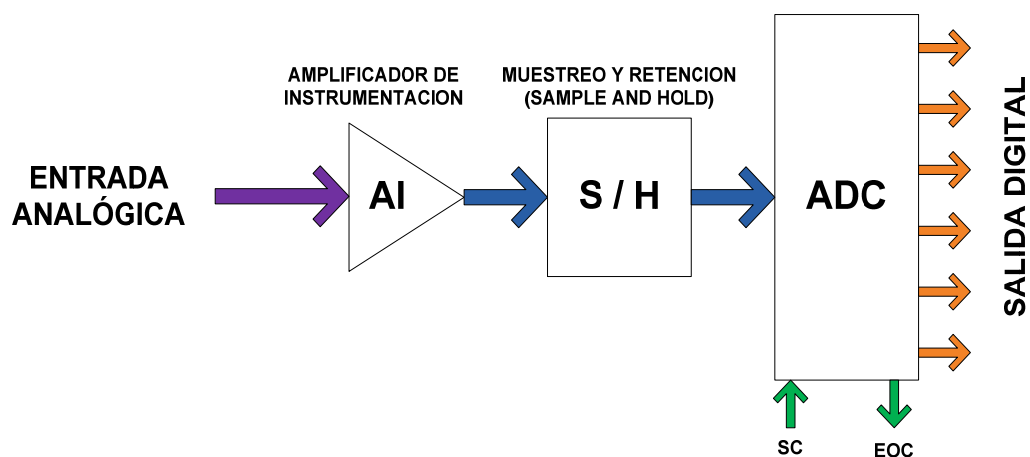
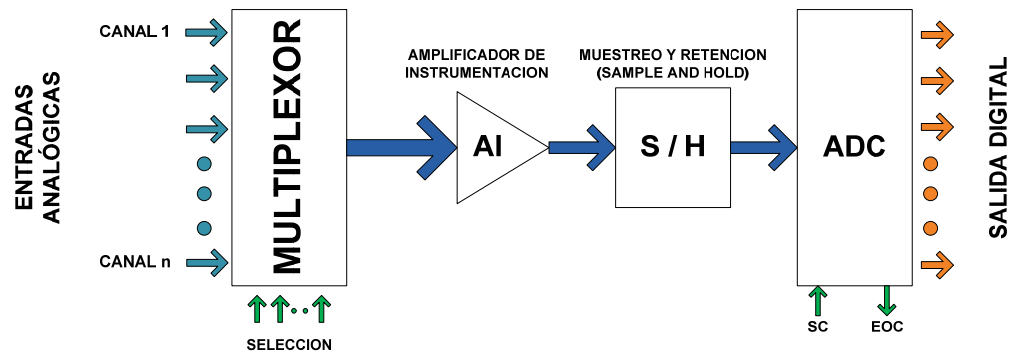


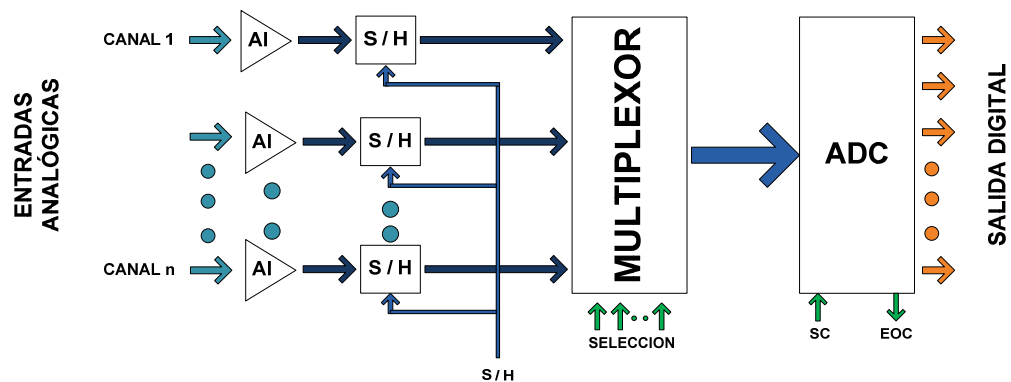
Figura. 2.8. Sistema de Adquisición de Datos Monocanal

- **Sistemas de Adquisición de Datos Multicanales**

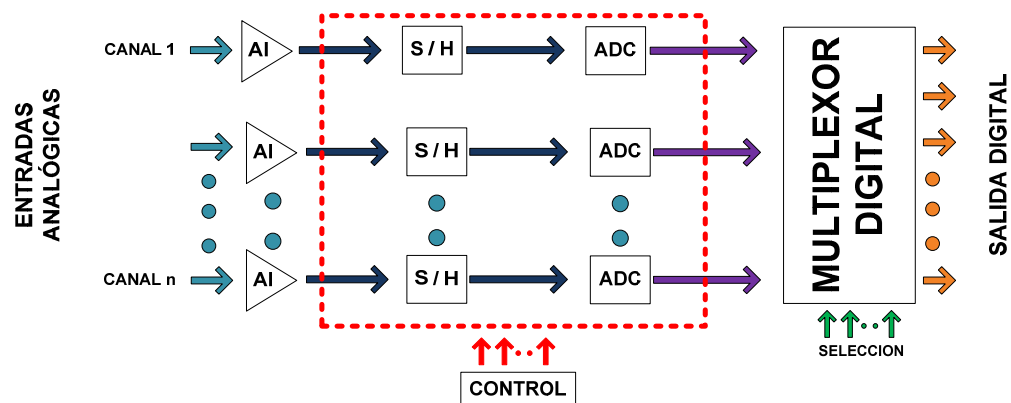
Diseñados para la conversión de varias señales analógicas de entrada, las variantes en su configuración dependen de la característica de la señal de entrada, la información que se desea obtener, velocidad de conversión, coste, etc. En la Figura. 2.9, se muestran los distintos tipos de SAD multicanales.



a) SAD Multicanal con muestreo secuencia de canales



b) SAD Multicanal con muestreo simultaneo de canales



c) SAD Multicanal en paralelo

Figura. 2.9. Sistemas de Adquisición de Datos Multicanales

2.4 INSTRUMENTACIÓN VIRTUAL

El desarrollo de la informática y la ingeniería electrónica en las dos últimas décadas revolucionó los campos de la instrumentación, mediciones y automatización.

Un importante desarrollo resultante de la universalidad de la computadora personal es el concepto de instrumentación virtual, el cual ofrece variados beneficios a ingenieros y científicos que requieran mayor productividad, precisión y rendimiento.

2.4.1 Definición

“El concepto de instrumentación virtual implica adquisición de señales, el procesamiento, análisis, almacenamiento, distribución y despliegue de los datos e información relacionados con la medición de una o varias señales, HMI, visualización, monitoreo y supervisión remota del proceso, la comunicación con otros equipos, etc.

Un sistema de instrumentación virtual está enfocado a los instrumentos encargados de medir señales, registrar datos y decidir las acciones de control, evidentemente, se requiere de una etapa de actuación, que conforma la interfaz entre la computadora y el sistema a controlar, por tanto esta etapa implicará drivers de potencia o transductores de señal especiales. Además, existen otras etapas auxiliares que no intervienen en el proceso de medida, como es el caso del subsistema de alimentación.” [4]

2.4.2 La instrumentación virtual en la Ingeniería Electrónica

En la Ingeniería Electrónica la enseñanza y el aprendizaje tienen que ir de la mano con la tecnología; lamentablemente para países subdesarrollados como el Ecuador el problema más relevante no es la rapidez del cambio tecnológico en el mundo, si no el alto costo de los equipos especialmente en el área de laboratorios.

Una solución a este problema es emplear PC's, en las cuales se reemplacen equipos convencionales por instrumentos virtuales y sistemas de adquisición de datos, permitiendo realizar adquisición, procesamiento y control de señales físicas en tiempo real a costos menores, con capacidad de realización de experimentos para distintos eventos sin la necesidad de contar con instrumentos o equipos físicos para cada uno de ellos.

2.4.3 Diseño de instrumentos virtuales

Para el diseño de un instrumento virtual se requiere, de un computador, un SAD, y el software apropiado. Un instrumento virtual debe realizar como mínimo las tres funciones básicas de un instrumento convencional: adquisición, análisis y presentación de datos.

La instrumentación virtual puede también ser implementada en equipos distribuidos en campo (RS-485), equipos a distancia (conectados vía radio, Internet, etc.), o equipos industriales.

Existe una tarjeta de adquisición de datos para casi cualquier bus o canal de comunicación en PC (ISA, PCI, USB, serial RS-232, RS-422, paralelo EPP, PCMCIA, CompactPCI, PC/104, VMEbus, CAMAC, PXI, VXI GPIB, etc.), y existe un driver para casi cualquier sistema operativo.

Entre los programas especializados para la instrumentación virtual destacan LabVIEW, Agilent-VEE, Cyber Tools, Beta Instruments Manager, Matlab Simulink, etc.

CAPÍTULO 3

SENSORES MTS310, NODOS MPR2400 Y MODULO MIB520

3.1 CARACTERÍSTICAS GENERALES

En el presente capítulo se describe el hardware utilizado en el proyecto, como son los sensores MTS310, los nodos MPR2400, la estación base MIB520CB de comunicación Serial USB, el lenguaje y software de programación.

Los módulos utilizados en este proyecto funcional bajo el sistema operativo TinyOS⁵ desarrollado por la Universidad de Berkeley, el software de fabrica MoteView también viene basado en este sistema, cuyo lenguaje de programación llamado NesC⁶, es muy similar al C y es soportado por las plataformas comerciales de Windows, principalmente XP Home, Profesional, y 2000 con service pack 4.

3.1.1 TinyOS y NesC

TinyOS es un sistema operativo de código abierto para trabajar con WSN, puede ser visto como un conjunto de programas avanzados que contiene varios algoritmos que permiten generar enrutamientos así como también posee diferentes aplicaciones para las redes de sensores inalámbricos, puede ser aplicado a otros sistemas embebidos y soporta diferentes plataformas de nodos de sensores.

TinyOS está desarrollado para trabajar bajo Linux, sin embargo para trabajar en Windows es necesaria la instalación de **Cygwin** que es un simulador de plataformas Unix que se ejecuta bajo Windows.

“**NesC** es un meta-lenguaje de programación basado en C, orientado a sistemas embebidos que incorporan el manejo de red. Además soporta un modelo de programación

⁵ TinyOS: Tiny microthreading Operative System

⁶ NesC: Network Embedded Systems C

que integra el manejo de comunicaciones, las concurrencias que provocan las tareas y eventos y la capacidad de reaccionar frente a sucesos que puedan ocurrir en los ambientes donde se desempeña.” [5]

3.1.2 MoteView

MoteView es el software de fábrica provisto por Crossbow, está diseñado para ser una interfaz entre el usuario y la red de sensores inalámbricos; este software viene provisto de herramientas para facilitar el monitoreo de la Red, en la Figura. 3.1, se muestra la ventana del programa en la cual se despliegan los datos de los distintos nodos, la información visualizada en esta ventana posteriormente será utilizada para el análisis del proyecto en el capítulo 6.

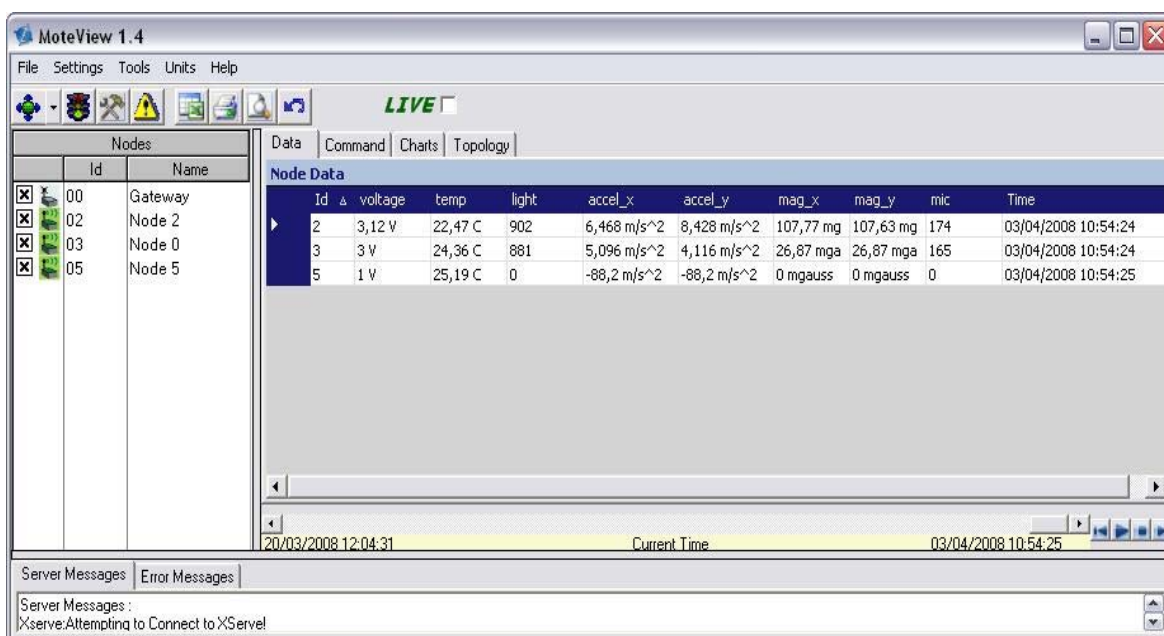


Figura. 3.1. Ventana principal de MoteView

3.2 MPR2400 Y MTS310

Como se describió en la sección 2.2.2 entre los componentes que conforma una WSN se encuentran las motas y sensores, en este proyecto las motas o nodos son los módulos MPR2400 (Mote Processor Radio) que trabajan sobre MICAz⁷ y los sensores MTS310 (Motes Sensors) que soportan esta plataforma poseen sensores de luz, temperatura, acelerómetro de 2 ejes, magnetómetro de 2 ejes, micrófono y un zumbador.

⁷ MICAz: Última generación en Plataformas de hardware para motas de Crossbow Technology.

3.2.1 Descripción

A continuación se detallará las características de las motas y sensores utilizados.

- **Nodos MPR2400**

Los nodos MPR2400 trabajan en el rango de frecuencias de 2400MHz a 2483.5MHz y a nivel industrial son conocidos únicamente como MICAz funcionan dentro de la norma IEEE 802.14.5, consta de Chipcon CC2420, transmisor/receptor de radio frecuencia ZigBee con un Microcontrolador Atmega128L, un conector de 51 pines I/O, una memoria flash de 512Kb y para su funcionamiento necesita de dos baterías AA.

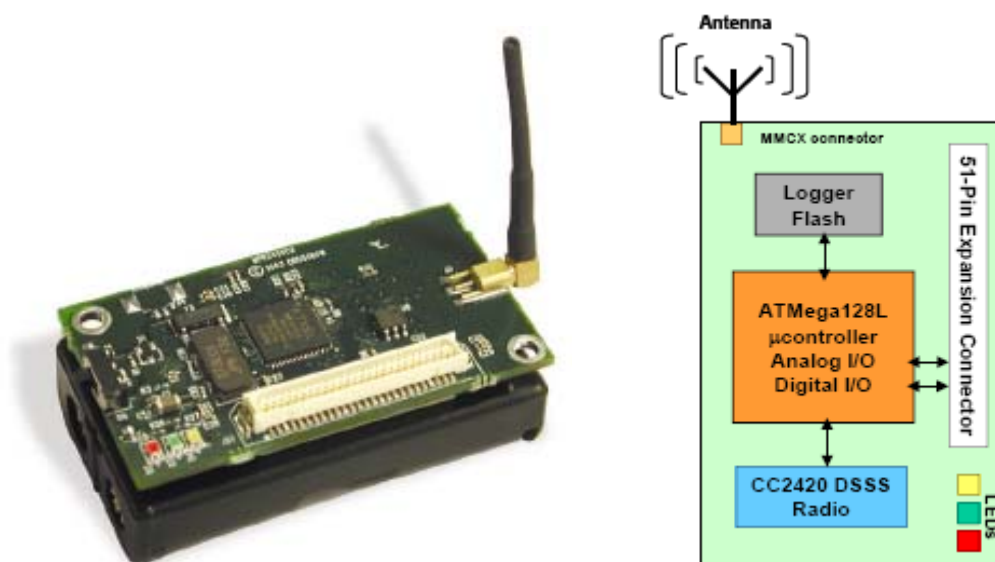


Figura. 3.2. Diagrama de bloques MPR2400/MICAz

- **Sensores MTS310**

Es un modulo flexible de sensores con varias modalidades de sensamiento entre las que se incluyen un acelerómetro y magnetómetro de dos ejes, luz, temperatura, sonido y zumbador; en la Figura. 3.3 se muestra el modulo de sensores MTS310.

MTS310 cuenta con un conector de 51 pines I/O para ser acoplado en el nodo MPR2400.

Para este proyecto los únicos elementos que no se utilizaran son el magnetómetro de dos ejes y el zumbador.

Micrófono. La circuitería de este dispositivo consta de dos operaciones, una que reconoce amplitudes y distancias a través de rangos acústicos, y otra operación que es para grabado y medición de audio. Este circuito consta de un pre-amplificador y un amplificador de segundo nivel con control de variación de resistencia digital.

Luz. El sensor de luz es una simple fotocelda de CdSe, tiene una máxima sensibilidad a la longitud de la onda de la luz de 690[nm]. La resistencia expuesta a la luz es de 2K Ω mientras que a la oscuridad la resistencia es de 520K Ω .

Temperatura. Para el sensamiento de temperatura MTS310 cuenta con un termistor configurado de tal manera que es comparable a un divisor de tensión con una lectura nominal de media escala hasta 25°C.

Acelerómetro. El acelerómetro de dos ejes es una superficie MEMS⁸ tiene una gran capacidad de detección y una resolución de 10 bits. Las características de este sensor se describen en la Tabla. 3.1

Acelerómetro de dos ejes	
Rango	$\pm 2 \text{ g}$ ($1 \text{ g} = 9.81 \text{ m/s}^2$)
Ancho de Banda	DC-50 Hz (controlado)
Resolución	2 mG (0.002 G) RMS
Sensibilidad	167 mV/G $\pm 17 \%$
Offset	2.5 V $\pm 0.4 \text{ V}$

Tabla. 3.1. Características acelerómetro de dos ejes.



Figura. 3.3. Sensores MTS310 con acelerómetro y magnetómetro

⁸ MEMS: Micro Electro Mechanic System

3.3 MIB520

La estación base utilizada para este proyecto es la MIB520CB (Mote Interface Board), este módulo actúa como interfaz entre el computador y los nodos.

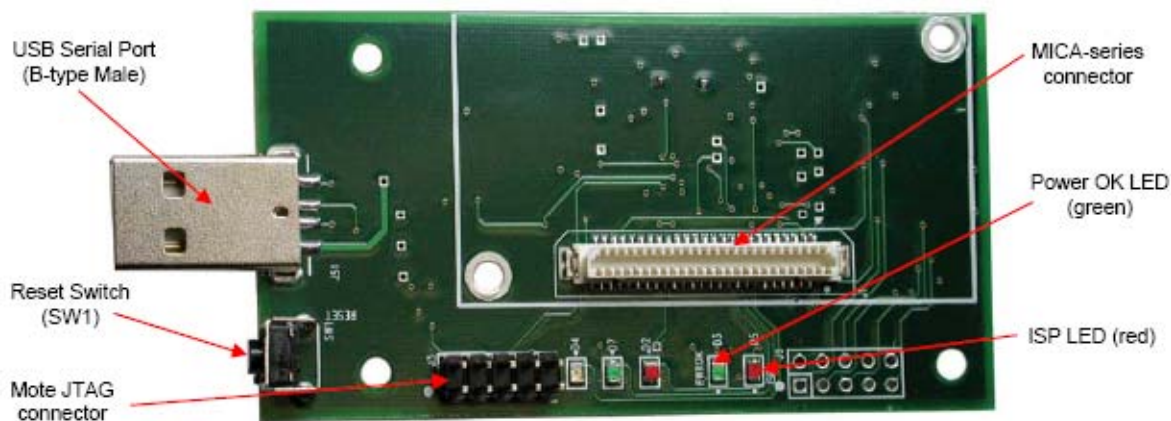


Figura. 3.4. Vista superior de MIB520CB⁹

3.3.1 Descripción

La placa MIB520CB es un dispositivo de comunicación serial USB, con capacidad de programación para distintas plataformas de hardware Crossbow.

Para su funcionamiento es necesaria la instalación del driver FTDI FT2232C provisto en los discos de instalación del software de fábrica, la función de este driver es usar el puerto USB físico como dos puertos COM virtuales COM_n y COM_{n+1} , el primero de ellos se utiliza para la programación de los nodos y el segundo para la recepción de la información de la red de sensores inalámbricos.

Tiene un procesador ISP¹⁰ Atmega16L, posee un conector de 51 pines, LEDs indicadores de nodos (amarillo, verde, rojo), LEDs indicadores ISP (Funcionamiento correcto: verde, Programación en progreso: rojo), un botón de RESET que resetea el ISP y el monitoreo del software que corre bajo el PC, un conector J3 (Mote JTAG) el cual permite la conexión al JTAG del microcontrolador Atmel para la generación de una depuración dentro del circuito mismo, la tasa de transferencia es de 57.6 Kbps y no requiere de fuente de alimentación externa. [6]

⁹ MPR-MIB_Series_Users_Manual_7430-0021-07_B.pdf

¹⁰ ISP: in-system processor

3.4 PROGRAMACIÓN NODOS Y ESTACIÓN BASE

Durante la instalación del software de fábrica provisto por Crossbow se instalan varios paquetes entre los cuales se encuentra **MoteConfig**, **TinyOS** y **Cygwin**, como en el proyecto se utilizan los nodos MPR2400 la plataforma para programación será MICAz; existen dos formas de programación de nodos y estación base, detalladas posteriormente.

Para cargar un programa en un nodo o mota, se debe acoplar el conector de 51 pines del MPR2400 a la base MIB520, los MPR no deben estar con las baterías colocadas tal como muestra la Figura. 3.5



Figura. 3.5. MIB520CB con MPR2400 acoplado

3.4.1 Programación mediante Cygwin

El código del programa para la estación base y los nodos se descargan a la PC con la instalación del software provisto por Crossbow, estos se encuentran en la dirección: *C:\Crossbow\cygwin\opt\MoteWorks\apps\xmesh*. Para los nodos dentro de la carpeta *XMTS310CB* y para la estación base dentro de *XMeshBase*.

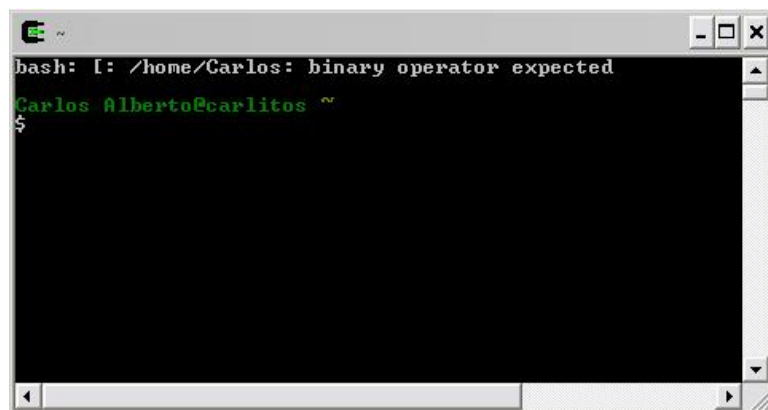



Figura. 3.6. Ventana terminal de Cygwin

En la Figura. 3.6, se muestra la ventana terminal de Cygwin, cuyo icono de acceso directo  se encuentra en el escritorio; una vez abierta la aplicación, para programar los nodos y la estación base se utilizan los comandos descritos a continuación.

Para la estación base se utiliza los siguientes comandos:

```
cd c:/Crossbow/cygwin/opt/moteworks/apps/xmesh/xmeshbase.  
make micaz install, 0 mib520,com <n>
```

Donde <n> representa el puerto COM virtual para programación.

Para la programación de los nodos se utilizan los comandos siguientes:

```
cd c:/Crossbow/cygwin/opt/moteworks/apps/xmesh/xmts310cb.  
make micaz install, <x> mib520,com <n+1>
```

Donde <x> es un número entero que representa al identificador de nodo.

3.4.2 Programación mediante MoteConfig 2.0

Como se menciono anteriormente dentro de los paquetes que se instalan con el software de fábrica se encuentra **MoteConfig 2.0**, esta aplicación se encuentra ubicada en: INICIO→TODOS LOS PROGRAMAS→CROSSBOW, la Figura. 3.7se muestra la ventana principal de MoteConfig 2.0.

En el menú SETTINGS→INTERFACE BOARD SETTINGS se debe configurar la estación base y el puerto para la programación a utilizarse.

Al seleccionar la opción anterior, aparecerá la ventana mostrada en Figura. 3.8; los parámetros a seleccionarse para este proyecto son:

- ✓ **Estación Base:** MIB520's
- ✓ **Puerto COM :** n (Primer Puerto Serial)
- ✓ **Bits por segundo:** 57600

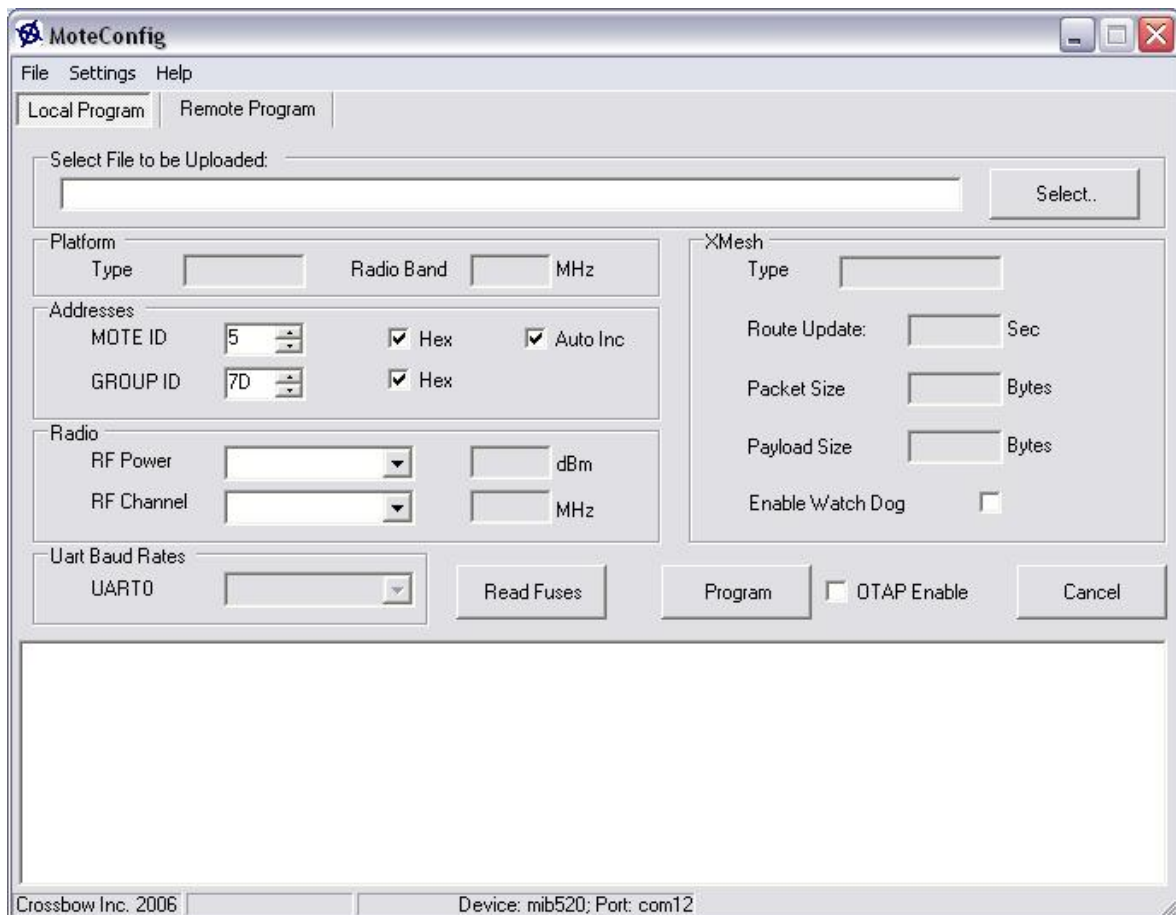


Figura. 3.7. Ventana de MoteConfig 2.0

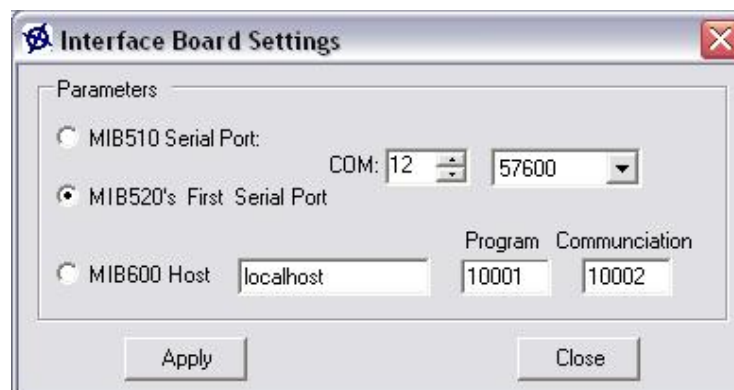


Figura. 3.8. Ventana Interface Board Settings, con COM 12 puerto serial virtual

Luego, en **Select File to be Uploaded** de la Figura. 3.7 se introduce la dirección de la aplicación de los nodos o de la estación base. Para los nodos la dirección se muestra en la Figura. 3.9 y para la estación base en Figura. 3.10; posteriormente hay que presionar el botón **Program** para subir la aplicación a la mota.

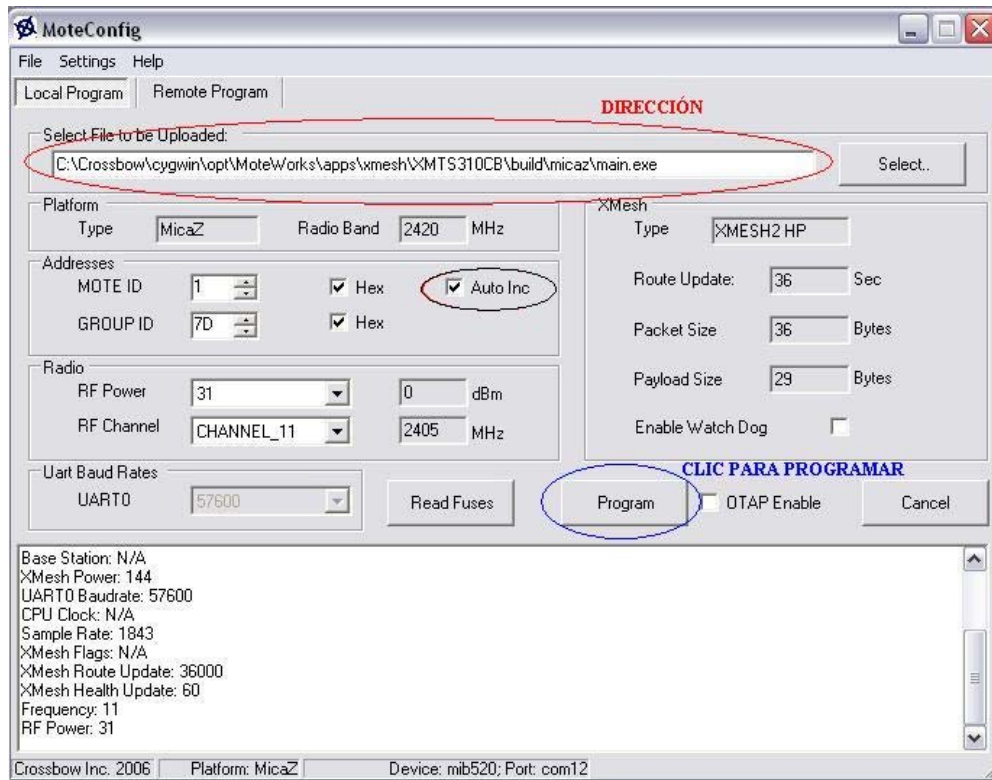


Figura. 3.9. Programación de un nodo en MoteConfig

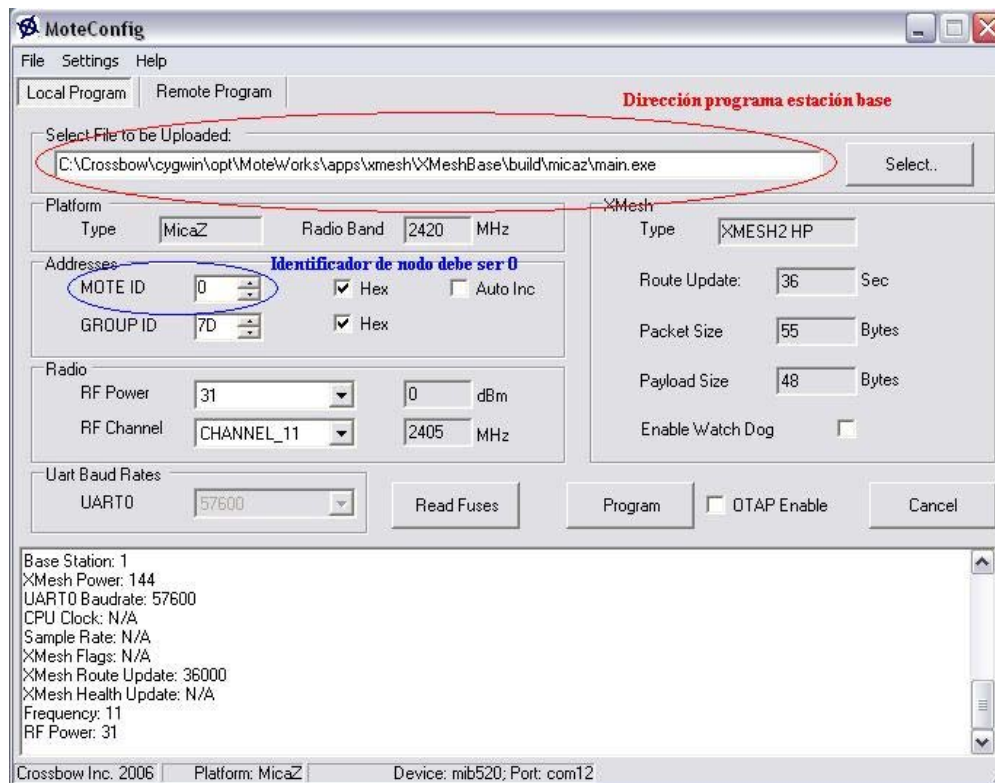


Figura. 3.10. Programación de la estación en MoteConfig

3.5 COMUNICACIÓN SERIAL USB

USB¹¹ es una interfaz para transmisión de datos y distribución de energía que ha sido introducida en el mercado de PCs y periféricos para mejorar las lentas interfaces paralelo y serie (RS-232).

Esta interfaz también permite la utilización física de un puerto USB como puerto virtual RS-232 (COM) o paralelo (LPT).

Entre las características de esta comunicación se encuentran:

- ✓ Permite la conexión de hasta 127 periféricos con un solo puerto, de baja y media velocidad. 1.5 Mbps y 12Mbps para la transmisión de datos.
- ✓ Transfiere las señales y energía a través de 4 hilos, para trenzado para datos (28AWG) y par trenzado para alimentación (20 a 28 AWG).
- ✓ Tensión nominal de 5V.
- ✓ Los bits son enviados empezando por el byte menos significativo, hasta llegar al byte más significativo de la trama.

3.6 TRAMA DE DATOS

Como se menciono con anterioridad si bien la estación base (MIB520) físicamente consta de un conector USB, los datos de la red de sensores inalámbricos enviados a la PC se los realizan de forma serial; durante el estudio de la trama de datos se determino que la longitud de la misma es de 39 Bytes, la misma que se detalla en la Tabla. 3.2

<i>Nombre del Paquete</i>	<i>Tamaño (Bytes)</i>
Cabecera TOS	7
Cabecera XMESH	7
Cabecera XSENSOR	6
Datos Sensor	16
Fin de trama	3

Tabla. 3.2. Descripción de paquetes dentro de trama de datos recibidos

¹¹ USB: Siglas en ingles de Bus Serial Universal (Universal Serial Bus)

Como se detalla en la Tabla. 3.2 el paquete **Datos Sensor** tiene un tamaño de 16 Bytes, en el cuales se encuentran distribuidas las distintas magnitudes físicas que poseen los sensores MTS310, las mismas que se detallan en la Tabla. 3.3

Variable	Tamaño (Bytes)
Estado Bateria	2
Temperatura	2
Luz	2
Micrófono	2
Acelerómetro	4 (2 x, 2 y)
Magnetómetro	4 (2 x, 2 y)

Tabla. 3.3. Descripción paquete Datos Sensor

En la Figura. 3.11 muestra la posición de los paquetes en la trama de datos recibidos por la estación base.

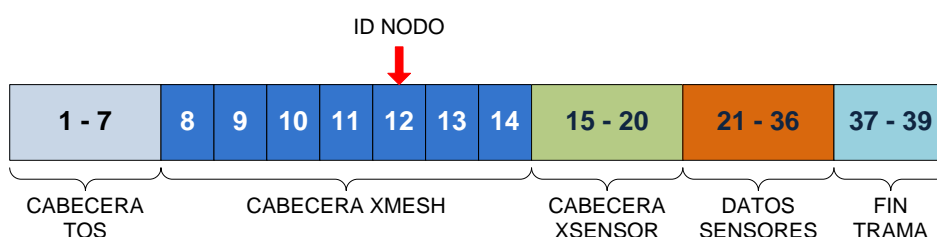


Figura. 3.11. Trama de Datos

La Figura. 3.12 detalla la distribución de los Bytes de magnitudes físicas de los sensores MTS310 dentro del paquete *Datos Sensores*.

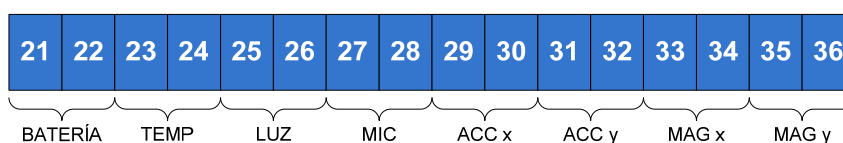


Figura. 3.12. Distribución de Bytes de magnitudes físicas en el paquete Datos Sensores

Cada Byte en la trama de datos está representado por un número hexadecimal de 2 dígitos; la conversión a unidades crudas de las magnitudes se muestra en la Figura. 3.13

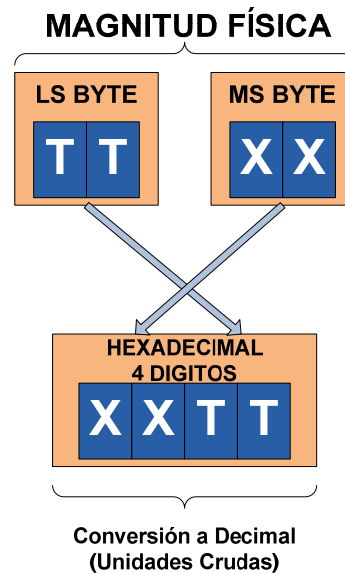


Figura. 3.13. Validación para obtención de datos sensados

3.7 CONVERSIÓN A UNIDADES DE INGENIERÍA

Una vez determinada la información de la trama de datos, y conocida la forma en la que esta información se convierte a unidades crudas, el siguiente paso es transformar estos datos a unidades de ingeniería, para lo cual se utilizarán las ecuaciones o algoritmos que a continuación se especifican. [7]

Como se detalla en la Figura. 3.12, la primera de las variables físicas es el estado de la batería de los sensores, para convertir esos datos a unidades de ingeniería se utiliza la Ecuación 3.1.

$$BV = RV \times \text{ADC_FS}/\text{data} \quad \text{Ecuación 3.1}$$

donde,

BV = Voltaje de la batería

ADC_FS = 1023

RV = 1.223[V] (Voltaje de Referencia para micaz)

data = Bytes 22-21 convertidos a decimal.

BV [V] = BV

BV [mV] = BV × 1000

La Ecuación 3.2, es utilizada para convertir los datos de los Bytes 24-23 (unidades crudas de temperatura) a grados Kelvin (unidades de ingeniería de temperatura).

$$T(K) = \frac{1}{a + b \times \ln(R_{thr}) + c \times [\ln(R_{thr})]^3} \quad \text{Ecuación 3.2}$$

donde,

$$a = 0.00130705$$

$$b = 0.000214381$$

$$c = 0.000000093$$

$$R_{thr} = R1 \times (ADC_FS - ADC) / ADC \quad \text{Ecuación 3.3}$$

$$R1 = 10 \text{ k}\Omega$$

$$ADC_FS = 1023$$

$$ADC = \text{Bytes 24-23 convertidos a decimal.}$$

$$T(^{\circ}\text{C}) = T(K) - 273.15$$

Los Bytes 26-25 y 28-27 de la trama de datos representan las magnitudes físicas de luz y sonido respectivamente, estas dos magnitudes no se las pudo convertir a unidades de ingeniería (Lux y decibelios) debido a que la información no es proporcionada por el fabricante, por tal motivo en el software se visualizará únicamente las unidades crudas.

Sin embargo para el caso de los bytes 26-25 correspondiente a la información recibida de la fotoresistencia detectora de la variación de Luz, en la Figura. 3.14 se muestra el circuito correspondiente a un divisor de tensión cuya variación de voltaje (ADC) se debe a la variación que sufre la resistencia (R Photoresistor).

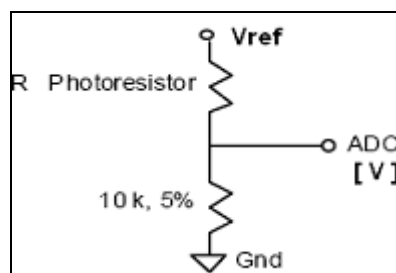


Figura. 3.14. Circuito Fotoresistencia

La Ecuación 3.4, indica la fórmula utilizada para obtener la variación de voltaje en el circuito divisor de tensión.

$$\text{ADC[V]} = \text{BV} \times \text{DATA}/\text{ADC_FS} \quad \text{Ecuación 3.4}$$

donde,

BV = Voltaje de la Batería (Ecuación 3.1)

DATA = Datos 26-25 convertidos a decimal

ADC_FS = 1023

Finalmente para el acelerómetro tanto la componente en x y la componente en y, la Ecuación 3.5, se utiliza para transformar los datos de los Bytes 30-29, 32-31.

$$\text{ACC} = 1 - (\text{CP1} - \text{ADC})/\text{ADC_FS} \quad \text{Ecuación 3.5}$$

donde,

CP1 = 500

CN1 = 400

ADC_FS = (CP1-CN1)/2

ADC = Bytes convertidos a decimal.

ACC(m/s²) = ACC × 9.81

CAPÍTULO 4

DISEÑO DEL SOFTWARE

4.1 LABVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control.

LabVIEW constituye un revolucionario sistema de programación gráfica las aplicaciones creadas son llamadas VI's (Instrumentos Virtuales) y entre sus aplicaciones principales se tiene:

- ✓ Mediciones de Alta Velocidad de un Amplio Rango de Sensores.
- ✓ Monitoreo de Vibración y Mantenimiento Predictivo.
- ✓ Registro de Datos y Control Estadístico de Procesos.
- ✓ Compartir Datos entre PLCs y otros Dispositivos de Automatización.
- ✓ Desarrollar Interfaces Gráficas de Usuario.
- ✓ Alarmas, Registro de Datos, Aplicar Escalas y Reportes.
- ✓ Algoritmos Avanzados y Control de Movimiento Personalizado.

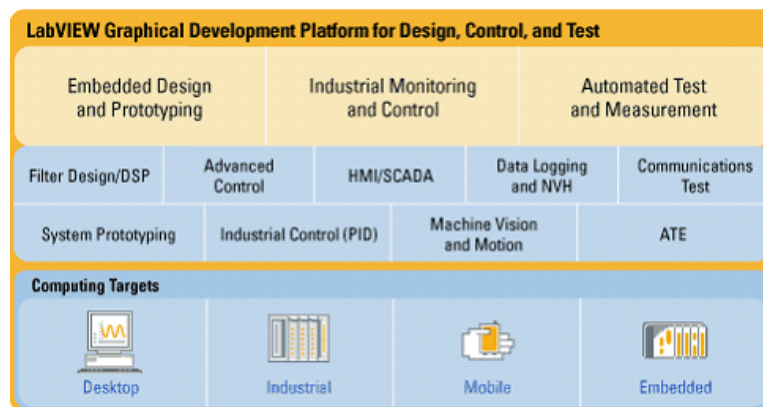


Figura. 4.1. ¿Qué es LabVIEW?

4.1.1 VISA¹²

Arquitectura de Software para Instrumentos Virtuales (VISA) es una librería de interfaz simple para controlar dispositivos e instrumentos en las plataformas de LabVIEW como VXI, GPIB, RS-232, USB. VISA es el estándar utilizado por la Alianza de Sistemas de VXI plug & play, en las cuales se encuentran más de 35 compañías en la industria de instrumentación. El estándar VISA unifica la industria para hacer software que pueda ser interpretado y reutilizado por más tiempo, sin importar el tipo de operación de su instrumento.

NI-VISA es un software controlador que está disponible en la página web de National Instruments: <http://ni.com/visa>. [8]



Figura. 4.2. ¿Qué es VISA?

4.1.2 Configuración Serial USB

Como se menciona en el CAPÍTULO 3, la estación base utilizada en este proyecto es la MIB520CB, que físicamente consta de un conector macho USB, pero la transmisión de los datos se realiza de manera serial.

En LabVIEW para poder tomar los datos de la WSN Crossbow se utiliza la librería NI-VISA la cual permite trabajar con el puerto COM_{n+1} virtual.

El diagrama de bloques utilizado para la comunicación se muestra en la Figura. 4.3.

¹² VISA: Virtual Instrument Software Architecture

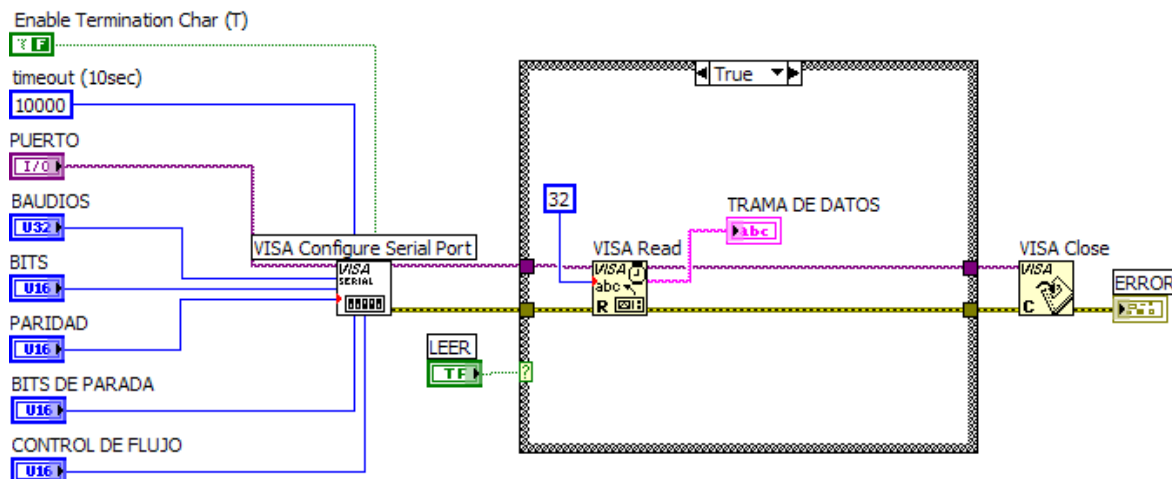


Figura. 4.3. Configuración de la comunicación Serial-USB

4.2 DISEÑO DE LA INTERFAZ HMI

Como ya se explicó en el CAPÍTULO 3, en el presente proyecto se van a utilizar sensores que detecten: temperatura, luz, sonido y vibración, de acuerdo a ello se diseñará la interfaz en LabVIEW para la visualización de la información proveniente de la red de sensores inalámbricos.

4.2.1 Características Generales

Una vez que se logró el reconocimiento de la información en la trama de datos, algoritmos de conversión y configuración serial con LabVIEW, se procedió a determinar las principales características con las que contaría la interfaz a ser diseñada, las mismas que se detallan a continuación:

- ✓ Ventana con pestañas desplegadas.
- ✓ Información de estado de los nodos.
- ✓ Visualización amigable de temperatura, variación de voltaje, sonido, vibración.
- ✓ Presentación de graficas históricas de los procesos de medición para las distintas variables.
- ✓ Exportación de datos adquiridos a una hoja de cálculo.
- ✓ Presentación de resultados de un proceso de adquisición.
- ✓ Control simulado para temperatura y luz.
- ✓ Acceso restringido a ciertas ventanas y código del programa, mediante contraseña.

4.2.2 Parámetros

De las características generales detalladas en la sección anterior, dan inicio a los siguientes parámetros:

- **Pantalla Principal**

Al iniciar la aplicación, esta es la primera pantalla que se mostrará, en esa pestaña se configura el puerto de comunicación a utilizarse con sus parámetros respectivos y el valor correspondiente (**Puerto COM_{n+1} , Baudios 57600, Bits 8, Paridad None, Bits de Parada 1, Control de Flujo None**), se visualiza el estado de la batería y un gráfico histórico del voltaje de los nodos, un botón para dar inicio al sensamiento (**START**), un botón para detener la ejecución de la aplicación (**STOP**), un Led para indicar que existen datos en el buffer (**LISTO**), un botón para borrar la pantalla del gráfico (**Limpiar Pantalla**) y un reloj digital, la Figura. 4.4 muestra esta ventana en la cual también constan los valores por defecto para la configuración del puerto COM_{n+1} virtual a utilizarse.

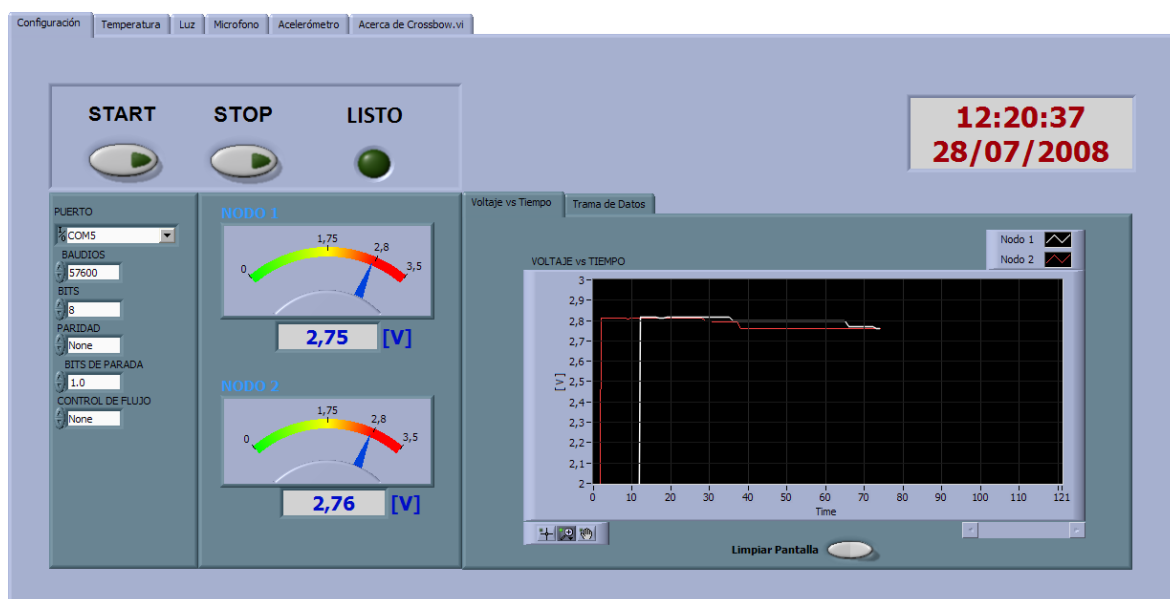


Figura. 4.4. Pantalla principal – Configuración

- **Pantalla Temperatura**

La pantalla de temperatura está dividida en dos secciones, una de ella se subdivide en tres secciones más; la primera sección consta de un indicador de temperatura en grados centígrados (**Temperatura °C**), un selector para el número de muestras a ser adquiridas (**# MUESTRAS**), un botón de inicio de muestreo (**START**).

La segunda sección de esta pantalla, consta de una pestaña con configuración de alarmas donde se puede setear parámetros (**MUY ALTA, ALTA, BAJA, MUY BAJA**), indicadores luminosos para cada uno de los niveles, y una simulación de la acción a tomarse cuando la temperatura medida se encuentra fuera del rango. La Figura. 4.5 y la Figura. 4.6 muestran las acciones a tomarse cuando existe una alarma alta o baja respectivamente.



Figura. 4.5. Pantalla Temperatura – Alarma Alta



Figura. 4.6. Pantalla Temperatura – Alarma Baja

La siguiente pestaña perteneciente a la segunda sección de la ventana de temperatura muestra un grafico histórico del proceso de muestreo, dispone de controles para modificar las escalas del gráfico en ambos ejes que se ejecutan cuando se acciona el botón **Activar**, y un botón para limpiar la pantalla del grafico (**Limpiar Pantalla**), esta pestaña se muestra en la Figura. 4.7. Finalmente en la pestaña de resultados (Figura. 4.8 se visualiza los valores máximos y mínimos del muestreo, y se puede modificar la dirección para guardar el archivo con los datos adquiridos.

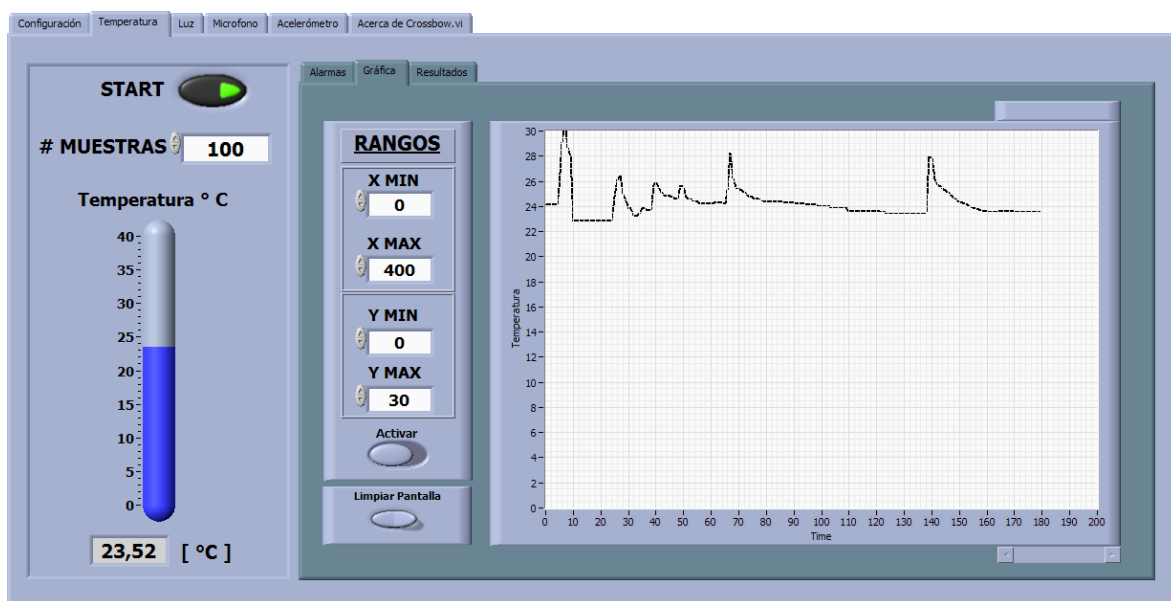


Figura. 4.7. Pantalla Temperatura – Gráfica

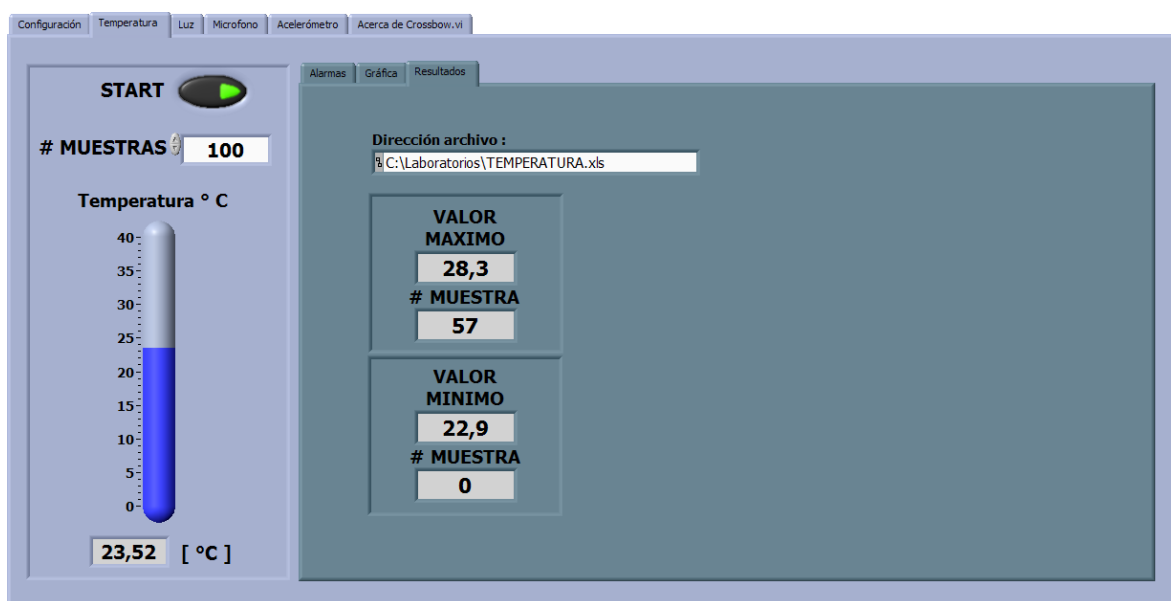


Figura. 4.8. Pantalla Temperatura – Resultados

- **Pantalla Luz**

De la misma manera que la pantalla de temperatura, la ventana correspondiente al sensamiento de la cantidad de luz recibida por la fotoresistencia, consta de dos secciones, una en la cual se visualiza la variación del voltaje debido al cambio del valor de resistencia en el divisor de tensión (**VOLTAJE CONVERSOR**), el voltaje de la batería del nodo (**VOLTAJE DE REFERENCIA**), un indicador vertical del valor de unidades crudas del Conversor (**Luz y U.C**), un controlador numérico para la cantidad de muestras en el proceso de adquisición (**# MUESTRAS**), y un botón para iniciar la toma del número de muestras seleccionado.

La segunda sección de esta ventana es muy similar a la temperatura, es decir se divide en tres sub secciones, la primera sub sección consta con cuatro controladores numéricos para distintos niveles de alarmas (**MUY ALTA, ALTA, BAJA, MUY BAJA**), indicadores visuales y auditivos para cuando la medida este fuera de cualesquiera de estos rangos, también con un simulador visual de la acción que se puede ejecutar con el valor medido.

La Figura. 4.9 muestra la pantalla de Luz en la cual, se visualiza la primera sección descrita en el primer párrafo y la segunda sección con la alarma ALTA activada y la simulación de la acción de control a ejecutarse, para este caso una bombilla apagada.

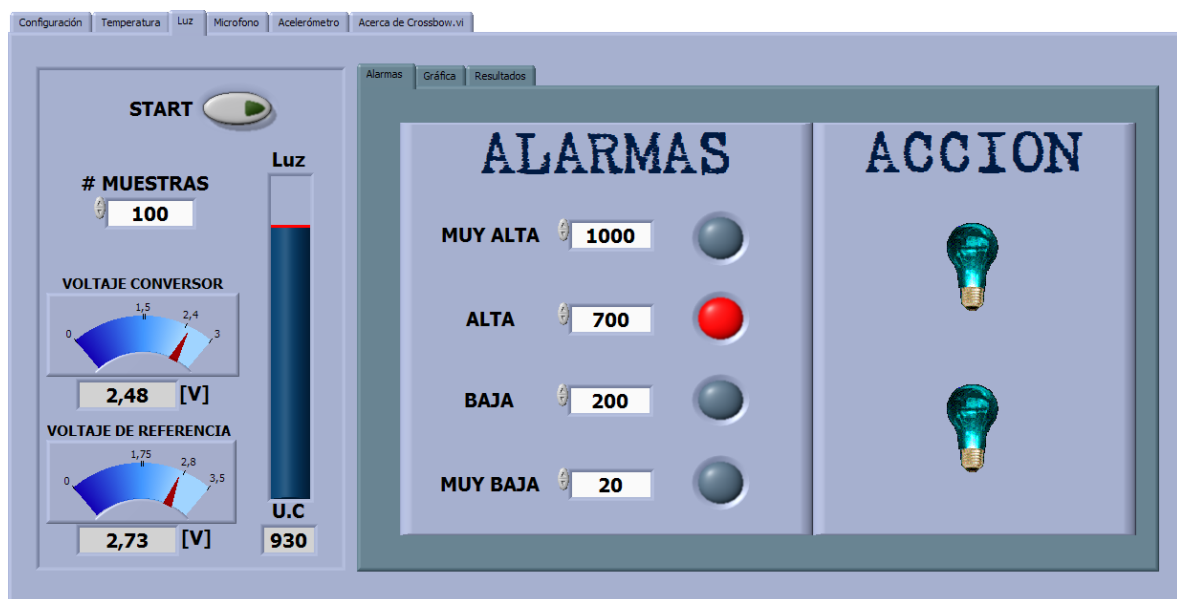


Figura. 4.9. Pantalla Luz – Alarma Alta

En la Figura 4.10 se visualiza la otra condición de simulación de la acción de control para cuando el nivel de Luz sentido este en los niveles bajos, para este caso una bombilla encendida.

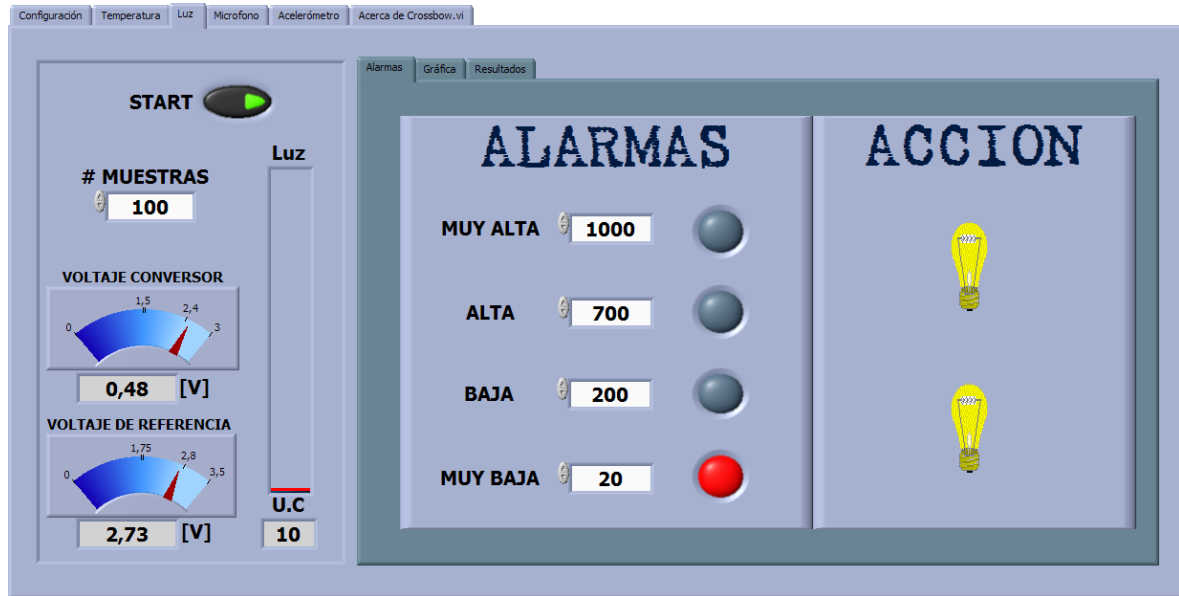


Figura. 4.10. Pantalla Luz – Alarma MUY BAJA

La Figura. 4.11 muestra la pantalla correspondiente a la gráfica de VOLTAJE CONVERTOR vs TIEMPO, con sus respectivos controles (**RANGOS**) y botones para modificación de escalas y limpieza del área de gráfico (**Limpiar Pantalla**).



Figura. 4.11. Pantalla Luz – Gráfica

La última opción disponible en la ventana de Luz, es la de **Resultados** mostrada en la Figura 4.12, que tiene la capacidad para editar la dirección del archivo de datos adquiridos, visualización de valores máximos y mínimos medidos, y el circuito de referencia que se utilizaran para los cálculos respectivos en las prácticas de laboratorio.

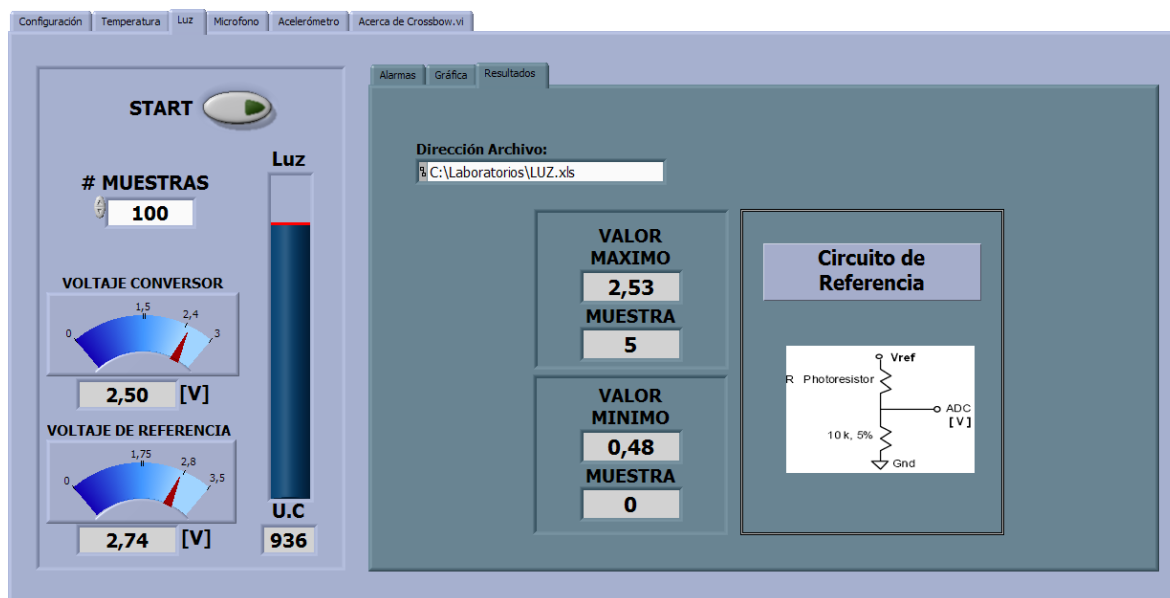


Figura. 4.12. Pantalla Luz - Resultados

- **Pantalla Micrófono**

De la misma manera que para Temperatura y Luz, la pantalla de microfono esta dividida en dos secciones, una en la cual consta con botones similares para inicio del proceso de muestreo (**START**), un indicador del nivel de sonido del nodo que muestra el valor en decimal del dato proveniente del canal de conversión (**Nivel de Sonido Nodo 2**), y un controlador numérico para seleccionar el número de muestras (**# MUESTRAS**).

En la segunda sección de esta ventana se encuentra el gráfico del valor muestreado vs tiempo, con sus respectivos controles para modificar las escalas y limpiar la pantalla (**Activar, Limpiar Pantalla**) y también una ventana de resultados en la cual está la dirección en la cual se guarda el archivo con los datos de la adquisición, y visualización de los valores máximos y mínimos. La Figura 4.13 y Figura 4.14 muestran las dos opciones disponibles en la pantalla de micrófono.

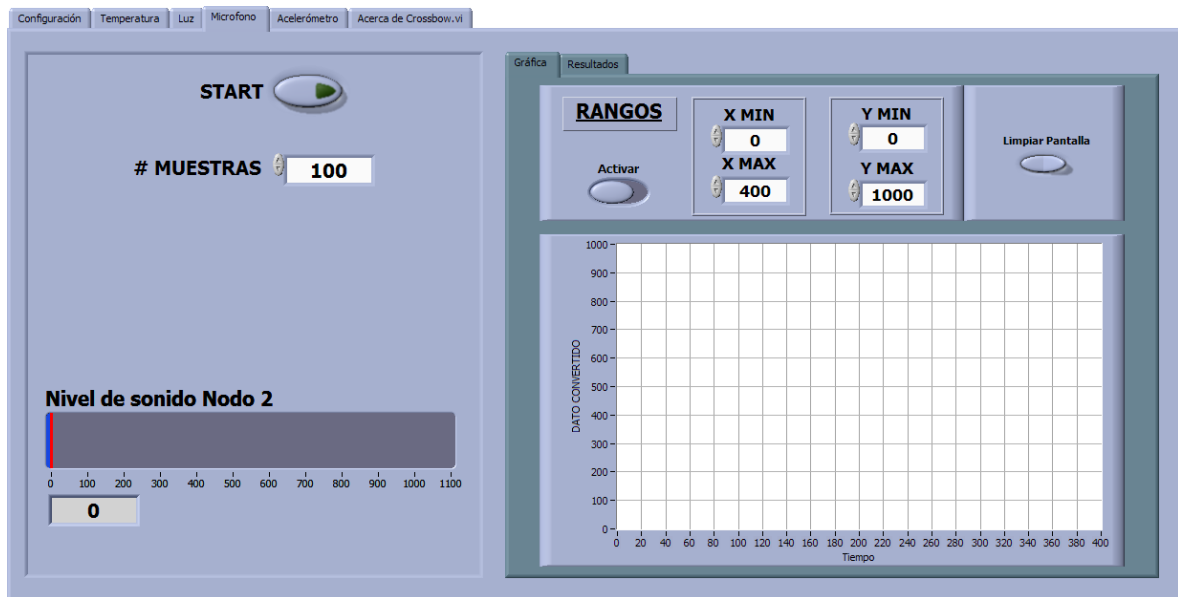


Figura. 4.13. Pantalla Micrófono – Gráfica

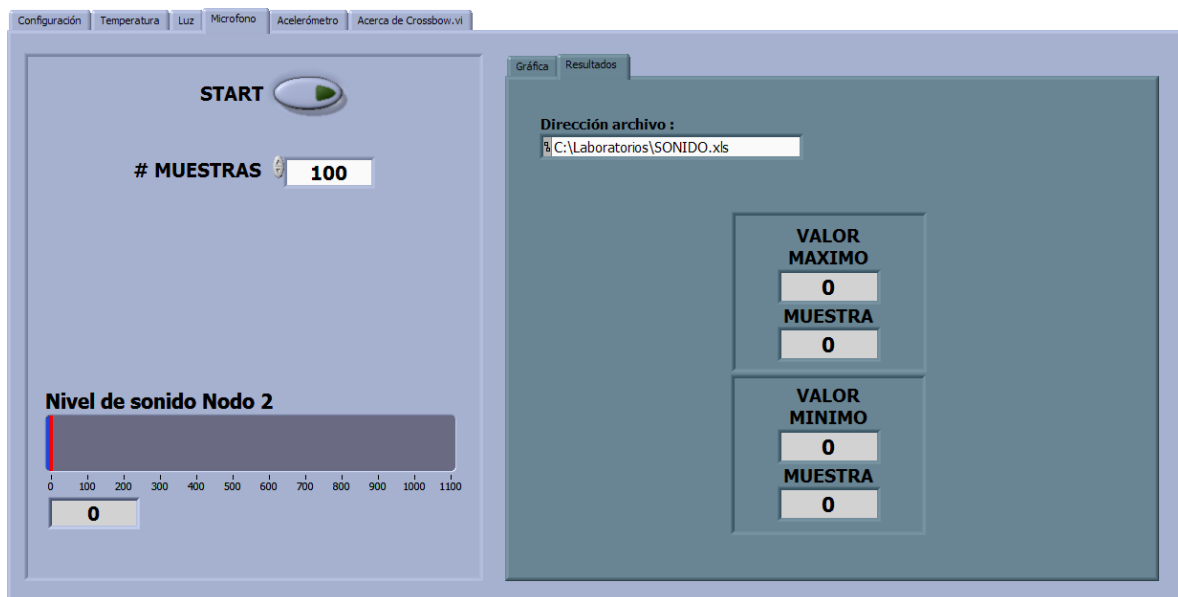


Figura. 4.14. Pantalla Micrófono – Resultados

- **Pantalla Acelerómetro**

Esta es la última pantalla en la cual se van a realizar adquisición de datos, el acelerómetro del MTS310CB va ser utilizado como detector de vibraciones en ambos ejes, en esta ventana se encuentran botones para iniciar la adquisición (**START**), para modificación de rangos y borrar la pantalla gráfica (**Activar** y **Limpiar Pantalla**).

Un área de gráfico para visualizar la vibración de los dos ejes vs tiempo, y una pestaña con opciones de un editor de texto para modificar o seleccionar la dirección del archivo y valores máximos y mínimos muestreados.

Las Figuras 4.15 y Figura 4.16 muestran las ventanas de Acelerómetro, el área izquierda de las dos Figuras es común y allí se visualiza la variación de la vibración en $[m/s^2]$.

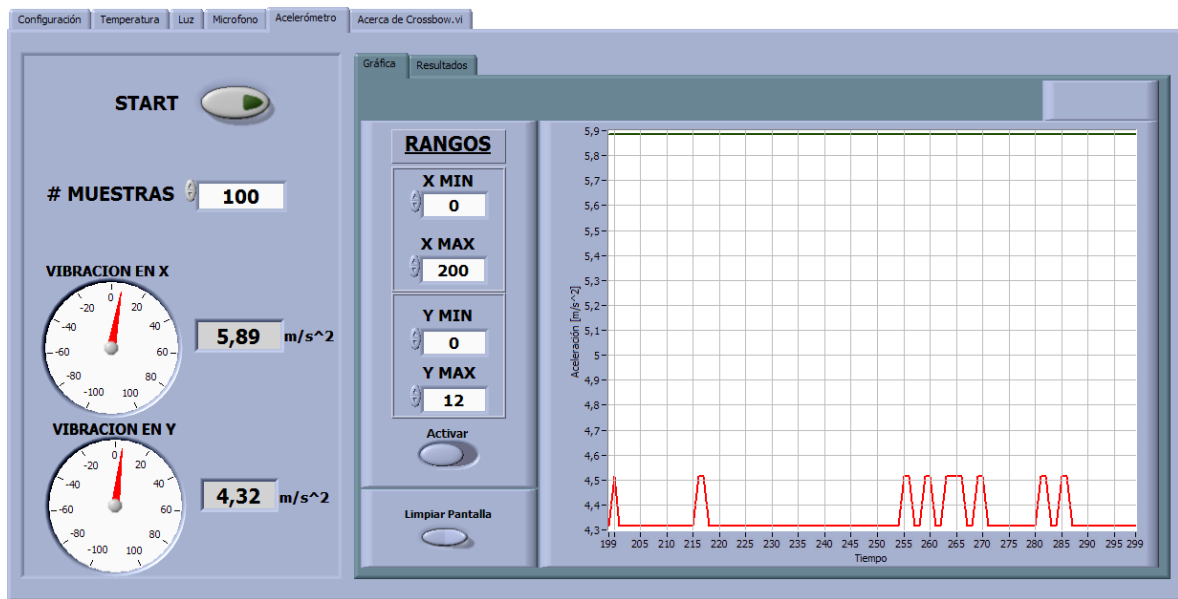


Figura. 4.15. Pantalla Acelerómetro – Gráfica

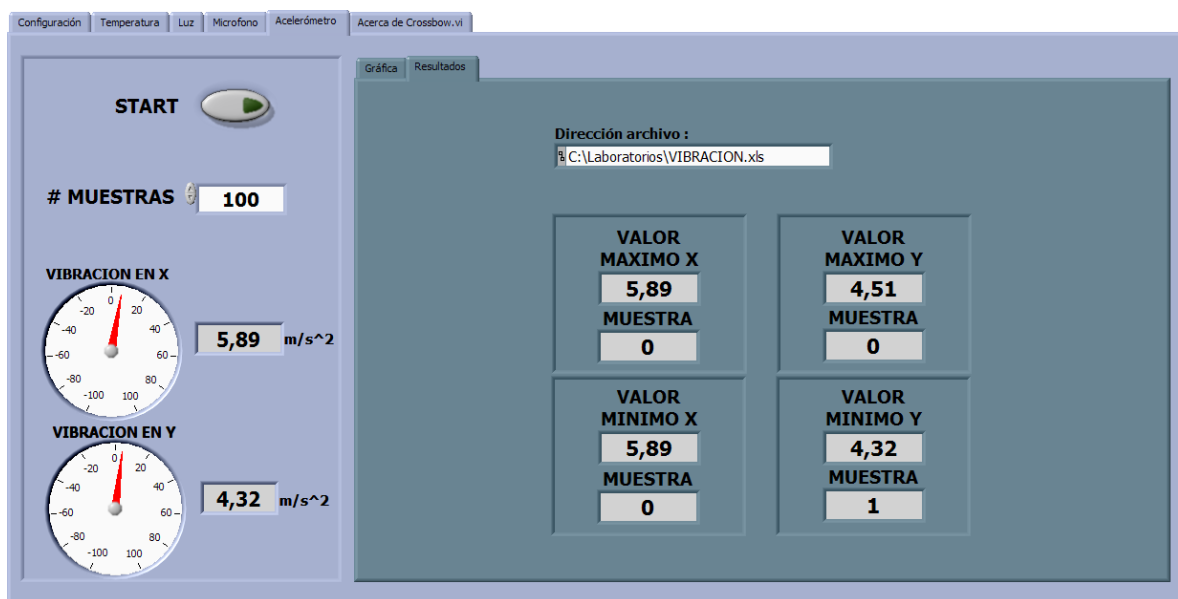


Figura. 4.16. Pantalla Acelerómetro – Resultados

- **Pantalla Acerca de Crossbow.Vi**

Como pantalla final, únicamente muestra información general del Software y del autor.



Figura. 4.17. Pantalla – Acerca de Crossbow.vi

4.3 PROGRAMACIÓN EN LABVIEW

Con los parámetros de las ventanas detallados en la sección anterior se realizó la programación en LabVIEW, para lo cual antes y durante el desarrollo se realizaron las consideraciones pertinentes para su correcto funcionamiento.

Como se explico en la sección 4.1, LabVIEW es un software de instrumentación virtual cuya programación se la realiza en bloques, es decir este programa dispone de varias herramientas para ser conectadas entre sí de acuerdo a las necesidades de la aplicación, esta aplicación en ejecución (Run Time) es conocida como Instrumento Virtual.

En la elaboración del software se utilizaron herramientas sencillas y básicas de tal manera que el programa pueda ser ejecutado en cualquier versión de LabVIEW a partir de la 7.1, cabe recalcar que la Escuela Politécnica del Ejército posee las licencias de varias versiones de este programa de instrumentación virtual, disponibles en el Departamento de Eléctrica y Electrónica.

4.3.1 Diagramas de Flujo

Desde la Figura 4.18 hasta la Figura 4.28 se muestran los diagramas de flujo del programa desarrollado.

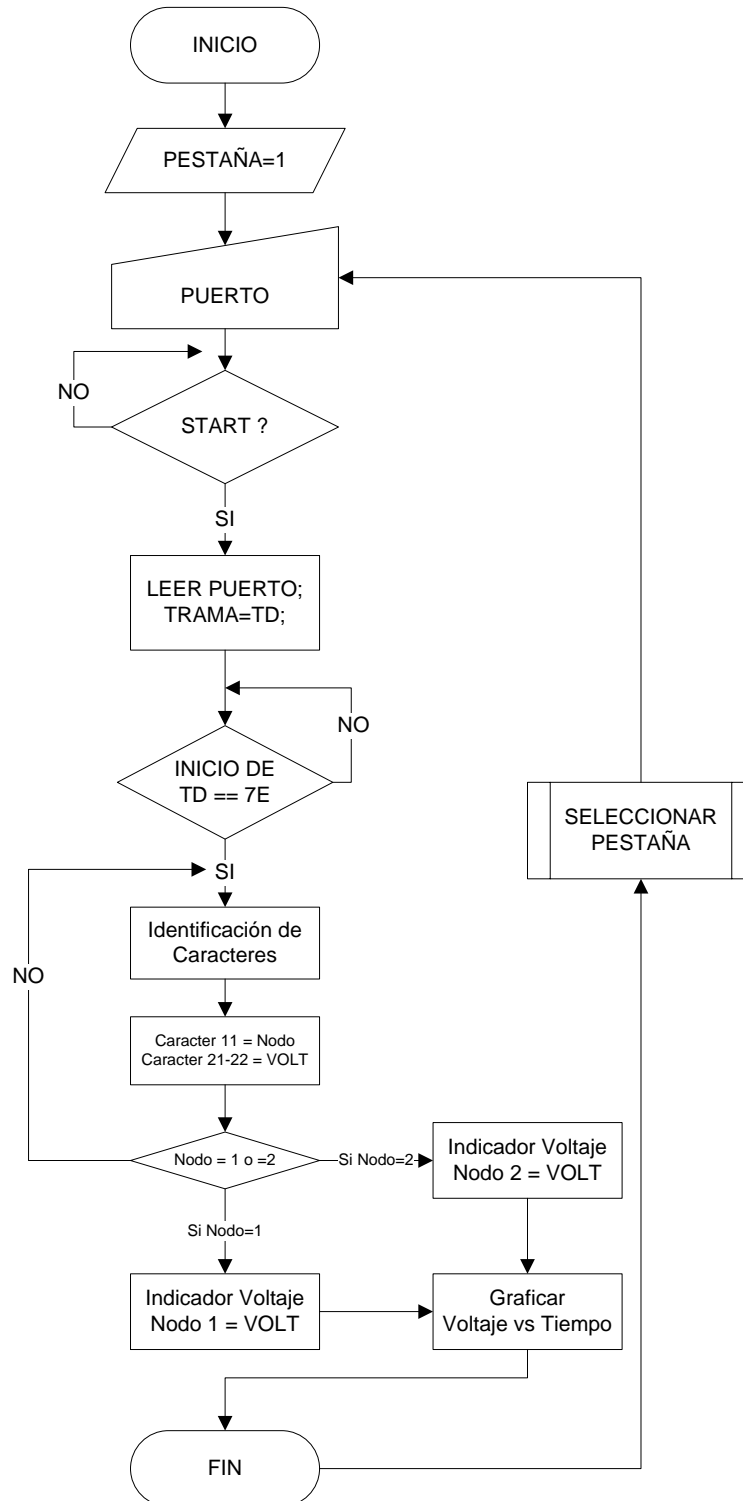


Figura. 4.18. Diagrama de Flujo – Pantalla Principal

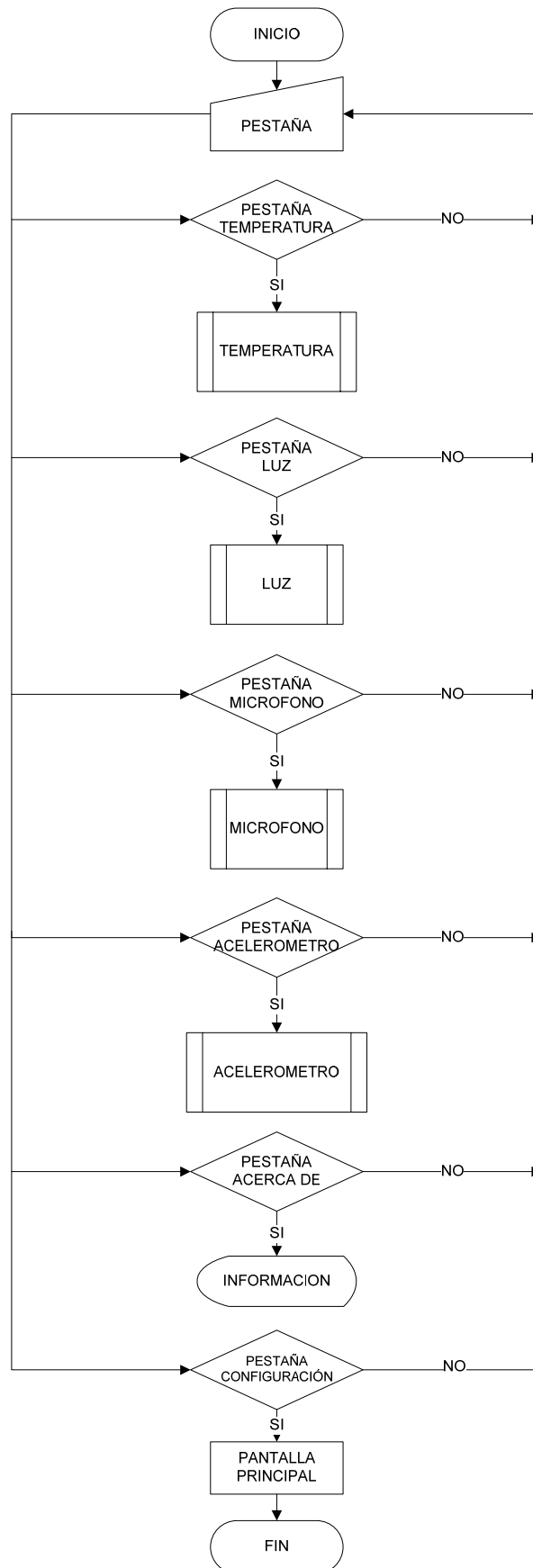


Figura. 4.19. Diagrama de Flujo – Proceso Seleccionar Pestaña

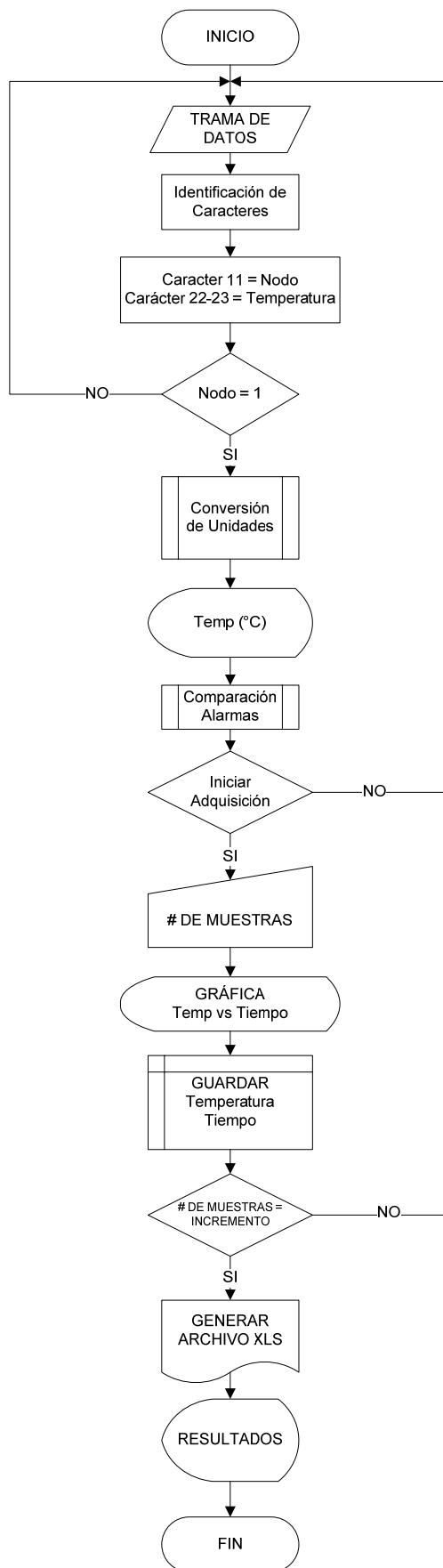


Figura. 4.20. Diagrama de Flujo – Proceso Temperatura

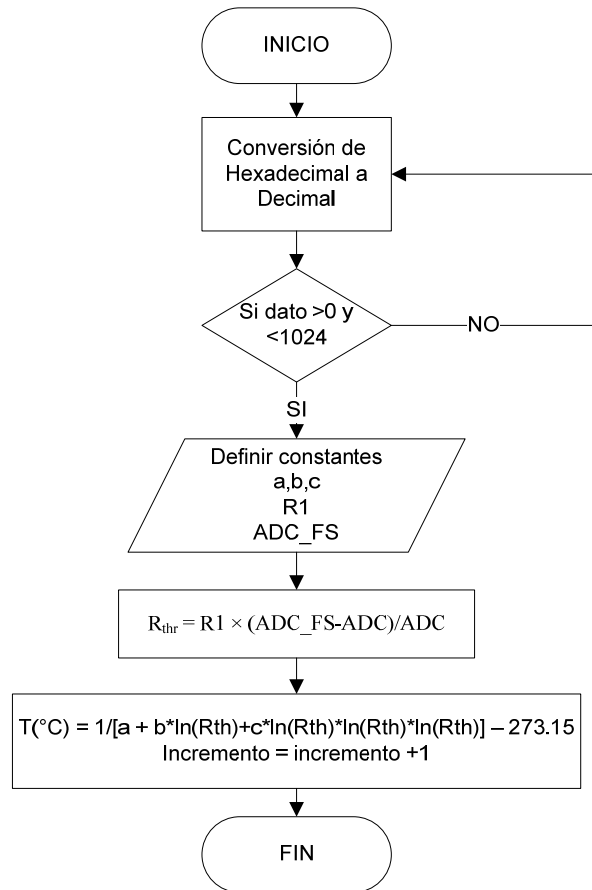


Figura. 4.21. Diagrama de Flujo – Proceso Conversión Unidades Temperatura

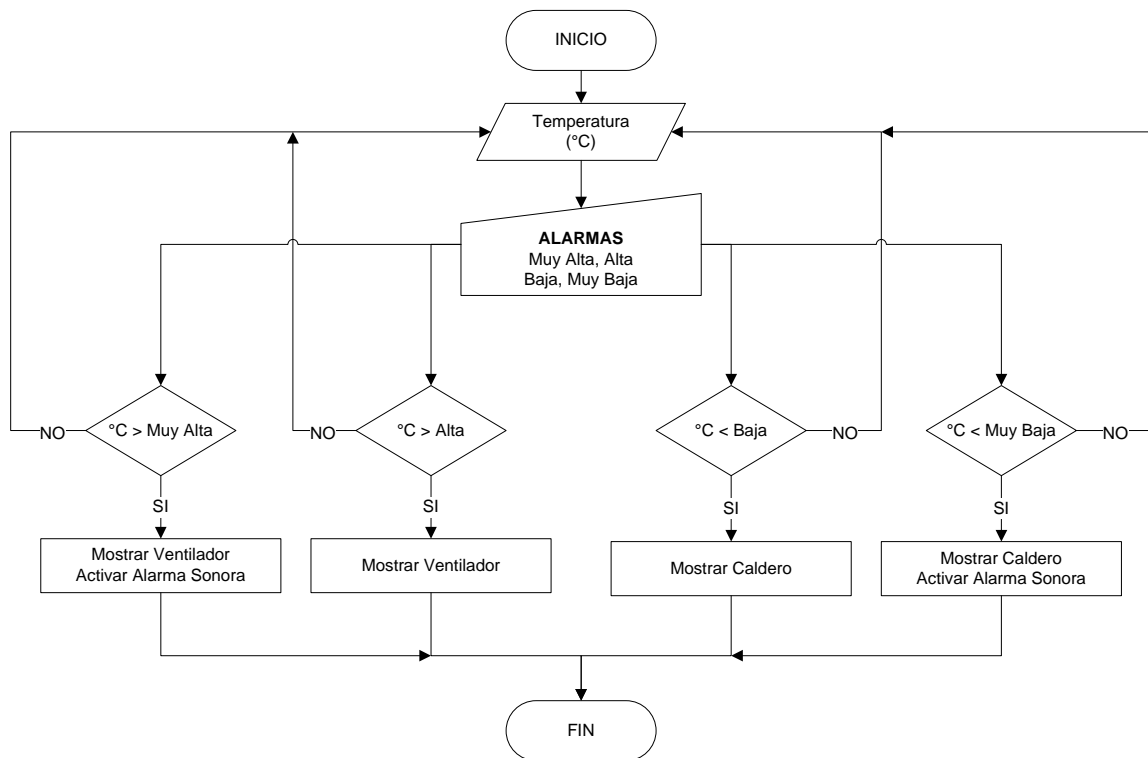


Figura. 4.22. Diagrama de Flujo – Proceso Comparación Alarmas Temperatura

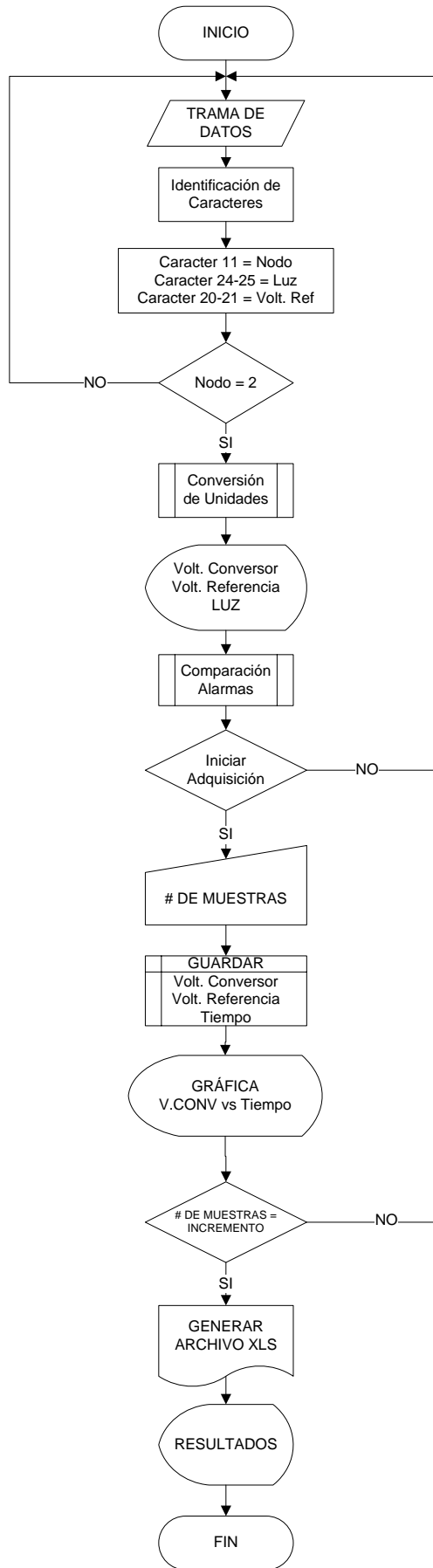


Figura. 4.23. Diagrama de Flujo – Proceso Luz

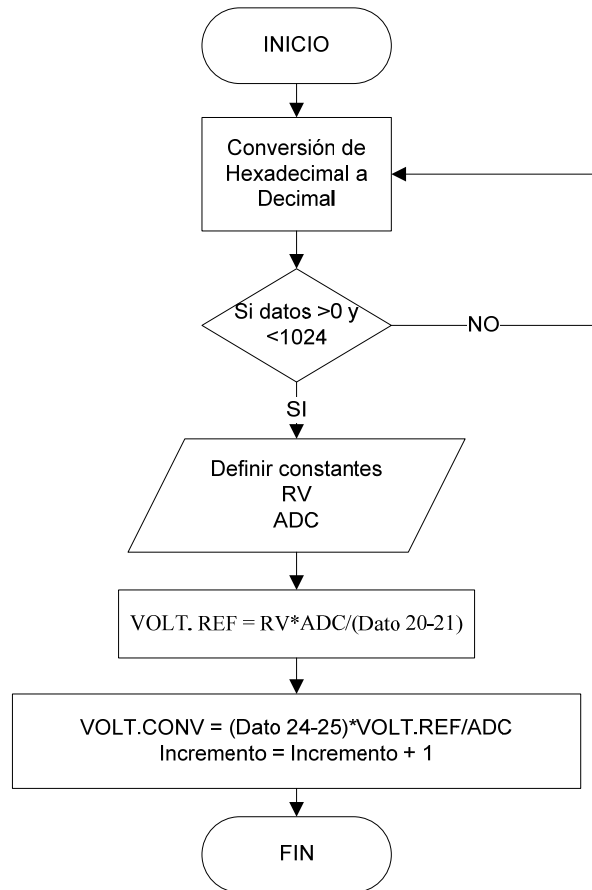


Figura. 4.24. Diagrama de Flujo – Proceso Conversión Unidades Luz

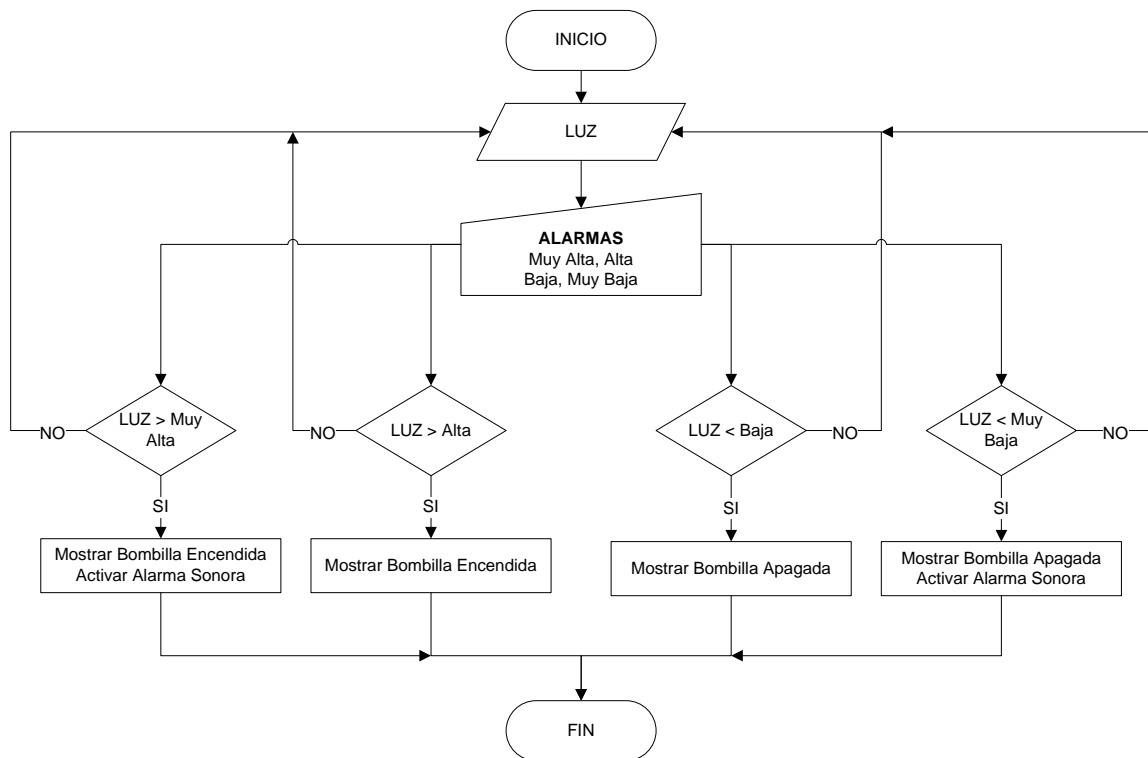


Figura. 4.25. Diagrama de Flujo – Proceso Comparación Alarmas Luz

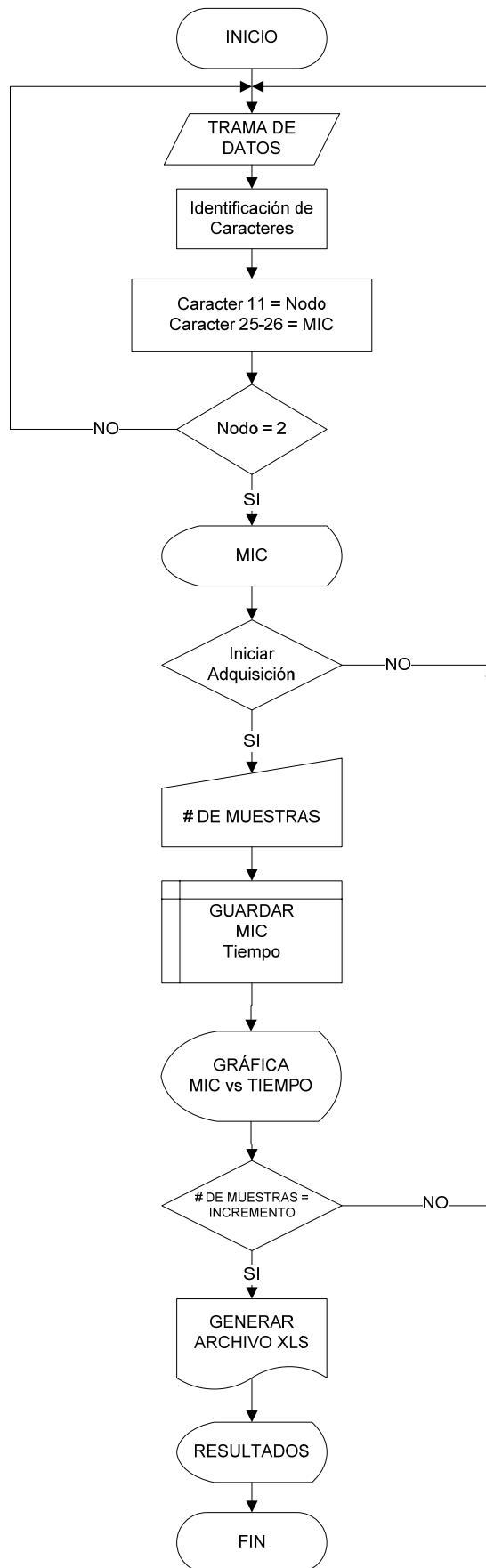


Figura. 4.26. Diagrama de Flujo – Proceso Micrófono

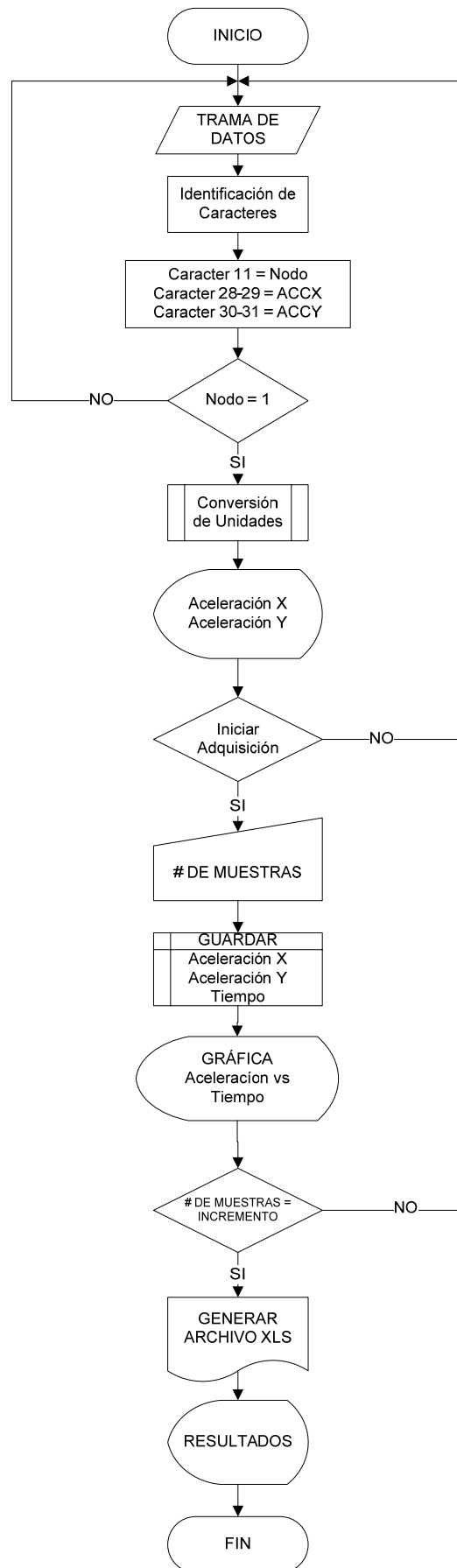


Figura. 4.27. Diagrama de Flujo – Proceso Acelerómetro

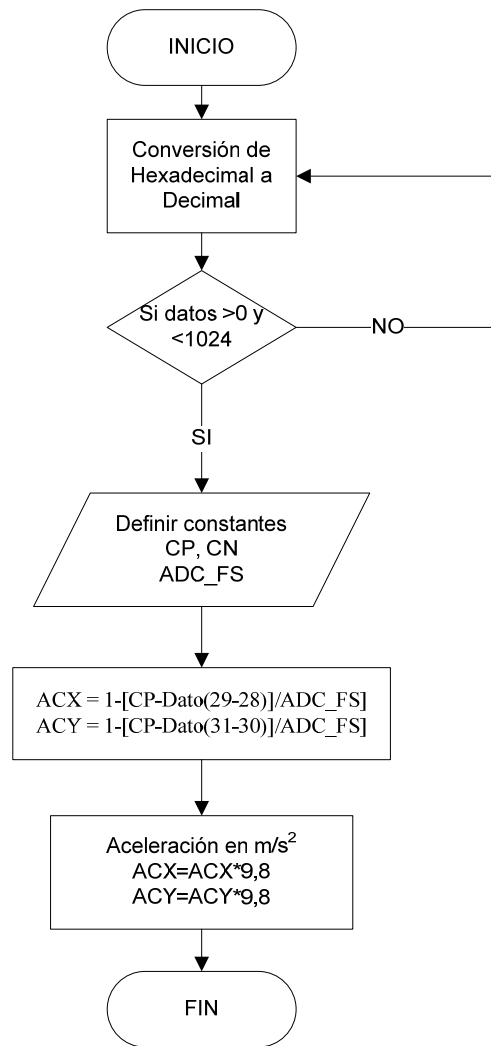


Figura. 4.28. Diagrama de Flujo – Proceso Conversión Unidades Acelerómetro

4.3.2 Descripción de Herramientas Utilizadas

LabVIEW dispone de muchas herramientas de programación para ser utilizadas en la elaboración de un instrumento virtual, para la descripción de las que se utilizaron en este proyecto se las clasificaran de acuerdo a su función como se detalla en la siguiente lista:

- ✓ Controladores.
- ✓ Indicadores.
- ✓ Constantes.
- ✓ Estructuras.
- ✓ Propiedades del nodo.
- ✓ Elementos de Interconexión (funciones).

- **Controladores**

Son herramientas que permiten el ingreso de diferentes tipos de datos sean estos: numéricos, cadenas de caracteres, booleanos, dispositivos de entrada o direcciones; pueden ser de ingreso manual o selección múltiple; estas herramientas son visibles en el panel frontal y en el diagrama de bloques el pin de interconexión está ubicado a la derecha de la herramienta.

En la Tabla 4.1, se detallan las características y visualizaciones de los controladores utilizados en el proyecto.

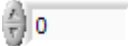







Nombre	Visualización Panel Frontal	Visualización Diagrama de Bloques	Descripción
Controlador Numérico Incremento/Decremento			Para ingresar valores numéricos
Controlador de Dispositivo de entrada			Permite selección del puerto de comunicación serial
Controlador Booleano			Al estar presionado entrega TRUE, sin presión FALSE
Controlador de Dirección			Permite el ingreso de una ruta de dirección para abrir o guardar un archivo.

Tabla. 4.1. Herramientas Utilizadas – Controladores

- **Indicadores**

Los indicadores son herramientas que permiten la visualización de datos en el panel frontal, de la misma manera que los controladores estas herramientas están diseñadas para cualquier tipo de datos.

El pin o pines de interconexión de estos elementos se encuentran ubicados a la izquierda, pueden recibir los datos directamente de los controladores, elementos de interconexión, variables, referencias o constantes. En la Tabla 4.2 se describen los indicadores utilizados en el proyecto.





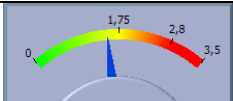

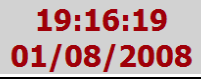



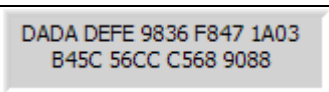



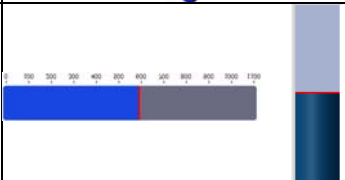

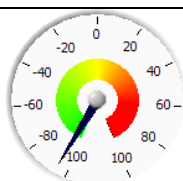

Nombre	Visualización Panel Frontal	Visualización Diagrama de Bloques	Descripción
Indicador Numérico			Muestra datos numéricos
LED			Se visualiza encendido cuando recibe TRUE, apagado cuando FALSE
Indicador Numérico Métrico			Muestra datos numéricos de forma más amigable al operador
Reloj			Muestra la hora y fecha actual
Waveform Chart			Muestra una curva del dato o datos ingresados en función del tiempo
Indicador Cadena de caracteres			Despliega datos provenientes de una cadena de caracteres
Termómetro			Muestra datos numéricos simulando un termómetro
Indicadores Numéricos Horizontal y Vertical			Permiten visualizar valores numéricos como una barra progresiva
Indicador numérico			Permite simular un Gauge

Tabla. 4.2. Herramientas Utilizadas – Indicadores

- **Constantes**

En la elaboración de cualesquier programa es necesaria la utilización de constantes; en LabVIEW así como en otros lenguajes de programación se dispone de valores predefinidos y también se pueden crear constantes o asignar valores fijos a variables para la utilización dentro del desarrollo del software, estas pueden ser creadas por código cuando se utilizan estructuras diseñadas para este fin o pueden ser creadas como un nuevo elemento de LabVIEW (Bloque) de lectura.

Las principales constantes con las que cuenta LabVIEW son matemáticas entre las que se encuentran: PI, e, $\log_{10}(e)$, $1/e$, $\ln(2)$, $\ln(10)$, número de avogadro, constante gravitacional y varias constantes numéricas adicionales. También se dispone de constantes tipo **String** como: constantes de espacio, cadena en blanco, salto de línea, fin de línea, espacio tabulado. En el caso booleano las constantes pueden ser TRUE o FALSE.

En el proyecto se utilizó constantes dentro de estructuras **Math Script**, se utilizaron las constantes tipo String predefinidas y se crearon bloques de constantes numéricas, cadena de caracteres y booleanos.

- **Estructuras**

Al momento de programar, muchas veces es necesario ejecutar un mismo conjunto de sentencias un número determinado de veces, o que éstas se repitan mientras se cumplan ciertas condiciones, también puede ocurrir que se desee ejecutar una u otra sentencia dependiendo de las condiciones fijadas, ejecutar un bloque de mucho código matemático para ahorrar espacio en la ventana de programación y reducir el tiempo de procesamiento, ejecutar sentencias por un determinado tiempo o simplemente forzar que unas se ejecuten siempre antes que otras.

Para ello LabVIEW dispone de algunas estructuras fácilmente diferenciables por su apariencia disponibles en la pantalla de Diagrama de Bloques.

Estas estructuras ayudan a reducir el área de programación, organizar de mejor manera los diagramas de bloques, reducir el tiempo del procesamiento en la ejecución de una aplicación ya que dependiendo de la condición o condiciones de las variables de entrada a estas estructuras sean estas numéricas o booleanas ciertas áreas de código no son utilizadas.

En la Tabla 4.3 se describen las estructuras utilizadas en la elaboración del software de adquisición de datos, la visualización gráfica de estas estructuras en la pantalla del diagrama de bloques y las características más relevantes de cada una de estas herramientas de programación.

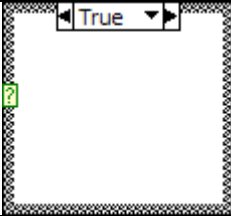
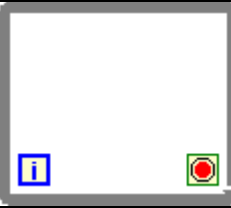
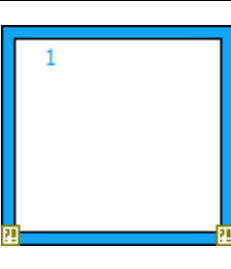
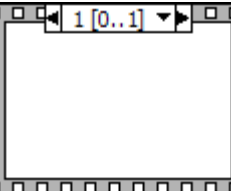
Nombre	Representación	Descripción
Case		Para situaciones en las que el número de alternativas disponibles sean dos o más. Según qué valor tome el selector (?) dentro de los n valores posibles, se ejecutará en correspondencia uno de los n subdiagramas El selector puede ser booleano o numérico.
While		Para operaciones que se repitan mientras una determinada condición sea cierta. Si su valor es TRUE (verdadero) continuará, pero por el contrario si su valor es FALSE (falso) detendrá la ejecución.
Math Script		Contiene una o más fórmulas separadas por un punto y coma, para ejecutar fórmulas matemáticas que serían complicadas de crear utilizando las diferentes herramientas matemáticas que LabVIEW incorpora en sus librerías. Una vez escrita la fórmula en el interior del rectángulo sólo se añaden los terminales que harán la función de variables de entrada o de salida
Stacked Sequence		Consiste de uno o más subdiagramas o marcos que se ejecutan secuencialmente. Se pueden agregar secuencias antes y después, eliminar, duplicarlas.

Tabla. 4.3. Herramientas Utilizadas – Estructuras

- **Propiedades del Nodo**

Propiedades del nodo se refieren a características que pueden obtener o se pueden modificar en las herramientas excluyendo a estructuras o funciones especiales de LabVIEW.

Las propiedades de los nodos pueden ser de lectura o escritura y dependen del nodo seleccionado; para controladores e indicadores es posible setear y leer valores máximos y mínimos, incrementos, el formato del número, esconder o visualizar el indicador o etiquetas, inicializar valores etc. Para indicadores gráficos es posible modificar su color de fondo, color de línea, rangos, escalas, etc.

Estas herramientas son de gran importancia al momento de contar con una interfaz más amigable, obtener y alterar características de herramientas sin programación adicional.

- **Elementos de interconexión**

A estas herramientas también se las puede denominar funciones y permiten realizar operaciones especiales que ya están predefinidas en LabVIEW, o es posible crear una nueva función (subVI) que cumpla un propósito similar.

A estas herramientas se las puede clasificar en tres grupos: elementos de interconexión o funciones que poseen solo entradas, elementos que únicamente poseen salidas y elementos de interconexión que poseen entradas y salidas. En la Tabla. 4.4 se detallan las herramientas utilizadas.

















Nombre	Representación	Descripción
VISA Configuración puerto serial		Inicializa y configura los parámetros del puerto de comunicación serial
VISA Lectura		Recibe la información del puerto, lee el número de bytes recibidos y retorna el dato recibido
VISA Close		Cierra el puerto de comunicación serial
String to Byte Array		Convierte una cadena de caracteres en un arreglo de bytes
Index Array		Retorna un elemento de un arreglo de acuerdo al índice ingresado
Bundle		Ensambla un cluster de elementos individuales
Build Array		Concatena múltiples arreglos o elementos en un nuevo arreglo
Get Date/Time		Retorna un dato del tiempo y fecha del computador Este Elemento tiene únicamente salidas
Format Date/String		Permite modificar el formato de fecha y hora Retorna el tiempo en una cadena de caracteres
Number to Fractional String		Convierte un número en una cadena Fraccional
Concatenate String		Concatena cadenas o arreglos tipo string de una dimensión
Array Max & Min		Retorna los valores máximos y mínimos de un arreglo
Write to Text File		Escribe una cadena de caracteres o un arreglo de cadenas como líneas en un archivo.
Write to Spradsheet File		Convierte arreglos de distintos tipos (numéricos, cadenas booleanos) en un archivo de texto
Greater?		Retorna TRUE cuando se cumple la condición
Or		Retorna FALSE si no cumple la condición

Tabla. 4.4. Herramientas Utilizadas – Funciones

Las herramientas descritas desde la Tabla 4.1 a la Tabla 4.4 son los objetos y elementos que en conjunto forman el software de adquisición de datos, el código general del programa se encuentra disponible en el Anexo 1.

Finalizada la programación de la aplicación el siguiente paso es la comprobación del funcionamiento del software bajo distintas condiciones y eventos que simulen lo que se va a realizar en las prácticas de laboratorio y de la misma manera la comprobación del funcionamiento del hardware independiente a la aplicación y el funcionamiento en conjunto (Hardware-Software), esto se va a detallar con más claridad en el siguiente capítulo.

CAPÍTULO 5

PRUEBAS Y RESULTADOS

5.1 PRUEBAS DE TRANSMISIÓN

Una vez programados los elementos que componen la Red de Sensores Inalámbricos (Nodos y Estación Base) se procedió a verificar que estos funcionen correctamente con el Software de fábrica MoteView, para lo cual en este software se configuraron los siguientes parámetros mostrados en la Figura. 5.1.

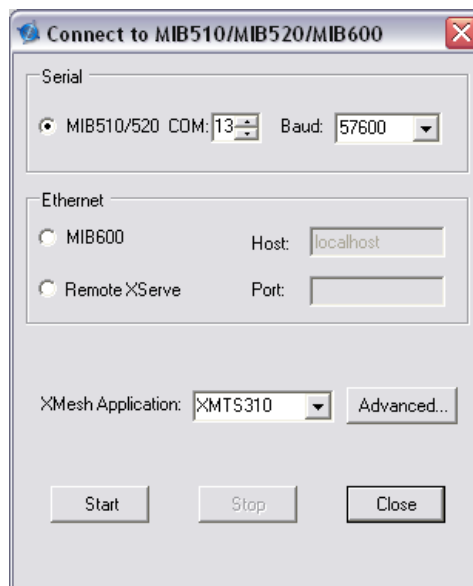


Figura. 5.1. Parámetros de Configuración MoteView

En los parámetros Setados el puerto **COM** debe ser el $n+1$, la velocidad (**Baud**) 57600bps y la **Xmesh Application** XMTS310. Una vez iniciada la aplicación, la ventana principal de MoteView muestra la información de los sensores de cada nodo.

La Figura. 5.2. muestra la ventana de MoteView con la comprobación de transmisión de datos de los nodos de la WSN.

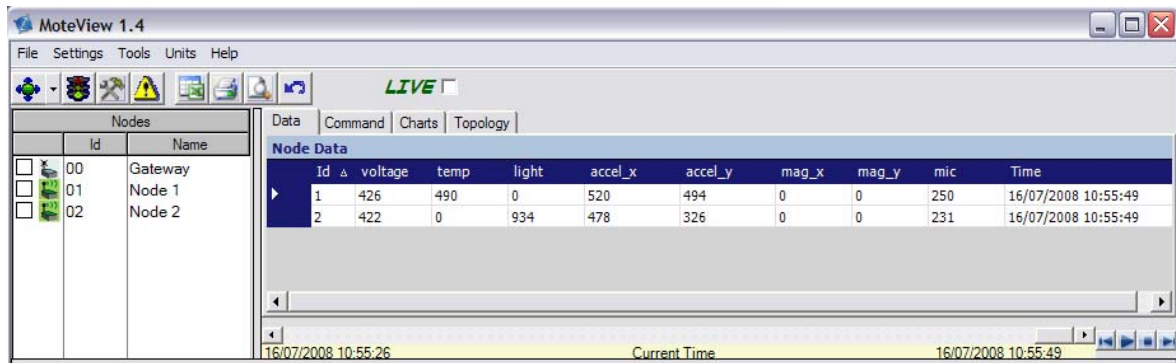


Figura. 5.2. Prueba de transmisión utilizando MoteView

El siguiente paso fue la comprobación de transmisión de la información de los nodos con LabVIEW para lo cual se realizó una aplicación sencilla que verifique el correcto funcionamiento.

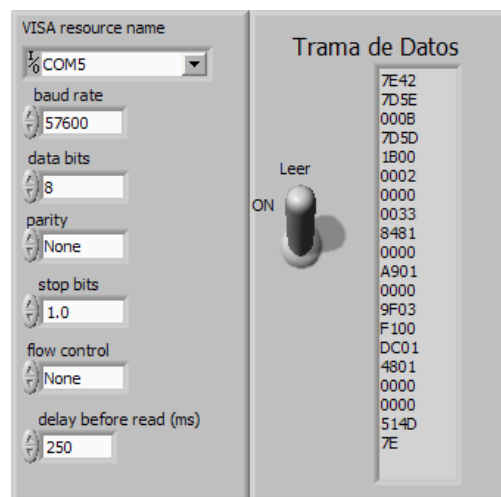


Figura. 5.3. Prueba de transmisión exitosa utilizando LabVIEW

La Figura. 5.3. muestra la aplicación que se realizó en LabVIEW para la comprobación de transmisión, esta también se aprovechó para identificar los elementos que constituyen la trama de datos descritos en la Tabla 3.2.

Durante las pruebas realizadas en LabVIEW se comprobó que la trama de datos inicia con la cabecera **7E42 7D5E 000B 7D5D 1B00**, sin embargo no siempre se recibe esta información en el programa, por lo que fue necesaria la realización de un algoritmo para evitar errores de adquisición.

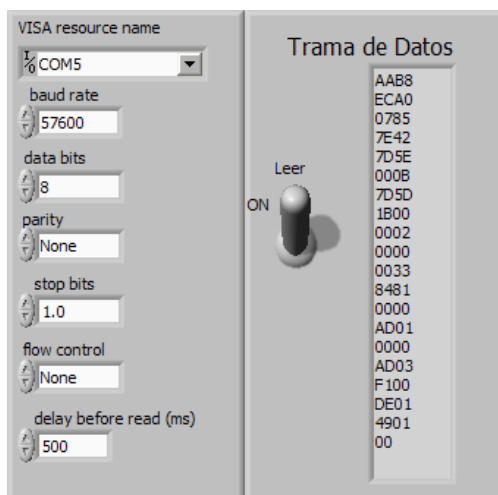


Figura. 5.4. Prueba de transmisión fallida utilizando LabVIEW

La Figura. 5.4. muestra una trama de datos recibida con errores, de no corregir este error la información recibida puede ser errónea, pese a que el resto del programa también consta con validaciones para la adquisición de cada magnitud física.

5.2 PRUEBAS DE ADQUISICIÓN DE DATOS

Durante y luego de la elaboración del software se sometió los nodos a distintas alteraciones de acuerdo a la magnitud física, ya sea aumentando o disminuyendo la cantidad de temperatura, luz, sonido, o realizando vibraciones para visualizar el cambio de cada una de ellas en las pantallas correspondientes en la interfaz.

Para comparar los datos adquiridos en LabVIEW se utilizó el software de fabrica MoteView, para lo cual se ubico a los nodos en un sector fijo y se procedió a visualizar la información de los sensores en cada uno de estos programas. Esta operación no se la puede realizar al mismo tiempo debido al conflicto que se produce por la utilización del puerto serial de lectura.

En la Figura. 5.5. y Figura. 5.6. se muestra la adquisición de datos en las pantallas de MoteView y LabVIEW respectivamente para el estado de la batería de los dos nodos utilizados en este proyecto, como se puede observar los datos adquiridos en LabVIEW son los mismos que se visualizan en la ventana de MoteView.

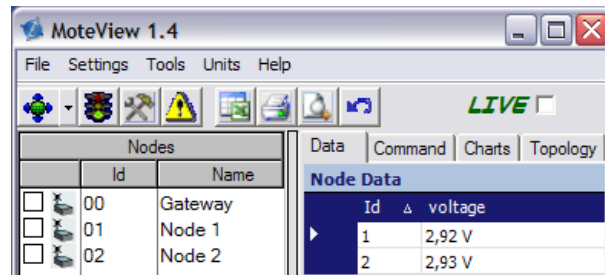


Figura. 5.5. Dato de voltaje adquirido en MoteView



Figura. 5.6. Dato de voltaje adquirido en LabVIEW

El siguiente dato que se comparó fue la temperatura, en la Figura. 5.7. se observa la información de los nodos 1 y 2; como se aprecia ahora el valor medido en el nodo 2 es cero, esto debido a que como se indico en la Sección 3.4 en este nodo se deshabilito dicho sensor para no tener inconvenientes con el canal de conversión.

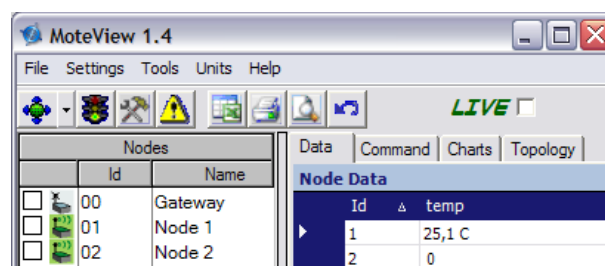


Figura. 5.7. Dato de Temperatura adquirido con MoteView

Bajo idénticas condiciones se realizó la medida de temperatura utilizando la aplicación de LabVIEW, los resultados se muestran en la Figura. 5.8.

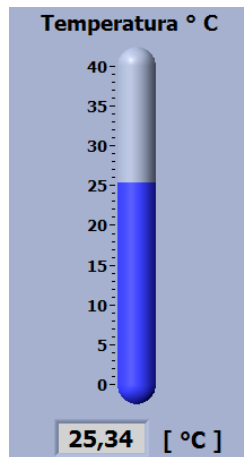


Figura. 5.8. Dato de Temperatura adquirido con LabVIEW

Así como en el nodo 2 se deshabilitó el sensor de temperatura para no tener errores al momento de utilizar el canal de conversión; en el nodo 1 se deshabilitó la fotoresistencia que detecta la cantidad de luz en el ambiente, y los resultados de comparación se indican en la Figura. 5.9. para MoteView y Figura. 5.10. para LabVIEW.

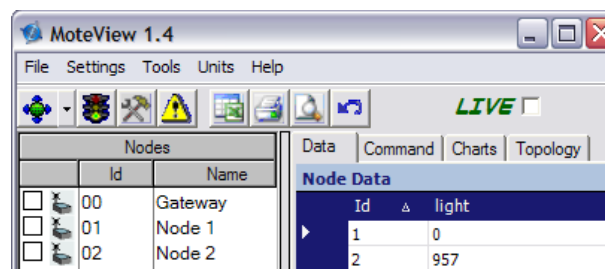


Figura. 5.9. Dato de Luz adquirido con MoteView

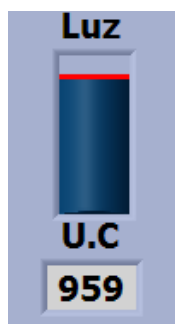


Figura. 5.10. Dato de Luz adquirido con LabVIEW

Las variables físicas de sonido y vibración provenientes del detector de tono y acelerómetro, se compararon de la misma manera que las anteriores y los resultados se muestran en las figuras mostradas a continuación.

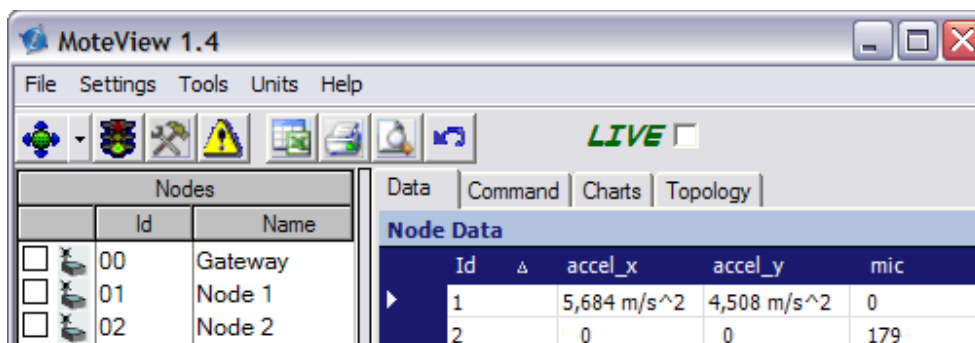


Figura. 5.11. Dato de vibración y sonido adquirido con MoteView



Figura. 5.12. Dato de vibración y sonido adquirido con LabVIEW

Si bien el detector de tono y el acelerómetro poseen canales de conversión independientes, en los nodos 1 y 2 se deshabilitaron los sensores que no se utilizan, para tener mejor eficiencia energética.

5.3 ANÁLISIS DE RESULTADOS

De los resultados obtenidos mientras se realizaron las pruebas del software de adquisición de datos, se logró determinar que este funciona correctamente; los datos obtenidos de las diferentes magnitudes físicas están acorde a la realidad y tienen la comprobación adecuada utilizando el software de fábrica.

Si bien en las figuras comparativas entre los resultados obtenidos en el software de fábrica MoteView y el software desarrollado en LabVIEW para este proyecto se presentan pequeñas variaciones, estas se deben a que se trataron de realizar mediciones bajo idénticas condiciones el tiempo que toma en cambiar la adquisición entre las interfaces hace que existan pequeñas variaciones en las lecturas, sin embargo estas son no son significativas y poseen un error menor al 0.01% para datos de los distintos sensores; para datos que no varían con respecto a las situaciones ambientales como es el caso del voltaje de la batería de los nodos, en las pruebas realizadas no se tiene errores de medición.

Una de las importantes características que presentan las Redes de Sensores Inalámbricos como se explico con anterioridad es el bajo consumo energético esta cualidad se comprobó con la realización del proyecto ya durante todo el tiempo de elaboración y comprobación se utilizaron únicamente tres pares de baterías AA para ambos nodos.

Con la red de sensores inalámbricos correctamente configurada y el software de adquisición de datos funcionando, el último paso del desarrollo del proyecto es la elaboración de las prácticas de laboratorio las mismas que se encuentran planteadas en el próximo capítulo de este documento.

CAPÍTULO 6

PRACTICAS DE LABORATORIO

6.1 PRÁCTICA 1: SONIDO

TEMA: INTERPRETACIÓN DE FRECUENCIA

6.1.1 Objetivos

Determinar la palabra digital correspondiente a una muestra de frecuencia analógica.

Encontrar un modelo matemático para la frecuencia.

6.1.2 Marco Teórico

- **Circuitos detectores de tono**

Los detectores de tono son circuitos integrados que se encargan de verificar si en la entrada del mismo hay una frecuencia igual o distinta a la que está sintonizado.

- **Frecuencia y notas musicales**

La característica fundamental de los sonidos musicales es su “altura” o cantidad de veces que vibra por segundo (frecuencia). Cuanto más oscilaciones por segundo tenga la frecuencia, más aguda o “alta” será la nota musical. En la música es muy importante la relación que existe entre la frecuencia de los distintos sonidos, a esta relación se le llama intervalo. La relación entre las notas musicales parten de la nota musical **LA** que tiene una frecuencia de 440Hz.

El oído humano puede llegar a captar frecuencias entre los 20Hz hasta 20.000Hz, la frecuencia de las notas musicales llega solamente a los 4.500Hz. En la Figura. 6.1, se muestra la frecuencia de las notas musicales.

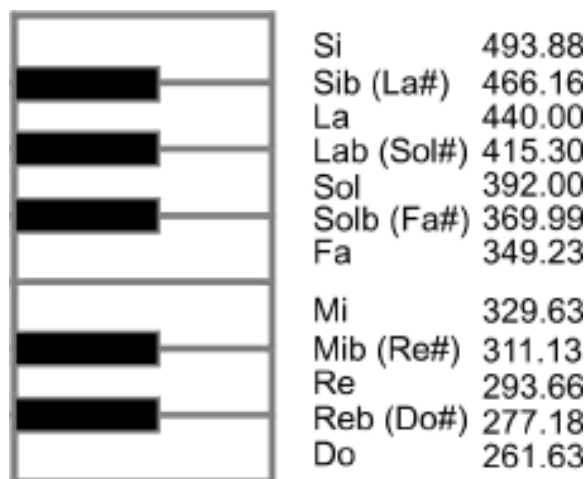


Figura. 6.1. Frecuencias de las Notas Musicales

6.1.3 Equipo Necesario

1. Estación Base MIB520CB.
2. Nodo 2: MPR2400.
3. Sensores MTS310CB.
4. Un par de baterías AA.
5. Computador Personal con drivers de estación base instalados y software de adquisición de datos Crossbow.vi (ver manual de usuario).
6. Equipo generador de frecuencia sonora.

6.1.4 Desarrollo de la práctica

1. Conectar correctamente la estación base (MIB520CB) al computador.
2. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
3. Identificar el NODO 2 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
4. Conectar las baterías AA al nodo.
5. Encender el tablero NODO-SENSORES.
6. Abrir y ejecutar la aplicación Crossbow.vi
7. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
8. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido).



Figura. 6.2. Verificación nodo Activo

9. Calibrar el equipo generador de frecuencias.
10. En la pestaña **Micrófono** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará.



Figura. 6.3. Ejemplo de ruta para generación de archivos

11. Acerque el generador de frecuencia (sonido) configurado a la frecuencia F1, al micrófono del sensor MTS310.
12. Presione al botón **START** de la ventana de sonido.

13. Finalizada la adquisición en la ventana de resultados se desplegarán los picos máximos y mínimos de la adquisición, estos resultados deben ser anotados en la siguiente tabla.

F1 =		F2 =		F3 =		F4 =		F5 =	
MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN

Tabla. 6.1. Resultados Practica 1

14. Repetir los pasos 9 a 12 para cada una de las frecuencias de la Tabla anterior. *Debe generarse un archivo para cada una de las frecuencias.*

15. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el grafico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione *Export Simplified Image* y posteriormente en *Save to File* seleccione la ubicación del archivo.

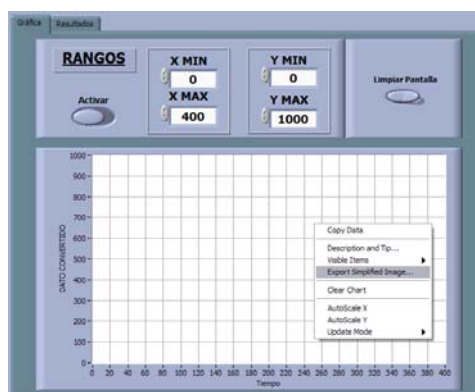


Figura. 6.4. Exportar Gráfico Historio

6.1.5 Cuestionario

1. ¿Qué son los detectores de tono?

- Indique 5 aplicaciones de los detectores de tono

- Enumere 3 elementos adicionales que cumplan la función de detector de tono y señale sus principales características.

6.1.6 Gráficas y Análisis de Resultados

- Utilizando Excel o Matlab realice las Gráficas DATO vs TIEMPO para cada una de las frecuencias tomadas en la Tabla 6.1.
- Obtenga el promedio del dato analógico convertido a digital para cada una de las frecuencias de la Tabla 6.1, realice una Gráfica PROMEDIO vs FRECUENCIA en Excel o Matlab y con la ayuda de las herramientas de estos programas obtenga la ecuación de la curva.
- Con los promedios obtenidos en el paso 5 complete la siguiente Tabla.

Frecuencia	Promedio	Palabra Digital
F1 :		
F2 :		
F3 :		
F4 :		
F5 :		

Tabla. 6.2. Cálculos – Práctica 1

4. Inserte la Grafica obtenida en el paso 15 del desarrollo.
5. Analice los resultados obtenidos en los pasos anteriores.

6.1.7 Conclusiones y Recomendaciones

6.1.8 Bibliografía

6.2 PRÁCTICA 2: ILUMINACIÓN

TEMA: VARIACIONES DE RESISTENCIA DEBIDO A CAMBIO DE LUZ

6.2.1 Objetivos

- Calcular la variación de resistencia debido al cambio de luz en la fotoresistencia del modulo de sensores Crossbow MTS310CB.
- Analizar la mínima y máxima variación de tensión debido al cambio de luz con respecto al voltaje de referencia.
- Determinar la palabra digital de cada una de las muestras realizadas.

6.2.2 Marco Teórico

- **Fotorresistencias**

Fotorresistencias, son resistencias sensibles a la luz cuyo valor ohmico puede variar fuertemente en función de la luz incidente.

El valor de la resistencia en la oscuridad es típicamente de alrededor de unos $M\Omega$. Al contrario, con 10 Lux se encuentra en la zona de pocos $K\Omega$. La ventaja en comparación con los fotodiodos de silicio consiste en la sensibilidad espectral de los LDRs. Mientras los fotodiodos son sensibles entre 400nm (azul) y 1100nm (Infrarrojo), los LDRs solo detectan luz en el espectro visible. Por ello los LDRs son especialmente aptos como sensores de luz porque no detectan la parte infrarroja que se encuentra normalmente en la luz natural y que puede falsear la medición.



Figura. 6.5. Fotorresistencia

Propiedades

- ✓ Solo componentes con dos terminales
- ✓ Sensores de diferente tamaño
- ✓ Impermeables o sobre sustrato de cerámica.
- ✓ Sensibles en el espectro visible

Aplicaciones

- ✓ Control de iluminación
- ✓ Retrovisor de automóvil automático

- **Circuito de Referencia**

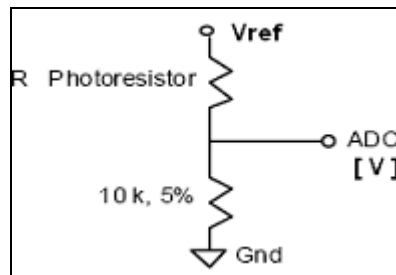


Figura. 6.6. Circuito Fotoresistencia

La ecuación de conversión es:

$$\text{ADC}[\text{V}] = \text{BV} \times \text{DATA}/\text{ADC_FS} \quad \text{Ecuación 7.1}$$

donde,

BV = Voltaje de la Batería

DATA = Datos ADC convertido a decimal

ADC_FS = 1023

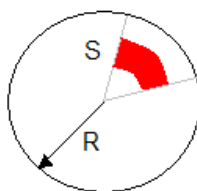
La Ecuación para hallar el valor de la fotoresistencia es:

$$R_{PH} = 10\text{K}\Omega \frac{V_{REF}}{V_{CONV}} - 10\text{K}\Omega \quad \text{Ecuación 7.2}$$

- **Luminotecnia**

Es la ciencia que estudia las diferentes formas de producir luz, su control y su aplicación. Su objetivo es permitir el funcionamiento de las tareas visuales en un determinado lugar, el confort visual y que exista un nivel de iluminación que permita distinguir las cosas.

Angulo Solido (ω). Relaciona el área de un casquete esférico respecto al radio del mismo.



$$\omega = \frac{S}{R^2}$$

Ecuación 7.3

Flujo Luminoso (F). Cantidad de luz emitida por una fuente luminosa por unidad de tiempo, su unidad de medida es el Lumen \rightarrow Lm

Intensidad Luminosa (I_L). Es el flujo luminoso por unidad de ángulo sólido radiado en una sola dirección, su unidad de medida es la candela (Cd).

$$I_L = \frac{F}{\omega} [Cd]$$

Ecuación 7.4

Nivel de Iluminación (E). Llamado simplemente iluminación, es el flujo luminoso incidente por unidad de superficie su unidad de medida es el Lux \rightarrow Lx.

$$E = \frac{F}{S}$$

Ecuación 7.5

Capacidad Visual. Es el término que se utiliza para describir la velocidad del funcionamiento del ojo y la exactitud con que se lleva a cabo una tarea.

Luminancia (L). Es la intensidad luminosa por unidad de superficie aparente de una fuente de luz primaria o secundaria.

Eficiencia Luminosa (η). Es la relación entre el flujo luminoso total de la fuente respecto de la potencia que consume la misma fuente.

$$\eta = \frac{F}{P} \left[\frac{lum}{W} \right] \quad \text{Ecuación 7.6}$$

Curvas ISOLUX. Es el lugar geométrico de puntos de una superficie donde la iluminancia tiene el mismo valor.

6.2.3 Equipo Necesario

1. Estación Base MIB520CB.
2. Nodo 2: MPR2400.
3. Sensores MTS310CB.
4. Un par de baterías AA.
5. Computador Personal con drivers de estación base instalados y software de adquisición de datos Crossbow.vi (ver manual de usuario).

6.2.4 Desarrollo de la Práctica

1. Conectar correctamente la estación base (MIB520CB) al computador.
2. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
3. Identificar el NODO 2 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
4. Conectar las baterías AA al nodo.
5. Encender el tablero NODO-SENSORES.
6. Abrir y ejecutar la aplicación Crossbow.vi
7. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
8. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido Figura 6.2).

9. En la pestaña **Luz** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará. (Figura 6.3)
10. Visualmente identifique cuatro lugares dentro del laboratorio en los cuales exista niveles distintos de iluminación.
11. Ubique el sensor en uno de los lugares identificados y presione al botón **START** de la ventana de luz.
12. Finalizada la adquisición en la ventana de resultados se desplegarán los picos máximos y mínimos de la adquisición, estos resultados deben ser anotados en la siguiente tabla.

L1		L2		L3		L4		V _{REF}
MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	[V]

Tabla. 6.3. Resultados 1 – Práctica 2

13. Repetir los pasos 11 y 12 para cada uno de los lugares del paso 10.
Debe generarse un archivo para cada una de las adquisiciones.
14. De acuerdo a los resultados de la Tabla 6.3, realice dos adquisiciones largas (100 a 200 muestras) en la cual el sensor pase por todos los 4 lugares de la Tabla 6.3, una desde el lugar con menos iluminación hasta el lugar con mayor iluminación y otra adquisición desde el lugar de mayor iluminación hacia el lugar de menos iluminación.
Debe generarse un archivo para cada una de las adquisiciones.
15. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el gráfico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione **Export Simplified Image** y posteriormente en **Save to File** seleccione la ubicación del archivo.
16. Utilizando su mano varíe el nivel de iluminación sobre la fotoresistencia y active cada una de las alarmas de la pestaña **Alarmas** y complete la Tabla. 6.4.

MUY BAJA		BAJA		ALTA		MUY ALTA	
V_{CONV}	V_{REF}	V_{CONV}	V_{REF}	V_{CONV}	V_{REF}	V_{CONV}	V_{REF}

Tabla. 6.4. Resultados 3 – Practica 2

6.2.5 Cuestionario

1. ¿Cuáles son las características de las fotoresistencias?

2. Indique 5 elementos detectores de luz.

3. Consulte los flujos luminosos para:

Lámpara incandescente 60 W: _____

Lámpara incandescente 100 W: _____

Lámpara fluorescente 20 W: _____

Lámpara fluorescente 40 W: _____

4. Consulte los niveles de iluminación recomendados para:

Cancha de Futbol: _____

Corredores: _____

Escaleras: _____

Dormitorios: _____

Salas de Clase: _____

5. Determine los valores de resistencia para los siguientes casos:

$$V_{CONV} = V_{REF}/4$$

$$V_{CONV} = V_{REF}/2$$

$$V_{CONV} = 3V_{REF}/4$$

6. Con los datos de la Tabla. 6.4, determine:

Rangos entre alarmas en Voltios.

Rangos entre alarmas en Resistencia.

Rangos entre alarmas en Unidades Crudas.

6.2.6 Gráficas y Análisis de Resultados

1. Utilizando Excel o Matlab realice las Gráficas V_{CONV} vs TIEMPO para cada uno de los archivos obtenidos.
2. Realice una gráfica V_{CONV} vs RESISTENCIA para los primeros datos obtenidos en el paso 14.
3. Realice una gráfica de histéresis con los datos obtenidos en el paso 14.
4. Con los datos de la Tabla 6.3 y la ecuación de la fotoresistencia complete la siguiente tabla.

LUGAR	Valor Máximo [V]	Resistencia [KΩ]	Valor Mínimo [V]	Resistencia [KΩ]
1				
2				
3				
4				

Tabla. 6.5. Cálculos 1 – Práctica 2

5. Con los datos de los archivos del paso 14 del desarrollo de la práctica complete la siguiente tabla:

Medición	Promedio $V_{CONV}[V]$	Promedio $V_{REF}[V]$	Resistencia [K Ω]	Palabra Digital (BIN)
1 – 4				
4 – 1				

Tabla. 6.6. Cálculos 2 – Práctica 2

6. Con los valores de resistencia calculados en la Tabla 6.5, realice una gráfica Diferencia de Resistencia vs Lugar de Muestra.
7. Inserte la grafica obtenida en el paso 15.
8. Analice los resultados obtenidos en los pasos anteriores.

6.2.7 Conclusiones y Recomendaciones

6.2.8 Bibliografía

6.3 PRÁCTICA 3: TEMPERATURA

TEMA: VARIACIÓN DE TEMPERATURA

6.3.1 Objetivos

Medir de la variación de temperatura sobre el termistor del MTS310CB, producida por una secadora de cabello.

Determinar la relación entre la variación de temperatura y la variación de la resistencia del termistor.

Calcular el valor digital del dato de temperatura.

6.3.2 Marco Teórico

- **Termistores**

Son resistores variables con la temperatura, basados en semiconductores. Su rango de medición de temperatura está entre los -50°C y los 150°C no obstante las unidades encapsuladas pueden alcanzar los 300°C . En la mayoría de aplicaciones el valor de resistencia a 25°C está entre 100Ω . Y $100\text{K}\Omega$. Aunque se pueden producir con resistencias tan bajas como 10Ω o tan altas como $40\text{M}\Omega$. Se pueden clasificar en NTC y PTC.

NTC. Esta resistencia se caracteriza por su disminución del valor resistivo a medida que aumenta la temperatura, por tanto presenta un coeficiente de temperatura negativo.

PTC. Se caracteriza por el coeficiente de temperatura positivo, de forma que su resistencia aumentará como consecuencia del aumento de la temperatura.

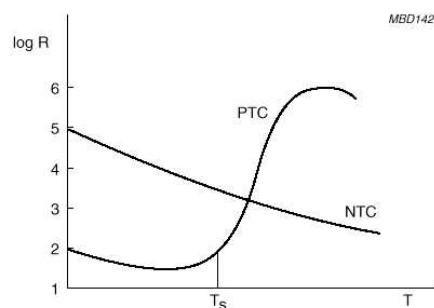


Figura. 6.7. Resistencia vs Temperatura Termistor PTC y NTC

- **Ecuaciones de conversión**

Las ecuaciones que se detallan a continuación son las que se utilizan para convertir los datos digitales a unidades de ingeniería.

$$T(K) = \frac{1}{a + b \times \ln(R_{thr}) + c \times [\ln(R_{thr})]^3} \quad \text{Ecuación 7.7}$$

donde,

$$a = 0.00130705$$

$$b = 0.000214381$$

$$c = 0.000000093$$

Ecuación de Resistencia

$$R_{thr} = R1 \times (ADC_FS - ADC) / ADC \quad \text{Ecuación 7.8}$$

$$R1 = 10 \text{ K}\Omega$$

$$ADC_FS = 1023$$

ADC = Dato digital convertido a decimal.

$$T(^{\circ}\text{C}) = T(K) - 273.15$$

6.3.3 Equipo Necesario

1. Estación Base MIB520CB.
2. Nodo 1: MPR2400.
3. Sensores MTS310CB.
4. Un par de baterías AA.
5. Computador Personal con drivers de estación base instalados y software de adquisición de datos Croosbow.vi (ver manual de usuario).
6. Secadora de cabello con distintos niveles de temperatura.
7. Termómetro de mercurio.

6.3.4 Desarrollo de la Práctica

1. Conectar correctamente la estación base (MIB520CB) al computador.
2. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
3. Identificar el NODO 1 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
4. Conectar las baterías AA al nodo.
5. Encender el tablero NODO-SENSORES.
6. Abrir y ejecutar la aplicación Crossbow.vi
7. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
8. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido Figura 6.2).
9. Ubique el sensor MTS310CB en una posición estable y coloque junto a él, el termómetro, y complete la siguiente Tabla.

TEMPERATURA	[° C]
Interfaz	
Termómetro	

Tabla. 6.7. Resultados 1 – Practica 3

10. En la pestaña **Temperatura** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará. (Figura 6.3)
11. Encienda la secadora de cabello calibrada a la menor temperatura que está disponga y acerque al sensor y termómetro a 5cm de distancia.
12. Presione el botón **START** en la ventana de temperatura.
13. Una vez finalizada la adquisición en la ventana resultados se desplegaran los valores máximos y mínimos adquiridos durante el proceso de muestreo, anote estos resultados y la medición del termómetro en la Tabla 6.8.
14. Repita los pasos 11 al 13 para otros niveles de temperatura de la secadora de cabello. *Debe generarse un archivo para cada muestra.*

Temperatura	Max (°C)	Min (°C)	Termómetro (°C)
T1			
T2			
T3			
T4			

Tabla. 6.8. Resultados 2 – Práctica 3

15. De acuerdo a los resultados de la Tabla 6.8, en la pestaña **Alarmas** configure los valores de acuerdo a su criterio de tal manera que realice dos adquisiciones largas (100 a 200 muestras) en las que varíe la temperatura de la secadora de cabello sobre el sensor, una adquisición de menor a mayor temperatura y otra adquisición de mayor a menor, y se activen todas las alarmas.

Debe generarse un archivo para cada adquisición.

ALARMA	TEMPERATURA [° C]
Muy Baja	
Baja	
Alta	
Muy Alta	

Tabla. 6.9. Resultados 3 – Práctica 3

16. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el grafico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione *Export Simplified Image* y posteriormente en *Save to File* seleccione la ubicación del archivo.

6.3.5 Cuestionario

1. ¿Cuáles son las características de los termistores?

2. ¿Enumere 5 aplicaciones en las cuales son utilizados los termistores?

3. Enumere 3 dispositivos que permitan sensor temperatura.

4. Compare a los termistores con los dispositivos de la pregunta 3.

5. Utilizando la ecuación de conversión determine el valor de la resistencia del termistor para los siguientes valores de temperatura.

$$T = 10 \text{ } ^\circ \text{ C}$$

$$T = 15 \text{ } ^\circ \text{ C}$$

$$T = 20 \text{ } ^\circ \text{ C}$$

$$T = 25 \text{ } ^\circ \text{ C}$$

6. Utilizando la ecuación de resistencia obtenga el valor de la palabra digital (BIN) para los valores de resistencia obtenidos en el paso anterior.

$$T = 10 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

$$T = 15 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

$$T = 20 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

$$T = 25 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

6.3.6 Gráficas y Análisis de resultados

1. Utilizando Excel o Matlab realice las gráficas TEMPERATURA vs TIEMPO, para cada uno de los archivos adquiridos.
2. Realice la gráfica TEMPERATURA vs DATO DIGITAL, para los primeros pasos obtenidos en el paso 15 del desarrollo.
3. Realice una curva de histéresis con los datos obtenidos en el paso 14.
4. Analice los resultados de la Tabla 6.8.
5. Con los datos del paso 15 complete la siguiente Tabla.

Medición	Promedio T [°C]	Promedio U.C	Resistencia [KΩ]	Palabra Digital (BIN)
Menor a Mayor				
Mayor a Menor				

Tabla. 6.10. Cálculos 1 – Practica 3

6. Con los datos obtenidos en el paso 5, obtenga el tiempo que se demora el proceso en activar las alarmas, y los intervalos de las muestras de este tiempo.
7. Inserte y analice la grafica obtenida en el paso 16.

6.3.7 Conclusiones y Recomendaciones

6.3.8 Bibliografía

6.4 PRACTICA 4: MOVIMIENTO

TEMA: MEDICIÓN DE LA VIBRACIÓN EN TELÉFONOS CELULARES

6.4.1 Objetivos

Analizar el efecto que tiene el dispositivo de vibración en los teléfonos celulares.

Determinar y diferenciar la vibración que producen distintos teléfonos celulares.

Conocer las características de los dispositivos detectores de vibraciones.

6.4.2 Marco Teórico

- **Acelerómetro**

El acelerómetro es un transductor versátil, basado en el principio de que cuando se comprime un retículo cristalino piezoeléctrico, se produce una carga eléctrica proporcional a la fuerza aplicada.

Generalmente los elementos piezoeléctricos son de circonato de plomo y se encuentran comprimidos por una masa, sujeta al otro lado por un muelle y todo el conjunto dentro de una caja metálica.

El acelerómetro sometido a vibración, el disco piezoeléctrico se ve sometido a una fuerza variable, proporcional a la aceleración de la masa. Son capaces de medir la aceleración en uno, dos o tres ejes.

Actualmente los acelerómetros son utilizados en teléfonos celulares para aplicaciones de medición de distancias recorridas y shake control en teléfonos Sony Ericsson, aplicaciones en iPhone y en controles de Nintendo Wii.

El acelerómetro que se encuentra en el tablero de sensores MTS310 de dos ejes es una superficie MEMS (Micro Electro Mechanic System) tiene una gran capacidad de detección y una resolución de 10 bits.

Las características de este sensor se describen en la Tabla 6.11.

Acelerómetro de dos ejes	
Rango	$\pm 2 \text{ g}$ ($1 \text{ g} = 9.81 \text{ m/s}^2$)
Ancho de Banda	DC-50 Hz (controlado)
Resolución	2 mG (0.002 G) RMS
Sensibilidad	167 mV/G $\pm 17 \%$
Offset	2.5 V $\pm 0.4 \text{ V}$

Tabla. 6.11. Características Acelerómetro del MTS310

- **Ecuaciones de Conversión**

La ecuación de conversión utilizada

$$\text{ACC} = 1 - (\text{CP1} - \text{ADC}) / \text{ADC_FS} \text{ (Gravedad Relativa)}$$

donde,

$$\text{CP1} = 500$$

$$\text{CN1} = 400$$

$$\text{ADC_FS} = (\text{CP1} - \text{CN1}) / 2$$

ADC = Dato digital en decimal.

$$\text{ACC}(\text{m/s}^2) = \text{ACC} \times 9.81$$

Ecuación 7.9

6.4.3 Equipo Necesario

1. Estación Base MIB520CB.
2. Nodo 1: MPR2400.
3. Sensores MTS310CB.
4. Un par de baterías AA.
5. Computador Personal con drivers de estación base instalados y software de adquisición de datos Croosbow.vi (ver manual de usuario).
6. Cuatro teléfonos celulares distintos.

6.4.4 Desarrollo de la Práctica

1. Conectar correctamente la estación base (MIB520CB) al computador.
2. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
3. Identificar el NODO 1 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
4. Conectar las baterías AA al nodo.
5. Encender el tablero NODO-SENSORES.
6. Abrir y ejecutar la aplicación Crossbow.vi
7. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
8. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido Figura 6.2).
9. Ubique un teléfono celular en modo vibración sobre una superficie plana, y sobre esté el tablero de sensores.
10. En la ventana acelerómetro anote los valores del sensor en reposo en la siguiente tabla.

Vibración en X	Vibración en Y

Tabla. 6.12. Resultados 1 – Práctica 4

11. En la pestaña **Acelerómetro** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará. (Figura 6.3)
12. Con la ayuda de otro teléfono celular, realice una llamada al celular que se encuentra bajo el tablero de sensores y presione el botón **START** de la ventana acelerómetro.
13. Una vez finalizada la aplicación en la pestaña **Resultados** se desplegarán los datos máximos y mínimos obtenidos en la adquisición, anote estos resultados en la Tabla 6.13.
14. Repita los pasos 11 al 13 con otros teléfonos celulares.
Debe generarse un archivo para cada muestreo.

15. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el grafico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione **Export Simplified Image** y posteriormente en **Save to File** seleccione la ubicación del archivo.

Vibración	Max (m/s ²)		Min (m/s ²)	
	X	Y	X	Y
Celular 1				
Celular 2				
Celular 3				
Celular 4				

Tabla. 6.13. Resultados 2 – Práctica 1

6.4.5 Cuestionario

1. ¿Cuáles son las características de los acelerómetros?

2. Enumere 4 aplicaciones en las cuales son utilizados acelerómetros.

3. Utilizando la ecuación de conversión determine la palabra digital (BIN) para los siguientes valores:

$$5 \text{ m/s}^2$$

$$9.81 \text{ m/s}^2$$

$$12 \text{ m/s}^2$$

6.4.6 Graficas y Análisis de Resultados

1. Utilizando Excel o Matlab realice las gráficas VIBRACIÓN x,y vs TIEMPO, para cada uno de los archivos adquiridos.
2. Realice una gráfica PROMEDIO DE LAS MUESTRAS vs TELÉFONO y analice el resultado obtenido.
3. Analice los resultados de la Tabla 6.13.
4. Inserte y analice la gráfica obtenida en el paso 15 del desarrollo de la práctica.

6.4.7 Conclusiones y Recomendaciones

6.4.8 Bibliografía

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

Mediante las diferentes pruebas y análisis realizados durante la elaboración y finalización del proyecto se han generado varias conclusiones sobre la manipulación de los elementos de una WSN y la programación en LabVIEW, las mismas que se detallan a continuación:

- ✓ Se desarrolló el software de adquisición de datos utilizando LabVIEW para la Red de Sensores Inalámbricos Crossbow, que permite la realización de prácticas de laboratorio para Instrumentación y Sensores.
- ✓ Del análisis del hardware de la Red de Sensores Inalámbricos se determinó que los sensores de Temperatura y Luz comparten el mismo canal de conversión.
- ✓ Del estudio realizado sobre la trama de datos serial se determinó que posee una longitud máxima de 39 bytes, dentro de la cual la información de cada magnitud física viene dada en grupos de 2 bytes; y para obtener el valor medido del estado de la batería, temperatura, luz, sonido o vibración en unidades crudas estos bytes deben cambiar sus ubicaciones.
- ✓ Se determinó que la comunicación entre la estación base y el computador tiene una tasa de transferencia de 57600 bps, y el puerto COM virtual para la comunicación es el n+1 instalado con el driver FTDI FT2232C.
- ✓ Se realizó una aplicación para medir Temperatura en el rango de 0 °C a 50 °C, con una precisión de 0,01 °C, con la capacidad de control del número de muestras a ser adquiridas, y simulación para cuatro niveles de alarmas, utilizando LabVIEW.

- ✓ Se realizó una aplicación para medir la variación de tensión debido al cambio de resistencia producida por la variación de luz incidente en la fotoresistencia del tablero de sensores MTS310CB del nodo dos, con la capacidad de configuración de alarmas y simulación visual de acciones de control.
- ✓ Se desarrolló una aplicación en LabVIEW para realizar adquisiciones para el detector de tono y acelerómetro del tablero de sensores MTS310CB, con la capacidad de exportación de información a un archivo de datos y una representación gráfica del proceso de medición.
- ✓ Se desarrollaron guías de laboratorio para las magnitudes físicas tratadas utilizando las aplicaciones en LabVIEW creadas, programadas para una duración de tres horas de clase.
- ✓ Se elaboró un manual de usuario para la programación y utilización de los elementos que conforman la Red de Sensores inalámbricos Crossbow.

7.2 RECOMENDACIONES

- ✓ Para la realización de futuros proyectos se recomienda una etapa de familiarización con los equipos que conforman la Red de Sensores Inalámbricos, mediante la lectura, análisis y comprensión de las hojas técnicas de cada uno de los elementos y su manipulación en funcionamiento.
- ✓ Antes de utilizar la aplicación de LabVIEW y la red de sensores para realizar las prácticas de laboratorio es recomendable leer el manual de usuario de este proyecto, principalmente para la programación de los nodos y la estación base en las que un mal acoplamiento en la programación podrían causar daños en los equipos así como también no se debe programar los nodos con las baterías colocadas.
- ✓ Al momento de acoplar del tablero MTS310 al nodo MPR2400 verificar que el switch de encendido y apagado del nodo este en la posición OFF, no retirar el tablero MTS310 con el nodo encendido, y no acoplar o retirar el nodo MPR2400 en la estación base MIB520 cuando esta esté conectado al puerto USB del computador.

- ✓ Por problemas de compatibilidad con algunas herramientas de LabVIEW no se recomienda la utilización del proyecto con el sistema operativo Windows Vista.
- ✓ Para la realización de las prácticas de laboratorio antes de proceder a adquirir datos verificar que el software está recibiendo información de la Red mediante la visualización del LED indicador de sistema LISTO, u observando la respuesta del sensor en cada una de las ventanas cuando se modifique la magnitud física medida.
- ✓ Escribir correctamente la dirección en la cual se va a generar los archivos de datos, para que al momento de la adquisición la aplicación no se detenga por este error, teniendo que repetir nuevamente el proceso.
- ✓ Para evitar que el sistema reciba únicamente la información de un solo nodo, comprobar que el nivel de voltaje de la batería de cada uno de ellos sea el mismo o que máximo difiera en 0.05 [V].

REFERENCIAS BIBLIOGRÁFICAS

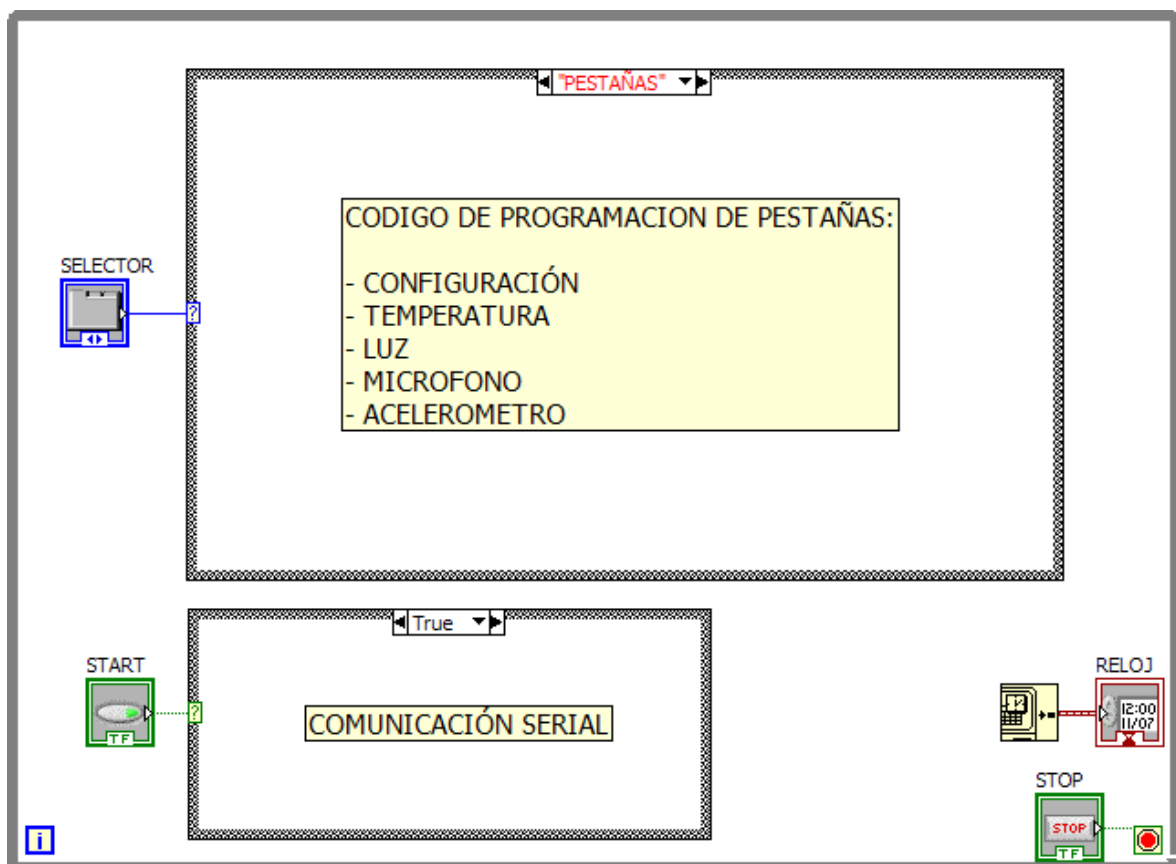
- [1] ORTIZ, Hugo, *Instrumentación y Sistemas de Control*, 1998, Págs. 4-5.
- [2] BOLTON, Bill, *Mediciones y pruebas eléctricas y electrónicas*, segunda edición, Editorial Alfaomega, México 1996, Pág. 293.
- [3] La tecnología ZigBee ha sido ratificada como estándar del IEEE,
<http://www.casadomo.com/noticiasDetalle.aspx?id=zigbee>
- [4] MENDIBURO, Henry, *Instrumentación Virtual Industrial (PDF)*.
- [5] ULLOA, José, *Estudio y Análisis de las características de TinyOS*.
<http://alumnos.elo.utfsm.cl/~julloa/titulo/download/UnoSeccionDos.doc>
- [6] CROSSBOW, *MPR-MIB Users Manual (PDF)*, Revisión B, Junio 2006.
- [7] CROSSBOW, *Xserve Users Manual (PDF)*, Revisión B, Abril 2006.
- [8] NATIONAL INSTRUMENTS, *National Instruments VISA*, <http://ni.com/visa>.
- [9] MOHAMMAD, Ilyas y IMAD, Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, Clearence Center, EEUU 2005, Págs. 23-26.
- [10] NATIONAL INSTRUMENTS, *La Instrumentación Virtual (PDF)*.
- [11] NATIONAL INSTRUMENTS, *Instrumentos Virtuales (PDF)*.
- [12] *Definiciones básicas sobre instrumentación*.
http://weblogs.madrimasd.org/medida_informacion_escepticismo/archive/2007/04/21/64086.aspx
- [13] *Exactitud y Trazabilidad*, <http://www.catlab.com.ar>
- [14] SANCHEZ, Lourdes, *Teoría de errores*.
<http://luda.uam.mx/curso2/tema1/errores.html>

- [15] CHACON, Rafael, “La instrumentación virtual en la enseñanza de la Ingeniería Electrónica”, *Acción Pedagógica*, Vol. 11 No. 1, Págs. 74-84, 2002.
- [16] ESCOLAR, Soledad, *Wireless Sensor Networks (PDF)*.
- [17] LEWIS, F, *Wireless Sensor Networks (PDF)*, <http://www.arri.uta.edu/acs>.
- [18] MORENO, Ignacio, *Sistemas de Adquisición de Datos (PDF)*.
- [19] LESTER, Jason, *System Architecture for Wireless Sensor Networks*, University of California, 2003.
- [20] NATIONAL INSTRUMENTS, *10 razones para agregar LabVIEW a su PLC*.
<http://zone.ni.com/devzone/cda/tut/p/id/6329#toc0>
- [21] ROCANCIO, Henry CIFUENTES, Hector, *Tutorial de LabVIEW*, Universidad Distrital Francisco José de Caldas, 2001.
- [22] NATIONAL INSTRUMENTS, *Introducción a LabVIEW (PDF)*. Septiembre 2003.
- [23] *Termistores: consideraciones mecánicas y eléctricas*.
http://www.sapiensman.com/medicion_de_temperatura/termistores.htm
- [24] *Sensores de Temperatura*.
<http://proton.ucting.udg.mx/temas/control/memo/termistores.html>
- [25] *Las Notas musicales*. <http://www.musicagospel.com.ar/blog/?p=14>
- [26] *Acelerómetros MEMS para aplicaciones de producción*.
<http://www.suframa.gov.br/minapim/news/visArtigo.cfm?Ident=66&Lang=ES>
- [27] *Detector de Frecuencia*, <http://www.x-robotics.com/hardware.htm>
- [28] *Fotoresistencias*, <http://www.endrich.com/es/site.php/303>.
- [29] BONET, Jordi, *Los acelerómetros, una nueva vía de interacción*, Febrero 2008.
<http://es.onsoftware.com/p/programas-acelerometros-nokia>

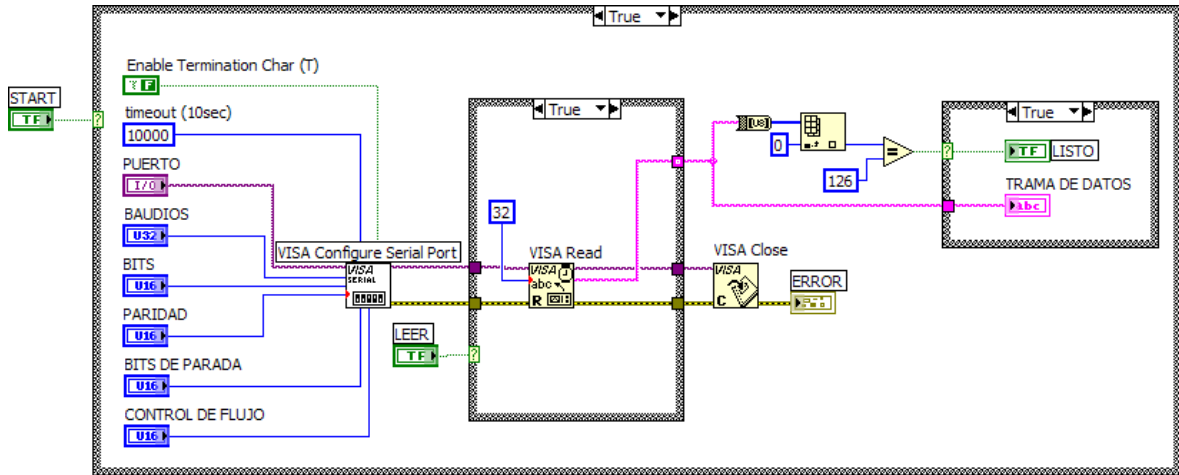
ANEXO 1

CÓDIGO INTERFAZ HMI

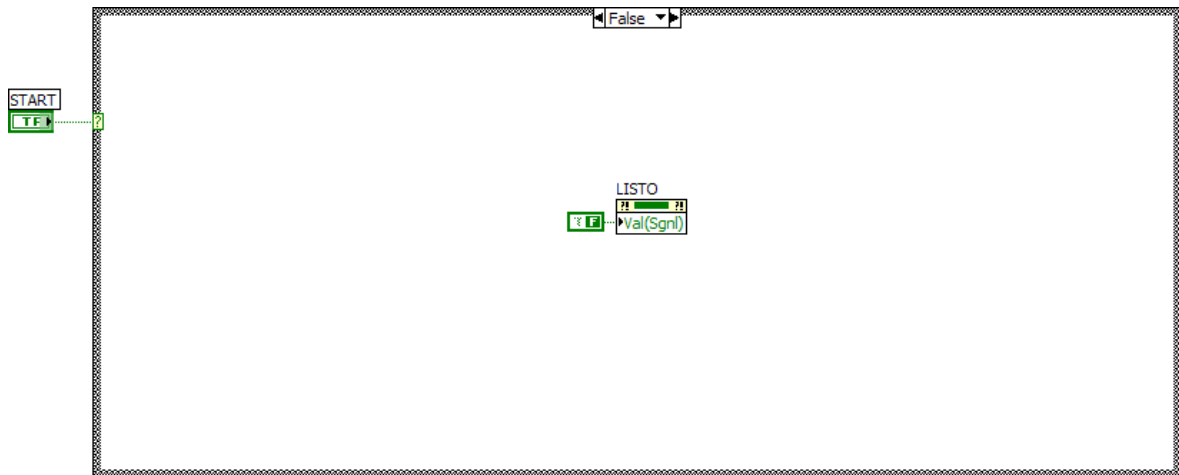
- **Distribución del Código**



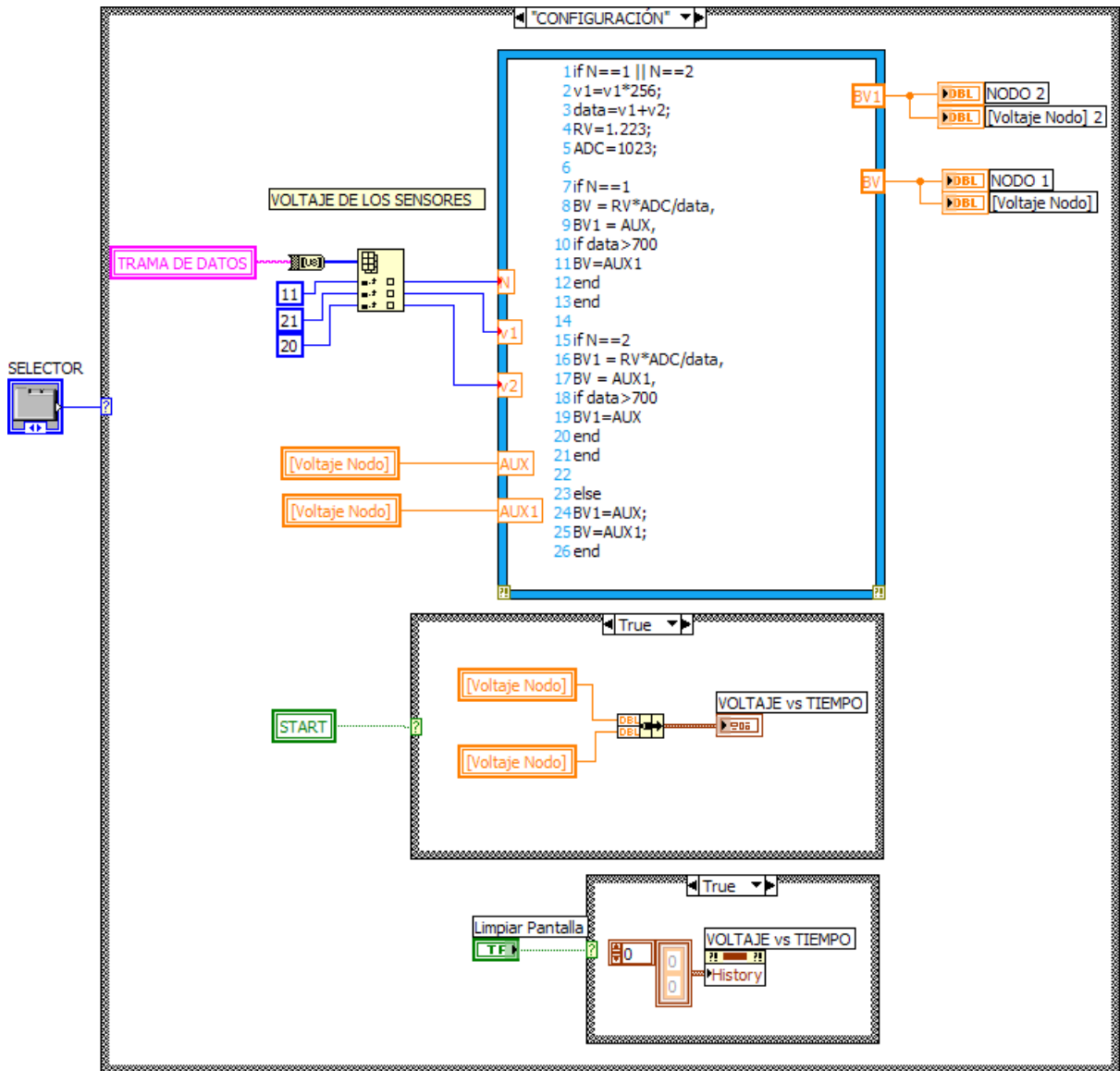
- Comunicación Serial TRUE**



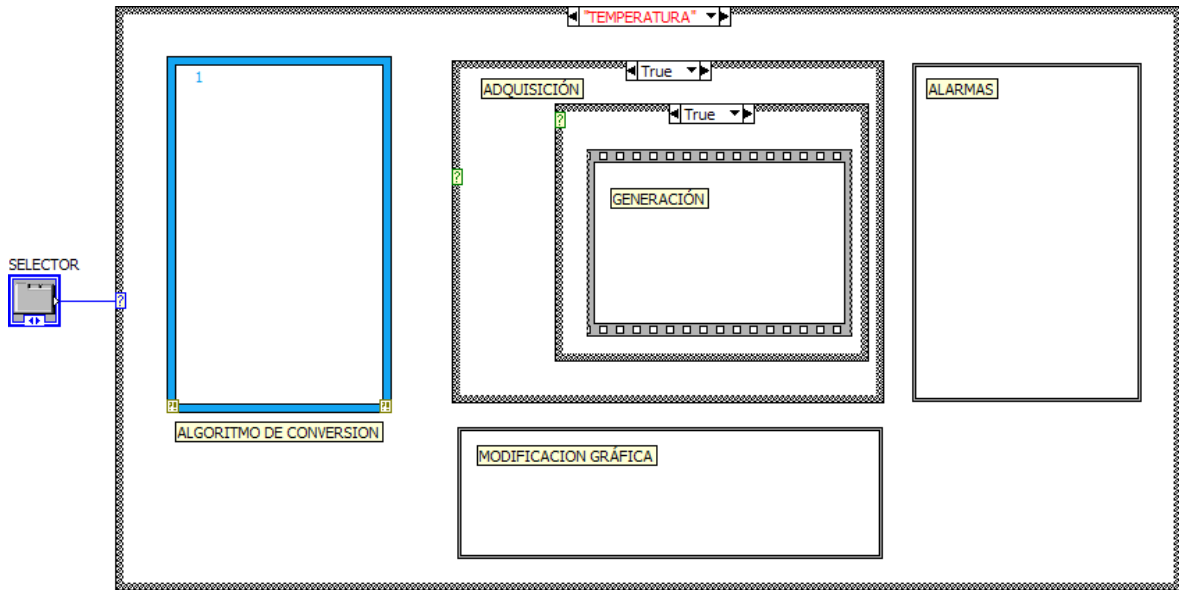
- Comunicación Serial FALSE**



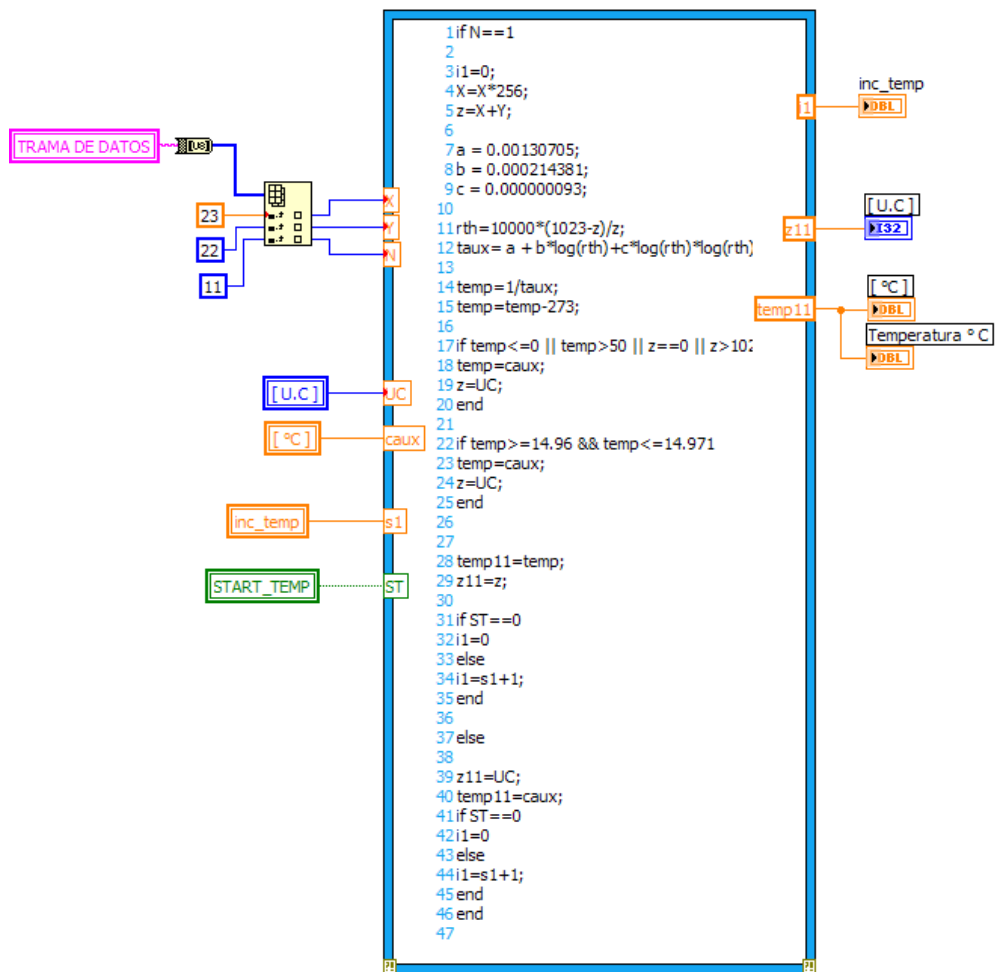
- **Código Pestaña Configuración**



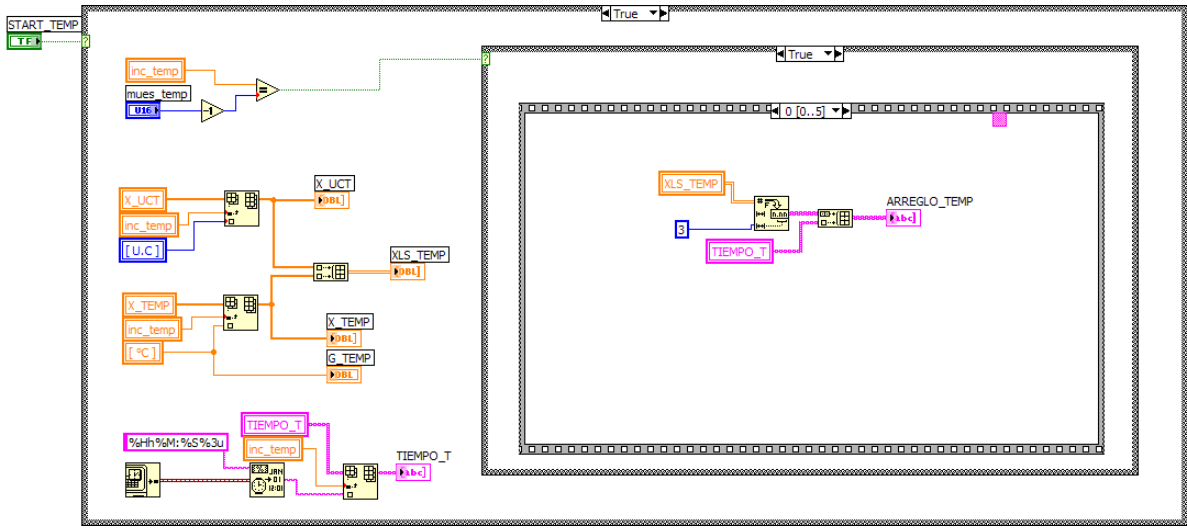
➤ **DISTRIBUCIÓN CÓDIGO PESTAÑA TEMPERATURA**



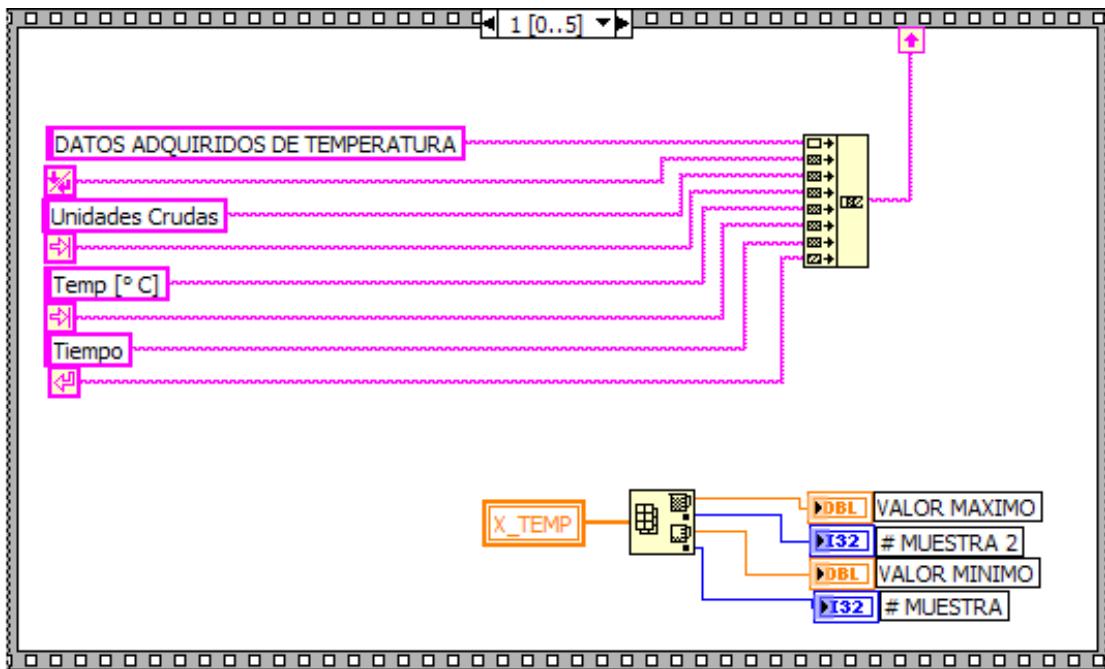
• **Algoritmo de conversión de Temperatura**



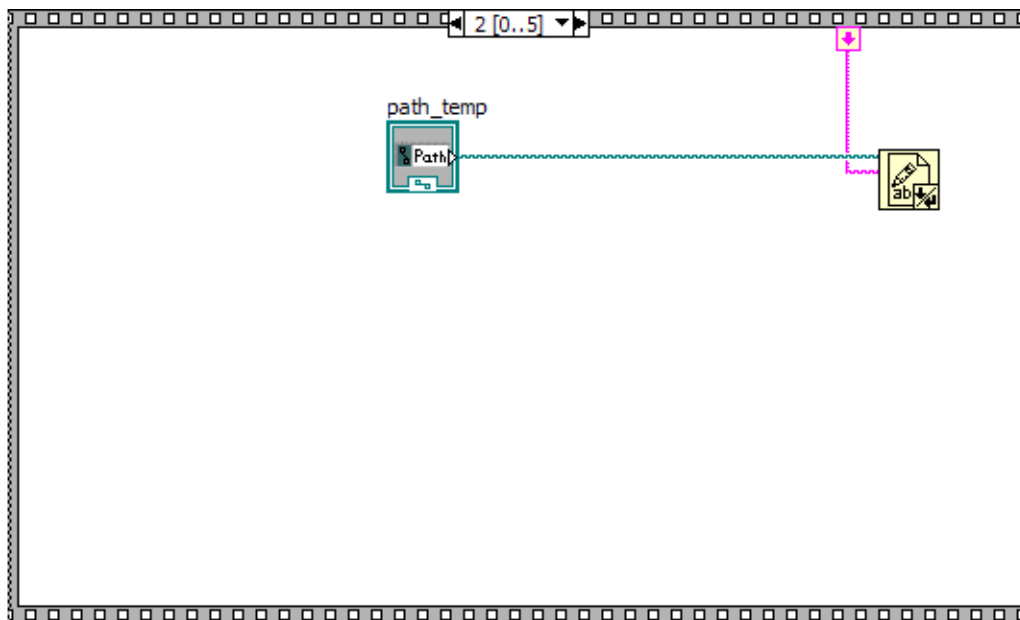
- **Código Adquisición Temperatura y Código Generación 0**



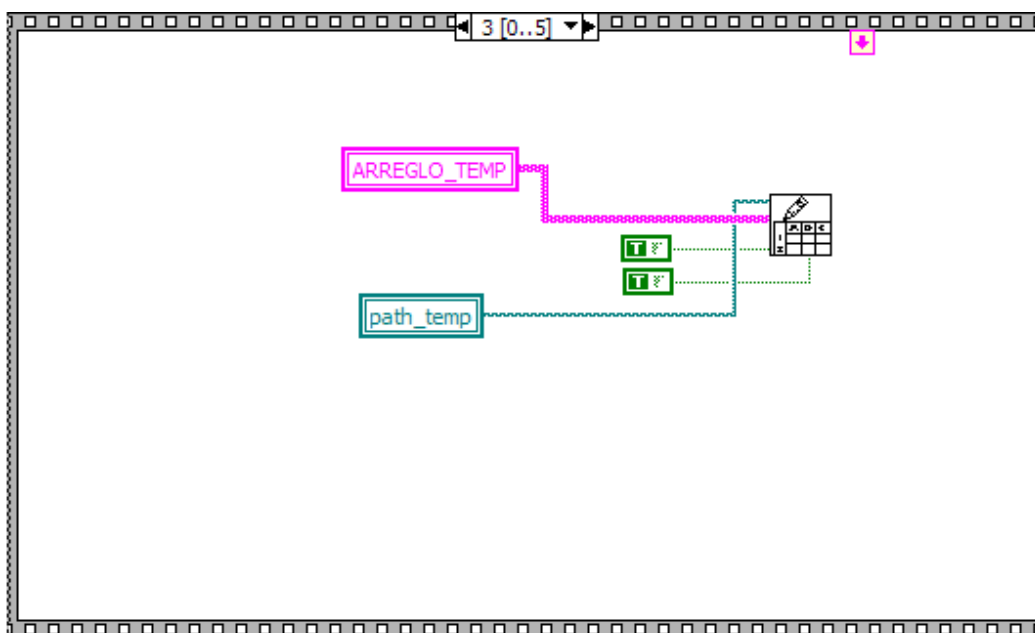
- **Código Generación Temperatura 1**



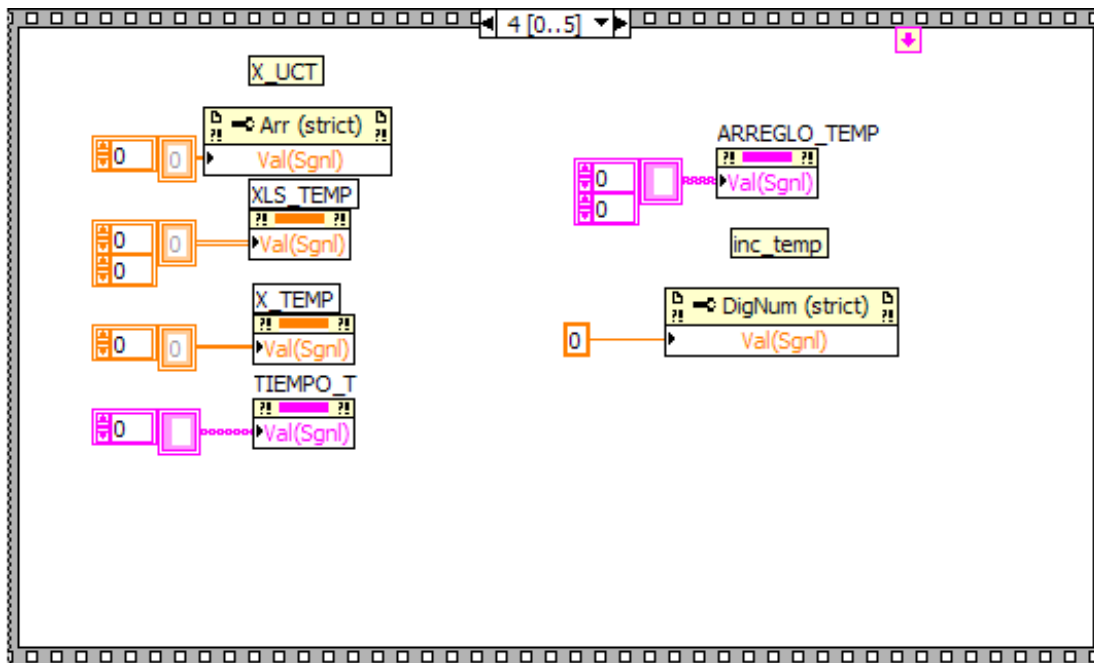
- **Código Generación Temperatura 2**



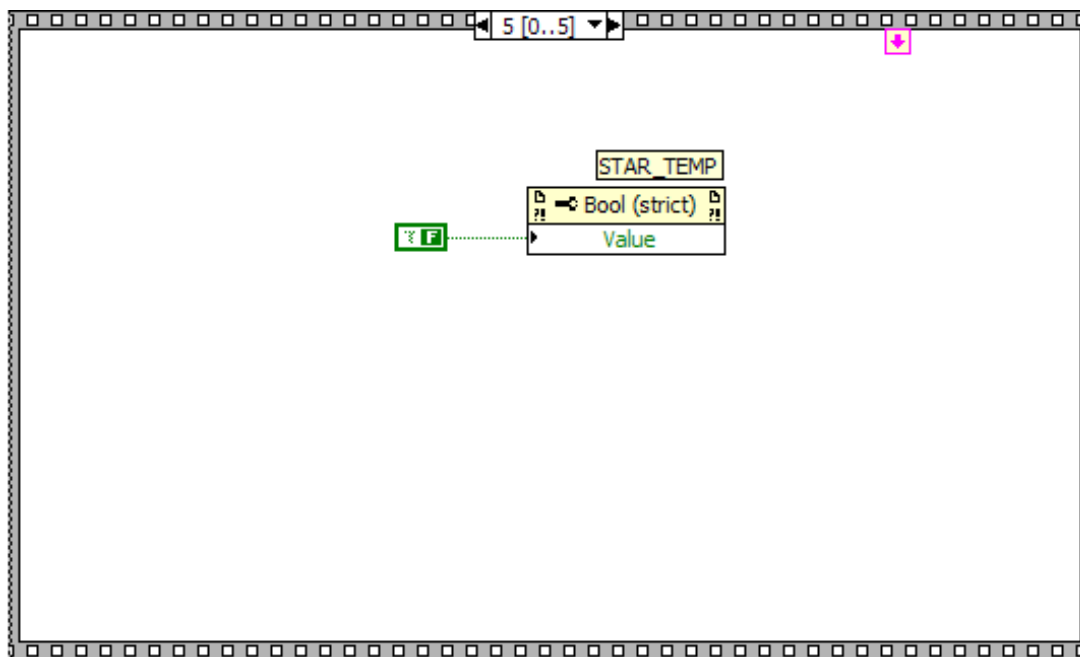
- **Código Generación Temperatura 3**



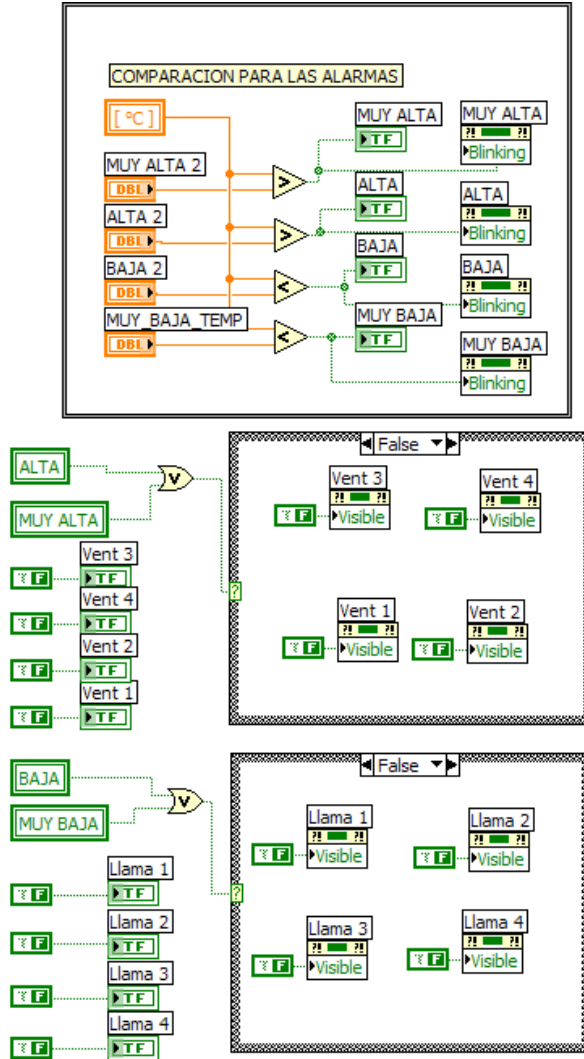
- **Código Generación Temperatura 3**



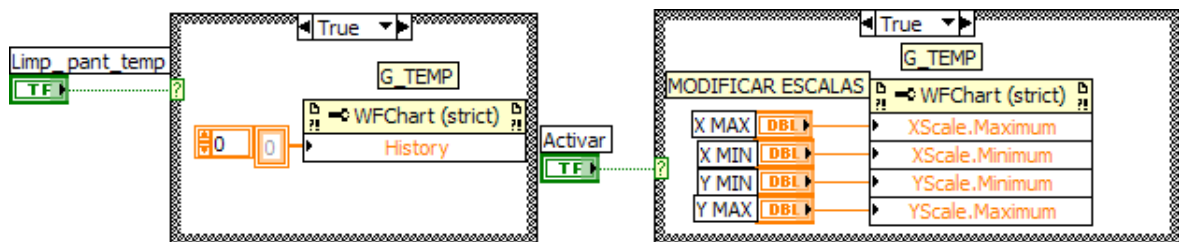
- **Código Generación Temperatura 4**



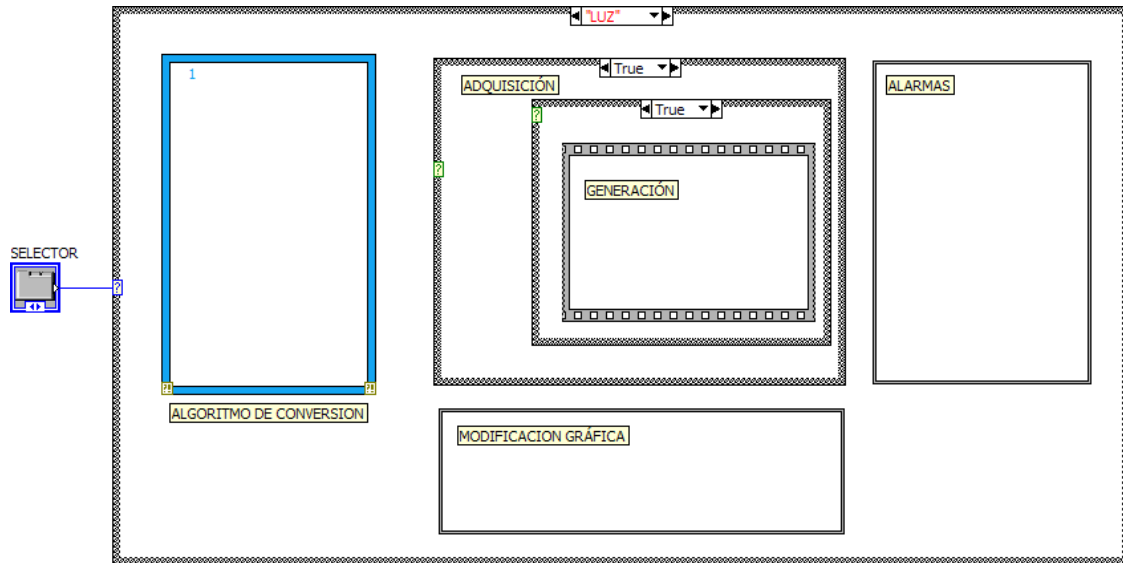
- **Código Alarmas Temperatura**



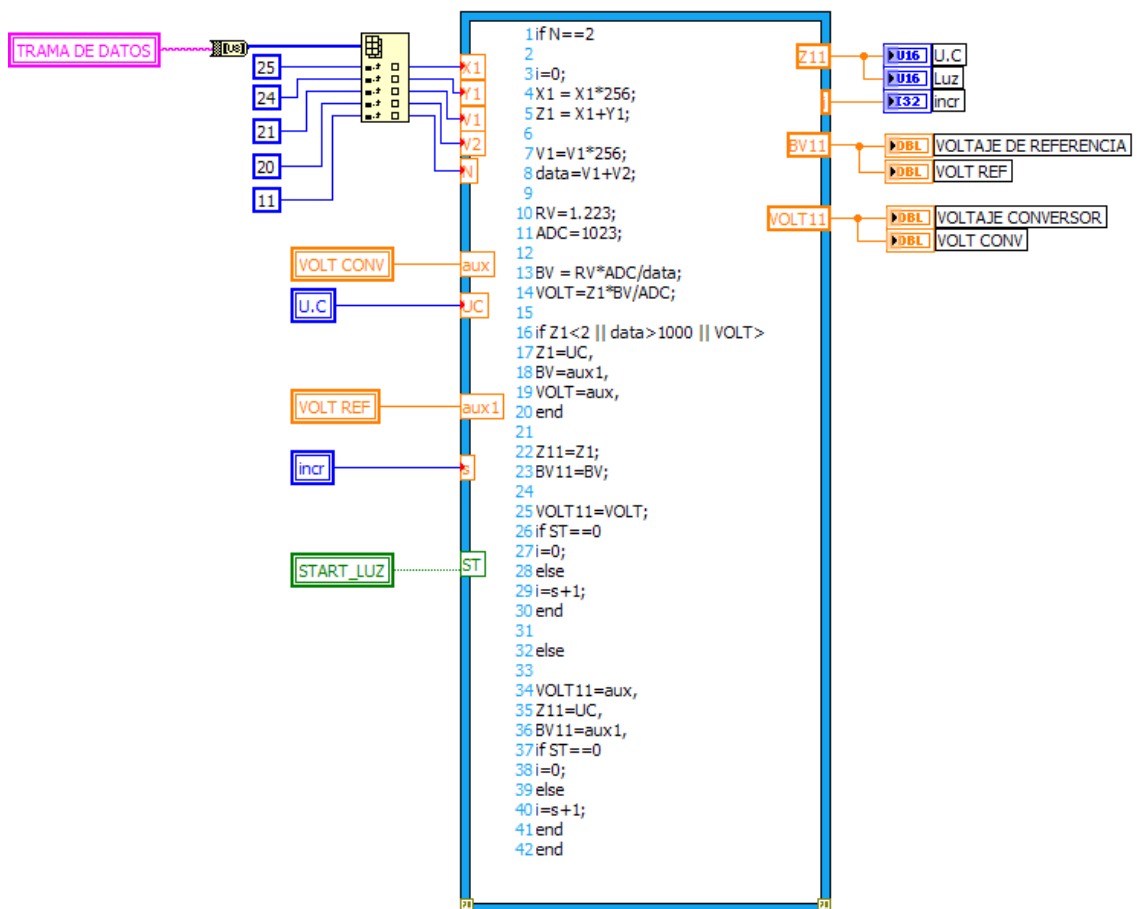
- **Código Modificación Gráfica Temperatura**



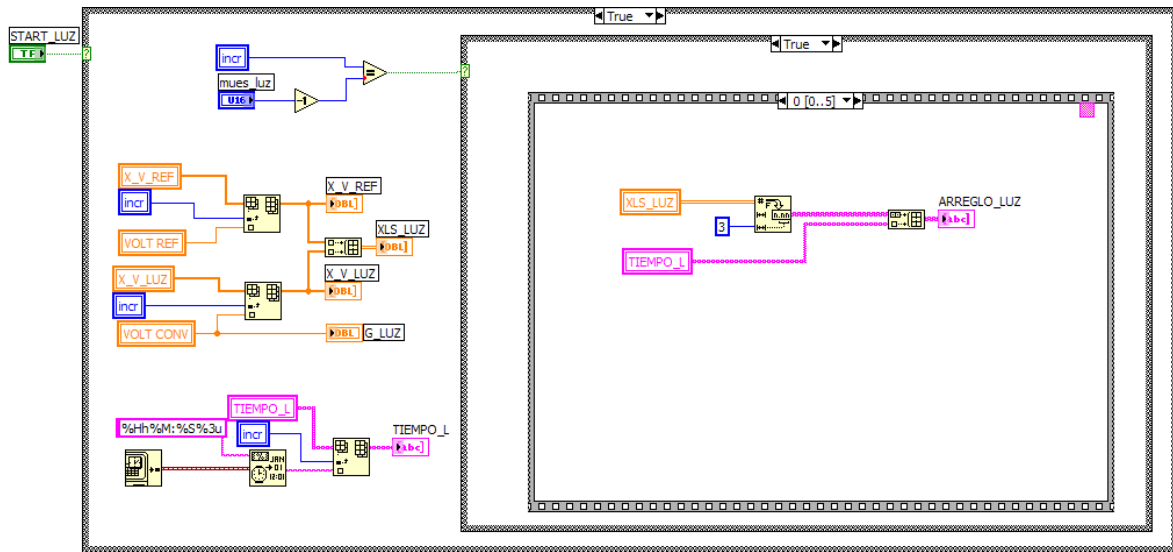
➤ **DISTRIBUCIÓN CÓDIGO PESTAÑA LUZ**



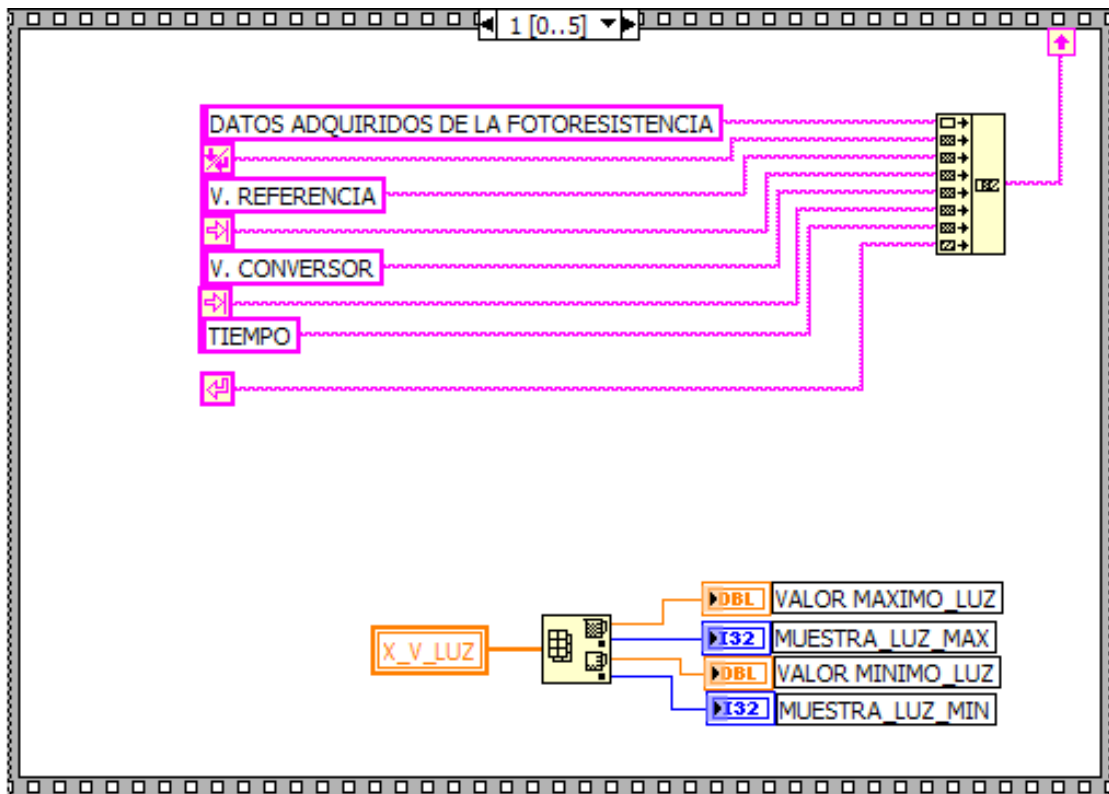
• **Algoritmo de Conversión de Luz**



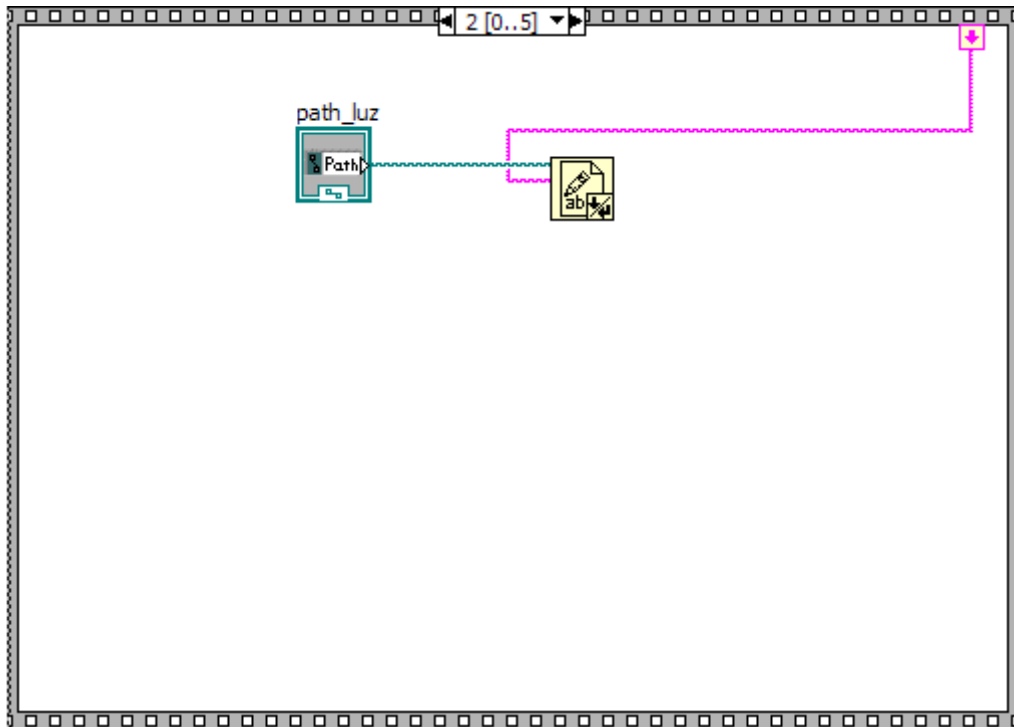
- **Código Adquisición Luz y Código Generación 0**



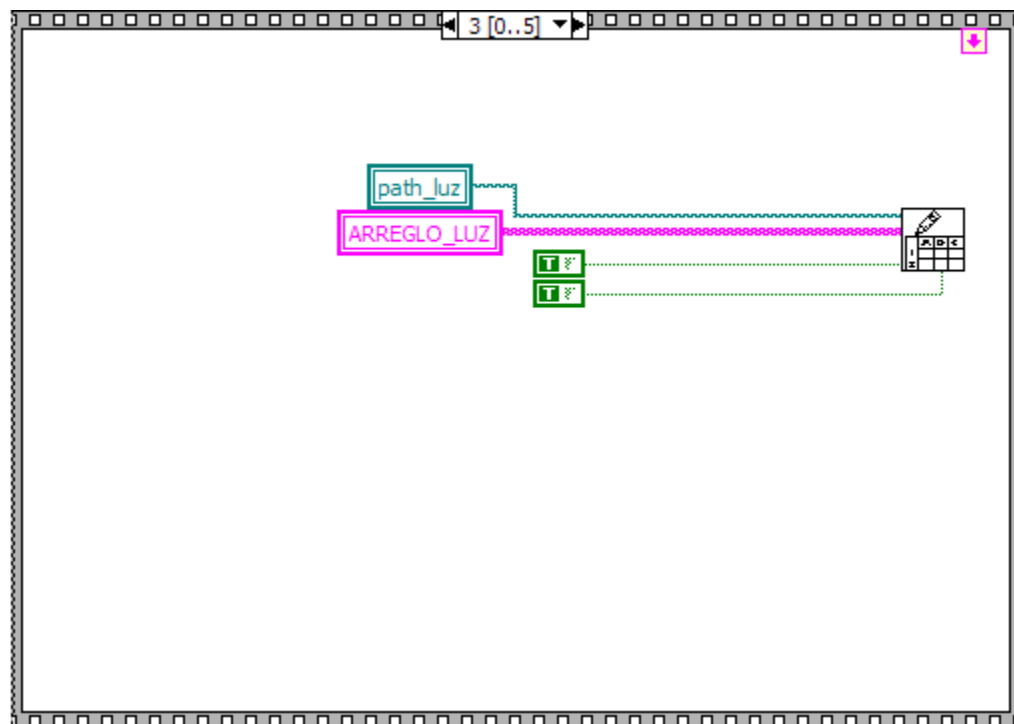
- **Código Generación Luz 1**



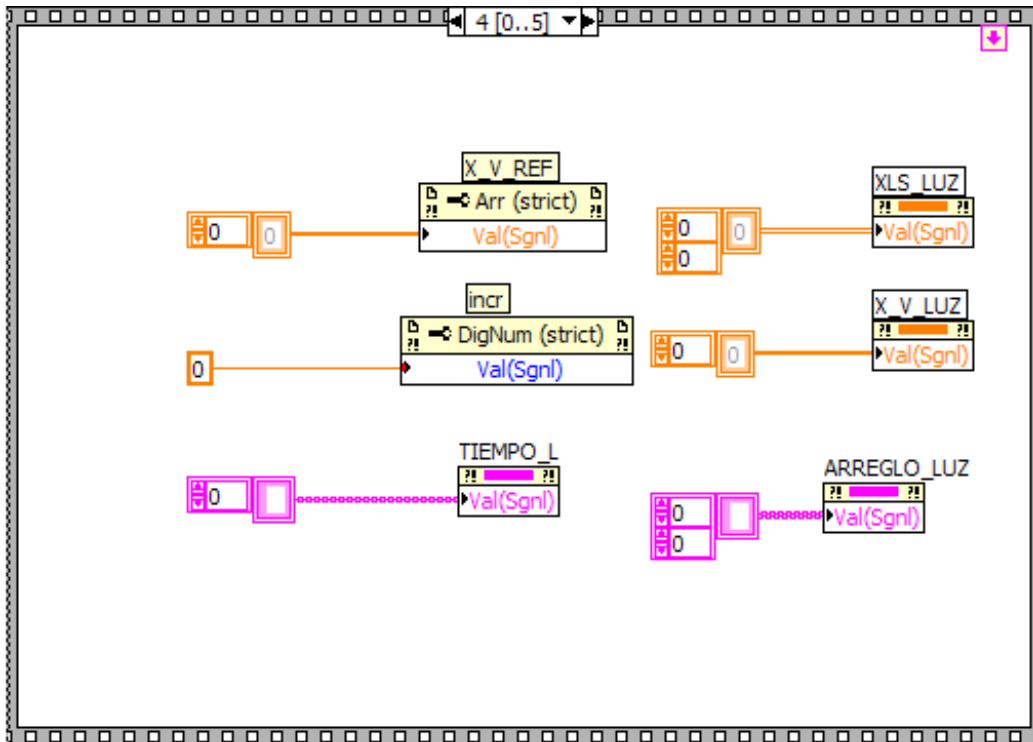
- **Código Generación Luz 2**



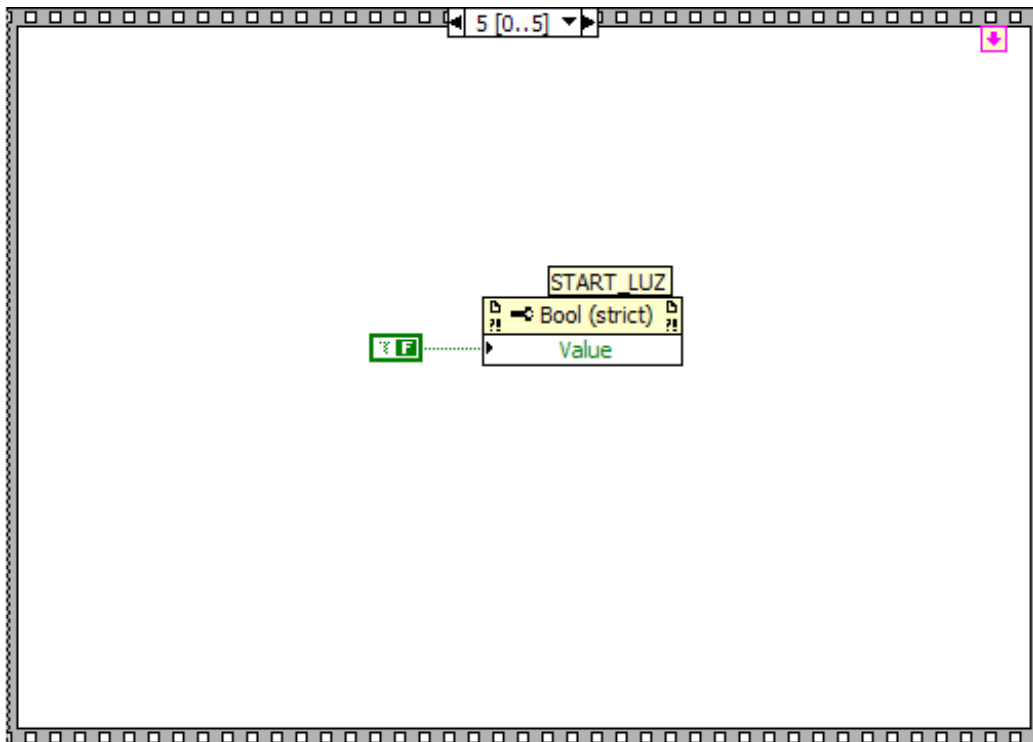
- **Código Generación Luz 3**



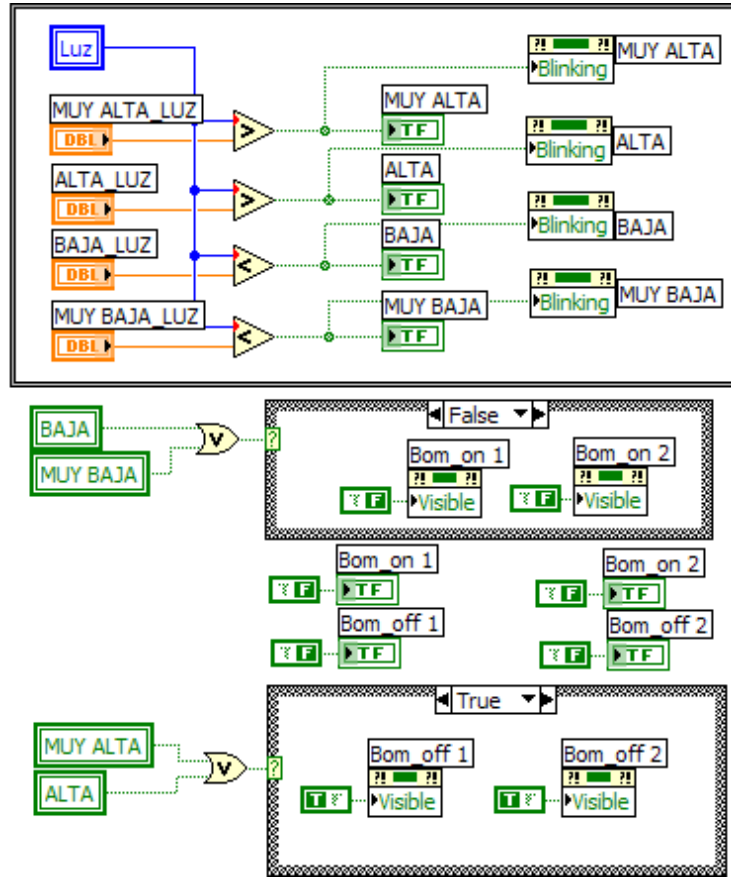
- **Código Generación Luz 4**



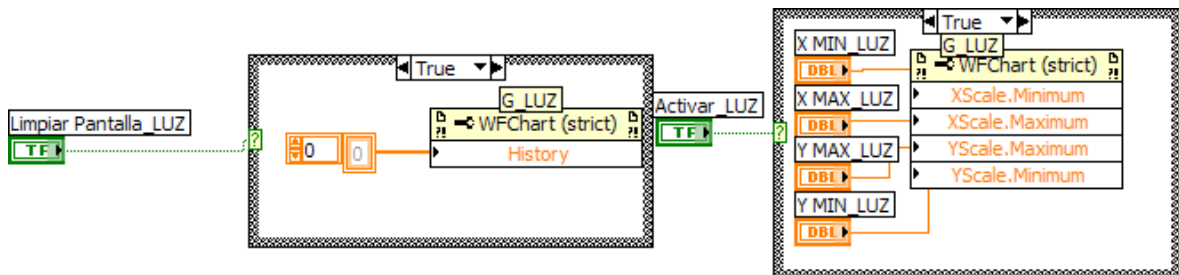
- **Código Generación Luz 5**



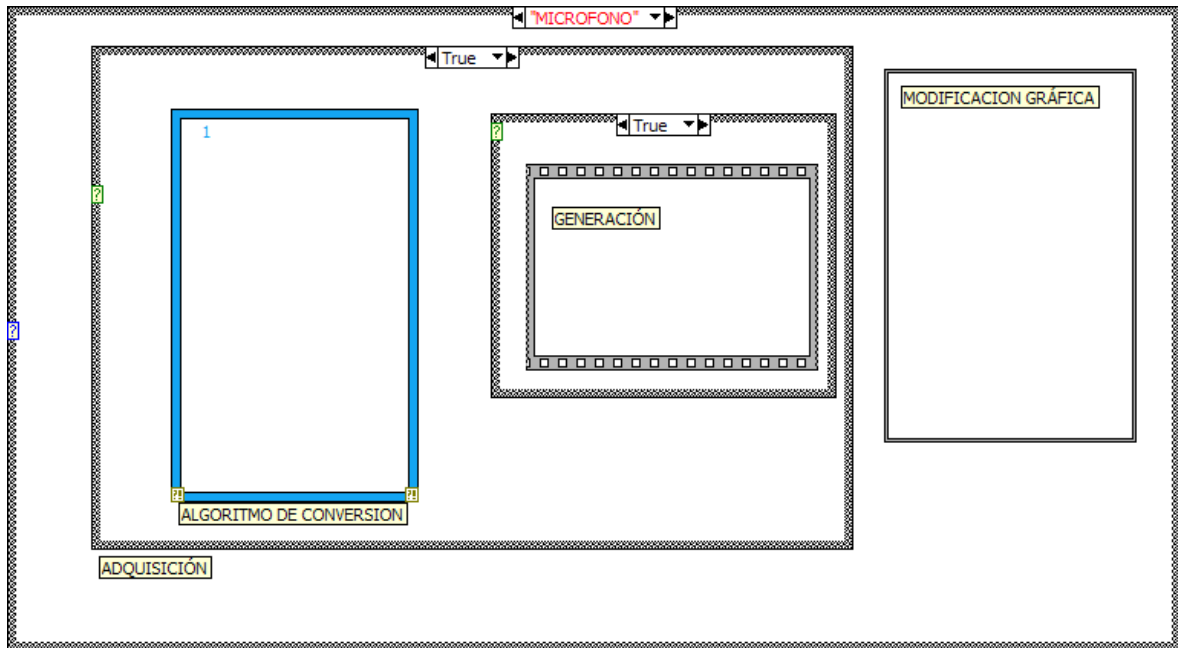
• Código Alarmas Luz



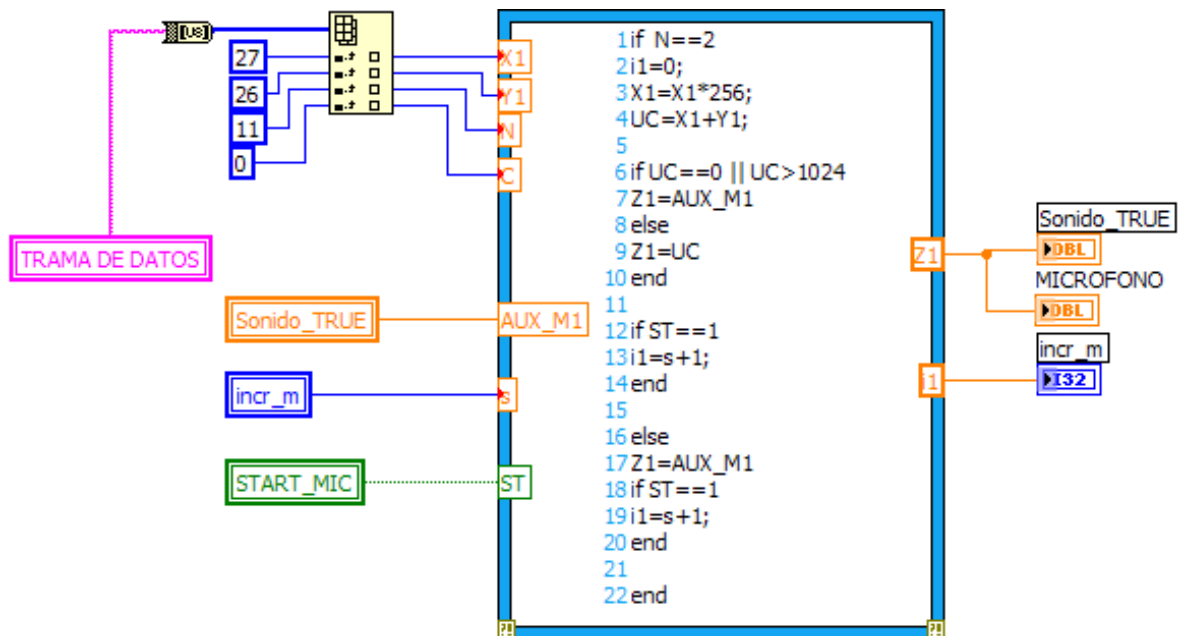
• Código Modificación Gráficas Luz



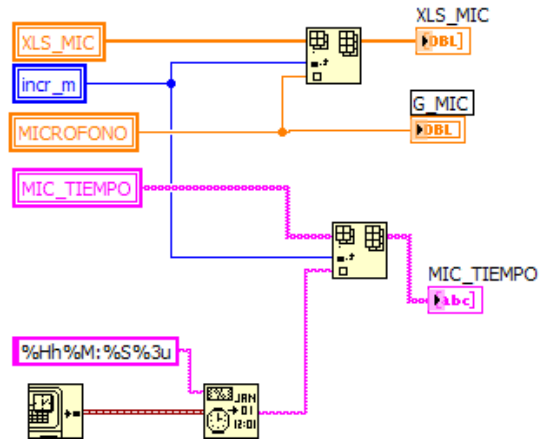
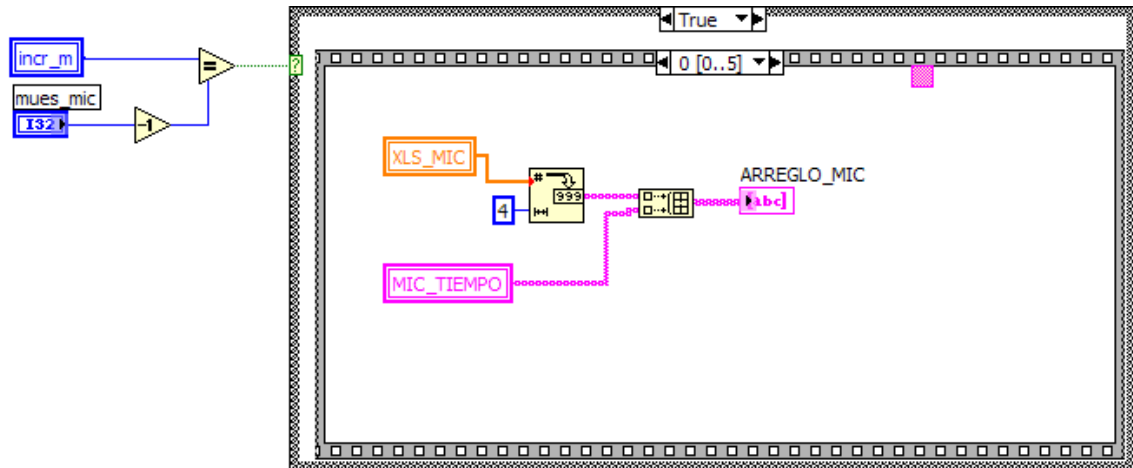
➤ **DISTRIBUCIÓN DE CÓDIGO PESTAÑA MICRÓFONO**



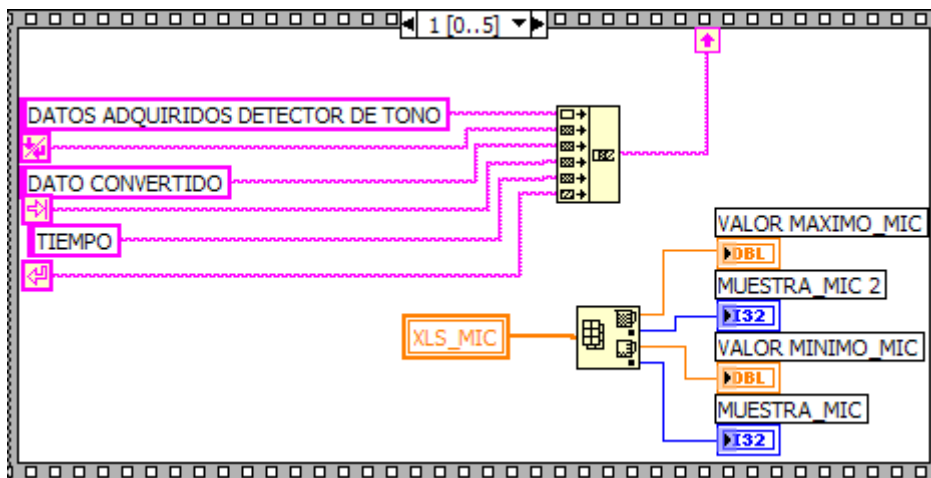
• **Código Algoritmo de conversión micrófono**



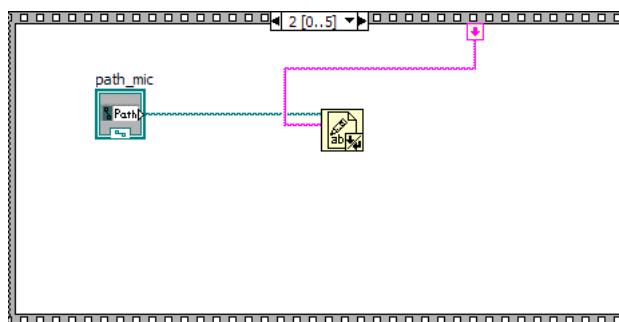
- **Código Adquisición Micrófono y Código Generación 0**



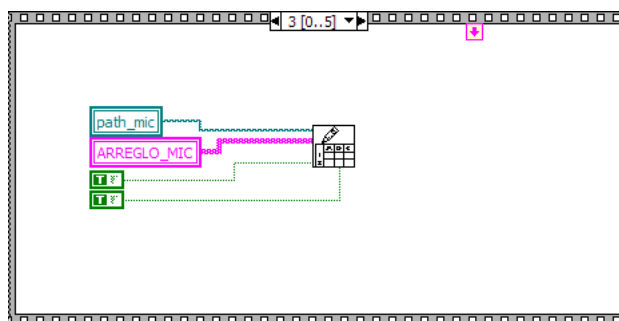
- **Código Generación Luz 1**



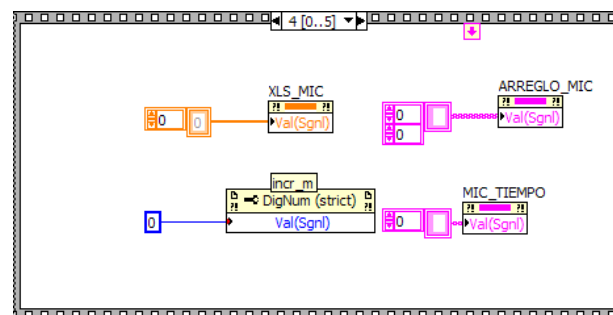
- **Código Generación Luz 2**



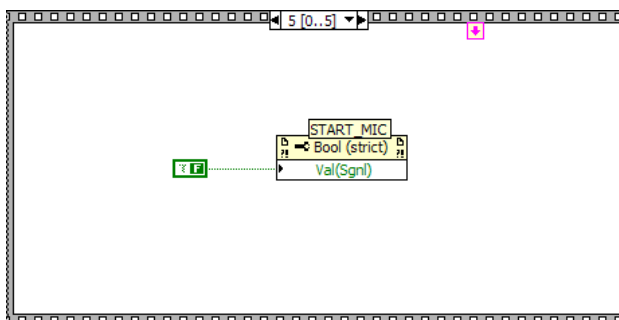
- **Código Generación Luz 3**



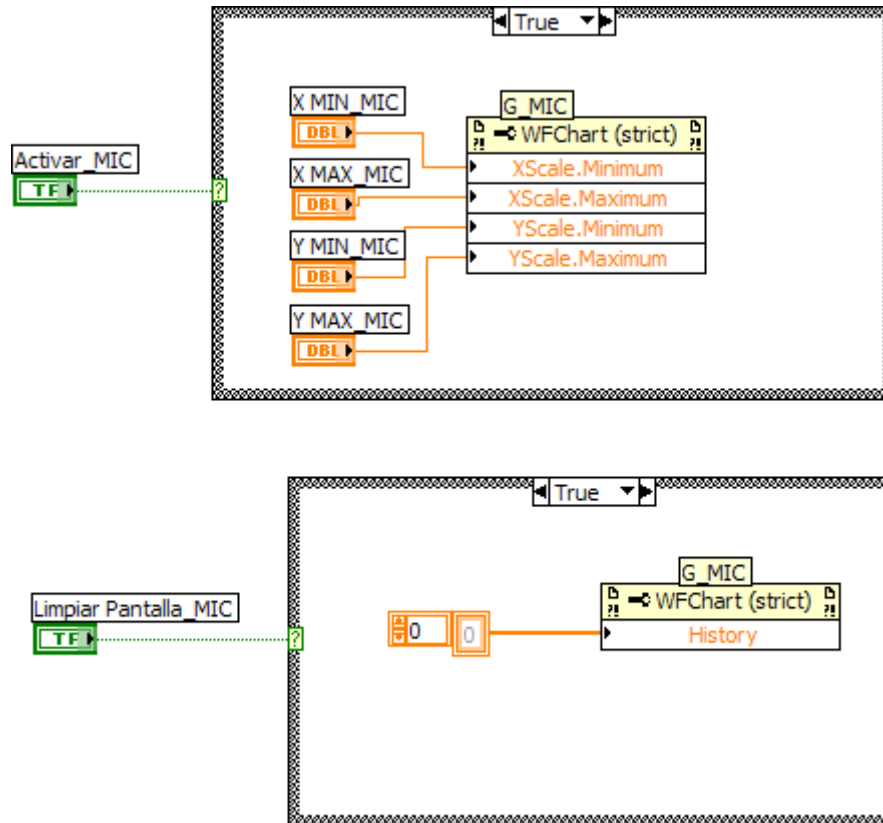
- **Código Generación Luz 4**



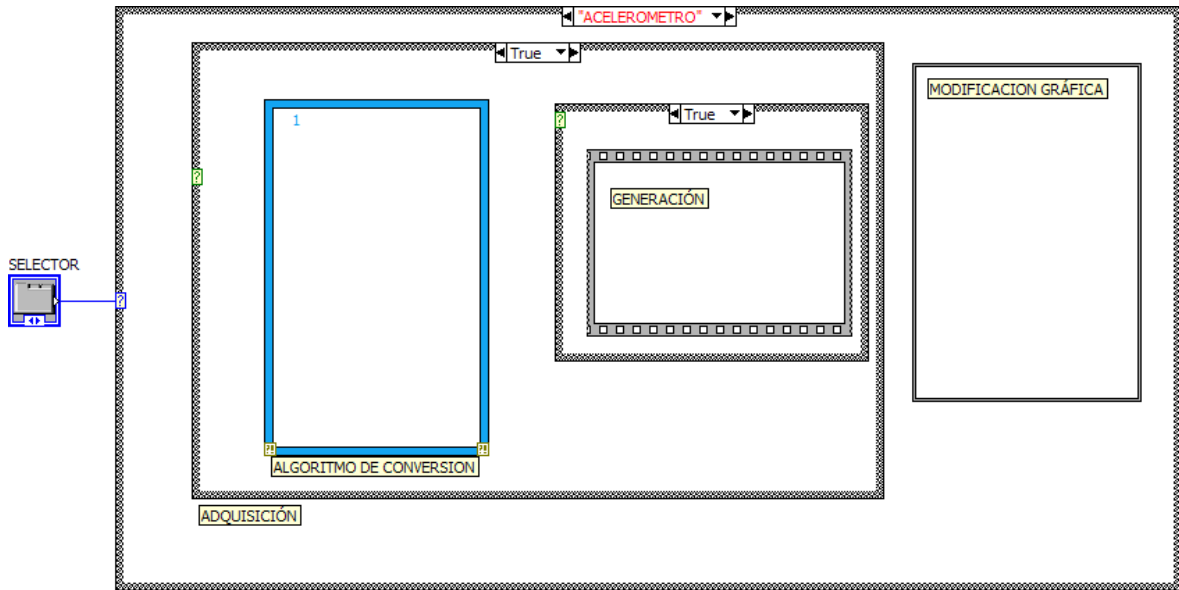
- **Código Generación Luz 5**



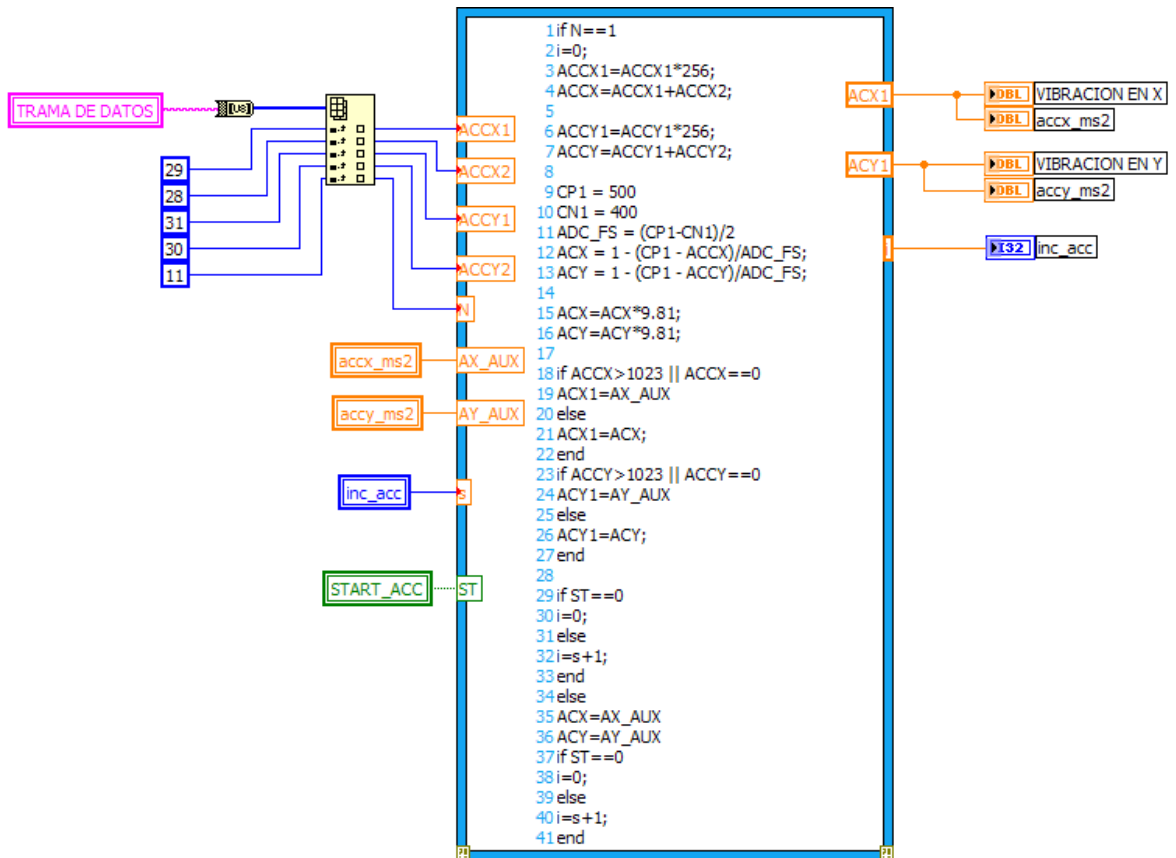
- **Código Modificación Gráfica Luz**



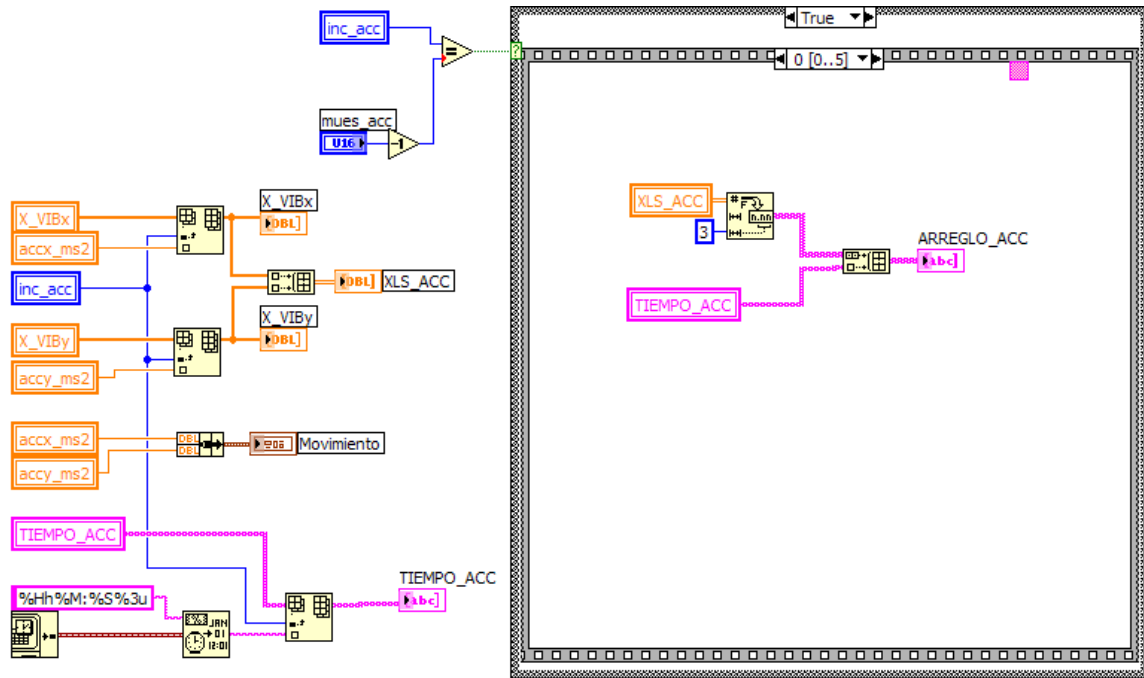
➤ **DISTRIBUCIÓN DE CÓDIGO PESTAÑA ACELERÓMETRO**



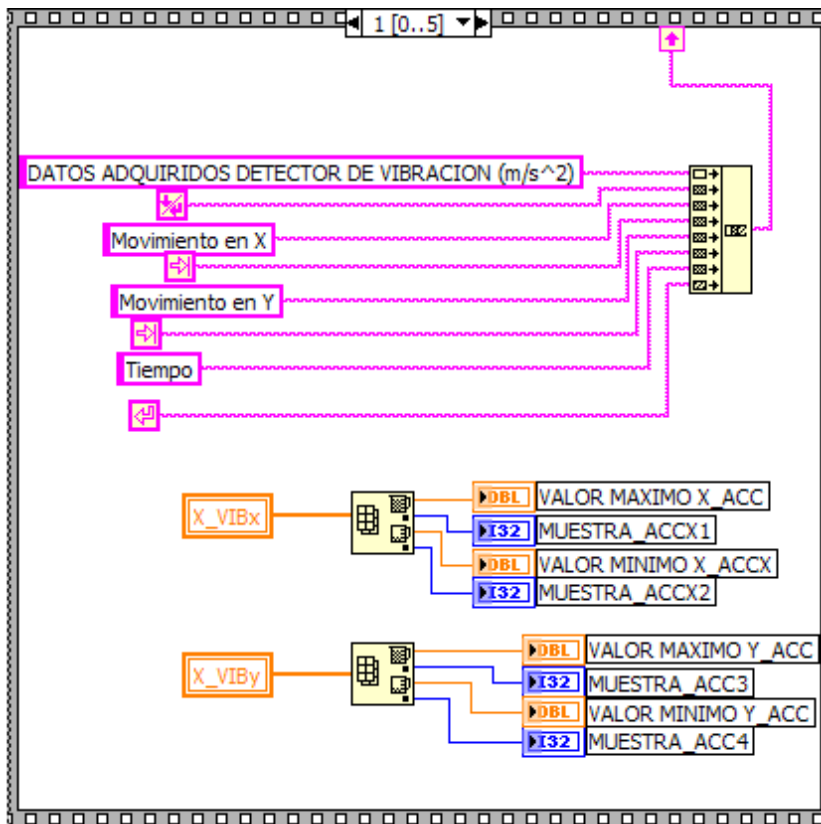
• **Código Algoritmo de Conversión acelerómetro**



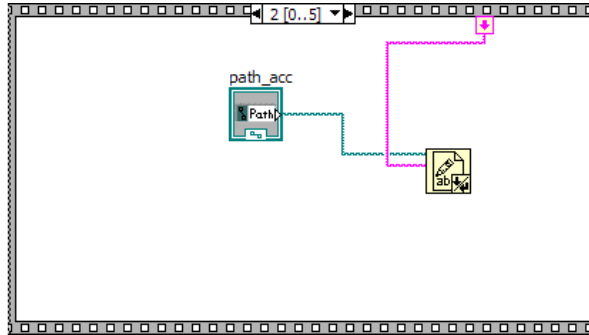
- **Código Adquisición Acelerómetro y Código Generación 0**



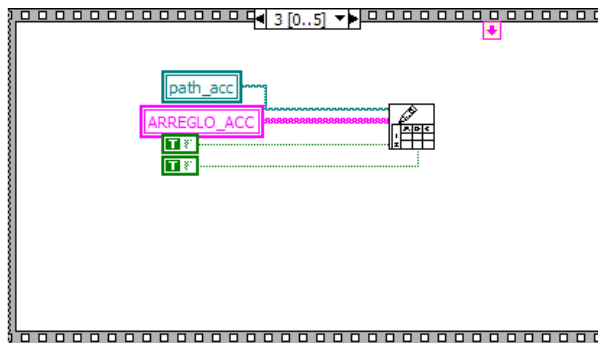
- **Código Generación Acelerómetro 1**



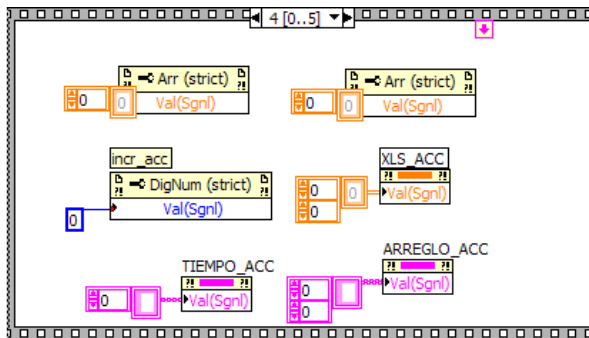
- **Código Generación Acelerómetro 2**



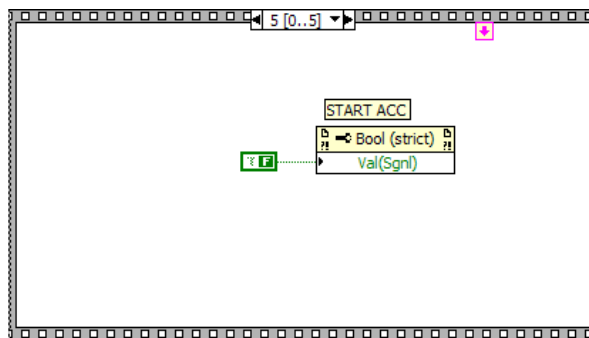
- **Código Generación Acelerómetro 3**



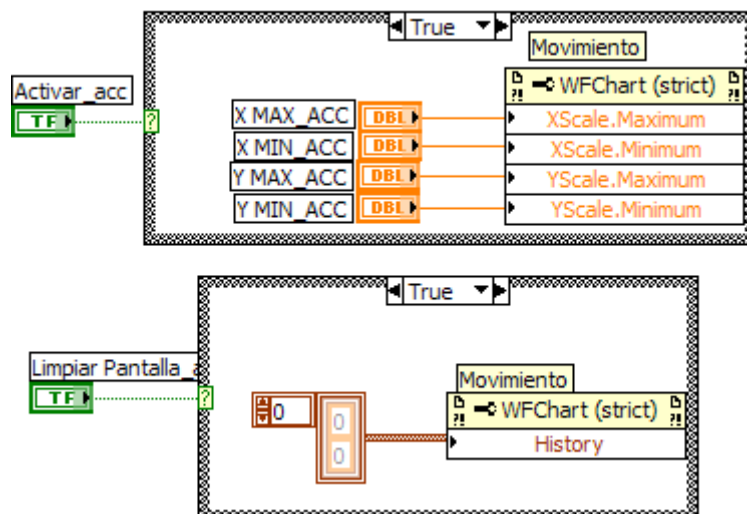
- **Código Generación Acelerómetro 4**



- **Código Generación Acelerómetro 5**



- **Código Modificación Gráfica Acelerómetro**



ANEXO 2

CÓDIGO DE PROGRAMACIÓN NODOS Y ESTACIÓN BASE

➤ PROGRAMACIÓN NODOS PARA SENSORES MTS310CB

• XMTS310M.NC

```
#include "appFeatures.h"
//includes XCommand;
includes sensorboard;
module XMTS310M {
    provides {
        interface StdControl;
    }
    uses {

        interface MhopSend as Send;           // setea funciones de networking tipo malla para RF
        interface RouteControl;              //interface de envio de tramas
                                              //interface de control y ruta de las tramas

        interface XCommand;                  //interfaz de comandos
        interface XEEControl;                //interfaz de control de la red tipo malla
        // interface ReceiveMsg as Bcast;     //interfaz de recepcion de tramas como broadcast

// Bateria
        interface ADC as ADCBATT;            //definicion de la interface para baterias
        interface StdControl as BattControl; // control de baterias
//Temp
        interface StdControl as TempControl; //definicion interfaz de control de temperatura
        interface ADC as Temperature;       //interfaz de recepcion Analoga de temperatura
//Light
        interface StdControl as PhotoControl; //definicion de la interfaz de control de luz
        interface ADC as Light;              //adquiere datos del ADC para la luz
// Mic
        interface StdControl as MicControl;  //definicion function microfono
        interface Mic;                       //interface para microfono
        interface ADC as MicADC;             // obtiene la informacion del ADC para el microfono
// Sounder
        interface StdControl as Sounder;    //función de control para sonido
// Accel
        interface StdControl as AccelControl; //función de control de la aceleración
        interface ADC as AccelX;
        interface ADC as AccelY;
// Mag
        interface StdControl as MagControl;
        interface ADC as MagX;
        interface ADC as MagY;

        //interface ADCControl;
        interface Timer;
        interface Leds;
```

```

#if (FEATURE_UART_SEND | TOS_LOCAL_ADDRESS==0) //
#if FEATURE_UART_SEND
    interface SendMsg as SendUART;          //envia por el uart el paquete en casi de tener el permiso
#endif
    command result_t PowerMgrEnable();      //Funci'on de activaci'on de potencia para TX de msj
    command result_t PowerMgrDisable();//
#endif
    command void health_packet(bool enable, uint16_t intv); //2 bytes de dentro de la cabecera informaci'on del
    paquete
    }
}
}
implementation {
    enum { START, BUSY, SOUND_DONE}; //inicio del sensor, prendido, ocupado, y sonido de inicio una
    vez listo

/* should debe ser movido a xmesh/bin una vez realizado a cambio de ---sfmao
    enum {
        MODE_UPSTREAM,
        MODE_UPSTREAM_ACK,
        MODE_DOWNSTREAM,
        MODE_DOWNSTREAM_ACK,
        MODE_ANY2ANY,
        MODE_ONE_HOP_BROADCAST
    };
    enum {
        BASE_STATION_ADDRESS = 0, //reconoce la estacion base como 0
    };
*/
#define MSG_LEN 29
    TOS_Msg gMsgBuffer;          //longitud del mensaje como 29 envia al buffer
    TOS_Msg msg_buf_radio;      //revisa el la frecuencia con la que se envia el msj por el buffer
    TOS_MsgPtr msg_radio;      //frecuencia de transmission de mensajes
    bool sleeping;             // application command state
    norace XDataMsg readings;
    char main_state;
    norace bool sound_state, sending_packet,sensinginssession;
    norace uint8_t miccnt;
    norace uint32_t val;

/*****
* Task to xmit radio message
*
* msg_radio->addr = TOS_BCAST_ADDR;
* msg_radio->type = 0x31;
* msg_radio->length = MSG_LEN;
* msg_radio->group = TOS_AM_GROUP;
*****/
task void send_radio_msg() {
    uint8_t i;
    uint16_t len;
    XDataMsg *data;

    call Leds.yellowOn();
    // Fill the given data buffer.
    data = (XDataMsg*)call Send.getBuffer(msg_radio, &len);

    for (i = 0; i <= sizeof(XDataMsg)-1; i++)
        ((uint8_t*)data)[i] = ((uint8_t*)&readings)[i];

    data->board_id = SENSOR_BOARD_ID;

```

```

    data->packet_id = 1;
    //data->node_id = TOS_LOCAL_ADDRESS;
    data->parent = call RouteControl.getParent();
    data->packet_id = data->packet_id | 0x80;
#if FEATURE_UART_SEND
    if (TOS_LOCAL_ADDRESS != 0) {
        call PowerMgrDisable();
        TOSH_uwait(1000);
        if (call SendUART.send(TOS_UART_ADDR, sizeof(XDataMsg),
                               msg_radio) != SUCCESS)
            {
                atomic sending_packet = FALSE;
                call Leds.yellowOff();
                call PowerMgrEnable();
            }
    }
    else
#endif
    {
        // Send the RF packet!
        if (call Send.send(BASE_STATION_ADDRESS,MODE_UPSTREAM,msg_radio,
                           sizeof(XDataMsg)) != SUCCESS) {
            atomic sending_packet = FALSE;
            call Leds.yellowOff();
        }
    }
    return;
}

static void initialize() {
    atomic {
        sleeping = FALSE;
        main_state = START;
        sound_state = TRUE;
        sending_packet = FALSE;
#ifdef APP_RATE
        timer_rate = XSENSOR_SAMPLE_RATE;
#else
#ifdef USE_LOW_POWER
        timer_rate = XSENSOR_SAMPLE_RATE + ((TOS_LOCAL_ADDRESS%255) << 7);
#else
        timer_rate = XSENSOR_SAMPLE_RATE + ((TOS_LOCAL_ADDRESS%255) << 2);
#endif
#endif
        miccnt=0;
        val=0;
        sensinginsession=FALSE;
    }
}

static void start()
{
    call BattControl.start();
    call Mic.gainAdjust(64); // Set the gain of the microphone. (refer to Mic)
    call MicControl.start();
    call TempControl.start();
    call PhotoControl.start();
#ifdef MTS310
    call AccelControl.start();
    call MagControl.start();
#endif
}

```

```

task void battstop()
{
    call BattControl.stop();
}
task void tempstop()
{
    call TempControl.stop();
}
task void photostop()
{
    call PhotoControl.stop();
}
task void Micstop()
{
    call MicControl.stop();
}
task void Accelstop()
{
    call AccelControl.stop();
}

/*****
* Initialize the component. Initialize ADCControl, Leds
*
*****/
command result_t StdControl.init() {

    atomic msg_radio = &msg_buf_radio;

    // MAKE_BAT_MONITOR_OUTPUT();      // enable voltage ref power pin as output
    // MAKE_ADC_INPUT();                // enable ADC7 as input
    call BattControl.init();
    // usart1 is also connected to external serial flash
    // set usart1 lines to correct state
    // TOSH_MAKE_FLASH_SELECT_OUTPUT();
    TOSH_MAKE_FLASH_OUT_OUTPUT();      //tx output
    TOSH_MAKE_FLASH_CLK_OUTPUT();      //usart clk
    // TOSH_SET_FLASH_SELECT_PIN();
    // Llamada de funciones para control de las variables del sensor
    call Leds.init();
    call TempControl.init();
    call PhotoControl.init();
    call MicControl.init();
    call Mic.muxSel(1);                 // Set the mux so that raw microhpone output is selected
    call Mic.gainAdjust(64);            // Set the gain of the microphone. (refer to Mic)
#ifdef FEATURE_SOUNDER
    call Sounder.init();
#endif

#ifdef MTS310
    call AccelControl.init();
    call MagControl.init();
#endif

#ifdef (TOS_LOCAL_ADDRESS==0)
    call PowerMgrDisable();
#endif
    initialize();
    return SUCCESS;
}

```

```

/*****
* Start the component. Start the clock.
*
*****/
command result_t StdControl.start()
{
    call StdControl.stop();
    call health_packet(TRUE,TOS_HEALTH_UPDATE);
    call Timer.start(TIMER_REPEAT, timer_rate);
    return SUCCESS;
}
/*****
* Stop the component.
*
*****/
command result_t StdControl.stop() {
    call BattControl.stop();
    call TempControl.stop();
    call PhotoControl.stop();
    call MicControl.stop();
#ifdef MTS310
    call AccelControl.stop();
    call MagControl.stop();
#endif
    return SUCCESS;
}
/*****
* Measure Temp, Light, Mic, toggle sounder
*
*****/
event result_t Timer.fired() {
    if (sending_packet)
        return SUCCESS;           //don't overrun buffers
    if(main_state==BUSY)
        return SUCCESS;
    start();
    atomic main_state = BUSY;
    if (!sensinginsession){
        call ADCBATT.getData();
        atomic sensinginsession = TRUE;
    }
    return SUCCESS;
}
/*****
* Battery Ref or thermistor data ready
*****/
async event result_t ADCBATT.dataReady(uint16_t data) {
    if (!sensinginsession) return FAIL;
    readings.vref = 0;                //readings.vref=0; para desactivar sensor
                                    //readings.vref=data; para activar sensor

    atomic sensinginsession = FALSE;
    post battstop();
    //call TempControl.start();
    call Temperature.getData();
    return SUCCESS;
}

/*****

```

```

* Temperature ADC data ready
* Read and get next channel.
*****/
async event result_t Temperature.dataReady(uint16_t data) {
    readings.thermistor = data;                //readings.termistor=0; para desactivar sensor
                                              //readings.termistor=data; para activar sensor

    //call PhotoControl.start();
    call Light.getData();
    return SUCCESS;
}
/*****
* Photocell ADC data ready
* Read and get next channel.
*****/
async event result_t Light.dataReady(uint16_t data) {
    readings.light =0;                        //readings.light=0; para desactivar sensor
                                              //readings.light=data; para activar sensor

    post photostop();
    post tempstop();
    call MicADC.getData();
    return SUCCESS;
}

/*****
* MicroPhone ADC data ready
* Read and toggle sounder.
* send uart packet
*****/
async event result_t MicADC.dataReady(uint16_t data) {
    if(miccnt<50)
    {
        atomic miccnt=miccnt+1;
        TOSH_uwait(1000);
        if(val<data)
        {
            atomic val=0;
            //data;
        }
        call MicADC.getData();
        return SUCCESS;
    }
    else
    {
        atomic miccnt=0;
    }
    readings.mic = val;//data;
    post Micstop();
    val=0;
#ifdef MTS310
    call AccelX.getData();
#else
    // This is the final sensor reading for the MTS300...
    atomic {
        if (!sending_packet) {
            sending_packet = TRUE;
            post send_radio_msg();
        }
    }
#endif
#ifdef FEATURE_SOUNDER
    if (sound_state) call Sounder.start();

```

```

    else call Sounder.stop();
    atomic {
        sound_state = SOUND_STATE_CHANGE;
    }
#endif
#endif
    return SUCCESS;
}
/*****
* ADC data ready
* Read and toggle sounder.
* send uart packet
*****/
async event result_t AccelX.dataReady(uint16_t data) {
    readings.accelX = 0; //readings.=0; para desactivar sensor
                        //readings.=data; para activar sensor

    call AccelY.getData();
    return SUCCESS;
}
/*****
* ADC data ready
* Read and toggle sounder.
* send uart packet
*****/
async event result_t AccelY.dataReady(uint16_t data) {
    readings.accelY = 0; //readings.accelY=0; para desactivar sensor
                        //readings. accelY =data; para activar sensor

    post Accelstop();
    call MagX.getData();
    return SUCCESS;
}

/**
* In response to the <code>MagX.dataReady</code> event, it stores the
* sample and issues command to sample the magnetometer's Y axis.
* (Magnetometer B pin)
* @return returns <code>SUCCESS</code>
*/
async event result_t MagX.dataReady(uint16_t data){
    readings.magX = 0;
    //data;

    call MagY.getData(); //get data for MagnetometerB
    return SUCCESS;
}

/**
* In response to the <code>MagY.dataReady</code> event, it stores the sample and issues a task to filter
and process the stored magnetometer data It also has a schedule which starts sampling the Temperture and
Accelormeter depending on the stepdown counter. @return returns <code>SUCCESS</code> */

async event result_t MagY.dataReady(uint16_t data){
    readings.magY = 0; //readings.magY=0; para desactivar sensor
                    //readings. magY =data; para activar sensor

    atomic {
        if (!sending_packet) {
            sending_packet = TRUE;
            post send_radio_msg();
        }
    }
}

```

```

#if FEATURE_SOUNDER
    if (sound_state) call Sounder.start();
    else call Sounder.stop();
    atomic {
        sound_state = SOUND_STATE_CHANGE;
    }
#endif
return SUCCESS;
}
/**
 * Handles all broadcast command messages sent over network.
 * NOTE: Bcast messages will not be received if seq_no is not properly
 * set in first two bytes of data payload. Also, payload is
 * the remaining data after the required seq_no.
event result_t XCommand.received(XCommandOp *opcode) {

    switch (opcode->cmd) {
        case XCOMMAND_SET_RATE:
            // Change the data collection rate.
            timer_rate = opcode->param.newrate;
            call Timer.stop();
            call Timer.start(TIMER_REPEAT, timer_rate);
            break;
        case XCOMMAND_SLEEP:
            // Stop collecting data, and go to sleep.
            sleeping = TRUE;
            call Timer.stop();
            call StdControl.stop();
            call Leds.set(0);
            break;
        case XCOMMAND_WAKEUP:
            // Wake up from sleep state.
            if (sleeping) {
                initialize();
                call Timer.start(TIMER_REPEAT, timer_rate);
                call StdControl.start();
                sleeping = FALSE;
            }
            break;

        case XCOMMAND_RESET:
            // Reset the mote now.
            break;
        case XCOMMAND_ACTUATE: {
            uint16_t state = opcode->param.actuate.state;
            if (opcode->param.actuate.device != XCMD_DEVICE_SOUNDER) break;
            // Play the sounder for one period.
            sound_state = state;
            if (sound_state) call Sounder.start();
            else call Sounder.stop();
            atomic {
                sound_state = SOUND_STATE_CHANGE;
            }
            break;
        }
        default:
            break;
    }
    return SUCCESS;
}

```



```

#if FEATURE_UART_SEND
/**
 * Handle completion of sent UART packet.
 *
 event result_t SendUART.sendDone(TOS_MsgPtr msg, result_t success)
 {
 // if (msg->addr == TOS_UART_ADDR) {
 atomic msg_radio = msg;
 msg_radio->addr = TOS_BCAST_ADDR;
 if (call Send.send(BASE_STATION_ADDRESS,MODE_UPSTREAM,msg_radio, sizeof(XDataMsg))
 != SUCCESS) {
 atomic sending_packet = FALSE;
 call Leds.yellowOff();
 }
 if (TOS_LOCAL_ADDRESS != 0) // never turn on power mgr for base
 call PowerMgrEnable();
 //}
 return SUCCESS;
 }
#endif
/**
 * Handle completion of sent RF packet.
 *
 event result_t Send.sendDone(TOS_MsgPtr msg, result_t success)
 {
 atomic {
 msg_radio = msg;
 main_state = START;
 sending_packet = FALSE;
 call StdControl.stop();
 }
 call Leds.yellowOff();
#if FEATURE_UART_SEND
 if (TOS_LOCAL_ADDRESS != 0) // never turn on power mgr for base
 call PowerMgrEnable();
#endif
 return SUCCESS;
 }
 event result_t XEEControl.restoreDone(result_t result)
 {
 if(result) {
 call Timer.stop();
 call Timer.start(TIMER_REPEAT, timer_rate);
 }
 return SUCCESS;
 }
}

```

- **XMTS310.NC**

```

#include "appFeatures.h"

includes sensorboardApp;

configuration XMTS310 {
// this module does not provide any interface
}
implementation
{
    components Main,
        TimerC,
            GenericCommPromiscuous as Comm,
            MULTIHOPRouter,XMTS310M, QueuedSend,
            Voltage, MicC, PhotoTemp, Accel, Mag, Sounder,

            XCommandC,

#if (FEATURE_UART_SEND | TOS_LOCAL_ADDRESS==0)
        HPLPowerManagementM,
#endif

        LEDS_COMPONENT
        Bcast;

    Main.StdControl -> XMTS310M;
    Main.StdControl -> QueuedSend.StdControl;

    Main.StdControl -> MULTIHOPRouter.StdControl;
    Main.StdControl -> Comm;
    Main.StdControl -> TimerC;

    LEDS_WIRING(XMTS310M)

#if (FEATURE_UART_SEND | TOS_LOCAL_ADDRESS==0)
    // Wiring for UART msg.
    XMTS310M.PowerMgrDisable -> HPLPowerManagementM.Disable;
    XMTS310M.PowerMgrEnable -> HPLPowerManagementM.Enable;
#endif
#if FEATURE_UART_SEND
    XMTS310M.SendUART -> QueuedSend.SendMsg[AM_XDEBUG_MSG];
#endif
#endif

    XMTS310M.Timer -> TimerC.Timer[unique("Timer")];

    // Wiring for Battery Ref
    XMTS310M.BattControl -> Voltage;
    XMTS310M.ADCBATT -> Voltage;

    XMTS310M.TempControl -> PhotoTemp.TempStdControl;
    XMTS310M.Temperature -> PhotoTemp.ExternalTempADC;

    XMTS310M.PhotoControl -> PhotoTemp.PhotoStdControl;
    XMTS310M.Light -> PhotoTemp.ExternalPhotoADC;

```

```
XMTS310M.Sounder -> Sounder;

XMTS310M.MicControl -> MicC;
XMTS310M.MicADC -> MicC;
XMTS310M.Mic -> MicC;

XMTS310M.AccelControl -> Accel;
XMTS310M.AccelX -> Accel.AccelX;
XMTS310M.AccelY -> Accel.AccelY;

XMTS310M.MagControl-> Mag;
XMTS310M.MagX -> Mag.MagX;
XMTS310M.MagY -> Mag.MagY;

XMTS310M.XCommand -> XCommandC;
XMTS310M.XEEControl -> XCommandC;

// Wiring for RF mesh networking.
XMTS310M.RouteControl -> MULTIHOPROUTER;
XMTS310M.Send -> MULTIHOPROUTER.MhopSend[AM_XMULTIHOP_MSG];
MULTIHOPROUTER.ReceiveMsg[AM_XMULTIHOP_MSG] -
>Comm.ReceiveMsg[AM_XMULTIHOP_MSG];
XMTS310M.health_packet -> MULTIHOPROUTER;
}
```

- **XMeshBaseM.NC**

```
#include "appFeatures.h"
```

```
module XMeshBaseM {
  provides
    interface StdControl; }
  uses
    {
    interface RouteControl;
    interface XCommand;
    interface Leds;
    }
}

implementation {
  command result_t StdControl.init() {
    call Leds.init();
    atomic{ TOS_LOCAL_ADDRESS = 0; }
    return SUCCESS;
  }
  command result_t StdControl.start() {
    return SUCCESS;
  }
  command result_t StdControl.stop() {
    return SUCCESS;
  }

  event result_t XCommand.received(XCommandOp *opcode)
  {
    switch (opcode->cmd) {
      case XCOMMAND_SET_RATE:
        break;
      case XCOMMAND_SLEEP:
        break;
      case XCOMMAND_WAKEUP:
        break;
      case XCOMMAND_RESET:
        break;
      default:
        break;
    }
    return SUCCESS;
  }
}
```

- **XMeshBase.NC**

```
#include "appFeatures.h"

includes sensorboardApp;

/** XMeshBase is the gateway firmware for all XMesh suite applications. */

configuration XMeshBase {
// this module does not provide any interface
}
implementation
{
    components Main,
        MULTIHOPROUTER,XMeshBaseM,
        JOIN_COMPONENT
        LEDS_COMPONENT
        HEARTBEAT_COMPONENT
        XCommandC;

    Main.StdControl -> XMeshBaseM;
    JOIN_WIRING()
    HEARTBEAT_WIRING()

    LEDS_WIRING(XMeshBaseM)

    XMeshBaseM.XCommand -> XCommandC;

    // Wiring for RF mesh networking.
    XMeshBaseM.RouteControl -> MULTIHOPROUTER;
}
```

ANEXO 3

MANUAL DE USUARIO

INFORMACIÓN

El siguiente documento guía que explica los pasos necesarios para la utilización del hardware que conforman la Red de Sensores Inalámbricos de la compañía Crossbow, programación de nodos, estación base y visualización de datos en el software de fábrica.

También se explica la operación del software de Adquisición de Datos desarrollado para el proyecto: “Desarrollo de un Software de Adquisición de Datos en LabVIEW para la Red de Sensores Inalámbricos Crossbow orientado a la realización de prácticas en el Laboratorio de Instrumentación y Sensores del Departamento de Eléctrica y Electrónica”.

INTRODUCCIÓN

Para la realización de estas prácticas de laboratorio se va utilizar los siguientes dispositivos electrónicos:

- ✓ Sensores MTS310CB.
- ✓ Nodos MPR2400.
- ✓ Estación Base MIB520CB.
- ✓ Computador Personal

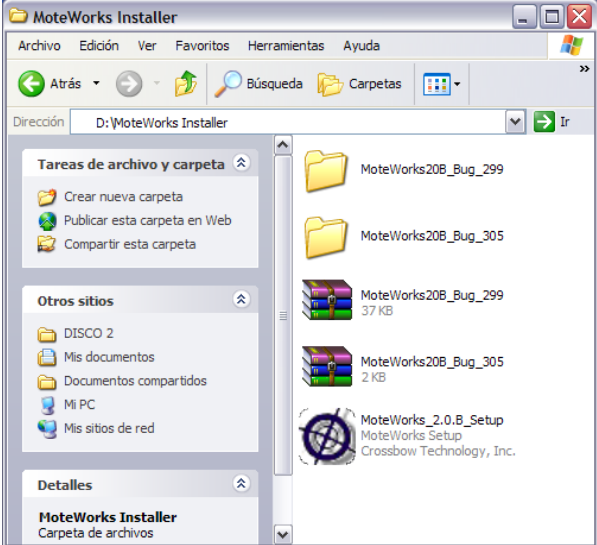
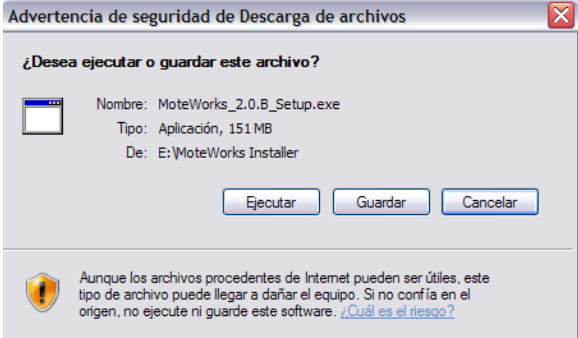
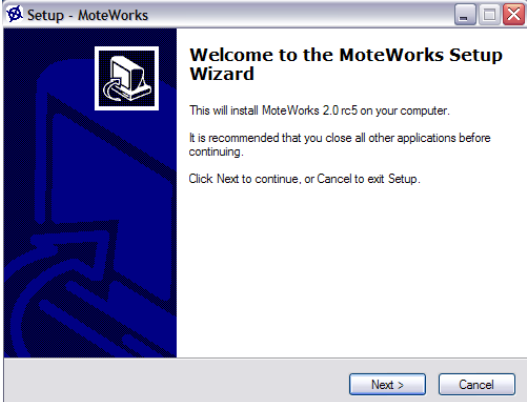
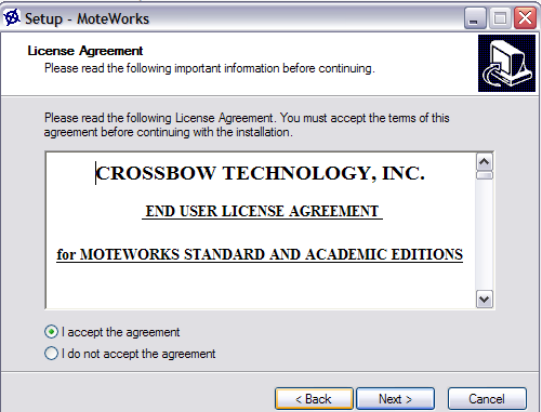
INSTALACIÓN

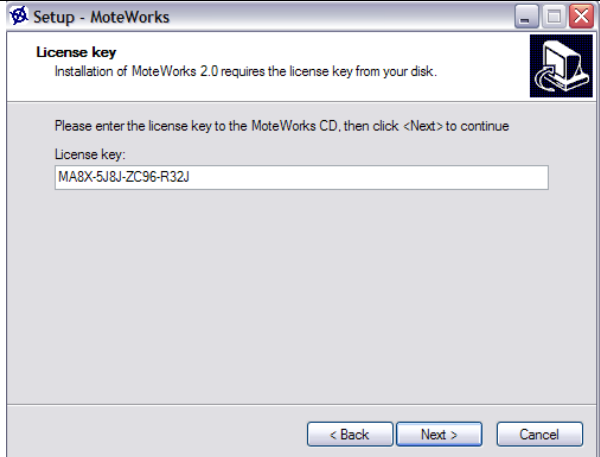
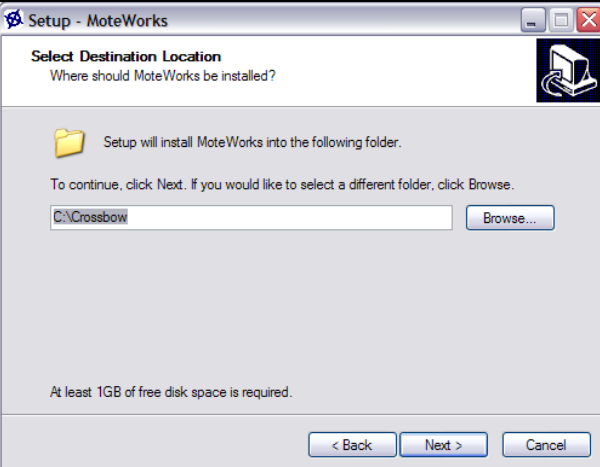
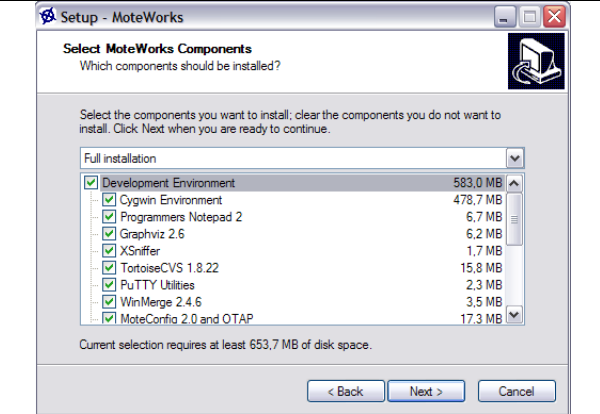
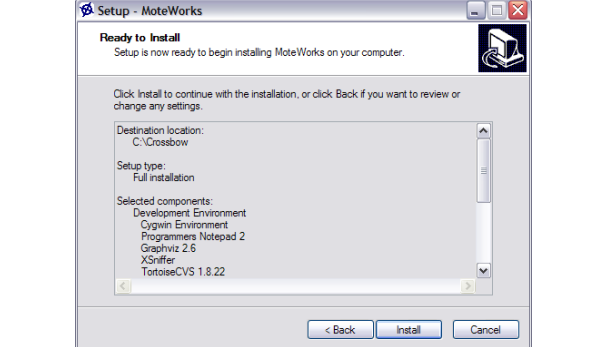
SISTEMA OPERATIVO

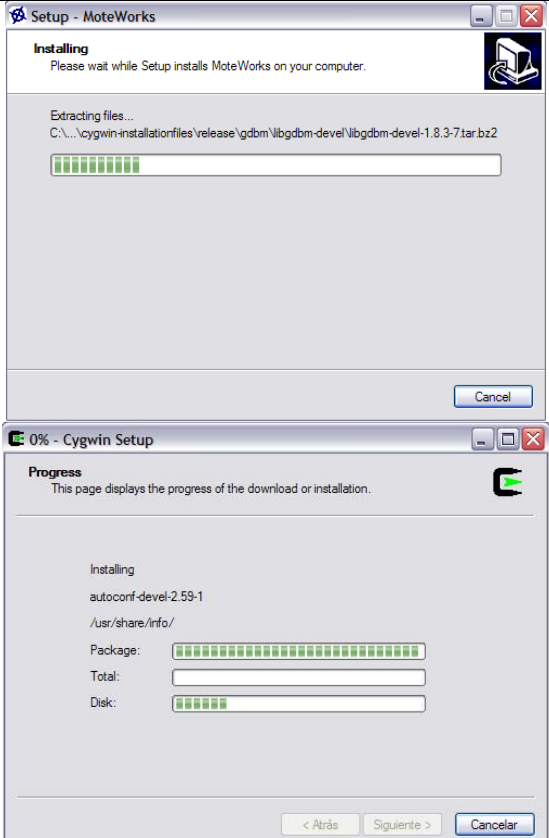
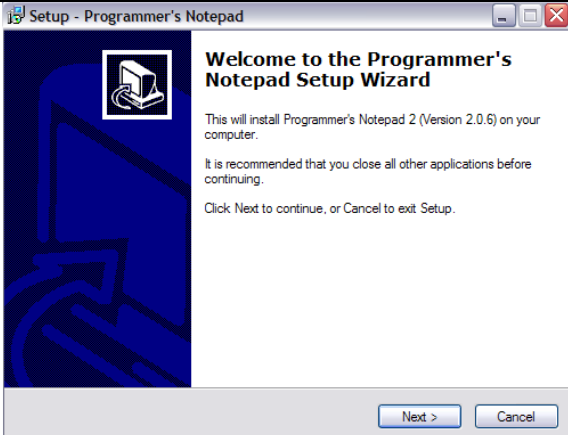
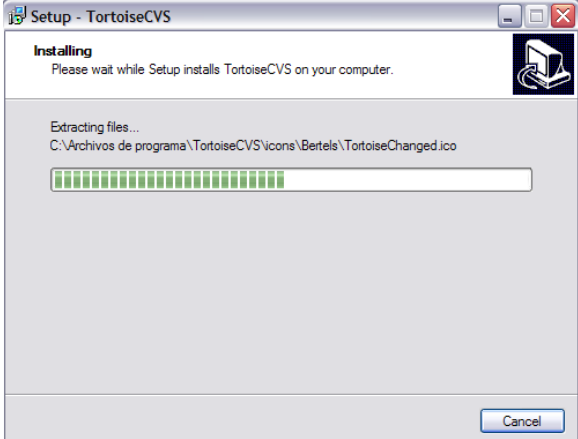
Para el correcto funcionamiento de los componentes Crossbow y la ejecución de LabVIEW es necesario que su sistema operativo sea: Windows XP Home o Windows XP Professional.

INSTALACIÓN SOFTWARE CROSSBOW

El paquete de Sensores Inalámbricos Crossbow, cuenta con dos discos para la instalación de sus componentes y configuración de la Red; para lo cual se va a utilizar el disco etiquetado MOTE WORKS, el cual dispone de las herramientas, programas y hojas técnicas de los elementos de la Wireless Sensor Network. A continuación se muestra los pasos necesarios para poder utilizar los equipos Crossbow.

<ol style="list-style-type: none"> 1. Insertar en el CD ROOM el disco Etiquetado MOTE WORKS. 2. En la carpeta MoteWorks Installer y haga doble clic en el archivo MoteWorks_2.0.B_Setup.exe 	
<ol style="list-style-type: none"> 3. En la ventana emergente que se despliega luego del paso 2, hacer clic en el botón ejecutar. 	
<ol style="list-style-type: none"> 4. Se desplegará una nueva ventana de bienvenida, hacer clic en Next. 	
<ol style="list-style-type: none"> 5. En la ventana de acuerdo de licencia seleccionar I accept the agreement, y luego hacer Clic en Next. 	

<p>6. En la ventana License Key ingresar el código clave que se encuentra en la parte posterior del estuche de los discos.</p>	
<p>7. Seleccionar el destino para la instalación y hacer clic Next.</p>	
<p>8. En la ventana de selección de componentes, seleccionar todos los que están disponibles y proceder a hacer clic en Next.</p>	
<p>9. En la ventana de información de la instalación hacer clic en Install.</p>	

<p>10. Durante el proceso de instalación del software, se instala también Cygwin.</p>	 <p>The image shows two overlapping windows. The top window is titled 'Setup - MoteWorks' and is in the 'Installing' phase, showing progress for extracting files from a tarball. The bottom window is titled '0% - Cygwin Setup' and is in the 'Progress' phase, showing progress for installing 'autoconf-devel-2.59-1'.</p>
<p>11. Finalizada la instalación de Cygwin instalar Programmer's Notepad.</p>	 <p>The image shows the 'Setup - Programmer's Notepad' window. It displays a 'Welcome to the Programmer's Notepad Setup Wizard' message, indicating that it will install Programmer's Notepad 2 (Version 2.0.6) and recommending that other applications be closed.</p>
<p>12. Luego de instalarse Programmer's Notepad también se instala TortoiseCVS.</p>	 <p>The image shows the 'Setup - TortoiseCVS' window in the 'Installing' phase, showing progress for extracting files from a specific icon file.</p>

<p>13. Luego de instalarse TortoiseCVS. Seleccionar No en la ventana emergente que aparece tras la instalación.</p>	 <p>A dialog box titled "Confirm" with a question mark icon. The text reads: "To complete the installation of TortoiseCVS, Setup must restart your computer. Would you like to restart now?" There are two buttons: "Sí" and "No".</p>
<p>14. Luego de hacer clic en No se completara la instalación de MoteWorks.</p>	 <p>A window titled "Setup - MoteWorks" showing the "Completing the MoteWorks Setup Wizard" screen. It features a blue graphic on the left and text on the right: "Completing the MoteWorks Setup Wizard", "Setup has finished installing MoteWorks on your computer.", and "Click Finish to exit Setup." A "Finish" button is at the bottom.</p>
<p>15. Finalizada esta instalación inserte el otro disco disponible en el paquete y en la dirección D:\MoteView y hacer doble clic en el archivo MoteView_1.4.C_Setup.exe</p>	 <p>A File Explorer window titled "MoteView" showing the contents of the directory "D:\MoteView". The file list contains "MoteView_1.4.C_Setup.exe" with a description: "MoteView 1.4.32 Setup, Crossbow Technology, Inc." The address bar shows "D:\MoteView".</p>
<p>16. Hacer clic en Next para completar la instalación de MoteView.</p>	 <p>A window titled "Setup - MoteView 1.4.11" showing the "Welcome to the MoteView 1.4.11 Setup Wizard" screen. It features a blue graphic on the left and text on the right: "Welcome to the MoteView 1.4.11 Setup Wizard", "This will install MoteView 1.4.11 on your computer.", "It is recommended that you close all other applications before continuing.", and "Click Next to continue, or Cancel to exit Setup." There are "Next >" and "Cancel" buttons at the bottom.</p>
<p>17. Seleccionar todas las opciones que aparecen en la figura y hacer clic en Next.</p>	 <p>A window titled "Setup - MoteView 1.4.11" showing the "Select Additional Tasks" screen. It asks "Which additional tasks should be performed?". Below, it lists several tasks with checked boxes: "Microsoft .NET Framework", "PostgreSQL 8.0 Database", "PostgreSQL ODBC Driver", "Reset Mote Database", and "MOTE-VIEW Application". Under "Additional icons:", "Create a desktop icon" is also checked. There are "< Back", "Next >", and "Cancel" buttons at the bottom.</p>

18. Finalizada la instalación de todos los componentes y programas, reiniciar el computador.
19. Para la instalación del hardware conectar la estación base MIB520 y automáticamente con cualquiera de los discos se instalaran los drivers para el dispositivo.
20. Para la programación de la estación y los nodos base retirar el modulo MIB520 y acoplar un nodo MPR2400 al conector de 51 pines y conectar nuevamente al puerto USB.
21. Ejecutar la ventana Cygwin y digitar los siguientes comandos para la instalación de la estación base:

```
cd
c:/Crossbow/cygwin/opt/moteworks/apps/xmesh/xmeshbase.
make micaz install, 0 mib520,com <n>
```

Donde <n> representa el puerto COM virtual para programación.

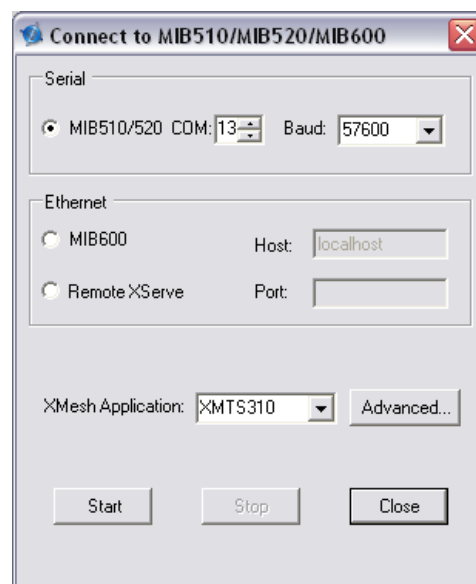
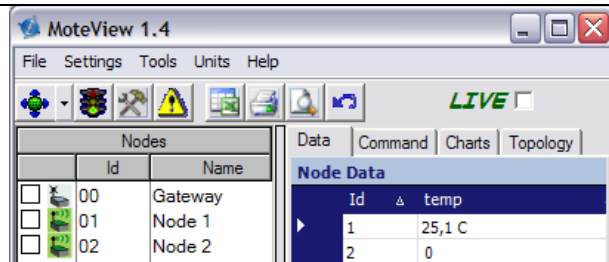
22. Para los nodos:

```
cd c:/Crossbow/cygwin/opt/moteworks/apps/xmesh/xmts310cb
make micaz install, <x> mib520,com <n+I>
```

Donde <x> es un número entero que representa al identificador de nodo.



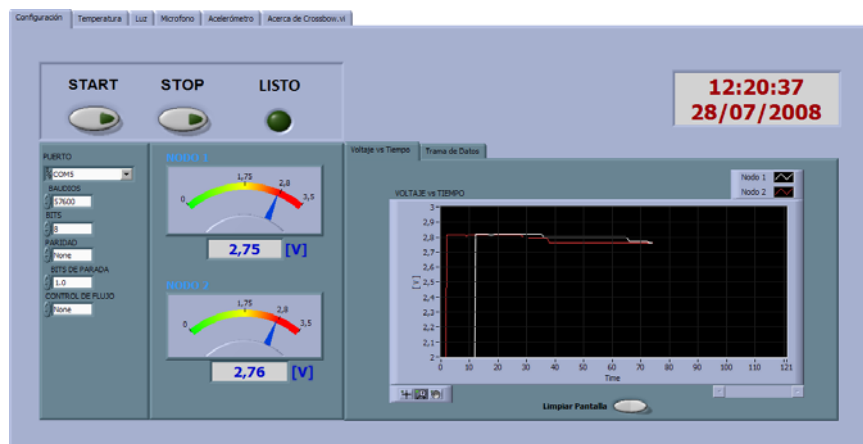
23. Para visualizar los datos adquiridos por los sensores y comprobar su funcionamiento ejecutar la aplicación **Mote View** instalada y configurar los parámetros que se encuentran en : *File* → *Connect* → *Connect to MIB*.....
24. Y presionar el botón **Start** para recibir los datos de los sensores.



OPERACIÓN DEL SOFTWARE

Para la operación del software ejecutar el archivo Crossbow.vi disponible en el disco del proyecto.

1. Configurar el puerto serial de comunicación.
2. Insertar la estación base MIB520 acoplada con el nodo MPR2400 al puerto USB del computador.
3. Encender los nodos 1 y 2, programados.
4. Haciendo clic en el Boton **RUN** de LabVIEW ejecutar la aplicación.
5. Presionar el botón **START** de la ventana Configuración.



6. Esperar a que el Led LISTO se encienda para poder adquirir datos.
7. Para la adquisición de datos en las distintas pestañas, configurar el número de muestras a ser adquiridas, la ubicación de destino del archivo de datos y presionar el botón START que se encuentra en cada una de las pestañas para Temperatura, Luz, Micrófono y Acelerómetro.



8. El botón START volverá a su posición inicial una vez finalizada la adquisición del número de muestras ingresado.
9. Para detener la ejecución de la aplicación presionar el botón **STOP** de la pestaña de configuración, retirar la estación base del computador y apagar los sensores.

RECOMENDACIONES

Para un correcto funcionamiento del sistema tomar las siguientes consideraciones.




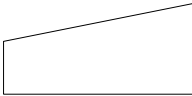
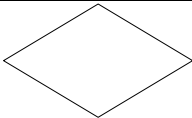
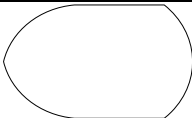



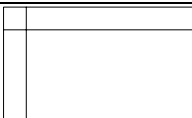

- ✓ Al momento de acoplar del tablero MTS310 al nodo MPR2400 verificar que el switch de encendido y apagado del nodo este en la posición OFF, no retirar el tablero MTS310 con el nodo encendido, y no acoplar o retirar el nodo MPR2400 en la estación base MIB520 cuando esta esté conectado al puerto USB del computador.

- ✓ Por problemas de compatibilidad con algunas herramientas de LabVIEW no se recomienda la utilización del proyecto con el sistema operativo Windows Vista.



ANEXO 4

FORMAS PARA DIAGRAMAS DE FLUJO

FORMA	DESCRIPCIÓN
	Inicio / Fin
	Proceso
	Datos
	Entrada Manual
	Decisión
	Pantalla
	Proceso Predefinido
	Documento
	Datos Almacenados
	Almacenamiento Interno
	Referencia

ANEXO 5

GUÍAS DE LABORATORIO

PRÁCTICA 1: SONIDO

TEMA: INTERPRETACIÓN DE FRECUENCIA

1. *Objetivos*

Determinar la palabra digital correspondiente a una muestra de frecuencia analógica.

Encontrar un modelo matemático para la frecuencia.

2. *Marco Teórico*

- **Circuitos detectores de tono**

Los detectores de tono son circuitos integrados que se encargan de verificar si en la entrada del mismo hay una frecuencia igual o distinta a la que esta sintonizado.

- **Frecuencia y notas musicales**

La característica fundamental de los sonidos musicales es su “altura” o cantidad de veces que vibra por segundo (frecuencia). Cuanto más oscilaciones por segundo tenga la frecuencia, más aguda o “alta” será la nota musical. En la música es muy importante la relación que existe entre la frecuencia de los distintos sonidos, a esta relación se le llama intervalo. La relación entre las notas musicales parten de la nota musical **LA** que tiene una frecuencia de 440Hz.

El oído humano puede llegar a captar frecuencias entre los 20Hz hasta 20.000Hz, la frecuencia de las notas musicales llega solamente a los 4.500Hz. En la Figura. 6.1, se muestra la frecuencia de las notas musicales.

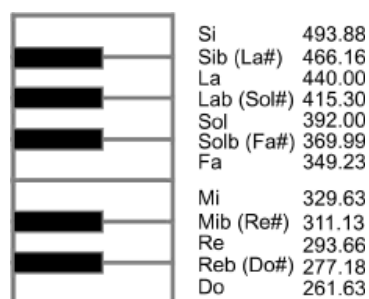


Figura 1. Frecuencias de las Notas Musicales

3. *EQUIPO NECESARIO*

1. Estación Base MIB520CB.
2. Nodo 2: MPR2400.
3. Sensores MTS310CB.
4. Un par de baterías AA.
5. Computador Personal con drivers de estación base instalados y software de adquisición de datos Crossbow.vi (ver manual de usuario).
6. Equipo generador de frecuencia sonora.

4. *Desarrollo de la práctica*

16. Conectar correctamente la estación base (MIB520CB) al computador.
17. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
18. Identificar el NODO 2 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
19. Conectar las baterías AA al nodo.
20. Encender el tablero NODO-SENSORES.
21. Abrir y ejecutar la aplicación Crossbow.vi
22. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
23. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido).



Figura 2. Verificación nodo Activo

- 24. Calibrar el equipo generador de frecuencias.
- 25. En la pestaña **Micrófono** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará.

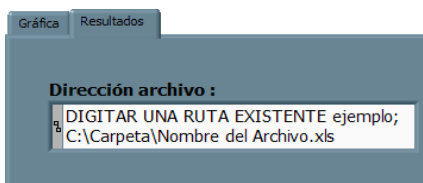


Figura 3. Ejemplo de ruta para generación de archivos

- 26. Acerque el generador de frecuencia (sonido) configurado a la frecuencia F1, al micrófono del sensor MTS310.
- 27. Presione al botón **START** de la ventana de sonido.
- 28. Finalizada la adquisición en la ventana de resultados se desplegaran los picos máximos y mínimos de la adquisición, estos resultados deben ser anotados en la siguiente tabla.

F1 =		F2 =		F3 =		F4 =		F5 =	
MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN

Tabla 1. Resultados Practica 1

- 29. Repetir los pasos 9 a 12 para cada una de las frecuencias de la Tabla anterior. *Debe generarse un archivo para cada una de las frecuencias.*
- 30. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el grafico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione **Export Simplified Image** y posteriormente en **Save to File** seleccione la ubicación del archivo.

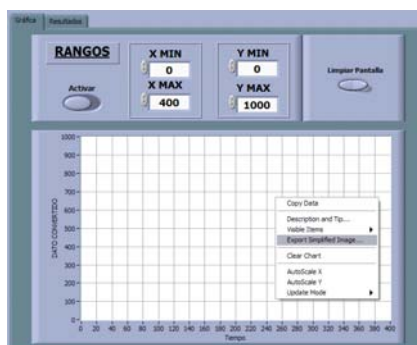


Figura 4. Exportar Gráfico Historio

5. *Informe*

8. ¿Qué son los detectores de tono?

9. Indique 5 aplicaciones de los detectores de tono

10. Enumere 3 elementos adicionales que cumplan la función de detector de tono y señale sus principales características.

11. Utilizando Excel o Matlab realice las Gráficas DATO vs TIEMPO para cada una de las frecuencias tomadas en la Tabla Resultados Práctica 1.

12. Obtenga el promedio del dato analógico convertido a digital para cada una de las frecuencias de la Tabla Resultados Práctica 1.

13. Realice una Gráfica PROMEDIO vs FRECUENCIA en Excel o Matlab y con la ayuda de las herramientas de estos programas obtenga la ecuación de la curva.

PRÁCTICA 2: ILUMINACIÓN

TEMA: VARIACIONES DE RESISTENCIA DEBIDO A CAMBIO DE LUZ

1. *Objetivos*

- Calcular la variación de resistencia debido al cambio de luz en la fotoresistencia del modulo de sensores Crossbow MTS310CB.
- Analizar la mínima y máxima variación de tensión debido al cambio de luz con respecto al voltaje de referencia.
- Determinar la palabra digital de cada una de las muestras realizadas.

2. *Marco Teórico*

• **Fotorresistencias**

Fotorresistencias, son resistencias sensibles a la luz cuyo valor ohmico puede variar fuertemente en función de la luz incidente.

El valor de la resistencia en la oscuridad es típicamente de alrededor de unos $M\Omega$. Al contrario, con 10 Lux se encuentra en la zona de pocos $K\Omega$. La ventaja en comparación con los fotodiodos de silicio consiste en la sensibilidad espectral de los LDRs. Mientras los fotodiodos son sensibles entre 400nm (azul) y 1100nm (Infrarrojo), los LDRs solo detectan luz en el espectro visible. Por ello los LDRs son especialmente aptos como sensores de luz porque no detectan la parte infrarroja que se encuentra normalmente en la luz natural y que puede falsear la medición.



Figura 5. Fotorresistencia

Propiedades

- ✓ Solo componentes con dos terminales
- ✓ Sensores de diferente tamaño
- ✓ Impermeables o sobre sustrato de cerámica.
- ✓ Sensibles en el espectro visible

Aplicaciones

- ✓ Control de iluminación
- ✓ Retrovisor de automóvil automático

• Circuito de Referencia

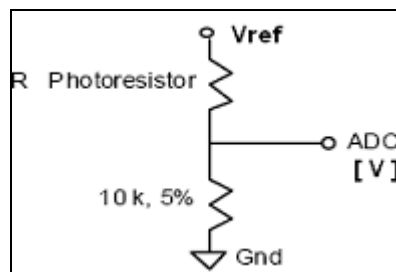


Figura 6. Circuito Fotoresistencia

La ecuación de conversión es:

$$\text{ADC}[\text{V}] = \text{BV} \times \text{DATA}/\text{ADC_FS} \quad \text{Ecuación 1}$$

donde,

BV = Voltaje de la Batería

DATA = Datos ADC convertido a decimal

ADC_FS = 1023

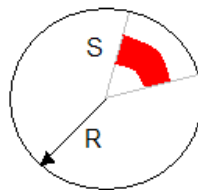
La Ecuación para hallar el valor de la fotoresistencia es:

$$R_{PH} = 10\text{K}\Omega \frac{V_{REF}}{V_{CONV}} - 10\text{K}\Omega \quad \text{Ecuación 2}$$

- **Luminotecnia**

Es la ciencia que estudia las diferentes formas de producir luz, su control y su aplicación. Su objetivo es permitir el funcionamiento de las tareas visuales en un determinado lugar, el confort visual y que exista un nivel de iluminación que permita distinguir las cosas.

Angulo Solido (ω). Relaciona el área de un casquete esférico respecto al radio del mismo.



$$\omega = \frac{S}{R^2}$$

Ecuación 3

Flujo Luminoso (F). Cantidad de luz emitida por una fuente luminosa por unidad de tiempo, su unidad de medida es el Lumen \rightarrow Lm

Intensidad Luminosa (I_L). Es el flujo luminoso por unidad de ángulo sólido radiado en una sola dirección, su unidad de medida es la candela (Cd).

$$I_L = \frac{F}{\omega} [Cd]$$

Ecuación 4

Nivel de Iluminación (E). Llamado simplemente iluminación, es el flujo luminoso incidente por unidad de superficie su unidad de medida es el Lux \rightarrow Lx.

$$E = \frac{F}{S}$$

Ecuación 5

Capacidad Visual. Es el término que se utiliza para describir la velocidad del funcionamiento del ojo y la exactitud con que se lleva a cabo una tarea.

Luminancia (L). Es la intensidad luminosa por unidad de superficie aparente de una fuente de luz primaria o secundaria.

Eficiencia Luminosa (η). Es la relación entre el flujo luminoso total de la fuente respecto de la potencia que consume la misma fuente.

$$\eta = \frac{F}{P} \left[\frac{lum}{W} \right] \quad \text{Ecuación 6}$$

Curvas ISOLUX. Es el lugar geométrico de puntos de una superficie donde la iluminancia tiene el mismo valor.

3. *Equipo Necesario*

17. Estación Base MIB520CB.
18. Nodo 2: MPR2400.
19. Sensores MTS310CB.
20. Un par de baterías AA.
21. Computador Personal con drivers de estación base instalados y software de adquisición de datos Crossbow.vi (ver manual de usuario).

4. *Desarrollo de la Práctica*

17. Conectar correctamente la estación base (MIB520CB) al computador.
18. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
19. Identificar el NODO 2 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
20. Conectar las baterías AA al nodo.
21. Encender el tablero NODO-SENSORES.
22. Abrir y ejecutar la aplicación Crossbow.vi
23. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
24. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido Figura 2).

25. En la pestaña **Luz** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará. (Figura 3)
26. Visualmente identifique cuatro lugares dentro del laboratorio en los cuales exista niveles distintos de iluminación.
27. Ubique el sensor en uno de los lugares identificados y presione al botón **START** de la ventana de luz.
28. Finalizada la adquisición en la ventana de resultados se desplegarán los picos máximos y mínimos de la adquisición, estos resultados deben ser anotados en la siguiente tabla.

L1		L2		L3		L4		V _{REF}
MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	[V]

Tabla 3. Resultados 1 – Práctica 2

29. Repetir los pasos 11 y 12 para cada uno de los lugares del paso 10.
Debe generarse un archivo para cada una de las adquisiciones.
30. De acuerdo a los resultados de la Tabla 3, realice dos adquisiciones largas (100 a 200 muestras) en la cual el sensor pase por todos los 4 lugares de la Tabla 3, una desde el lugar con menos iluminación hasta el lugar con mayor iluminación y otra adquisición desde el lugar de mayor iluminación hacia el lugar de menos iluminación.
Debe generarse un archivo para cada una de las adquisiciones.
31. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el gráfico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione **Export Simplified Image** y posteriormente en **Save to File** seleccione la ubicación del archivo.
32. Utilizando su mano varíe el nivel de iluminación sobre la fotoresistencia y active cada una de las alarmas de la pestaña **Alarmas** y complete la Tabla. 4.

MUY BAJA		BAJA		ALTA		MUY ALTA	
V _{CONV}	V _{REF}	V _{CONV}	V _{REF}	V _{CONV}	V _{REF}	V _{CONV}	V _{REF}

Tabla 4. Resultados 3 – Practica 2

5. Cuestionario

22. ¿Cuáles son las características de las fotoresistencias?

23. Indique 5 elementos detectores de luz.

24. Consulte los flujos luminosos para:

Lámpara incandescente 60 W: _____

Lámpara incandescente 100 W: _____

Lámpara fluorescente 20 W: _____

Lámpara fluorescente 40 W: _____

25. Consulte los niveles de iluminación recomendados para:

Cancha de Futbol: _____

Corredores: _____

Escaleras: _____

Dormitorios: _____

Salas de Clase: _____

26. Determine los valores de resistencia para los siguientes casos:

$$V_{CONV} = V_{REF}/4$$

$$V_{CONV} = V_{REF}/2$$

$$V_{CONV} = 3V_{REF}/4$$

27. Con los datos de la Tabla. 6.4, determine:

Rangos entre alarmas en Voltios.

Rangos entre alarmas en Resistencia.

Rangos entre alarmas en Unidades Crudas.

6. Gráficas y Análisis de Resultados

9. Utilizando Excel o Matlab realice las Gráficas V_{CONV} vs TIEMPO para cada uno de los archivos obtenidos.
10. Realice una gráfica V_{CONV} vs RESISTENCIA para los primeros datos obtenidos en el paso 14.
11. Realice una gráfica de histéresis con los datos obtenidos en el paso 14.
12. Con los datos de la Tabla 3 y la ecuación de la fotoresistencia complete la siguiente tabla.

LUGAR	Valor Máximo [V]	Resistencia [KΩ]	Valor Mínimo [V]	Resistencia [KΩ]
1				
2				
3				
4				

Tabla 5. Cálculos 1 – Práctica 2

13. Con los datos de los archivos del paso 14 del desarrollo de la práctica complete la siguiente tabla:

Medición	Promedio $V_{CONV}[V]$	Promedio $V_{REF}[V]$	Resistencia [K Ω]	Palabra Digital (BIN)
1 – 4				
4 – 1				

Tabla 6. Cálculos 2 – Práctica 2

14. Con los valores de resistencia calculados en la Tabla 5, realice una gráfica Diferencia de Resistencia vs Lugar de Muestra.
15. Inserte la grafica obtenida en el paso 15.
16. Analice los resultados obtenidos en los pasos anteriores.

7. Conclusiones y Recomendaciones

8. Bibliografía

PRÁCTICA 3: TEMPERATURA

TEMA: VARIACIÓN DE TEMPERATURA

1. *Objetivos*

- Medir de la variación de temperatura sobre el termistor del MTS310CB, producida por una secadora de cabello.
- Determinar la relación entre la variación de temperatura y la variación de la resistencia del termistor.
- Calcular el valor digital del dato de temperatura.

2. *Marco Teórico*

• **Termistores**

Son resistores variables con la temperatura, basados en semiconductores. Su rango de medición de temperatura está entre los -50°C y los 150°C no obstante las unidades encapsuladas pueden alcanzar los 300°C . En la mayoría de aplicaciones el valor de resistencia a 25°C está entre 100Ω . Y $100\text{K}\Omega$. Aunque se pueden producir con resistencias tan bajas como 10Ω o tan altas como $40\text{M}\Omega$. Se pueden clasificar en NTC y PTC.

NTC. Esta resistencia se caracteriza por su disminución del valor resistivo a medida que aumenta la temperatura, por tanto presenta un coeficiente de temperatura negativo.

PTC. Se caracteriza por el coeficiente de temperatura positivo, de forma que su resistencia aumentará como consecuencia del aumento de la temperatura.

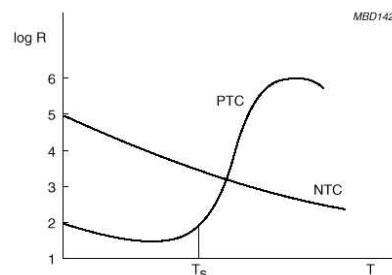


Figura 7. Resistencia vs Temperatura Termistor PTC y NTC

- **Ecuaciones de conversión**

Las ecuaciones que se detallan a continuación son las que se utilizan para convertir los datos digitales a unidades de ingeniería.

$$T(K) = \frac{1}{a + b \times \ln(R_{thr}) + c \times [\ln(R_{thr})]^3} \quad \text{Ecuación 7}$$

donde,

$$a = 0.00130705$$

$$b = 0.000214381$$

$$c = 0.000000093$$

Ecuación de Resistencia

$$R_{thr} = R1 \times (ADC_FS - ADC) / ADC \quad \text{Ecuación 8}$$

$$R1 = 10 \text{ K}\Omega$$

$$ADC_FS = 1023$$

ADC = Dato digital convertido a decimal.

$$T(^{\circ}\text{C}) = T(K) - 273.15$$

3. ***Equipo Necesario***

8. Estación Base MIB520CB.
9. Nodo 1: MPR2400.
10. Sensores MTS310CB.
11. Un par de baterías AA.
12. Computador Personal con drivers de estación base instalados y software de adquisición de datos Croosbow.vi (ver manual de usuario).
13. Secadora de cabello con distintos niveles de temperatura.
14. Termómetro de mercurio.

4. *Desarrollo de la Práctica*

17. Conectar correctamente la estación base (MIB520CB) al computador.
18. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
19. Identificar el NODO 1 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
20. Conectar las baterías AA al nodo.
21. Encender el tablero NODO-SENSORES.
22. Abrir y ejecutar la aplicación Crossbow.vi
23. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
24. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido Figura 2).
25. Ubique el sensor MTS310CB en una posición estable y coloque junto a él, el termómetro, y complete la siguiente Tabla.

TEMPERATURA	[° C]
Interfaz	
Termómetro	

Tabla 7. Resultados 1 – Practica 3

26. En la pestaña **Temperatura** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará. (Figura 3)
27. Encienda la secadora de cabello calibrada a la menor temperatura que está disponga y acerque al sensor y termómetro a 5cm de distancia.
28. Presione el botón **START** en la ventana de temperatura.
29. Una vez finalizada la adquisición en la ventana resultados se desplegaran los valores máximos y mínimos adquiridos durante el proceso de muestreo, anote estos resultados y la medición del termómetro en la Tabla 8.
30. Repita los pasos 11 al 13 para otros niveles de temperatura de la secadora de cabello. *Debe generarse un archivo para cada muestra.*

Temperatura	Max (°C)	Min (°C)	Termómetro (°C)
T1			
T2			
T3			
T4			

Tabla 8. Resultados 2 – Práctica 3

31. De acuerdo a los resultados de la Tabla 8, en la pestaña **Alarmas** configure los valores de acuerdo a su criterio de tal manera que realice dos adquisiciones largas (100 a 200 muestras) en las que varíe la temperatura de la secadora de cabello sobre el sensor, una adquisición de menor a mayor temperatura y otra adquisición de mayor a menor, y se activen todas las alarmas.

Debe generarse un archivo para cada adquisición.

ALARMA	TEMPERATURA [° C]
Muy Baja	
Baja	
Alta	
Muy Alta	

Tabla 9. Resultados 3 – Práctica 3

32. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el grafico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione **Export Simplified Image** y posteriormente en **Save to File** seleccione la ubicación del archivo.

5. Cuestionario

7. ¿Cuáles son las características de los termistores?

8. ¿Enumere 5 aplicaciones en las cuales son utilizados los termistores?

9. Enumere 3 dispositivos que permitan sensor temperatura.

10. Compare a los termistores con los dispositivos de la pregunta 3.

11. Utilizando la ecuación de conversión determine el valor de la resistencia del termistor para los siguientes valores de temperatura.

$$T = 10 \text{ } ^\circ \text{ C}$$

$$T = 15 \text{ } ^\circ \text{ C}$$

$$T = 20 \text{ } ^\circ \text{ C}$$

$$T = 25 \text{ } ^\circ \text{ C}$$

12. Utilizando la ecuación de resistencia obtenga el valor de la palabra digital (BIN) para los valores de resistencia obtenidos en el paso anterior.

$$T = 10 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

$$T = 15 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

$$T = 20 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

$$T = 25 \text{ } ^\circ \text{ C} \rightarrow R = \underline{\hspace{2cm}}$$

6. *Gráficas y Análisis de resultados*

8. Utilizando Excel o Matlab realice las gráficas TEMPERATURA vs TIEMPO, para cada uno de los archivos adquiridos.
9. Realice la gráfica TEMPERATURA vs DATO DIGITAL, para los primeros pasos obtenidos en el paso 15 del desarrollo.
10. Realice una curva de histéresis con los datos obtenidos en el paso 14.
11. Analice los resultados de la Tabla 8.
12. Con los datos del paso 15 complete la siguiente Tabla.

Medición	Promedio T [°C]	Promedio U.C	Resistencia [KΩ]	Palabra Digital (BIN)
Menor a Mayor				
Mayor a Menor				

Tabla 10. Cálculos 1 – Practica 3

13. Con los datos obtenidos en el paso 5, obtenga el tiempo que se demora el proceso en activar las alarmas, y los intervalos de las muestras de este tiempo.
14. Inserte y analice la grafica obtenida en el paso 16.

7. *Conclusiones y Recomendaciones*

8. *Bibliografía*

PRÁCTICA 4: MOVIMIENTO

TEMA: MEDICIÓN DE LA VIBRACIÓN EN TELÉFONOS CELULARES

1. *Objetivos*

- Analizar el efecto que tiene el dispositivo de vibración en los teléfonos celulares.
- Determinar y diferenciar la vibración que producen distintos teléfonos celulares.
- Conocer las características de los dispositivos detectores de vibraciones.

2. *Marco Teórico*

• **Acelerómetro**

El acelerómetro es un transductor versátil, basado en el principio de que cuando se comprime un retículo cristalino piezoeléctrico, se produce una carga eléctrica proporcional a la fuerza aplicada.

Generalmente los elementos piezoeléctricos son de circonato de plomo y se encuentran comprimidos por una masa, sujeta al otro lado por un muelle y todo el conjunto dentro de una caja metálica.

El acelerómetro sometido a vibración, el disco piezoeléctrico se ve sometido a una fuerza variable, proporcional a la aceleración de la masa. Son capaces de medir la aceleración en uno, dos o tres ejes.

Actualmente los acelerómetros son utilizados en teléfonos celulares para aplicaciones de medición de distancias recorridas y shake control en teléfonos Sony Ericsson, aplicaciones en iPhone y en controles de Nintendo Wii.

El acelerómetro que se encuentra en el tablero de sensores MTS310 de dos ejes es una superficie MEMS (Micro Electro Mechanic System) tiene una gran capacidad de detección y una resolución de 10 bits.

Las características de este sensor se describen en la Tabla 11.

Acelerómetro de dos ejes	
Rango	$\pm 2 \text{ g}$ ($1 \text{ g} = 9.81 \text{ m/s}^2$)
Ancho de Banda	DC-50 Hz (controlado)
Resolución	2 mG (0.002 G) RMS
Sensibilidad	167 mV/G $\pm 17 \%$
Offset	2.5 V $\pm 0.4 \text{ V}$

Tabla 11. Características Acelerómetro del MTS310

- **Ecuaciones de Conversión**

La ecuación de conversión utilizada

$$\text{ACC} = 1 - (\text{CP1} - \text{ADC}) / \text{ADC_FS} \text{ (Gravedad Relativa)}$$

donde,

$$\text{CP1} = 500$$

$$\text{CN1} = 400$$

$$\text{ADC_FS} = (\text{CP1} - \text{CN1}) / 2$$

ADC = Dato digital en decimal.

$$\text{ACC}(\text{m/s}^2) = \text{ACC} \times 9.81$$

Ecuación 9

3. *Equipo Necesario*

7. Estación Base MIB520CB.
8. Nodo 1: MPR2400.
9. Sensores MTS310CB.
10. Un par de baterías AA.
11. Computador Personal con drivers de estación base instalados y software de adquisición de datos Croosbow.vi (ver manual de usuario).
12. Cuatro teléfonos celulares distintos.

4. *Desarrollo de la Práctica*

16. Conectar correctamente la estación base (MIB520CB) al computador.
17. Verificar el puerto COM virtual de lectura (COM_{n+1}), mediante el administrador de dispositivos.
18. Identificar el NODO 1 (MPR2400), y acoplar correctamente el tablero de sensores MTS310CB al nodo.
19. Conectar las baterías AA al nodo.
20. Encender el tablero NODO-SENSORES.
21. Abrir y ejecutar la aplicación Crossbow.vi
22. En la pantalla principal de la aplicación Crossbow.vi, seleccionar el puerto COM_{n+1} virtual del paso 2.
23. Presionar el botón **START** y esperar a que el equipo esté listo para adquirir información de los sensores (Led **LISTO** encendido Figura 2).
24. Ubique un teléfono celular en modo vibración sobre una superficie plana, y sobre esté el tablero de sensores.
25. En la ventana acelerómetro anote los valores del sensor en reposo en la siguiente tabla.

Vibración en X	Vibración en Y

Tabla 12. Resultados 1 – Práctica 4

26. En la pestaña **Acelerómetro** fije el número de muestras a adquirirse; en pantalla de micrófono la pestaña **Resultados** fije la dirección en la cual se va a generar el archivo de datos, asegúrese de que la ruta digitada sea una dirección válida, caso contrario el archivo no se generará. (Figura 3)
27. Con la ayuda de otro teléfono celular, realice una llamada al celular que se encuentra bajo el tablero de sensores y presione el botón **START** de la ventana acelerómetro.
28. Una vez finalizada la aplicación en la pestaña **Resultados** se desplegarán los datos máximos y mínimos obtenidos en la adquisición, anote estos resultados en la Tabla 6.13.
29. Repita los pasos 11 al 13 con otros teléfonos celulares.
Debe generarse un archivo para cada muestreo.

30. Utilizando los controles de modificación de rangos en la pestaña **Gráfica** ajuste el grafico histórico, sobre la gráfica haga clic derecho y en el menú emergente seleccione **Export Simplified Image** y posteriormente en **Save to File** seleccione la ubicación del archivo.

Vibración	Max (m/s ²)		Min (m/s ²)	
	X	Y	X	Y
Celular 1				
Celular 2				
Celular 3				
Celular 4				

Tabla 13. Resultados 2 – Práctica 1

5. Cuestionario

4. ¿Cuáles son las características de los acelerómetros?

5. Enumere 4 aplicaciones en las cuales son utilizados acelerómetros.

6. Utilizando la ecuación de conversión determine la palabra digital (BIN) para los siguientes valores:

$$5 \text{ m/s}^2$$

$$9.81 \text{ m/s}^2$$

$$12 \text{ m/s}^2$$

6. *Graficas y Análisis de Resultados*

5. Utilizando Excel o Matlab realice las gráficas VIBRACIÓN x,y vs TIEMPO, para cada uno de los archivos adquiridos.
6. Realice una gráfica PROMEDIO DE LAS MUESTRAS vs TELÉFONO y analice el resultado obtenido.
7. Analice los resultados de la Tabla 6.13.
8. Inserte y analice la gráfica obtenida en el paso 15 del desarrollo de la práctica.

7. *Conclusiones y Recomendaciones*

8. *Bibliografía*

ÍNDICE DE FIGURAS

Figura. 2.1.	Sensores Crossbow MTS310	20
Figura. 2.2.	Nodos o Motas Crossbow MPR2400	20
Figura. 2.3.	Gateway	20
Figura. 2.4.	Estación Base MIB520.....	21
Figura. 2.5.	Arquitectura Wireless Sensor Network.....	22
Figura. 2.6.	ZigBee vs IEEE 802.15.4.....	23
Figura. 2.7.	Arquitectura Sistema de Adquisición de Datos.....	26
Figura. 2.8.	Sistema de Adquisición de Datos Monocanal.....	26
Figura. 2.9.	Sistemas de Adquisición de Datos Multicanales.....	27
Figura. 3.1.	Ventana principal de MoteView	31
Figura. 3.2.	Diagrama de bloques MPR2400/MICAz	32
Figura. 3.3.	Sensores MTS310 con acelerómetro y magnetómetro.....	33
Figura. 3.4.	Vista superior de MIB520CB.....	34
Figura. 3.5.	MIB520CB con MPR2400 acoplado	35
Figura. 3.6.	Ventana terminal de Cygwin.....	35
Figura. 3.7.	Ventana de MoteConfig 2.0	37
Figura. 3.8.	Ventana Interface Board Settings, con COM 12 puerto serial virtual.....	37
Figura. 3.9.	Programación de un nodo en MoteConfig	38
Figura. 3.10.	Programación de la estación en MoteConfig	38
Figura. 3.11.	Trama de Datos	40
Figura. 3.12.	Distribución de Bytes de magnitudes físicas en el paquete Datos Sensores	40
Figura. 3.13.	Validación para obtención de datos sensados	41
Figura. 3.14.	Circuito Fotorresistencia	42
Figura. 4.1.	¿Qué es LabVIEW?	44
Figura. 4.2.	¿Qué es VISA?.....	45
Figura. 4.3.	Configuración de la comunicación Serial-USB	46
Figura. 4.4.	Pantalla principal – Configuración.....	47
Figura. 4.5.	Pantalla Temperatura – Alarma Alta.....	48
Figura. 4.6.	Pantalla Temperatura – Alarma Baja	48
Figura. 4.7.	Pantalla Temperatura – Gráfica	49
Figura. 4.8.	Pantalla Temperatura – Resultados.....	49
Figura. 4.9.	Pantalla Luz – Alarma Alta.....	50
Figura. 4.10.	Pantalla Luz – Alarma MUY BAJA	51
Figura. 4.11.	Pantalla Luz – Gráfica.....	51
Figura. 4.12.	Pantalla Luz - Resultados.....	52
Figura. 4.13.	Pantalla Micrófono – Gráfica.....	53
Figura. 4.14.	Pantalla Micrófono – Resultados	53
Figura. 4.15.	Pantalla Acelerómetro – Gráfica.....	54
Figura. 4.16.	Pantalla Acelerómetro – Resultados	54
Figura. 4.17.	Pantalla – Acerca de Crossbow.vi.....	55
Figura. 4.18.	Diagrama de Flujo – Pantalla Principal	56
Figura. 4.19.	Diagrama de Flujo – Proceso Seleccionar Pestaña	57

Figura. 4.20.	Diagrama de Flujo – Proceso Temperatura.....	58
Figura. 4.21.	Diagrama de Flujo – Proceso Conversión Unidades Temperatura	59
Figura. 4.22.	Diagrama de Flujo – Proceso Comparación Alarmas Temperatura.....	59
Figura. 4.23.	Diagrama de Flujo – Proceso Luz.....	60
Figura. 4.24.	Diagrama de Flujo – Proceso Conversión Unidades Luz	61
Figura. 4.25.	Diagrama de Flujo – Proceso Comparación Alarmas Luz.....	61
Figura. 4.26.	Diagrama de Flujo – Proceso Micrófono	62
Figura. 4.27.	Diagrama de Flujo – Proceso Acelerómetro	63
Figura. 4.28.	Diagrama de Flujo – Proceso Conversión Unidades Acelerómetro.....	64
Figura. 5.1.	Parámetros de Configuración MoteView.....	71
Figura. 5.2.	Prueba de transmisión utilizando MoteView	72
Figura. 5.3.	Prueba de transmisión exitosa utilizando LabVIEW	72
Figura. 5.4.	Prueba de transmisión fallida utilizando LabVIEW	73
Figura. 5.5.	Dato de voltaje adquirido en MoteView	74
Figura. 5.6.	Dato de voltaje adquirido en LabVIEW.....	74
Figura. 5.7.	Dato de Temperatura adquirido con MoteView.....	74
Figura. 5.8.	Dato de Temperatura adquirido con LabVIEW	75
Figura. 5.9.	Dato de Luz adquirido con MoteView.....	75
Figura. 5.10.	Dato de Luz adquirido con LabVIEW	75
Figura. 5.11.	Dato de vibración y sonido adquirido con MoteView	76
Figura. 5.12.	Dato de vibración y sonido adquirido con LabVIEW.....	76
Figura. 6.1.	Frecuencias de las Notas Musicales.....	79
Figura. 6.2.	Verificación nodo Activo.....	80
Figura. 6.3.	Ejemplo de ruta para generación de archivos.....	80
Figura. 6.4.	Exportar Gráfico Historio	81
Figura. 6.5.	Fotoresistencia	84
Figura. 6.6.	Circuito Fotoresistencia	85
Figura. 6.7.	Resistencia vs Temperatura Termistor PTC y NTC.....	92

ÍNDICE DE TABLAS

Tabla. 2.1.	Significado de las letras de identificación funcional.....	17
Tabla. 2.2.	Símbolos de señales y/o conexiones	18
Tabla. 2.3.	Simbología de instrumentos generales.....	18
Tabla. 3.1.	Características acelerómetro de dos ejes.....	33
Tabla. 3.2.	Descripción de paquetes dentro de trama de datos recibidos.....	39
Tabla. 3.3.	Descripción paquete Datos Sensor.....	40
Tabla. 4.1.	Herramientas Utilizadas – Controladores	65
Tabla. 4.2.	Herramientas Utilizadas – Indicadores	66
Tabla. 4.3.	Herramientas Utilizadas – Estructuras	68
Tabla. 4.4.	Herramientas Utilizadas – Funciones.....	69
Tabla. 6.1.	Resultados Practica 1	81
Tabla. 6.2.	Cálculos – Práctica 1	82
Tabla. 6.3.	Resultados 1 – Práctica 2	88
Tabla. 6.4.	Resultados 3 – Practica 2	89
Tabla. 6.5.	Cálculos 1 – Práctica 2.....	90
Tabla. 6.6.	Cálculos 2 – Práctica 2.....	91
Tabla. 6.7.	Resultados 1 – Practica 3	94
Tabla. 6.8.	Resultados 2 – Práctica 3	95
Tabla. 6.9.	Resultados 3 – Práctica 3	95
Tabla. 6.10.	Cálculos 1 – Practica 3.....	97
Tabla. 6.11.	Características Acelerómetro del MTS310	99
Tabla. 6.12.	Resultados 1 – Práctica 4	100
Tabla. 6.13.	Resultados 2 – Práctica 1	101

HOJA DE RECEPCIÓN

Sangolquí, Agosto del 2008

ELABORADO POR:

Sr. Carlos Alberto Vivanco Albuja

COORDINADOR:

Ing. Victor Proaño Rosero
Coordinado de la carrera de
Ingeniería en Electrónica, Automatización y Control