



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DESARROLLO DE FIRMWARE Y SOFTWARE DE
CONTROL PARA UN DISPOSITIVO LECTOR Y
ALMACENADOR DE DATOS DE CUERDA VIBRANTE
CON COMUNICACIÓN ETHERNET Y/O USB

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

JORGE BLANCO FIGUEROA

DIRECTOR DE TESIS: Mtro. Miguel Ángel Mendoza
García

Ciudad Universitaria, México D.F.

Febrero 2016



“A mis Padres, por aguantar la luz encendida toda la noche, en muchas ocasiones, para que pudiera estudiar...”

“A mis Abuelos, por todo el azúcar que me regalaron para seguir adelante...”

“A Bibiana, por escucharme, aunque en muchas ocasiones no me comprendiera...”

“Al Maestro Miguel Ángel, por la oportunidad tan valiosa que me ha ofrecido...”

“Al Ingeniero Enrique, por su paciencia al explicarme diversas cosas y por los prototipos que me ha facilitado para el desarrollo de todo este proyecto...”

“A Alejandro, simplemente por aguantarme todo este tiempo...”

“A mis amigos, por apreciarme tal cual soy...”

A Todos, Gracias n.n !!!

Índice

Índice de Figuras	iii
Índice de Tablas	v
Introducción	1
Hipótesis	2
Objetivo	2
1 Antecedentes.....	3
1.1 El Transductor	3
1.1.1 Funcionamiento del Transductor de Cuerda Vibrante	4
1.1.2 Principio de Funcionamiento de la Cuerda Vibrante	6
1.2 El Microcontrolador.....	9
1.2.1 Computadoras, Microprocesadores, Microcomputadoras y Microcontroladores	9
1.2.2 Microprocesador y Microcontrolador.....	10
1.3 Protocolo RS232	11
1.3.1 Niveles de Voltaje del Estándar <i>ANSI/EIA-232</i>	12
1.4 Protocolo “Universal Serial Bus” (USB).....	12
1.4.1 Especificaciones físicas.....	12
1.4.2 Acceso al Medio (infraestructura USB)	14
1.4.3 Modelo lógico Funcional de USB.....	14
1.5 Transmisión de datos a través de la interfaz Ethernet	15
1.5.1 ARPANET y ALOHANET	15
1.5.2 Modelo OSI	18
1.5.3 Modelo TCP/IP y su comparación contra el Modelo OSI	19
1.5.4 Los elementos básicos de la interfaz Ethernet.....	19
1.6 Planteamiento del Problema.....	20
1.7 Prototipos e Interfaces.....	21
1.7.1 USB.....	21
1.7.2 Ethernet.....	22
2 Firmware y Algoritmos	23
2.1 Lector de “Cuerda Vibrante” (Microcontrolador de 8bits).....	24
2.1.1 Generación del Barrido de Frecuencias.....	24
Pseudocódigo para generar el barrido de frecuencias.....	27
2.1.2 Lectura del transductor	28
Pseudocódigo para la lectura del Transductor.....	29
2.1.3 Comunicación Serial.....	29

Pseudocódigo para selección de Comando	30
2.2 Funcionamiento del Programa de Conexión Directa mediante la interfaz USB ..	33
Pseudocódigo del Funcionamiento del Programa de conexión USB	33
2.3 Micro-servidor Web e interfaz Ethernet (Microcontrolador de 32 bits)	35
2.3.1 Intercomunicación Serial (RS232)	36
2.3.2 Funciones de Adquisidor de datos	40
2.3.2.1 Configuración de la Adquisición	40
Pseudocódigo para la captura de datos de la configuración desde la página llamada "Configuracion.html"	41
2.3.2.2 Funciones para la Adquisición de datos	42
2.3.2.2.1 Adquisición Continua	42
Pseudocódigo para la Adquisición Continua	42
2.3.2.2.2 Adquisición Programada	43
Pseudocódigo para la Adquisición Programada	44
2.3.3 Funciones como Servidor Web	45
2.3.3.1 Comunicación TCP/IP	45
Pseudocódigo del microservidor de HTTP	46
2.3.3.2 Rutinas para el control, lectura y configuración del Adquisidor de datos ..	47
Pseudocódigo de las Páginas Web	47
3 Interfaces Graficas para el Usuario	51
3.1 Programa de comunicación a través del puerto USB	51
3.1.1 Configuración de la Muestra	51
3.1.2 Adquisición manual	56
3.1.3 Adquisición Automática	56
3.2 Página Web para mostrar los datos y configurar el Dispositivo	63
3.2.1 Página Principal (index.htm)	63
3.2.2 Página de Configuración de la adquisición (configuracion.html)	66
4 Resultados	75
4.1 Lector de Cuerda Vibrante	75
4.2 Comunicación a través de la página web	81
Conclusiones	87
Apéndice A Pines del Lector de Cuerda Vibrante con Conexión USB	88
Apéndice B Comandos del Lector de Cuerda Vibrante	90
Glosario	95
Referencias	96

Índice de Figuras

Figura 1.1 Composición del Transductor de Cuerda Vibrante	4
Figura 1.2 Bobinas en Contrafase y Supresión de Ruido	4
Figura 1.3 Fuerza Electro Motriz Aplicada sobre la Cuerda Vibrante	5
Figura 1.4 Fuerza Electro Motriz Ejercida sobre las Bobinas	5
Figura 1.5 Cuerda Vibrante de Tubo	6
Figura 1.6 Armónica Fundamental	7
Figura 1.7 Componentes principales del microprocesador	10
Figura 1.8 Dirección del flujo de datos Figura 1.9 Pines del Conector RS232	11
Figura 1.10 Niveles de Voltaje	12
Figura 1.11 Composición del cable para la interfaz USB	13
Figura 1.12 Infraestructura típica de USB	13
Figura 1.13 Acceso al medio en la interfaz USB	14
Figura 1.14 Esquema del Modelo Lógico Funcional de USB	15
Figura 1.15 Capas del Modelo OSI	18
Figura 1.16 Comparativa entre el Modelo TCP/IP y el Modelo OSI	19
Figura 1.17 Ejemplo de Interfaz Ethernet	20
Figura 1.18 Diagrama de Bloques de la Interfaz USB	21
Figura 1.19 Diagrama de Bloques de la interfaz Ethernet	22
Figura 2.1 Comportamiento del Barrido de Frecuencias	25
Figura 2.2 Inconvenientes del cálculo del periodo de cada señal del barrido	27
Figura 2.3 Método para determinar la Frecuencia del Transductor	28
Figura 2.4 Convertidor RS232-USB para el Microcontrolador de 8 bits	30
Figura 2.5 Esquema de conexión para la comunicación RS232 entre los Microcontroladores	36
Figura 2.6 Cola de Datos Recibidos Vacía	37
Figura 2.7 Cola de Datos Recibidos con un dato nuevo	37
Figura 2.8 Cola de Datos Recibidos con varios datos Nuevos	37
Figura 2.9 Cola de Datos Recibidos antes de desencolar los nuevos datos	38
Figura 2.10 Cola de Datos Recibidos encolando nuevos datos antes de desencolar los anteriores	38
Figura 2.11 Cola de Datos Recibidos con los datos anteriores desencolados	38
Figura 2.12 Cola de Datos Para Transmitir vacía	39
Figura 2.13 Cola de Datos Para Transmitir con varios datos Nuevos	39
Figura 2.14 Cola de Datos Para Transmitir antes de encolar los nuevos datos	39
Figura 2.15 Cola de Datos Para Transmitir antes de Actualizar el apuntador a los nuevos datos	40
Figura 2.16 Cola de Datos Para Transmitir con los Nuevos Datos Encolados	40
Figura 2.17 Modelo TCP/IP	46
Figura 3.1 Mensaje de Conexión del puerto COM	52
Figura 3.2 Lista de ejemplo de Puertos COM de una PC	52
Figura 3.3 Problema con la Comunicación de uno de los puertos COM	53
Figura 3.4 Correcta Comunicación con el Puerto COM Seleccionado	53
Figura 3.5 Lista de Rangos de Frecuencia de Diferentes Cuerdas Vibrantes	54
Figura 3.6 Configuración de Lectura de Una sola cuerda	54
Figura 3.7 Configuración de Lectura a partir de un Multiplexor	55
Figura 3.8 Modo de Operación del Multiplexor	55
Figura 3.9 Lista de Canales del Multiplexor para una lectura Manual	56
Figura 3.10 Configuración de una Muestra Automatizada y Periódica	57
Figura 3.11 Selección de la Ruta para Crear los archivos de las Muestras Automáticas	58

Figura 3.12 Inicio de la Adquisición Automática	58
Figura 3.13 Aviso de modificación de datos de la actual muestra	59
Figura 3.14 Creación del archivo para almacenar los datos de la Muestra Automática ...	59
Figura 3.15 Ejemplo del archivo creado por el Programa para la Interfaz USB	60
Figura 3.16 Creación ó Sobrescritura de Archivo	60
Figura 3.17 Nuevo Archivo para el almacenamiento de Datos de la Muestra	61
Figura 3.18 Datos De la Nueva Muestra Programada	61
Figura 3.19 Datos Sobrescritos en el archivo	61
Figura 3.20 Configuración de una Lista de Canales Definida por el Usuario	62
Figura 3.21 Creación del Nuevo Archivo con la lista de Canales Definida por el Usuario	62
Figura 3.22 Mensaje del Microservidor	63
Figura 3.23 Tabla de Datos para la Lectura de una Cuerda Vibrante.....	64
Figura 3.24 Tabla de Datos Para la lectura de 16 canales de un Multiplexor	64
Figura 3.25 Tabla de Datos para la Lectura de 32 Canales de un Multiplexor	65
Figura 3.26 Adquisidor de Datos sin Configurar.....	66
Figura 3.27 Opciones de configuración del Adquisidor de Datos	67
Figura 3.28 Creación de los Controles para Configurar la lectura de una Cuerda Vibrante	67
Figura 3.29 Explicación de los Controles Creados para la Lectura de una Cuerda Vibrante	68
Figura 3.30 Creación de los Controles para Configurar la lectura de los 16 canales del Multiplexor	69
Figura 3.31 Explicación de los Controles Creados para la Lectura los 16 canales del Multiplexor	70
Figura 3.32 Tabla de datos al seleccionar todos los canales.....	71
Figura 3.33 Tabla de datos al deseleccionar todos los canales	71
Figura 3.34 Antes de la Selección Global	72
Figura 3.35 Desplegando las Opciones de la Selección Global	72
Figura 3.36 Luego de la Selección Global.....	73
Figura 3.37 Creación de los Controles para Configurar la lectura de los 32 canales del Multiplexor	74
Figura 4.1 Pruebas Con el Patrón de Señal.....	75
Figura 4.2 Pulsos Adquiridos	76
Figura 4.3 Representación en frecuencia de los pulsos	77
Figura 4.4 Dispositivo Geokon leyendo la Señal Patrón.....	77
Figura 4.5 Cuerda Vibrante Fijada a una barra de acero	78
Figura 4.6 Dispositivo Geokon GK-403.....	79
Figura 4.7 Dispositivo Lector de Cuerda Vibrante	79
Figura 4.8 Esquema del Dispositivo Adquisidor Vía Ethernet.....	81
Figura 4.9 Diagrama del Funcionamiento del Adquisidor de Datos Vía Ethernet	81
Figura 4.10 Ejecución de los programas Independientes en la PC	82
Figura 4.11 Explicación del programa de “Recepción y Envío de Datos”	83
Figura 4.12 Comportamiento de los programas de la PC, Sin ninguna Configuración Programada.....	84
Figura 4.13 Comportamiento de los programas de la PC, Durante la Configuración de la Muestra.....	84
Figura 4.14 Comportamiento de los programas de la PC, Durante la Adquisición de los Datos	85

Figura 4.15 Comportamiento de los programas de la PC, Explicación de la Adquisición de los Datos.....	85
Figura A.1 Pines del Lector de Cuerda Vibrante.....	88
Figura A.2 Multiplexor en Modo de 16 Canales.....	89
Figura A.3 Multiplexor en Modo de 32 Canales.....	89
Figura B.1 Programa de Comunicación Genérico	91
Figura B.2 Breve explicación del programa de Comunicación Genérico	91
Figura B.3 Comando de Lectura Directa	92
Figura B.4 Comando Fin de Lecturas	92
Figura B.5 Comando para Lectura de un Canal del Multiplexor	93
Figura B.6 Comando para Lectura de Varios Canales del Multiplexor	93
Figura B.7 Comando de Configuración del Multiplexor y Rango de Frecuencia	94
Figura B.8 Comando No Valido	94

Índice de Tablas

Tabla 1 Barrido de Frecuencia.....	25
Tabla 2 Barrido de frecuencia con una Nueva Delta de Tiempo	26
Tabla 3 Muestras del Dispositivo Geokon	80
Tabla 4 Muestras del Dispositivo Lector de Cuerda Vibrante	80

Introducción

En instrumentación permanente de edificios cada vez son más utilizadas las redes Ethernet y la Interfaz de Bus Serial Universal (o USB por las siglas en inglés de “*Universal Serial Bus*”) de las computadoras portátiles o “*Laptops*” que, si bien no son nuevas tecnologías, se están convirtiendo en estándares para la transferencia de información entre dispositivos con diferentes características de funcionamiento, pues en una red Ethernet pueden estar conectados dispositivos de fabricantes distintos y todos pueden comunicarse a través de un protocolo establecido por organismos internacionales. De manera similar a la transferencia de datos por la red Ethernet, es posible conectar diferentes dispositivos con interfaz USB hacia un “*host*” y transferir datos entre todos estos, aunque sean de diferentes fabricantes. Estas ventajas permiten realizar nuevos adquirentes de datos que aprovechen esta infraestructura Ethernet, regularmente incorporada en las nuevas construcciones, para evitar utilizar una nueva e independiente red de cableado especializada para la instrumentación de las edificaciones. La ventaja de la interfaz USB existe en la premisa de solo utilizar un programa especializado para poder realizar la lectura del adquirente de datos sin la necesidad de más instrumentos o algún tipo de alimentación externa.

La instrumentación de edificaciones puede recabar diferentes variables físicas, tales como temperatura, humedad del ambiente, consumo de corriente eléctrica y/o de agua de la edificación, pero para el caso del presente trabajo se seleccionó medir las deformaciones por esfuerzo que sufre la edificación que se planea estudiar en un periodo de tiempo indeterminado, por lo cual el adquirente de datos desarrollado en el presente trabajo utiliza los transductores de tipo “Cuerda Vibrante”.

El Instituto de Ingeniería, a través de la Coordinación de Electrónica, planteó el desarrollo y manufacturación de dispositivos que aprovecharan la infraestructura Ethernet de las edificaciones para instrumentación permanente y la interfaz USB de las computadoras portátiles para realizar pruebas a los diferentes transductores y verificar su correcto funcionamiento. El dispositivo con comunicación a través de la infraestructura Ethernet tiene como funciones principales la recolección de los datos de las deformaciones que sufre la edificación donde este se coloca y la presentación de dichos datos a través de una página web que reside en este dispositivo. Por otra parte, el dispositivo con comunicación a través de la interfaz USB tiene como objetivo realizar muestreos de prueba de los diversos transductores de “Cuerda Vibrante” que hayan sido conectados a dicho dispositivo y el programa especializado, que se comunica con el adquirente de datos vía USB y se ejecuta en la computadora portátil, tiene como objetivo el almacenamiento de los de configuración de la muestra y los datos que el adquirente transmita.

El presente trabajo está dividido en 4 capítulos:

- **Antecedentes:** Se habla de los estándares para la transferencia de información a través de la red Ethernet, la interfaz USB y entre microprocesadores mediante el protocolo “RS232” y el principio mecánico del Transductor de “Cuerda Vibrante” y el dispositivo a través del cual se realiza la lectura del esfuerzo de la edificación bajo estudio. También se mencionan los prototipos desarrollados en este trabajo.

- **Firmware y Algoritmos:** El dispositivo se basa en 2 partes principales, el adquisidor de datos y la interfaz para envío de comandos y conversión de datos. En este capítulo se muestran los algoritmos y los pseudocódigos en los que se basa la adquisición de datos del Transductor y cuáles son los comandos que se desarrollaron para el adquisidor de datos o Lector de “Cuerda Vibrante” y la manera en que las interfaces, USB y Ethernet, son utilizadas para almacenar y operar los datos que el adquisidor envía en información necesaria para un post procesamiento.
- **Interfaces Graficas para el Usuario:** Este capítulo muestra las interfaces grafica de usuario que utilizan cada interfaz para la transferencia de datos, USB y Ethernet, y la explicación de cómo se utiliza cada una de estas.
- **Resultados:** Este último capítulo muestra una comparativa de los datos adquiridos entre un dispositivo adquisidor de datos comercial y el que se ha desarrollado en el presente trabajo. Por causas ajenas al proyecto, las pruebas con el prototipo de interfaz Ethernet, como se muestra en el subcapítulo [1.6.2 Ethernet](#), no pudieron ser realizadas, pero se realizaron pruebas que justifican el funcionamiento de dicho prototipo final, con algunas adecuaciones que no afectan la idea original.

Hipótesis

Es posible construir un acondicionador de señal, un lector y registrador de datos para transductores de “Cuerda Vibrante” a partir de electrónica básica, un microcontrolador y el firmware adecuado.

Objetivo

El presente trabajo tiene como objetivo demostrar la posibilidad de realizar un adquisidor de datos que pueda ser programado para realizar muestreos periódicos y la transferencia de datos sea realizada mediante la red Ethernet o a través de una conexión directa a la interfaz USB. Esta necesidad surge gracias al conocimiento de las necesidades que la Coordinación de Estructuras y la Coordinación de Electrónica, del Instituto de Ingeniería, tienen con respecto a la adquisición de equipos comerciales para realizar las mediciones de diversas variables físicas. Se pretende aprovechar la experiencia que se ha tenido con estos equipos comerciales, con respecto a su funcionamiento, programación de muestras y almacenamiento de datos, y los nuevos equipos adquiridos para la manufacturación de Placas de Circuito Impreso (o PCB por las siglas en inglés de “*Printed Circuit Board*”) y el soldado de piezas de tamaños milimétricos para manufacturar dispositivos adquiredores de datos propios del Instituto de Ingeniería que satisfagan las necesidades específicas de ambos departamentos mediante todo el conocimiento en electrónica, manufacturación y programación de microcontroladores.

1 Antecedentes

Para comprender el presente trabajo son necesarios los siguientes conceptos básicos.

1.1 El Transductor

Un transductor se define como un elemento que transforma una magnitud física en una señal eléctrica y se pueden clasificar en 2 grupos, Activos y Pasivos. Los Activos son aquellos que requieren una fuente de alimentación externa para responder a la magnitud física, tales como las fotoresistencias y termoresistencias, y los transductores pasivos son aquellos que generan una señal eléctrica como respuesta a la magnitud física, como por ejemplo una celda fotovoltaica y fotodiodos. (Dr. Navasquillo Hervás).

El Transductor que se utiliza para cuantificar la deformación por esfuerzo es de tipo “Cuerda Vibrante” y se basa en un alambre pretensado entre dos bloques que se adosan a la estructura bajo estudio, los esfuerzos que sufre la estructura son también transferidos a la cuerda disminuyendo o aumentando la tensión sobre esta. Si se excita mecánicamente a la cuerda esta vibrará a una frecuencia que es función de la geometría de la cuerda, longitud entre los puntos de apoyo, diámetro de la cuerda, así como de la fuerza de tensión de la cuerda. Al considerarse fija la geometría, las variaciones de la frecuencia son exclusivas de las variaciones de la tensión que a su vez son función de los cambios de esfuerzo a los que la estructura es sometida.

Es importante tomar en cuenta que las variaciones dimensionales producidas por efectos térmicos no son despreciables con respecto las mecánicas por lo que es necesario conocer

la temperatura del arreglo a fin de poder descontar los esfuerzos por temperatura, por lo que normalmente se agrega un sensor de temperatura. Tanto la “Cuerda Vibrante” como el sensor de temperatura utilizados en este trabajo son pasivos, ya que ninguno utiliza una fuente externa, como se explica más adelante.

1.1.1 Funcionamiento del Transductor de Cuerda Vibrante

Se utiliza el transductor de “Cuerda Vibrante” en lugar de los deformímetros para cuantificar las deformaciones por esfuerzo debido a que la “Cuerda Vibrante” trabaja con variaciones de frecuencia para calcular dichas deformaciones y presenta una inmunidad mayor al ruido en cableados con longitudes mayores de 100[m] y casi hasta 1 [Km] lo cual puede causar una degradación apreciable en la lectura en comparación a los deformímetros, los cuales trabajan con variaciones de tensión, pudiendo presentar alteraciones en la señal debido a la resistencia presente en la longitud del cable.

El dispositivo con el que se realizan las mediciones de la “Cuerda Vibrante” cuenta con 2 bobinas, como se muestra en la figura 1.1, esto con el objetivo de evitar la interferencia en el rango de frecuencia de 50 a 60 [Hz] producida por lámparas y aparatos electrónicos conectados a la línea de alimentación. Este ruido eléctrico se evita debido a que las bobinas se encuentran en contra fase, por lo que la misma amplitud que se genera en una de las bobinas, se genera en la otra, pero con amplitud contraria, por lo que se anula esta señal, como se muestra en la figura 1.2.

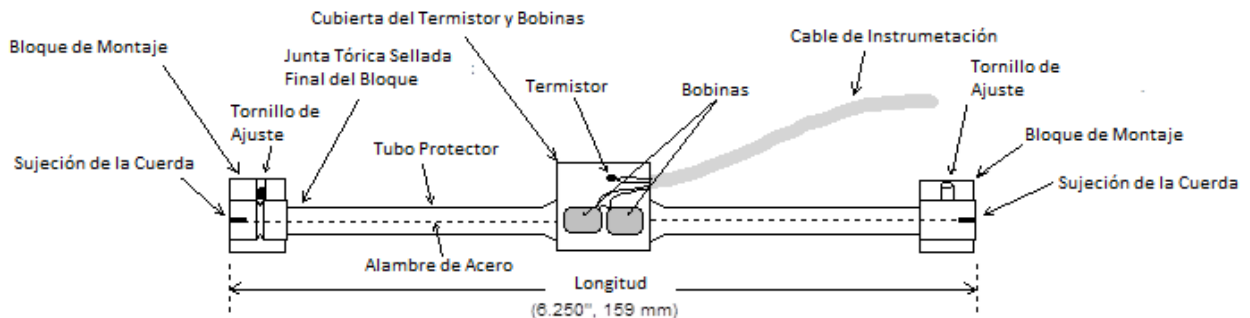


Figura 1.1 Composición del Transductor de Cuerda Vibrante

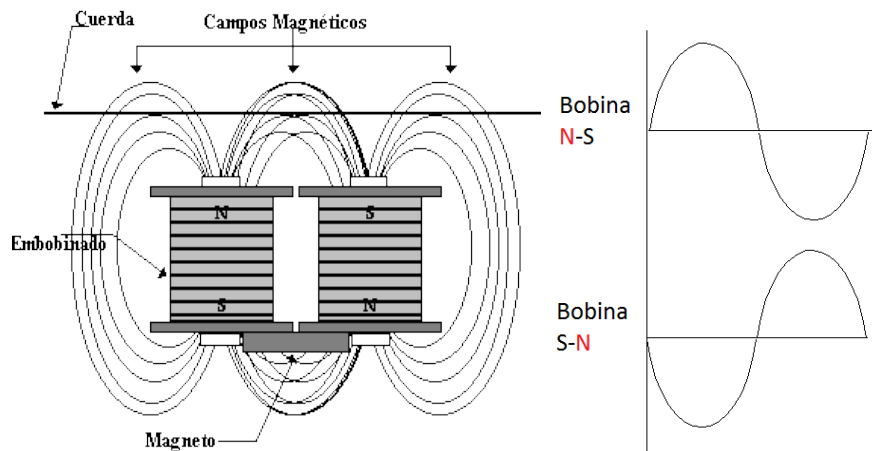


Figura 1.2 Bobinas en Contrafase y Supresión de Ruido

Para medir la frecuencia de resonancia de la cuerda vibrante es necesario aplicar una excitación, una fuerza electro-motriz, que se logra al hacer circular una corriente en las bobinas con el objetivo de hacer vibrar a la cuerda que es de material ferromagnético, la cual posteriormente ejercerá una fuerza electro-motriz sobre las bobinas con su respectiva frecuencia de resonancia, figura 1.3.

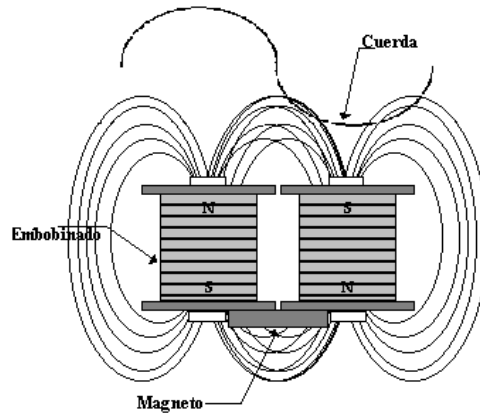


Figura 1.3 Fuerza Electro Motriz Aplicada sobre la Cuerda Vibrante

La corriente que se circula a través de las bobinas debe variar a una frecuencia cercana a la frecuencia de resonancia de la cuerda vibrante para que esta comience a moverse y genere una fuerza electro-motriz sobre las bobinas. Esta fuerza electro-motriz ejercida sobre las bobinas al estar en movimiento la cuerda vibrante y debido a la disposición de las bobinas en el transductor, como se muestra en la figura 1.1, provoca que la cresta de la señal sea captada primero por una de las bobinas y el valle de la misma señal por la otra bobina, por lo que la amplitud de la señal de la cuerda se duplica, como se muestra en la figura 1.4.

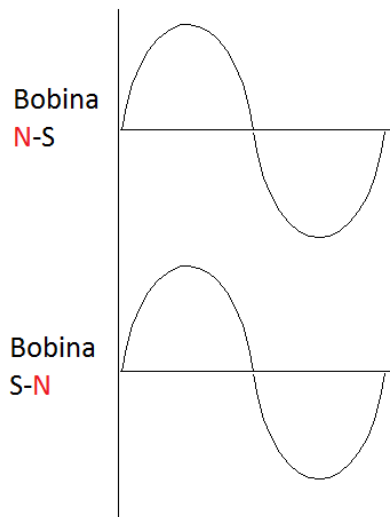


Figura 1.4 Fuerza Electro Motriz Ejercida sobre las Bobinas

La frecuencia que se aprecia en la figura 1.4 corresponde a la frecuencia de resonancia natural inducida por la fuerza electro-motriz de la cuerda sobre las bobinas del transductor

cuando esta se encuentra vibrando, por lo general esta señal es acondicionada para ser interpretada por los equipos adquirentes de datos.

El funcionamiento de este transductor se basa en el principio físico del comportamiento de las ondas, dado que la vibración de la cuerda es una onda. A continuación, se explica el comportamiento de dicho principio físico de una manera simple para el modelo de cuerda vibrante utilizado como referencia para este proyecto.

1.1.2 Principio de Funcionamiento de la Cuerda Vibrante

La figura 1.5 muestra la típica construcción de una “Cuerda Vibrante” de tubo, esta imagen corresponde al modelo de Cuerda Vibrante 4000 de la compañía GEOKON, al igual que en las cuerdas vibrantes utilizadas en los instrumentos musicales, la vibración de la cuerda es una onda cuya velocidad de propagación es función de la raíz cuadrada de la relación entre la tensión de la cuerda con respecto a la densidad lineal de la misma, como se muestra en la ecuación 1



Figura 1.5 Cuerda Vibrante de Tubo

$$v = \sqrt{\frac{T}{\mu}}$$

Ecuación 1 Velocidad de Propagación en función de la Tensión y Densidad Lineal

Donde:

v es la velocidad de propagación de la onda.

T es la tensión de la cuerda a la cual está sometida.

μ es la densidad lineal de la cuerda vibrante.

La velocidad de propagación de la onda también es función de la frecuencia multiplicado por la longitud de onda, como se muestra en la ecuación 2

$$v = \lambda f$$

Ecuación 2 Velocidad de Propagación en función de la Longitud de Onda y Frecuencia

Donde

v es la velocidad de propagación

λ es la longitud de onda, distancia real recorrida de un máximo, ya sea un valle (máximo negativo) o una cresta en una onda senoidal, a otro máximo.

f es la frecuencia a la cual se encuentra vibrando la cuerda

Igualando las ecuaciones 1 y 2 se tiene

$$\lambda f = \sqrt{\frac{T}{\mu}}$$

Despejando la frecuencia se obtiene la ecuación 3

$$f = \frac{1}{\lambda} \sqrt{\frac{T}{\mu}}$$

Ecuación 3 Frecuencia en Función de la Longitud de Onda, la Tensión y la Densidad lineal

La "Cuerda Vibrante" está fija en sus 2 extremos, si la longitud de la cuerda es constante (L) y su armónica fundamental, que se produce por la vibración, y cuyos nodos son los extremos fijos de la cuerda, como se muestra en la figura 1.6, permite sustituir el valor de la onda de longitud en la ecuación 3 como $2L$.

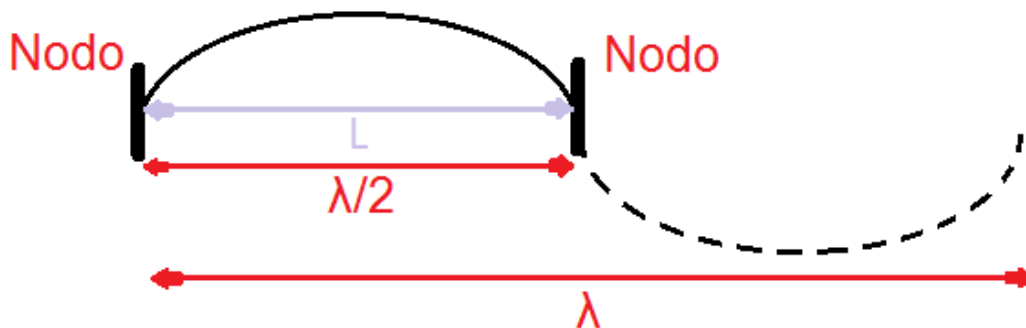


Figura 1.6 Armónica Fundamental

$$f = \frac{1}{2L} \sqrt{\frac{T}{\mu}}$$

Ecuación 4 Frecuencia en Función de la Longitud de la Cuerda Vibrante, la Tensión y la Densidad lineal

Donde:

f es la frecuencia a la que se encuentra vibrado la cuerda

L es la longitud de la cuerda

T es la tensión de la cuerda

μ es la densidad lineal

De la ecuación anterior se sabe que la densidad lineal es prácticamente constante y si bien es cierto que los cambios de la longitud L por efectos de la temperatura son pequeños, es también cierto que son comparables con los cambios de la longitud debidos a los esfuerzos a medir, por lo cual se incorpora un termistor para conocer la temperatura y corregir los efectos de la temperatura.

Para el caso específico de los transductores de “Cuerda Vibrante” que GEOKON manufactura, se hacen aclaraciones sobre las constante que cada Transductor presenta para transformar de frecuencia a microdeformaciones, como la temperatura puede afectar las lecturas de las microdeformaciones registradas dependiendo de donde haya sido instalada el Transductor y cuáles son los rangos de frecuencia donde cada modelo de “Cuerda Vibrante” presenta lecturas correctas, pues la tensión a la cual pueden estar sometidas puede causar lecturas erróneas, ya sea porque la cuerda está muy tensa o a una tensión muy baja.

Dado que es perfectamente bien conocida toda la física y matemática relacionada al comportamiento de las cuerdas vibrantes es un buen método para la fabricación de transductores en los que la tensión, esfuerzo, sea la variable a medir.

En conclusión se puede decir que las variaciones de frecuencia por cambios en los esfuerzos a los que es sometida la Cuerda Vibrante, es la variable que se mide para determinar los esfuerzos que sufre la estructura bajo estudio, son producidas por 3 factores, el material del cual este constituida la Cuerda Vibrante, su longitud y la tensión a la cual este siendo sometida la Cuerda, como se muestra en la ecuación 4, en este caso el material de la cuerda no cambia, la longitud es constante y la tensión es la única variable que provoca un cambio en la frecuencia de resonancia de la cuerda.

Para poder realizar la medición de esta frecuencia, que se muestra en la figura 1.4, es necesario inyectar una señal cuadrada que varíe su frecuencia dentro del rango de funcionamiento del transductor, dicho rango de frecuencia está determinado por el fabricante del transductor, posteriormente, cuando se haya alcanzado la frecuencia más alta del rango, es necesario interrumpir la señal debido a que la cuerda se encuentra vibrando e induciendo una fuerza electro motriz sobre las bobinas con una señal cuya frecuencia corresponde a la frecuencia de resonancia de la cuerda que, por último, está siendo adquirida por el lector para interpretar su frecuencia.

1.2 El Microcontrolador

Para comprender el surgimiento de los *microcontroladores* es necesario mencionar el desarrollo y evolución de los *microprocesadores* debido que no siempre es necesario procesar cantidades inmensas de información, sino responder en tiempo real a condiciones específicas en diversas situaciones, por lo cual a continuación se hace un repaso histórico de ambos componentes.

1.2.1 Computadoras, Microprocesadores, Microcomputadoras y Microcontroladores

Un sistema computacional está compuesto por la Unidad Central de Procesamiento (CPU), memoria, la cual contiene el programa y los datos, una interfaz de Entrada/Salida (Input/Output) asociada a dispositivos de Entrada y Salida y 3 buces que interconectan los elementos del sistema. La organización del programa y los datos en un solo bloque de memoria es llamada "*arquitectura Von Neuman*", luego de que Jon Von Neuman, quien describió este propósito general, almacenara el programa computacional de esta manera en 1945.

Hasta que la corporación Intel introdujo el primer microprocesador, el "4004", en 1971, la CPU fue construida a partir de muchos componentes. De hecho, en 1958 la computadora SAGE de la Fuerza Aérea requirió de 40,000 [ft²], 3 [MW] de poder y tenía 30,000 tubos con un núcleo magnético para una palabra de 4Kx32 bits de memoria (Bit Savers, 2013). El "PDP-8" de la Compañía de Equipo Digital fue la primera minicomputadora producida en masa y apareció en 1964. Este fue el inicio por crear un dispositivo más barato, pequeño y sustentable para aplicaciones que no fueran tradicionales y de procesamiento de datos.

La mayor contribución de Intel fue integrar las funciones de muchos elementos del CPU en un solo circuito Integrado, o al menos algunas. El término "*microprocesador*" vino primero por parte de Intel en 1972 y generalmente se refiere a la implementación de las Funciones de la Unidad Central de Procesamiento en un solo circuito integrado con una alta escala de Integración ("*Large Scale Integrated*"). Una "*microcomputadora*" entonces es una computadora construida usando un microprocesador y unos cuantos otros componentes para la Memoria y los periféricos de I/O. El Intel "4004" albergaba a 4 microcomputadoras y permitió el desarrollo del "8008", el primer microprocesador de 8 bits introducido en 1972. Este procesador tiene 45 instrucciones, 30 microsegundos por instrucción, en promedio, y podía almacenar 16 [KB] de memoria. (Fredrick, 1997, págs. 1 - 4)

Tiempo después, Intel mantuvo sus desarrollos y lanzo al mercado el microprocesador de 8 bits llamado "8080", el cual podía direccionar 64[KB] de memoria y tenía 75 instrucciones. Motorola comprendió lo que estaba sucediendo y se unió a la lucha por crear el microprocesador que se adueñara del mercado, por ello creo el "6800" creado principalmente por Chuck Peddle, además de crear otros periféricos como el "6802" y "6850". Poco tiempo después Chuck Peddle abandono Motorola para unirse a "*MOS Technology*" en la cual continuó trabajando intensamente en el desarrollo de microprocesadores.

“MOS Technology” anuncia en 1975 durante una exhibición sus microprocesadores, el “6501” y “6502”, con un precio de 25 USD cada uno, dado que Intel y Motorola estaban vendiendo sus microprocesadores a un precio de 179 USD. Dado este anuncio, Intel y Motorola bajan sus precios a 69.50 USD cada uno. En ese momento Motorola reclama a “MOS Technology” y a su creador, Chuck Peddle, el haber copiado su microprocesador, el “6800”, por lo cual “MOS Technology” suspende la creación del microprocesador “6501”, aunque sigue fabricando el “6502”, el cual era un microprocesador de 8 bits, 56 instrucciones y 64[KB] de memoria.

Durante el tiempo en que el “6502” se volvió muy popular y fue utilizado en diferentes computadoras, tales como: “KIM-1”, “Apple I”, “Apple II”, “Atari”, etc., Federico Faggin dejó Intel y fundó su propia empresa, “Zilog Inc.”, para crear su propio microprocesador. Fue entonces que, en 1976, “Zilog” anunció el “Z80”, el cual era compatible con los programas ya realizados para el “8080” de Intel, tenía una memoria de 64[KB], 176 instrucciones, un gran número de registros y una opción para poder refrescar la memoria RAM Dinámica además de una mayor velocidad de trabajo que el “8080”. Por esto es que muchos migraron del “8080” hacia el “Z80”, siendo incorporado en algunas computadoras tales como: “Spectrum, Partner”, “TRS703”, “Z-3” entre otras. (Universidad Autónoma de San Luis Potosí, 2015)

Tiempo después, en 1976, Intel lanza el primer microcontrolador denominado “8048” y Motorola el “6805R2”, con lo cual comienza la ruptura evolutiva entre cada uno de estos dispositivos, pues los microcontroladores incorporan mayor capacidad que les permite interactuar con el mundo físico en tiempo real y los microprocesadores se han enfocado en el manejo de grandes cantidades de información.

1.2.2 Microprocesador y Microcontrolador

El microprocesador es un pequeño procesador que está integrado en un solo encapsulado que está compuesto por un gran número de circuitos integrados y su arquitectura está compuesta por 3 partes principales, como se muestra en la figura 1.7

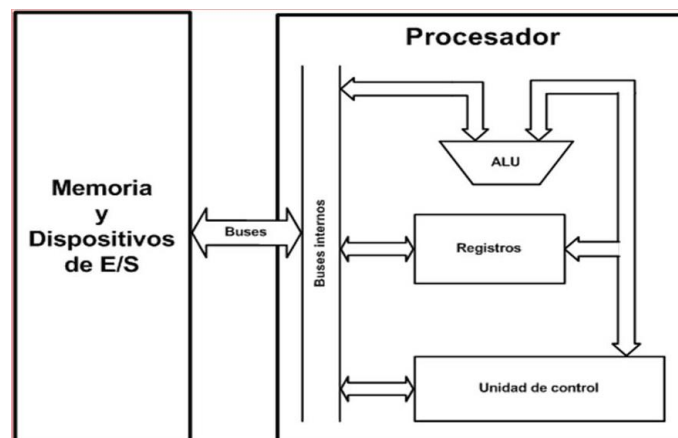


Figura 1.7 Componentes principales del microprocesador

Los 3 componentes principales del microprocesador son:

- “Los Registros”: son localidades de memoria utilizados para mantener información y/o una localidad de memoria durante la ejecución de una instrucción.

- “La Unidad Aritmética Lógica”: es la unidad que está encargada de realizar operaciones aritméticas y evaluar operaciones lógicas.
- “La Unidad de Control” contiene las instrucciones lógicas del hardware, está encargada de decodificar y monitorear la ejecución de las instrucciones.

Las limitaciones que el microcontrolador presenta para poder realizar su funcionamiento son:

- Necesidad de una memoria externa para ejecutar programas
- No puede ser conectado directamente a las interfaces de entrada/salida (Input/Output), es necesario utilizar circuitos periféricos para interactuar con los periféricos.
- Es necesario utilizar una lógica diferente para cada memoria externa e interfaz con los dispositivos de I/O que sea conectada al microprocesador, debido a que cada dispositivo es diferente.

Son estas limitaciones las que dan pie a la creación de microcontroladores, ya que simplifican estos problemas y facilitan el diseño de hardware de productos basados en microprocesadores. (Huang, 1996)

El microcontrolador es una computadora implementada en un solo chip pues incorpora un microprocesador, memoria para ejecutar los programas y memoria para los componentes de las interfaces de I/O, permitiendo así que las aplicaciones para este tipo de dispositivos sean muy variadas y que cada microcontrolador se adapte a las necesidades puntuales que cada proyecto necesita.

1.3 Protocolo RS232

Este protocolo de intercambio de datos se ha utilizado para comunicaciones punto a punto entre dispositivos, los datos se envían de manera serial, es decir, un bit a la vez, y son necesarias mínimo 3 líneas para realizar la comunicación, “TX”, “RX” y “GND”. En el estándar “ANSI/EIA-232” indica en su puerto de conexión algunos pines extra para realizar el “handshake” y el control de flujo de los datos, como se muestra en la figura 1.8 y 1.9.

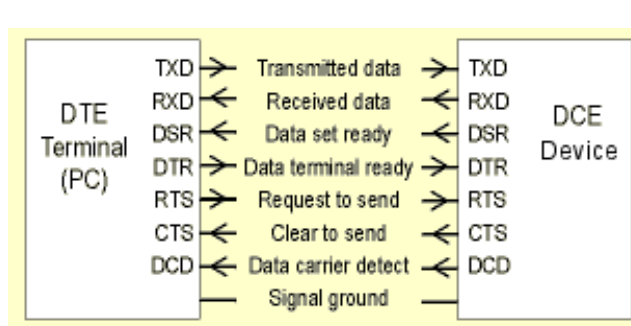


Figura 1.8 Dirección del flujo de datos

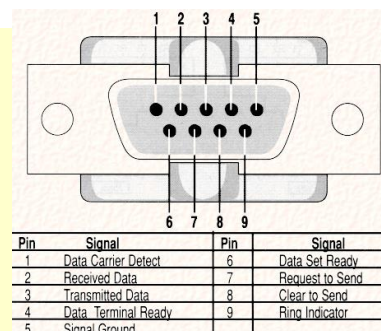


Figura 1.9 Pines del Conector RS232

1.3.1 Niveles de Voltaje del Estándar ANSI/EIA-232

La figura 1.10 muestra la manera en que se deben enviar los pulsos (que corresponden a los bits de la letra "J") tanto en niveles de voltaje TTL (0 a 5 [V]) como a niveles entre 2 Dispositivos Estándar (-12 a +12 [V]). El valor de "1" lógico en los Dispositivos Estándar se codifica como -12[V] y para los voltajes TTL se codifica como 5[V], en el caso del "0" lógico, en los primeros dispositivos, la codificación es +12[V] y para los voltajes TTL es de 0[V]. Los voltajes de -12 y +12 [V] acordados para la transferencia de datos entre dispositivos que soportan el protocolo "RS232" se debe a que así se asegura que el dato que se está enviando sea el que se desea enviar y no sea algún tipo de ruido externo debido a la longitud del cable o interferencias electromagnéticas. Los valores comprendidos entre -3 y +3 [V], en el caso de los Dispositivos Estándar, y los voltajes de 0 a 0.8 [V], para el caso de los voltajes TTL, se entienden como un valor indeterminado producido por algún ruido externo o falla en el dispositivo emisor o receptor. La corriente que manejan ambos niveles de voltaje debe ser máximo de 500[mA] con una longitud máxima de cable de 15[m] (Ing. Forero Saboya, 2012), según la norma "ANSI/EIA-232".

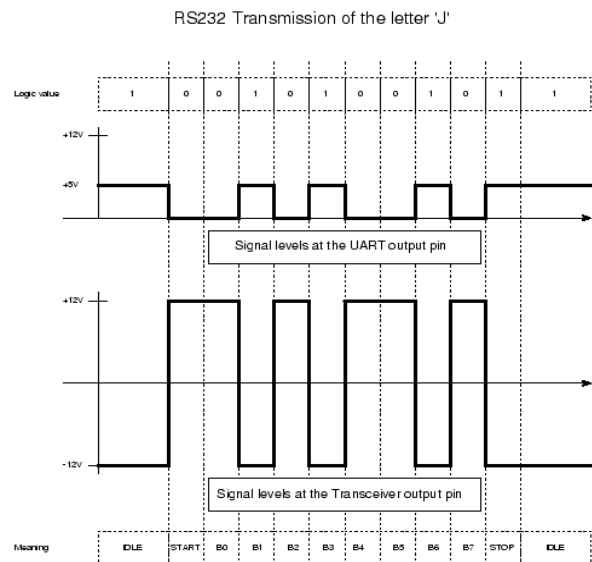


Figura 1.10 Niveles de Voltaje

1.4 Protocolo "Universal Serial Bus" (USB)

1.4.1 Especificaciones físicas

El cable USB cuenta con 4 hebras que cumplen una función específica, en la figura 1.11 se puede apreciar cada hebra y su función

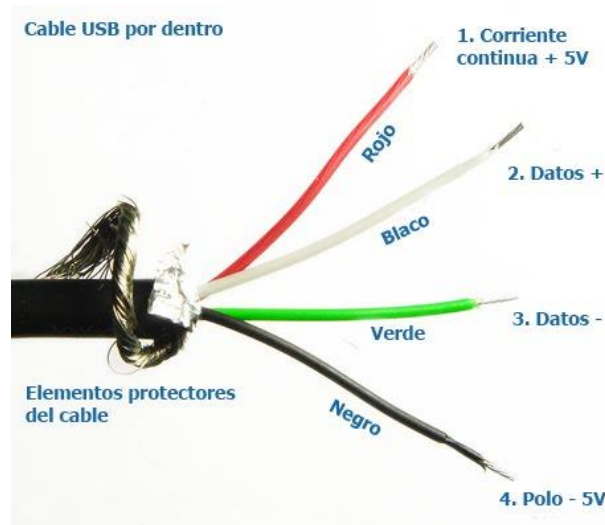


Figura 1.11 Composición del cable para la interfaz USB

Los cables USB pueden ser de hasta 5 [m] y para enviar un bit a través de la infraestructura USB es necesario que exista una diferencia entre las líneas de Datos+ y Datos-- dado que será un "0" o un "1" diferencial, si se desea enviar un "1" en el bus, es necesario que la línea de Datos+ tenga un voltaje superior a 2.8 [V] y la línea de Datos-- sea menor a 0.3 [V], el caso del "0" es inverso, deben existir 2.8 [V] en Datos-- y en Datos+ 0.3 [V] y para recibir un bit es necesario una diferencia entre las líneas de Datos de al menos 200 [mV], para recibir un "0", Datos-- debe ser 200 [mV] mayor a Datos+ y viceversa.

La infraestructura USB se asemeja a una arquitectura de árbol, como se muestra en la figura 1.12, en la cual las hojas son los periféricos, cada infraestructura USB puede tener solo un "Host" el cual planifica el ancho de banda y se encarga de las transacciones. Los "Hubs" son usados para aumentar el número de periféricos en la infraestructura, este se encarga de notificar la conexión o desconexión de nuevos periféricos (o también conocido como función) al "Host".

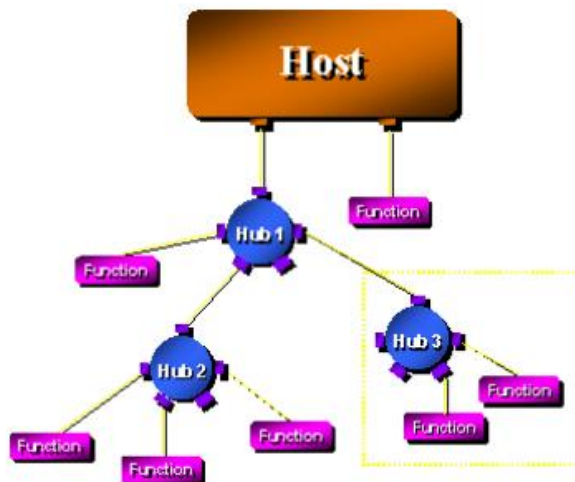


Figura 1.12 Infraestructura típica de USB

1.4.2 Acceso al Medio (infraestructura USB)

Para comenzar la comunicación entre la PC (“*Host*”) y el periférico (función), el “*Host*” envía un paquete denominado “*Token Packet*”, el cual especifica que periférico será el que pueda enviar datos a través del bus en ese momento. Al recibir este paquete, el cual especifica el número de identificación del periférico que fue otorgado previamente por el “*Host*” en el momento en que dicho periférico fue conectado al bus, el periférico comienza a realizar sus tareas propias y de transmisión de datos. Al finalizar de transmitir datos, el periférico enviara un paquete para notificar al “*Host*” que no tiene más datos que enviar.

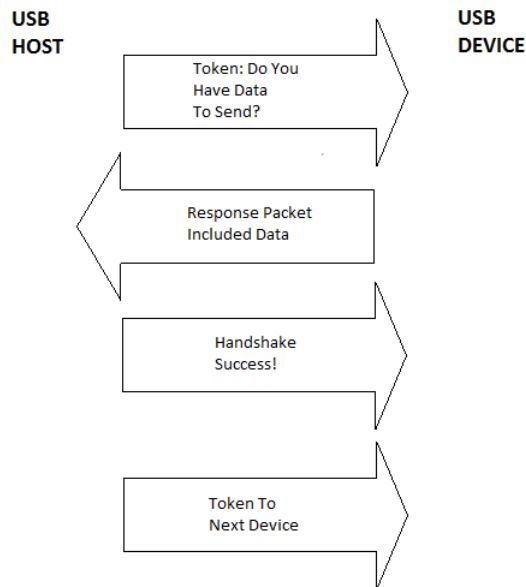


Figura 1.13 Acceso al medio en la interfaz USB

La figura 1.13 muestra, a grandes rasgos, como el “*host*” concede a los periféricos el uso de bus.

1.4.3 Modelo lógico Funcional de USB

Existen 3 niveles lógicos tanto en el “*Host*” como en el Periférico que se comunican entre sí para realizar las diferentes tareas que permiten la correcta comunicación y funcionamiento de los periféricos y sus aplicaciones. La figura 1.14 muestra las conexiones físicas y virtuales (“*Pipes*”) entre los diferentes niveles lógicos del “*Host*” y del periférico

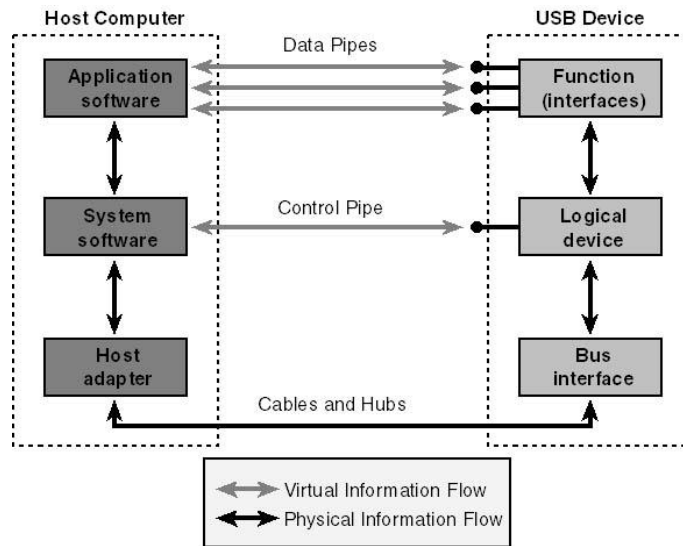


Figura 1.14 Esquema del Modelo Lógico Funcional de USB

La capa superior del modelo, la capa de “*Aplicación de Software*” está encargada de realizar peticiones a los periféricos quienes, en su capa superior, responderán y enviarán la información que el “*Host*” (o cada de “*Aplicación de Software*”) está solicitando.

La capa intermedia del “*Host*”, la capa del “*Sistema de Software*”, se encarga de atender los comandos, gestionar el control interno de los periféricos, así como su información de control y la respuesta de los diferentes periféricos con los cuales la capa de “*Aplicación de Software*” desea comunicarse para que esta capa solo se encargue de atender las solicitudes de información del usuario hacia el periférico.

Finalmente, la capa física, tiene como objetivo resolver los problemas relacionados con la modulación, voltajes de transmisión, saltos de fase, frecuencia y características de la transmisión de los datos a través del bus USB.

1.5 Transmisión de datos a través de la interfaz Ethernet

Los datos que se envían desde un punto a otro de la red necesariamente deben compartir el medio físico donde más dispositivos se encuentran conectados y listos para recibir y envía datos a través del mismo medio físico, para lo cual existen algoritmos que permiten una más eficaz transferencia de dichos datos, pero no solo es importante la transferencia, es igual de importante el significado de la misma, por ello existen protocolos que ayudan a la interpretación de cada paquete de datos enviado por la red ethernet, es por esto que se debe hablar de su historia y marcada interrelación.

1.5.1 ARPANET y ALOHANET

Durante la guerra fría, en 1958, Estados Unidos crea la Agencia de Proyectos de Investigación Avanzados (o ARPA, por las siglas en ingles de “*Advanced Research Projects*”

Agency”) en respuesta a los desafíos tecnológicos y militares de Rusia. E.U. encargo a ARPA diseñar una red confiable que uniera sus centros de datos, debido a que pensaba que las redes telefónicas que alquilaba no eran de fiar. Con esto se creó la ARPANET (red de ARPA o “*ARPA Network*”), en 1969, al principio utilizó la conmutación de la red telefónica para realizar el intercambio de información, pero era ineficiente, por lo cual se optó por utilizar la “conmutación de paquetes”, teoría que Leonard Kleinrock había publicado en 1961. Este mecanismo facilitaba el uso eficiente de rutas alternativas, para aumentar la fiabilidad en caso de pérdida de conexiones. En su primer etapa se unieron 4 nodos, los cuales tenían conectados dispositivos conocidos como Interfaces de Procesamiento de Mensajes o IMS (por las siglas en inglés de “*Interface Message Processors*”), los cuales eran ordenadores que almacenaban y reenviaban mensajes. En esta etapa se decidió documentar las especificaciones técnicas de todo lo que se desarrollaba en documentos llamados RFC (por las siglas en inglés de “*Request For Comment*”), en 1971 se publicaron las especificaciones de los primeros protocolos a nivel de aplicación, tales como FTP (por las siglas en inglés de “*File Transfer Protocol*”), TELNET (“*Telecommunication Network*”) y “*Mail Box Protocol*”.

ARPANET fue la primera Red de Área Local (o LAN, por las siglas de “*Local Area Network*”), pero la primera red de redes fue la NFSNET de la Fundación Nacional de Ciencia (o NFS, por las siglas en inglés de “*National Science Foundation*”), la cual era financiada por el propio gobierno de E.U. y comenzó a operar en 1985 con los protocolos de ARPANET. Fue hasta 1995 cuando dejó de dar servicio ya que se crearon numerosas compañías a nivel regional que proporcionaban acceso a Internet.

Por otra parte, en 1984, en Europa la Organización Internacional de Estandarización (u OSI, por las siglas en inglés de “*International Organization for Standardization*”) publicó el Modelo de Interconexión de Sistemas Abierto (u OSI, por las siglas de “*Open System Interconnection*”). Hasta finales de los años 80, los avances del protocolo TCP/IP, llevado a cabo en E.U., tuvieron repercusiones en Europa, ya que los protocolos de las diferentes capas del Modelo OSI aún no estaban listas y los sistemas operativos Unix comenzaron a implementar el protocolo americano de manera nativa. (Redes Telemáticas, 2013)

Hasta ahora se ha hablado de la manera en que los datos son transmitidos de una computadora a otra a través de los protocolos TCP/IP y el Modelo OSI, pero aún no se ha hablado del canal por el cual se envían los datos y algunos aspectos interesantes, todo esto debido a que ARPANET y ALOHANET se desarrollaron por caminos distintos.

En 1970, cuando ARPANET llevaba pocos meses funcionando, un equipo de la Universidad de Hawai, dirigido por Norman Abramson, quería poner en marcha una red para interconectar terminales ubicados en Kuwui, Maui y Hawai con un ordenador central en Honolulu, en la isla de Oahu. Dado que el costo de instalar redes telefónicas era muy elevado y de baja calidad, se optó por utilizar otro medio de comunicación, se utilizó una red de radio enlaces entre las islas. Para esta red se asignó un canal para que los ordenadores se comunicaran con el ordenador central en Oahu y otro para el sentido inverso, para permitir un ancho de banda mayor en contra posición a tener un canal exclusivo para que cada terminal se comunicara directamente con el ordenador central y otro para que el ordenador central se comunicara con cada terminal, lo cual requeriría 6 canales en total. El protocolo de Control de Acceso al Medio (o MAC por las siglas en inglés de “*Medium Access Control*”) se llamó ALOHA, este control comenzó siendo simple, si una terminal necesitaba enviar información, simplemente la enviaba y esperaba una

confirmación dentro de un tiempo razonable, en caso de no recibir la confirmación, se esperaba un tiempo aleatorio y se enviaba de nuevo.

Este protocolo MAC era bastante ineficiente, por lo cual en 1972 se propuso una mejora, en la cual se le asignaba un intervalo de duración constante para la emisión de datos. Las estaciones estarían sincronizadas y sabrían cuando comenzaría cada intervalo y no se encadenarían las colisiones, debido a que tienen un límite de tiempo. El estudiante Robert Metcalfe desarrollo como tesis de doctorado, la cual terminaría en 1973, una propuesta para una mejora al protocolo MAC de ALOHANET, en la cual las estaciones verificaban si el medio estaba libre antes de enviar información, este protocolo MAC mejorado por Robert, se denominaría Acceso Múltiple con Detección de Portadora y Detección de Colisiones, o más brevemente CSMA/CD (por las siglas en ingles de "*Carrier Sense Multiple Access / Collision Detect*").

Para 1972, Robert Metcalfe se mudó a California para trabajar en el Centro de Investigación Xerox, donde se le encomendó diseñar y construir una red que uniera todos los equipos que existentes, que eran los ordenadores e impresoras, para que se compartiera la información, con la ayuda de David Boggs, un estudiante de doctorado de Standford. Se mejoraron gradualmente los prototipos, denominaron a la nueva red como "Alto Aloha Network", hasta que el 23 de mayo de 1973, mediante un memorándum interno, se informaba de la nueva red y para que no solo se pensara que solo se podían conectar ordenadores, Metcalfe decidió cambiar el nombre de por el de "Ethernet", haciendo referencia a la teoría física según la cual las ondas electromagnéticas viajaban por un fluido llamado éter, haciendo alusión a que el éter en este caso era el cable coaxial por el cual se enviaba la información.

En 1975 se describió Ethernet en un artículo que Metcalfe y Boggso enviaron a Comunicaciones de la Asociación de Maquinaria Computacional (o "*Association for Computing Machinery*" por sus siglas en inglés), incluyendo el uso de repetidores para aumentar el alcance de la red y publicado en 1976. En esta época, Ethernet era propietario de Xerox y la mayoría de los fabricantes hacían arquitecturas de redes fundamentalmente jerárquicas, en el cual existía un mainframe al cual se conectaban las terminales de los usuarios, esta arquitectura era la filosofía que anunciaba IBM. Esta arquitectura era completamente opuesta a la que proponía Xerox, en la cual las trasferencias de información eran de un host a otro, sin controles centralizados.

Metcalfe le propuso a IBM la incorporación de la interfaz Ethernet a sus productos, a lo cual IBM declino la oferta, entonces la propuesta fue hecha a "Digital Equipment Corporation" o "DEC", la cual acepto. A petición de Xerox, Intel se incorporó al grupo de trabajo, para crear chips de bajo costo. La alianza DEC-Intel-Xerox (DIX) resulto en un aumento de la velocidad de Ethernet de 2.4 Mbps a 10 Mbps con precios razonables. (Montaña, 2007)

Estos son los aspectos más relevantes para el surgimiento de las redes Ethernet y el transporte de datos entre dispositivos, posterior a esto se dieron las reglas para la asignación de direcciones MAC y la estandarización de los protocolos y mejoras a los mismos. A continuación, se explica que es el modelo OSI, el protocolo TCP/IP y los dispositivos necesarios para la transferencia de información y acceso al medio físico.

1.5.2 Modelo OSI

Se trata de una normativa estandarizada útil debido a la existencia de muchas tecnologías, fabricantes y compañías dentro del mundo de las comunicaciones, y al estar en continua expansión, se tuvo que crear un método para que todos pudieran entenderse de algún modo, incluso cuando las tecnologías no coincidieran. De este modo, no importa la localización geográfica o el lenguaje utilizado. Todo el mundo debe seguir las normas mínimas para poder comunicarse entre sí. Esto es sobre todo importante cuando se habla de la red de redes, es decir, Internet.

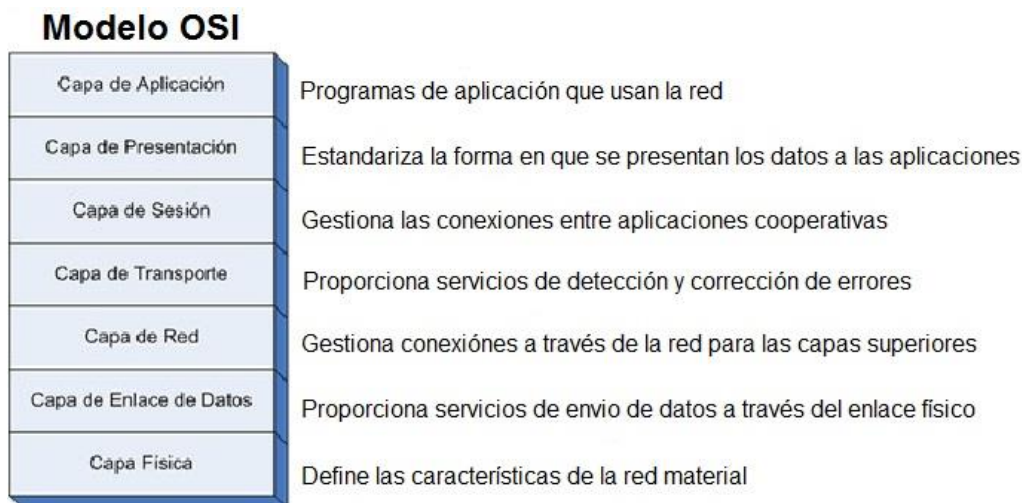


Figura 1.15 Capas del Modelo OSI

Esta representación en forma de pila mostrada en la figura 1.15 en la que cada capa reposa sobre la anterior suele llamarse pila de protocolos o simplemente pila.

En una capa no se define un único protocolo sino una función de comunicación de datos que puede ser realizada por varios protocolos. Así, por ejemplo, un protocolo de transferencia de ficheros y otro de correo electrónico facilitan, ambos, servicios de usuario y son ambos parte de la capa de aplicación.

Cada protocolo se comunica con su igual en la capa equivalente de un sistema remoto, cada protocolo solo ha de ocuparse de la comunicación con su gemelo, sin preocuparse de las capas superior o inferior. Sin embargo, también debe haber acuerdo en cómo pasan los datos de capa en capa dentro de un mismo sistema, pues cada capa está implicada en el envío de datos.

Las capas superiores delegan en las inferiores para la transmisión de los datos a través de la red subyacente. Los datos descienden por la pila, de capa en capa, hasta que son transmitidos a través de la red por los protocolos de la capa física. En el sistema remoto, irán ascendiendo por la pila hasta la aplicación correspondiente.

La ventaja de esta arquitectura es que, al aislar las funciones de comunicación de la red en capas, se minimiza el impacto de cambios tecnológicos en el juego de protocolos, es decir

que es posible añadir nuevas aplicaciones sin cambios en la red física y además añadir nuevo hardware a la red sin tener que reescribir el software de aplicación.

1.5.3 Modelo TCP/IP y su comparación contra el Modelo OSI

El modelo de arquitectura de estos protocolos es más simple que el modelo OSI, como resultado de la agrupación de diversas capas en una sola o bien por no usar alguna de las capas propuestas en dicho modelo de referencia. La figura 1.16 muestra una comparación entre la pila de protocolos TCP/IP y su equivalencia con el Modelo OSI.

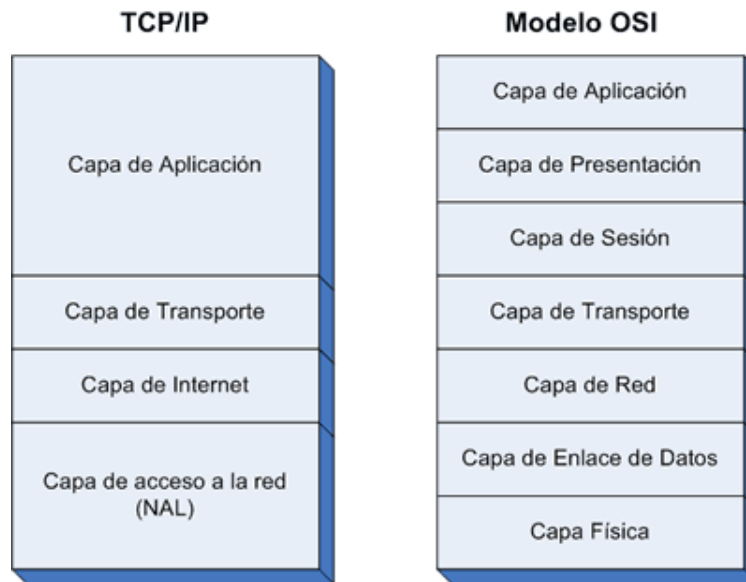


Figura 1.16 Comparativa entre el Modelo TCP/IP y el Modelo OSI

Al igual que en el modelo OSI, los datos descienden por la pila de protocolos en el sistema emisor y la escalan en el extremo receptor. Cada capa de la pila añade a los datos a enviar a la capa inferior, información de control para que el envío sea correcto. Esta información de control se denomina cabecera, pues se coloca precediendo a los datos. A la adición de esta información en cada capa se le denomina encapsulación. Cuando los datos se reciben tiene lugar el proceso inverso, es decir, según los datos ascienden por la pila, se van eliminando las cabeceras correspondientes.

1.5.4 Los elementos básicos de la interfaz Ethernet

Los elementos necesarios para poder intercambiar información a través de la “Capa de acceso a la red” del modelo TCP/IP son:

- Medio físico. - el cual está compuesto por los cables y otros elementos de hardware, como conectores, utilizados para transportar la señal entre las computadoras conectadas en red.
- Interfaz de acceso al medio. - Son dispositivos electrónicos estandarizados (“*transceivers*”) que envían y reciben señales sobre un canal Ethernet.

- Conjunto de reglas para acceder al medio. - protocolos utilizados por la interfaz (tarjeta de red) que controla el acceso al medio y que le permite a los dispositivos acceder (utilizar) de forma compartida el canal Ethernet. Existen 2 modo; unidireccional (“*half dúplex*”), en el cual se utiliza el mismo canal para transmitir o recibir datos, y bidireccional (“*full dúplex*”), en el cual se tienen 2 canales independientes, uno para transmitir y el otro para recibir datos.
- Paquete (o “*frame*” Ethernet). - conjunto de bits organizados de forma estándar. El paquete es utilizado para llevar los datos dentro del sistema Ethernet. También recibe el nombre de marco o trama.

El paquete es un conjunto de Bytes dividido en diferentes campos que indican la dirección física de las interfaces Ethernet, un campo variable de datos (entre 46 y 1500 bytes) y un campo de chequeo de error.



Figura 1.17 Ejemplo de Interfaz Ethernet

La figura 1.17 representa una interfaz Ethernet, que convierte los paquetes de datos del medio físico de la red a una comunicación “RS232”.

1.6 Planteamiento del Problema

El fin que se desea alcanzar con el desarrollo de este dispositivo de adquisición de datos es lograr una lectura correcta de las “Cuerdas Vibrantes”. Esta lectura se logra generando un barrido de frecuencia para crear una excitación al transductor de acuerdo a las especificaciones del fabricante, sobre un modelo específico de cuerda vibrante, ya que dichos valores son los máximos y mínimos que puede alcanzar el instrumento para ser considerados como datos válidos, posteriormente es necesario acondicionar la señal que está siendo generada por el transductor para realizar una correcta lectura.

Con esto es posible saber a qué frecuencia está oscilando la cuerda y por tanto que deformación se está presentando en la estructura que se está analizando.

Se crearon 2 prototipos con distintas interfaces de comunicación para realizar la adquisición de datos:

- USB
- Ethernet

La interfaz USB fue desarrollada en un principio para realizar las pruebas y el monitoreo de los equipos lectores de una manera sencilla y directa.

La interfaz Ethernet fue desarrollada para permitir el monitoreo remoto de los equipos a través de una página web y el almacenamiento de los datos en una memoria MicroSD en caso de no estar disponible la conexión con la red de datos interna de la construcción, realizando así las funciones de Adquisidor de Datos.

1.7 Prototipos e Interfaces

El Lector de “Cuerda Vibrante” no cuenta con una interfaz de usuario, por lo cual fue necesario implementar algún medio que permitiera enviar comandos de lectura y configuración y recibir los datos del Transductor, por lo cual se implementaron 2 interfaces para este propósito.

1.7.1 USB

La interfaz USB está presente en todas las computadoras, tanto PC como Laptops, en el mercado y permite conectar nuevos periféricos de una manera sencilla, es por esto que se decidió utilizar esta interfaz para realizar la verificación del funcionamiento de lector de Cuerda Vibrante. La figura 1.18 muestra el diagrama de bloques de la interfaz.

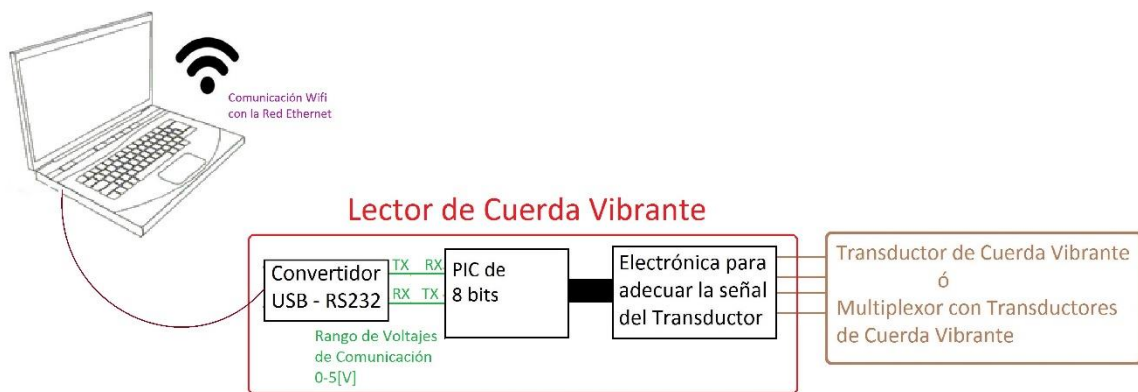


Figura 1.18 Diagrama de Bloques de la Interfaz USB

1.7.2 Ethernet

La interfaz Ethernet nos permite realizar un monitoreo remoto del equipo debido a que se está realizando un servidor Web con una página que permite manipular el comportamiento del lector de cuerda vibrante desde cualquier dispositivo con un navegador web y conexión a internet. La figura 1.19 muestra el diagrama de bloques del dispositivo que está siendo utilizado como servidor web e interfaz Ethernet.

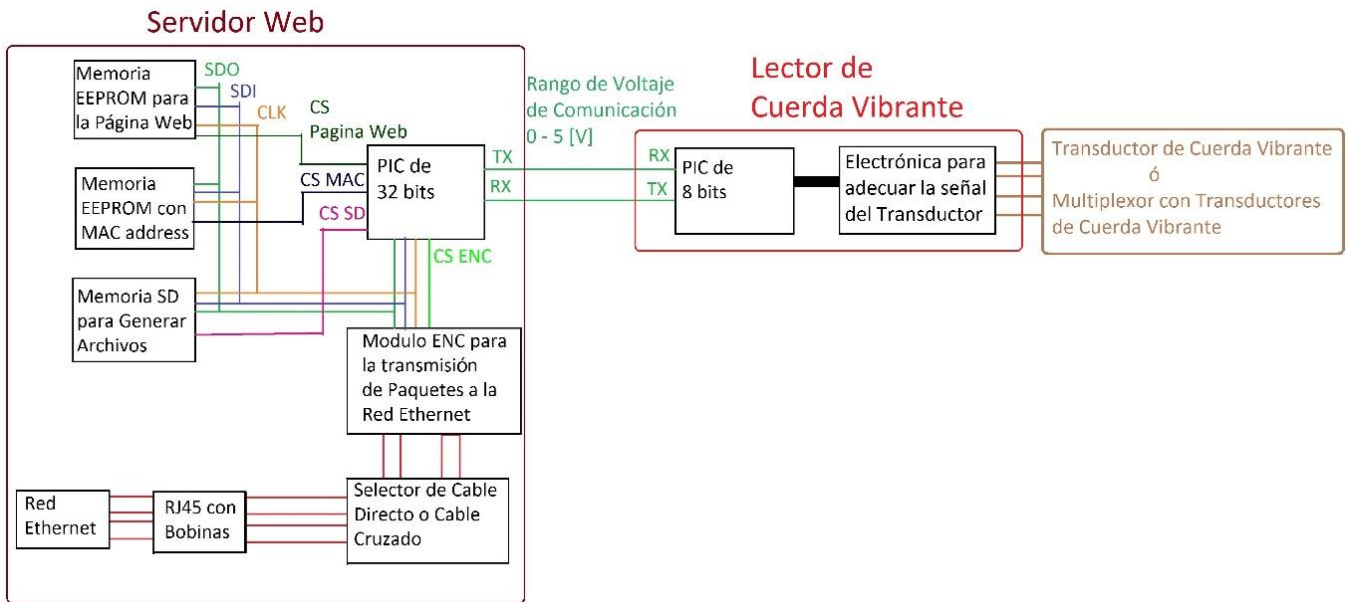


Figura 1.19 Diagrama de Bloques de la interfaz Ethernet

2 Firmware y Algoritmos

El Lector de Cuerda Vibrante está compuesto por:

- Un microcontrolador de 8 bits
- Componentes electrónicos para adecuar la señal del Transductor
- Un convertidor de RS232 a USB (Para la Interfaz USB, figura 1.18)

Estas partes permiten realizar la correcta lectura del Transductor de Cuerda Vibrante y la transferencia de información desde el lector de Cuerda Vibrante hacia los dispositivos que estén solicitando información, ya sea el microcontrolador de 32 bits que se está utilizando como servidor Web (figura 1.19) o la Laptop (figura 1.18).

Por otra parte, está el Servidor Web que está siendo utilizado como interfaz Ethernet, este dispositivo está compuesto por:

- 1 microcontrolador de 32 bits (Servidor Web)
- 2 Memorias EEPROM
 - Para almacenar la página web que el usuario puede consultar
 - Para almacenar la dirección MAC del dispositivo para comunicarse con la red Ethernet
- 1 Memoria MicroSD para almacenar los archivos
- 1 Modulo ENC para enviar y recibir los paquetes a la red Ethernet
- 1 interfaz Ethernet para interactuar con la capa Física compuesta por 2 partes
- 1 Conector RJ45 con magnetos para evitar las colisiones en la capa Física

- 1 selector de Cable Cruzado o Directo para poder enviar y recibir paquetes sin importar la norma de construcción del cable Ethernet

Estos componentes permiten realizar un servidor web en el microcontrolador de 32 bits para poder mostrar una página donde se muestren los datos enviados por el Lector de Cuerda Vibrante y la manipulación de su configuración para la adquisición de datos.

A continuación, se describen las tareas de cada microcontrolador y del programa para realizar la adquisición de datos mediante la interfaz USB.

2.1 Lector de “Cuerda Vibrante” (Microcontrolador de 8bits)

Las tareas del Lector de Cuerda Vibrante son:

- Generar un Barrido de Frecuencia. - Esto se realiza con el objetivo de localizar la frecuencia a la que el Transductor se encuentra debido a la deformación que presenta la estructura
- Lectura del Transductor. - Una vez generada la excitación necesaria para hacer vibrar la Cuerda, es necesario un mecanismo para determinar la frecuencia a la que la cuerda se encuentra vibrando en ese instante
- Comunicación Serial. - Una vez determinada la frecuencia, es necesario enviar esta información hacia el dispositivo que ha enviado el comando de lectura, esto se logra mediante el protocolo de comunicación “RS232” debido a su facilidad de implementación.

En las siguientes secciones se describen cada una de las tareas de este microprocesador.

2.1.1 Generación del Barrido de Frecuencias

Para poder verificar que el barrido de frecuencia que se está realizando sea correcto primero fue necesario seleccionar un modelo de cuerda para que, en un principio, se realizara una correcta lectura y luego se extrapolara el algoritmo a otras cuerdas con diferentes rangos de frecuencias a los cuales trabajaran.

Para este caso el modelo usado fue una Cuerda Vibrante de GEOKON de modelo 4000; el rango de frecuencias a la cual trabaja es de 400 a 1200 [Hz]

Se planteó que el barrido de frecuencia (de 400 a 1200 [Hz]) se realizara a lo largo de 500 [mS], pues el fabricante no especifica un tiempo dado, y al observar el comportamiento del barrido de frecuencias en una gráfica se logró caracterizar su comportamiento a lo largo de tiempo con la siguiente función

$$y = mx + b$$

Con:

$$y \Rightarrow \text{periodo actual}$$

$$b \Rightarrow \text{Tiempo Inicial } (T_i)$$

$$m = \frac{T_f - T_i}{500\,000[\mu S]}$$

$$x \Rightarrow \text{tiempo acumulado}$$

Para este tipo específico de cuerda, las frecuencias, inicial y final, son:

$$F_i = 400 \text{ [Hz]}$$

$$F_f = 1200 \text{ [Hz]}$$

Sustituyendo valores para obtener sus periodos:

$$T_i = b = \frac{1}{F_i} \Rightarrow \frac{1}{400} \Rightarrow b = 0.0025 \text{ [S]}$$

$$T_f = \frac{1}{F_f} \Rightarrow \frac{1}{1200} \Rightarrow 0.000833 \text{ [S]}$$

Por lo tanto:

$$m = \frac{T_f - T_i}{500\,000 [\mu\text{S}]} \Rightarrow \frac{833 - 2500 [\mu\text{S}]}{500\,000 [\mu\text{S}]} \Rightarrow -0.003334 \text{ [1]}$$

La función que describe el comportamiento del barrido de frecuencia es:

$$y = -0.003334x + 0.0025$$

Donde:

y. –representa el periodo de la frecuencia en el instante dado
x. –representa la delta de tiempo, en este caso son milisegundos

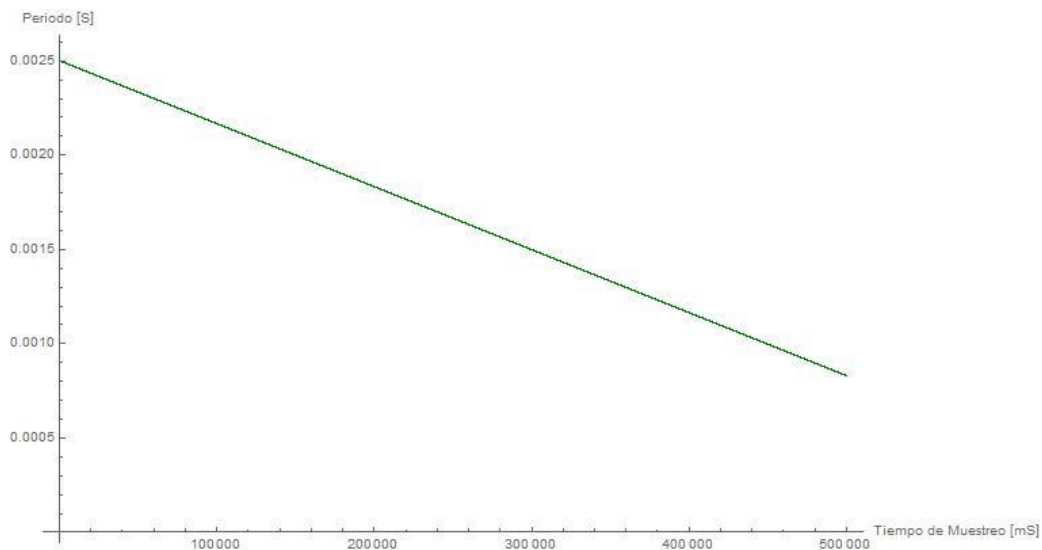


Figura 2.1 Comportamiento del Barrido de Frecuencias

Al observar la gráfica de la figura 2.1 que representa el comportamiento en el tiempo del barrido de frecuencia, se puede notar que los cambios entre cada periodo, con una base de tiempo de 1[μS], son muy pequeños, como se muestra en la Tabla 1.

Tabla 1 Barrido de Frecuencia

# de muestra	X	Y(Periodo)	y(frecuencia)
0	0	0.0025	400
1	0.000001	0.0025	400.000533
2	0.000002	0.00249999	400.001067
3	0.000003	0.00249999	400.0016
4	0.000004	0.00249999	400.002134
5	0.000005	0.00249998	400.002667

6	0.000006	0.00249998	400.003201
7	0.000007	0.00249998	400.003734
8	0.000008	0.00249997	400.004268
9	0.000009	0.00249997	400.004801
10	0.00001	0.00249997	400.005334
11	0.000011	0.00249996	400.005868
12	0.000012	0.00249996	400.006401
13	0.000013	0.00249996	400.006935

14	0.000014	0.00249995	400.007468
15	0.000015	0.00249995	400.008002
16	0.000016	0.00249995	400.008535
17	0.000017	0.00249994	400.009069
18	0.000018	0.00249994	400.009602
19	0.000019	0.00249994	400.010136
20	0.00002	0.00249993	400.010669
21	0.000021	0.00249993	400.011203
22	0.000022	0.00249993	400.011736
23	0.000023	0.00249992	400.012269
24	0.000024	0.00249992	400.012803
25	0.000025	0.00249992	400.013336
26	0.000026	0.00249991	400.01387
27	0.000027	0.00249991	400.014403
28	0.000028	0.00249991	400.014937
29	0.000029	0.0024999	400.01547
30	0.00003	0.0024999	400.016004
31	0.000031	0.0024999	400.016537

32	0.000032	0.00249989	400.017071
33	0.000033	0.00249989	400.017604
34	0.000034	0.00249989	400.018138
35	0.000035	0.00249988	400.018671
36	0.000036	0.00249988	400.019205
37	0.000037	0.00249988	400.019738
38	0.000038	0.00249987	400.020272
39	0.000039	0.00249987	400.020805
40	0.00004	0.00249987	400.021339
41	0.000041	0.00249986	400.021872
42	0.000042	0.00249986	400.022406
43	0.000043	0.00249986	400.022939
44	0.000044	0.00249985	400.023473
45	0.000045	0.00249985	400.024006
46	0.000046	0.00249985	400.02454
47	0.000047	0.00249984	400.025073
48	0.000048	0.00249984	400.025607
49	0.000049	0.00249984	400.02614

Por lo cual al revisar de nuevo la tabla 1 se logró observar que el cambio entre cada periodo en un instante dado puede ser despreciable, por lo cual se optó por generar un barrido de señal con una variación más amplia en los periodos de las señales. Al observar el patrón de las frecuencias, se puede apreciar que aproximadamente cada 3[μ S] se obtiene un cambio en el periodo de 0.001 [μ s] (Tabla 2), esto nos permite tener más tiempo para calcular y generar la frecuencia deseada con el microcontrolador en un instante dado sin generar un tono que afecte a la lectura del transductor.

Tabla 2 Barrido de frecuencia con una Nueva Delta de Tiempo

# de muestra	X	y(periodo)	y(frecuencia)
0	0	0.0025	400
1	0.000003	0.00249999	400.0016
2	0.000006	0.00249998	400.003201
3	0.000009	0.00249997	400.004801
4	0.000012	0.00249996	400.006401
5	0.000015	0.00249995	400.008002
6	0.000018	0.00249994	400.009602
7	0.000021	0.00249993	400.011203
8	0.000024	0.00249992	400.012803
9	0.000027	0.00249991	400.014403
10	0.00003	0.0024999	400.016004
11	0.000033	0.00249989	400.017604
12	0.000036	0.00249988	400.019205
13	0.000039	0.00249987	400.020805
14	0.000042	0.00249986	400.022406
15	0.000045	0.00249985	400.024006
16	0.000048	0.00249984	400.025607

17	0.000051	0.00249983	400.027207
18	0.000054	0.00249982	400.028808
19	0.000057	0.00249981	400.030408
20	0.00006	0.0024998	400.032009
21	0.000063	0.00249979	400.03361
22	0.000066	0.00249978	400.03521
23	0.000069	0.00249977	400.036811
24	0.000072	0.00249976	400.038411
25	0.000075	0.00249975	400.040012
26	0.000078	0.00249974	400.041613
27	0.000081	0.00249973	400.043213
28	0.000084	0.00249972	400.044814
29	0.000087	0.00249971	400.046415
30	0.00009	0.0024997	400.048015
31	0.000093	0.00249969	400.049616
32	0.000096	0.00249968	400.051217
33	0.000099	0.00249967	400.052818

Esto es muy conveniente porque así se logra tener un espacio más amplio para la generación del barrido de la señal, pues el generar el valor en microsegundos del periodo de una frecuencia toma algunos ciclos de reloj, y mientras esto se calcula no es posible generar una nueva frecuencia, como se muestra en la figura 2.2, y si el cálculo toma demasiados ciclos de reloj provoca una lectura incorrecta del transductor ya que se genera un tono en el barrido de frecuencia.

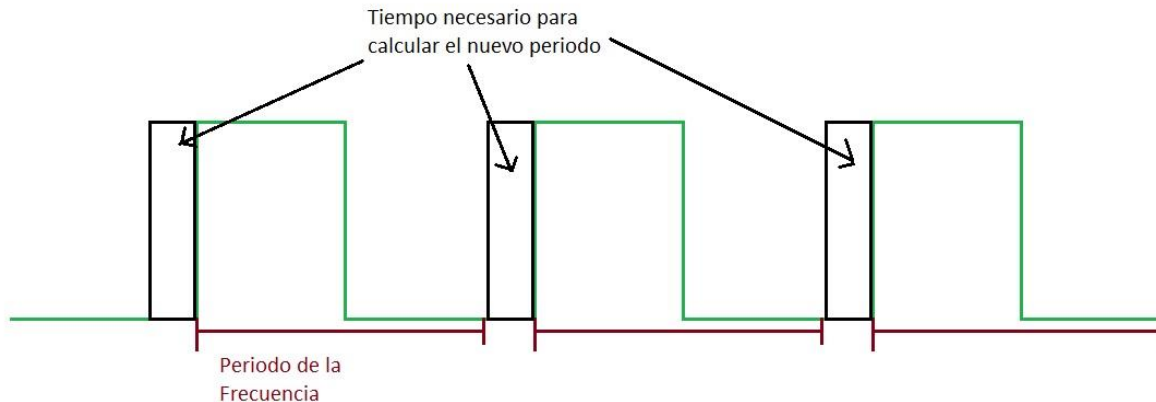


Figura 2.2 Inconvenientes del cálculo del periodo de cada señal del barrido

Ya que se tienen los valores de la función que describen el comportamiento del barrido de frecuencia, hay que notar que serían muchas las instrucciones que debería realizar el procesador si es que se desea programar la función para obtener cada valor del periodo de cada frecuencia que se desea alcanzar, por ello es más sencillo realizar una resta continua hasta alcanzar la frecuencia más alta, dado que el periodo (que es el valor que estamos obteniendo de la función) es inversamente proporcional a su frecuencia, solo se necesita contar con el valor del periodo inicial (que sería el periodo de la frecuencia más baja), la delta de tiempo y el periodo final (que estaría representando el valor de la frecuencia más alta) del barrido para poder generar la señal correcta para la lectura del transductor, a lo cual, la función de lo antes mencionado tendría un comportamiento similar a la función del comportamiento del barrido de la señal.

$$y = T_{inicial} - x \quad E \quad D \in [0, T_{final}] \quad x \in [R \mid x * \Delta t]$$

Se obtiene con esta función un rango de valores que corresponden a los periodos para la generación del barrido para lograr la lectura del transductor especificado al principio.

Pseudocódigo para generar el barrido de frecuencias

Por lo antes explicado es posible realizar el barrido de frecuencias mediante solo 3 valores, un periodo inicial, un periodo final y una delta, la cual fue calculada para este caso en la tabla 2 para permitir una menor carga de trabajo al dispositivo.

- ❖ Inicio
- ❖ Leer los valores de Periodo inicial, Periodo final y Delta
- ❖ PeriodoDeSeñal = PeriodoInicial / 2
- ❖ Mientras PeriodoDeSeñal sea menor que Periodofinal
 - Colocar el pin de salida en alto

- Esperar (PeriodoDeSeñal) milisegundos
 - Colocar el pin de salida en bajo
 - Esperar (PeriodoDeSeñal) milisegundos
 - $\text{PeriodoDeSeñal} = \text{PeriodoDeSeñal} - \text{Delta}$
- ❖ Activar la interrupción para la lectura del Transductor
- ❖ Fin.

2.1.2 Lectura del transductor

Después de generar el barrido de frecuencias, es necesario adecuar la señal que el transductor genera, la cual es la frecuencia de oscilación de la “Cuerda Vibrante”, posteriormente hay que cuantificar dicha frecuencia, para ello se utiliza un contador de pulsos y un contador de 16 bits con un cristal de una frecuencia más alta para poder realizar la comparación de los pulsos contra un tiempo definido.

Para comparar la frecuencia, fue necesario contabilizar los pulsos del contador de 16 bits entre el primer cambio de estado, de 0[V] a 5[V] del pulso que está enviando el transductor hasta el siguiente, la figura 2.3 muestra cómo se obtiene los datos.

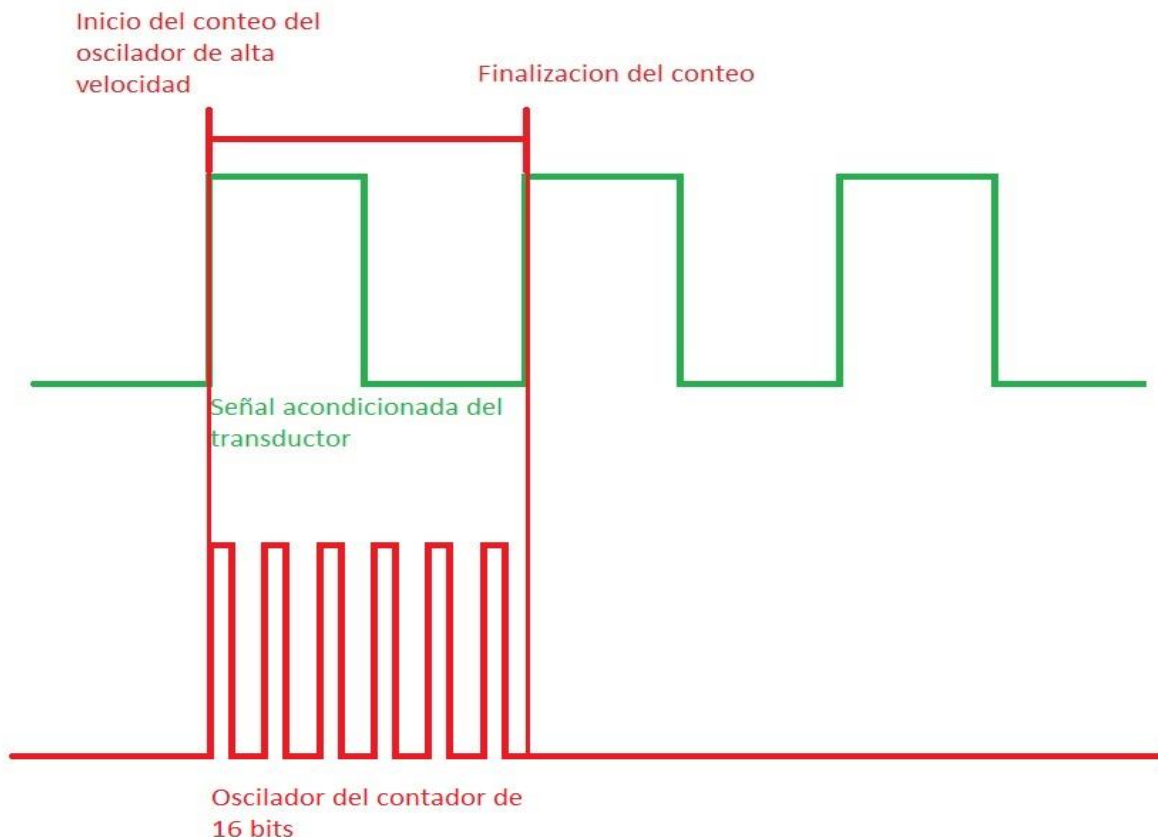


Figura 2.3 Método para determinar la Frecuencia del Transductor

Se sabe la velocidad del oscilador del contador de 16 bits, por lo tanto, al multiplicar la cantidad de pulsos que el contador ha registrado, es posible saber cuántos segundos transcurrieron desde un cambio de estado hasta el siguiente y así se sabrá el periodo de la señal y por tanto la frecuencia a la cual está vibrando la cuerda vibrante.

Pseudocódigo para la lectura del Transductor

El microcontrolador está realizando la cuenta de los pulsos mediante un pin dedicado a cuantificar los cambios de estado y estos serán contabilizados en un contador por hardware, cuando cambie de nivel, es decir de 0[V] a 5[V] la interrupción se llevará a cabo y es en este momento donde se realiza la contabilización de los pulsos, como se muestra en la figura 2.3.

- ❖ Inicio
- ❖ Inicializar el contador de 16 bits con 0 cuentas
- ❖ Activar la interrupción por cambio de estado en el pin donde se está recibiendo la señal adecuada desde el transductor de la cuerda vibrante
- ❖ Activar el contador de 16 bits Cuando se detecte el primer cambio de nivel en el pin de la señal de la cuerda vibrante
- ❖ ¿Se ha detectado otro cambio de nivel en el pin de la señal de la cuerda vibrante?
 - Si
 - ◆ Detener el contador de 16 bits
 - ◆ Desactivar la Interrupción por cambio de nivel
 - ◆ Transformar el número de cuentas del contador de 16 bits a caracteres ascii
 - ◆ Enviar los caracteres ascii mediante el protocolo RS232
 - No
 - ◆ Regresar al flujo de programa principal y esperar la interrupción
- ❖ Fin

2.1.3 Comunicación Serial

Tanto para la Interfaz USB como para la Interfaz Ethernet se está utilizando el protocolo “RS232” de comunicación, la única variación es que para la interfaz USB se añadió hardware adicional ya que el Microcontrolador de 8 bits no incorpora en su Hardware los requerimientos mínimos para establecer una comunicación USB maestro-esclavo, como se muestra en la figura 2.4.

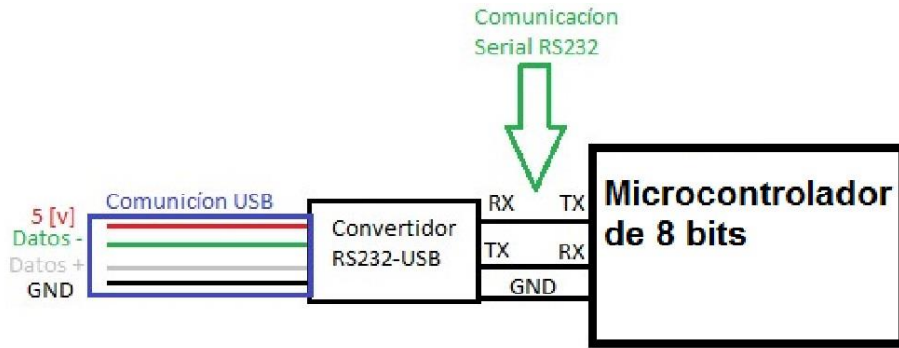


Figura 2.4 Convertidor RS232-USB para el Microcontrolador de 8 bits

Para la captura de datos a través de la conexión USB y la interfaz Ethernet es necesario un programa “Host” el cual solicite las lecturas, configure la lectura estableciendo el rango de frecuencias y el modo en que los multiplexores están programados (32 o 16 cuerdas) y además realice las conversiones de los pulsos leídos a una frecuencia o periodo y el almacenamiento de los datos adquiridos, por tanto el algoritmo no presenta modificaciones con respecto al firmware entre cada una de las interfaces utilizadas.

Pseudocódigo para selección de Comando

El microcontrolador recibe los bytes concernientes a los caracteres ascii de 8 bits del comando que el dispositivo está enviando, por lo cual es necesario discriminar estos caracteres para saber que comando se llevara a cabo y cuáles serán las instrucciones que se deben de ejecutar, para que el adquisidor reconozca y ejecute el comando es necesario incorporar los caracteres de “line feed” (‘\n’ ó 0x0A en hexadecimal) y “carriage return” (‘\r’ ó 0x0D en hexadecimal) al final de cada comando. Los comandos que reconoce el dispositivo son:

- LD\n\r
 - Lectura directa del transductor o de un canal del multiplexor que haya sido seleccionado
- M:##\tC:##\n\r
 - Lectura del multiplexor numero ##¹ y del canal ##, donde ## es un numero entre 01 hasta 99
- M:##\tC:##,##\n\r
 - Lectura del multiplexor numero ##¹ y de una serie de canales que abarcan del canal ## hasta el canal ##, donde ## es un numero entre 01 hasta 16 o 32, dependiendo de la configuración del multiplexor.
- F\n\r
 - Finalización de la adquisición de datos
- C:16\n\r ó C:32\n\r
 - Configuración del multiplexor, para lectura de 16 canales o 32 canales
- S:#\n\r
 - Configuración del rango de frecuencia del Transductor de Cuerda Vibrante, donde # puede ser alguno de los siguientes valores
 - Valor Rango[Hz]

¹ El adquisidor de datos aun no puede utilizar más de un multiplexor por ahora debido a los tiempos de desarrollo del proyecto.

- 0 450 – 1200
- 1 1200 – 3500
- 2 1000 – 3500
- 3 2500 – 6000
- 4 450 – 6000 (valor por default)

❖ Inicio

❖ Leer Carácter del buffer

- El carácter es igual al fin de cadena ('\r')?
 - No
 - ◆ Almacenar en el Buffer "*General*"
 - ◆ Regresar y esperar otro carácter
 - Si
 - ◆ Verificar el Primer Carácter del Buffer
 - El primer carácter es igual a 'M'
 - Si
 - ◆ Se Guarda en el Buffer del "*Multiplexor*" hasta que el carácter del Buffer "*General*" de recepción sea igual a "*tabulador*" ('\t')
 - ◆ Se descarta el carácter del "*tabulador*" ('\t')
 - ◆ Se Guarda en el Buffer del "*Canal inicial*" hasta que el carácter del Buffer "*General*" de recepción sea igual a ',' o fin de cadena '\r'
 - ◆ Si el carácter del Buffer "*General*" es igual a ',' entonces se Guarda en el Buffer del "*Canal Final*" hasta que el carácter del Buffer "*General*" de recepción sea igual a fin de cadena '\r'
 - ◆ Realizar las conversiones de caracteres ascii de los Buffer del "*Multiplexor*", "*Canal Inicial*" y "*Canal Final*" en su representación de un byte hexadecimal. Por ejemplo, el valor "20" en ascii representa el valor 0x14 en hexadecimal
 - ◆ Indicar al microcontrolador que va a realizar la lectura de un canal de un multiplexor o varios canales
 - ◆ Regresar al flujo principal del programa
 - No
 - El primer carácter es igual a 'L'
 - Si
 - ◆ ¿El siguiente carácter es igual a 'D'?
 - ◆ Si
 - ◆ Indicar al microcontrolador que lea el transductor de manera continua
 - ◆ No
 - ◆ Detener las lecturas del transductor y enviar al dispositivo maestro un mensaje de comando erróneo
 - No
 - El primer carácter es igual a 'F'
 - Si
 - ◆ Detener las lecturas del transductor
 - No

- El primer carácter es igual a 'C'
 - Si
 - ◆ Guardar en el Buffer de "Configuración" los caracteres que se encuentran después del carácter ':' hasta el fin de cadena '\r'
 - ◆ Realizar las conversiones de caracteres ascii del Buffer de "Configuración" en su representación de un byte hexadecimal
 - ◆ Verificar el numero en hexadecimal
 - ◆ Si es igual a 0x0F (16 en decimal)
 - ◆ Configurar el control de selección de canal del multiplexor para 16 canales
 - ◆ Si es igual a 0x1E (30 en decimal)
 - ◆ Configurar el control de selección de canal del multiplexor para 32 canales
 - ◆ Si no es ninguno de los anteriores
 - ◆ Configurar el control de selección de canal del multiplexor para 16 canales
 - No
- El primer carácter es igual a 'S'
 - Si
 - ◆ Saltar el carácter ':' y Verificar el siguiente carácter
 - ◆ Si el carácter es igual a '0'
 - ◆ Configurar el Control para realizar el barrido de frecuencia de 450 a 1200 [Hz]
 - Sino
 - ◆ Si el carácter es igual a '1'
 - ◆ Configurar el Control para realizar el barrido de frecuencia de 1200 a 3500 [Hz]
 - Sino
 - ◆ Si el carácter es igual a '2'
 - ◆ Configurar el Control para realizar el barrido de frecuencia de 1000 a 3500 [Hz]
 - Sino
 - ◆ Si el carácter es igual a '3'
 - ◆ Configurar el Control para realizar el barrido de frecuencia de 2500 a 6000 [Hz]
 - Sino
 - ◆ Si el carácter es igual a '4'
 - ◆ Configurar el Control para realizar el barrido de frecuencia de 450 a 6000 [Hz]
 - Sino
 - ◆ En cualquier otro caso
 - ◆ Configurar el Control para realizar el barrido de frecuencia de 450 a 6000 [Hz]
 - No
- En cualquier otro caso
 - ◆ Detener las lecturas del transductor y enviar al dispositivo maestro un mensaje de comando erróneo

❖ Fin

2.2 Funcionamiento del Programa de Conexión Directa mediante la interfaz USB

Dado que el adquirente de datos se comporta como un esclavo, ya que solo responde a peticiones del dispositivo maestro, la interfaz de comunicación con el dispositivo adquirente de datos se encarga de interpretar los datos y configuración del usuario en comandos para solicitar los datos del adquirente y posteriormente la generación de archivos en la máquina "host" donde se esté ejecutando el programa de Conexión Directa.

Pseudocódigo del Funcionamiento del Programa de conexión USB

El programa se basa en 3 partes primordiales:

1. Configuración
2. Adquisición Manual
3. Adquisición Automática

En las últimas 2 partes no es posible realizar ningún muestreo sin haber realizado una correcta configuración de la comunicación con el adquirente y el tipo de Muestreo.

1. Configuración

- ❖ Inicio
- ❖ Inicializar las variables de los datos necesarios para la adquisición de datos
- ❖ Verificación de la correcta configuración y comunicación con el Adquirente de datos
- ❖ ¿El Puerto de comunicación (COM) del Adquirente puede ser configurado correctamente?
 - Si
 - ◆ Continuar hacia la parte de configuración del muestreo
 - No
 - ◆ Avisar al usuario que el puerto COM no puede ser configurado
 - ◆ Continuar hacia la parte de configuración del muestreo
- ❖ Ir a la sección de Configuración del dispositivo
 - Al seleccionar un nuevo puerto COM
 - ◆ Configurar y revisar que se pueda realizar la comunicación
 - Si se logró configurar con éxito, esperar hasta que se seleccione algún tipo de adquisición de datos
 - En otro caso impedir la selección de algún tipo de adquisición de datos (Manual o Automática)
 - Adquirir los datos de Configuración del Multiplexor o la Cuerda Vibrante
 - ◆ Guardar el Rango de Frecuencia del Transductor
 - ◆ Guardar el Número de canales del Multiplexor, si este ha sido seleccionado
- ❖ Al cambiar de sección

- Si el Puerto COM este configurado correctamente, proceder con el cambio de sección
- Si no, impedir el cambio de sección y regresar a la sección de Configuración.

2. Adquisición Manual

- ❖ En la sección de Adquisición Manual
 - Mientras la Sección no Cambie
 - ◆ Enviar la Configuración del Rango de Frecuencia
 - ◆ Si No está seleccionado la adquisición mediante un multiplexor
 - Enviar el Comando de Lectura Directa Constante de la Cuerda Vibrante (solo una vez)
 - ◆ Si está seleccionado la adquisición mediante un Multiplexor
 - Enviar la Configuración del Multiplexor (16 canales (cuerda + termistor) o 32 canales (solo cuerdas)) (Solo una vez)
 - Si el Canal a adquirir cambia, actualizar el comando de Lectura de Canal
 - Enviar el Comando para Lectura de Canal
 - ◆ Leer Datos del Buffer
 - ◆ Realizar las operaciones necesarias para convertir la cantidad de Pulsos del Canal en una frecuencia y posteriormente en microdeformaciones
 - ◆ Realizar las operaciones necesarias para convertir los bits del convertidor A/D en una temperatura

3. Adquisición Automática

- ❖ En la sección de Adquisición Automática
 - Mientras la sección no Cambie
 - ◆ Obtener la Ruta donde los archivos serán Creados y las Muestras serán Almacenadas
 - ◆ Obtener el Tipo de Muestra deseado, ya sea una lista de canales secuenciales o sin una secuencia
 - ◆ Si el Tipo de Muestra es lista de Canales Secuenciales
 - Generar los Comandos y las cabeceras de archivo con los canales seleccionados
 - ◆ Si el Tipo de Muestra es lista de Canales No Secuencial
 - Verificar que los canales seleccionados estén en el rango de canales permitidos
 - Generar los Comandos y las cabeceras de archivo con los canales seleccionados
 - ◆ Obtener El tiempo de Muestreo
 - ◆ Si se inicializa el Muestreo
 - Si la Ruta no Existe o no fue configurada por el usuario
 - ◆ Preguntar al usuario si desea Utilizar una Ruta Predefinida o desea seleccionar una
 - ◆ Si desea Seleccionar alguna, regresar y Verificar si existe o no la nueva Ruta

- ◆ Si desea utilizar la Ruta Predefinida, verificar la existencia de los archivos en la Ruta Predefinida
- Si en la Ruta Seleccionada No existen los archivos
 - ◆ Crear las carpetas y los archivos para almacenar la Muestra
- En caso contrario
 - ◆ Preguntar al usuario si desea Crear Un nuevo Archivo o Sobrescribir el existente
 - ◆ Si desea Crear uno nuevo, colocar el mismo nombre del archivo y concatenar un nuevo subíndice numérico que no exista y almacenar los nuevos datos
 - ◆ Si desea Sobrescribir el archivo, almacenar los nuevos datos en el archivo
- ¿Se ha terminado el tiempo para realizar la siguiente muestra?
 - ◆ Si, se ha Terminado el tiempo
 - ◆ Por cada canal de la lista de canales
 - ◆ Enviar Comando de Lectura de canal
 - ◆ Leer Datos del Buffer
 - ◆ Realizar las Operaciones para la conversión de los datos en Microdeformaciones y Temperatura
 - ◆ Almacenar los datos en el archivo Previamente creado
 - ◆ No, aun no se Termina el tiempo
 - ◆ ¿Existe el archivo del día Actual?
 - ◆ Si existe
 - ◆ Continuar almacenando los datos de la muestra en el archivo
 - ◆ No existe
 - ◆ Crear un nuevo archivo y colocar las cabeceras de los datos de la muestra en el nuevo archivo y seleccionar este como archivo del día Actual
- Si algún dato de la muestra cambia
 - ◆ Preguntar al Usuario si desea cambiar algún dato de la muestra, como la ruta, la cantidad de canales o el orden de los mismos
 - ◆ Si desea cambiar algún dato
 - ◆ Cargar nuevo dato e Inicializar el Muestreo
 - ◆ Si No desea cambiar algún dato
 - ◆ Mantener los datos y continuar con la muestra

❖ Fin

2.3 Micro-servidor Web e interfaz Ethernet (Microcontrolador de 32 bits)

Este microcontrolador tendrá a su cargo la tarea de atender las peticiones de conexión que sean solicitadas por los navegadores web de los usuarios, almacenar las configuraciones realizadas por el usuario para las Muestras programadas y la adquisición de datos, de manera continua y programada, a través de la intercomunicación serial “RS232”. Por ello las tareas se pueden dividir en 2 grandes rutinas, las Funciones como Servidor Web y las Funciones de Adquisidor de datos pues la intercomunicación serial estará siendo únicamente atendida en el momento en que se necesite escribir o leer a partir de una interrupción del microcontrolador. A continuación, se describen a grandes rasgos estas rutinas y el método de comunicación entre los Microcontroladores.

2.3.1 Intercomunicación Serial (RS232)

El esquema de comunicación entre los 2 microcontroladores es una comunicación serial por el estándar “RS232”, por lo cual fue necesario conectar ambos microcontroladores como se muestra en la figura 2.5.

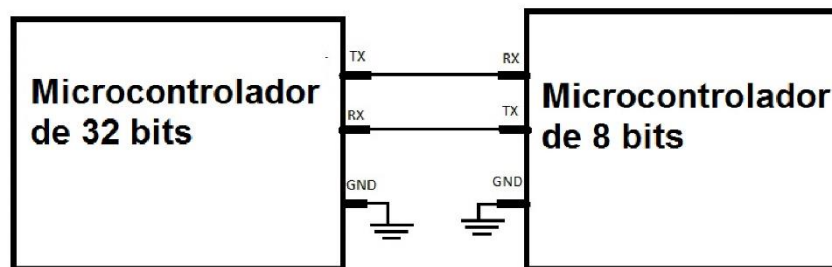


Figura 2.5 Esquema de conexión para la comunicación RS232 entre los Microcontroladores

El Microcontrolador de 32 bits debe realizar varias tareas para poder mantener la integridad del Protocolo TCP/IP, por lo que cada tarea debe ser dividida en pequeñas tareas que no mantengan el microprocesador ocupado e incapaz de recibir y enviar tramas a la red Ethernet, por lo que para esta aplicación es necesario utilizar una interrupción del microcontrolador de 32 bits para almacenar el byte que se envía desde el microcontrolador de 8 bits mediante “RS232” y colocarlo en una cola circular, la cual será leída por el microcontrolador de 32 bits y procesada para ser presentada en la página web.

El funcionamiento de la interrupción (al encolar los nuevos datos) y las funciones (al desencolar los datos) es el siguiente:

Las etiquetas de RX Cola, RX Cabeza, TX Cola y TX Cabeza se refieren a datos de tipo apuntador que se utilizaran para saber cuál fue el último dato encolado y cuál fue el último dato desencolado.

La interrupción encola un dato cada vez que se ejecuta, en un principio la cola está vacía (figura 2.6)

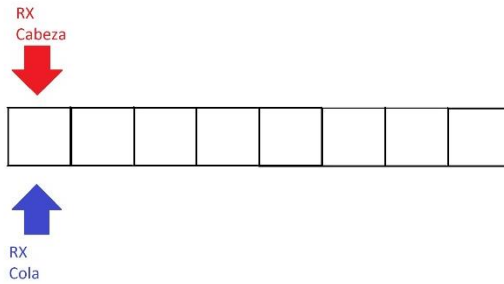


Figura 2.6 Cola de Datos Recibidos Vacía

Luego de la primera interrupción un dato ha sido encolado (figura 2.7)

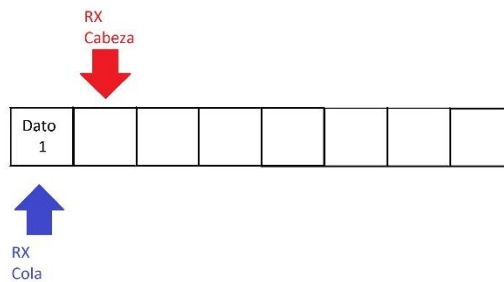


Figura 2.7 Cola de Datos Recibidos con un dato nuevo

Y así sucesivamente después de cada interrupción en donde se recibe un nuevo dato (figura 2.8)

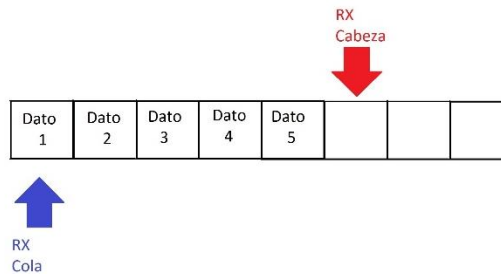


Figura 2.8 Cola de Datos Recibidos con varios datos Nuevos

Cuando el procesador realiza las tareas de lectura de la cola circular, primero realiza una copia de la localidad a la que apunta RX Cabeza, pues al estar siendo atendida la interrupción por la llegada de un nuevo dato, la localidad de RX Cabeza puede cambiar y dar lugar a errores y fallas en la interpretación de los caracteres, pues el algoritmo de lectura de la cola da datos comienza la lectura desde la localidad de RX Cola hasta RX Copia Cabeza.

En la figura 2.9 se observa el caso en el cual no se han encolado nuevos datos

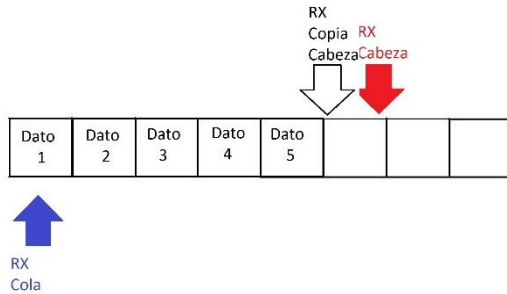


Figura 2.9 Cola de Datos Recibidos antes de desencolar los nuevos datos

La figura 2.10 muestra el caso en el cual se han encolado nuevos datos, pero el índice de RX Cabeza aun es mayor que RX Cola

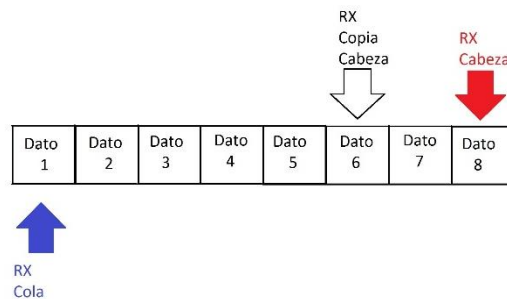


Figura 2.10 Cola de Datos Recibidos encolando nuevos datos antes de desencolar los anteriores

Una vez que se han leído los datos (es decir, que la localidad de RX Cola es igual a la de RX Copia Cabeza) se libera el procesador para ser ocupado por otra tarea y se pueden recibir nuevos datos desde el microcontrolador de 8 bits, tal como la figura 2.11 muestra.

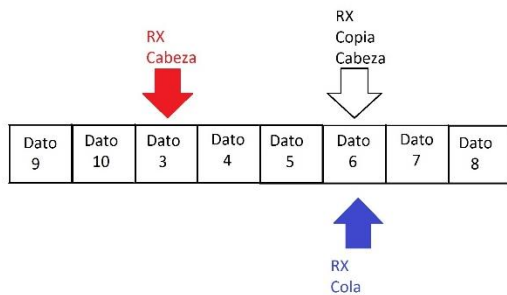


Figura 2.11 Cola de Datos Recibidos con los datos anteriores desencolados

Así los procesos no se interrumpen y los datos no se perderán, permitiendo que el procesador atienda otras tareas sin necesidad de mantenerse a la espera de todos los caracteres que el microcontrolador de 8 bits tenga que enviar.

Para enviar datos hacia el Microcontrolador de 8 bits el proceso es muy similar, primero la cola está vacía (figura 2.12)

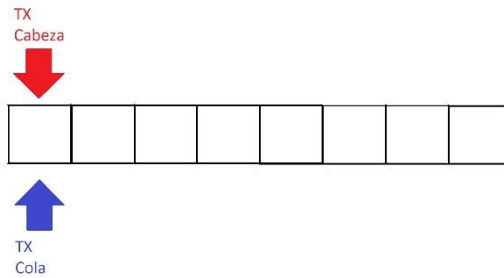


Figura 2.12 Cola de Datos Para Transmitir vacía

Encolamos los datos que se desean enviar mediante una función, por ahora la interrupción no está activada para no enviar aun los datos (figura 2.13)

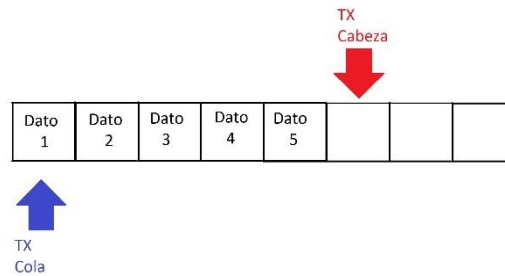


Figura 2.13 Cola de Datos Para Transmitir con varios datos Nuevos

Cuando los datos han terminado de encolarse, se activa la interrupción para enviar los datos y estos se enviarán lo más pronto posible mediante la interrupción de envío de datos. Si por alguna circunstancia se colocan más datos, es necesario realizar una copia de TX Cabeza para poder insertar nuevos datos y evitar algún error, como se muestra en la figura 2.14.

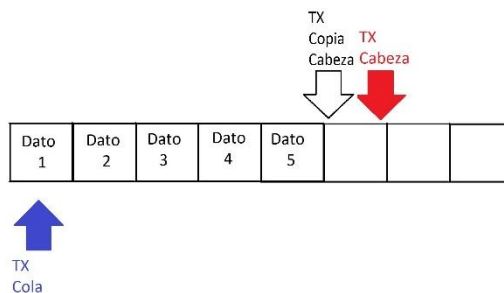


Figura 2.14 Cola de Datos Para Transmitir antes de encolar los nuevos datos

En el caso de la figura 2.15, TX Copia Cabeza está siendo usada para colocar los datos nuevos y posteriormente se actualizará la localidad de TX Cabeza para que los datos sean enviados desde TX Cola hasta TX Cabeza.

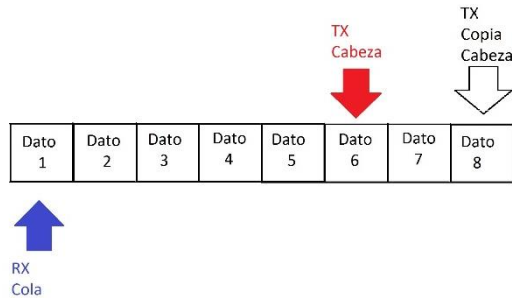


Figura 2.15 Cola de Datos Para Transmitir antes de Actualizar el apuntador a los nuevos datos

Actualizando la localidad de memoria (figura 2.16).

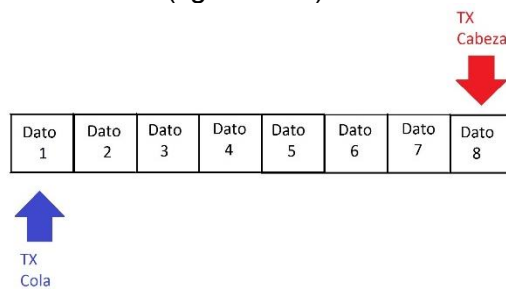


Figura 2.16 Cola de Datos Para Transmitir con los Nuevos Datos Encolados

De esta manera se pueden transmitir y recibir datos a través de la interfaz serial del microcontrolador de 32 bits mediante el protocolo “RS232”.

2.3.2 Funciones de Adquisidor de datos

2.3.2.1 Configuración de la Adquisición

Es necesario saber qué tipo de configuración está colocada en el multiplexor, que rango de frecuencia tiene cada cuerda colocada en el multiplexor o si solo una cuerda está siendo leída, además del tiempo en el que se desea que la lectura sea almacenada en una memoria MicroSD, para lo cual el microcontrolador cuenta con una función con la que recibe los datos de configuración desde la página web. Con esto se puede configurar la captura de datos continua y programada además de seleccionar la configuración del multiplexor o de la única cuerda.

Pseudocódigo para la captura de datos de la configuración desde la página llamada “Configuracion.html”

La secuencia en que se reciben los datos desde la página es la siguiente:

- Tipo de Adquisición: Ya sea utilizar el multiplexor con 16 o 32 cuerdas o la única Cuerda Vibrante
- Tiempo de Muestra: La cantidad de tiempo que debe transcurrir antes de realizar una Muestra y guardar los datos en memoria

- Todos los datos de la cuerda o cuerdas que se estén configurando de la siguiente manera
 - Primero se verifica que canal está siendo enviado y posteriormente se verifica si se ha encendido ese canal o si la única cuerda se ha enviado
 - La Frecuencia del canal ya verificado en el paso anterior

- ❖ Inicio
- ❖ La Rutina que atiende las sesiones de los clientes manda a llamar la función que permite la captura de datos
 - Leer el tipo de adquisición
 - Leer el Tiempo de Muestreo
 - Asignar el Tiempo de Muestreo a la Adquisición mediante Multiplexor y Cuerda Única
 - Verificar el tipo de adquisición
 - ◆ Si es Lectura Directa de la Única Cuerda
 - Activar la lectura continua y programada de la Cuerda Única
 - Leer el Rango de frecuencia de la Cuerda
 - ◆ Si es Lectura de Multiplexor configurado con 16 canales
 - Activar la lectura continua y programada de la lectura del multiplexor
 - Configurar el Adquisidor de datos para realizar la lectura en modo de 16 canales
 - Mientras $n \leq 16$
 - ◆ ¿El nombre del canal leído es igual a “canal” + n?
 - ◆ Si, verificar si el canal seleccionado está activado
 - ◆ Si está activado, leer el rango de frecuencia de este canal
 - ◆ Si No es igual, $n=n+1$
 - ◆ Si es Lectura de Multiplexor configurado con 32 canales
 - Activar la lectura continua y programada de la lectura del multiplexor
 - Configurar el Adquisidor de datos para realizar la lectura en modo de 32 canales
 - Mientras $n \leq 32$
 - ◆ ¿El nombre del canal leído es igual a “canal” + n?
 - ◆ Si, verificar si el canal seleccionado está activado
 - ◆ Si está activado, leer el rango de frecuencia de este canal
 - ◆ Si No es igual, $n=n+1$
 - ◆ En cualquier otro caso
 - Desactivar todas las lecturas

❖ Fin

2.3.2.2 Funciones para la Adquisición de datos

La adquisición de datos trabaja de 2 maneras distintas continua y programada, en las cuales se están enviando y recibiendo bytes de información desde el lector de cuerda vibrante de manera constante, por lo cual ha sido necesario realizar 2 rutinas independientes pero que trabajan de manera muy similar. La rutina de adquisición Continua está presente todo el tiempo antes de que se cumpla el tiempo de la muestra, permitiendo así una lectura de todos los instrumentos para verificar su funcionamiento o algún error de lectura. La rutina de adquisición Programada impide que se lleve a cabo la adquisición Continua para evitar cambios bruscos en el multiplexor.

2.3.2.2.1 Adquisición Continua

Esta rutina está enviando los comandos necesarios para realizar un Muestreo con la mayor velocidad posible para el adquirente de datos, pues es necesario un tiempo de adquisición para cada una de las cuerdas de aproximadamente 1[s].

Pseudocódigo para la Adquisición Continua

❖ Inicio

❖ ¿Se está ejecutando una Lectura programada?

➤ No,

➤ ¿Se activó la lectura Directa de la Única Cuerda?

▪ Si,

- Enviar la configuración del Rango de Frecuencia
- Enviar el Comando de Lectura Directa de la Cuerda
- ¿Hay bits en la cola de recepción de datos?

◆ No, regresar al proceso principal

◆ Si, leer los bits que existen en la cola y colocar en un buffer

➤ ¿Alguno de los bits es el fin de cadena (r)?

➤ No, regresar al proceso principal y esperar más bits

➤ Si,

▪ Enviar el Comando de Fin de lectura de canal

▪ Verificar que los datos del buffer cumplan con la estructura "F:%ul,%u\T:%u\r" [donde %ul significa un número sin signo]

▪ ¿Cumple la condición?

▪ No, limpiar el buffer y pedir nuevamente la lectura del dato

▪ Si,

• Convertir los datos en microdeformaciones y temperatura

• Almacenar los datos en el espacio de memoria asignado en el microcontrolador

• Regresar al Inicio del Algoritmo

▪ No, verificar siguiente condición

➤ ¿Se activó la lectura mediante el Multiplexor?

▪ Si,

- Enviar la configuración del Multiplexor (16 o 32 cuerdas)
 - Mientras $n \leq$ ultima cuerda configurada
 - ◆ Si $n \geq$ ultima cuerda configurada
 - $n =$ primera cuerda configurada
 - ◆ Desde $n = 1$ hasta $n \leq$ ultima cuerda configurada con incrementos $n = n + 1$
 - Si la cuerda n esta activada
 - Salir del ciclo
 - En otro caso
 - Continuar en el ciclo
 - ◆ Enviar la configuración del Rango de Frecuencia de la cuerda n
 - ◆ Enviar el Comando de Lectura del Canal n de la Cuerda
 - ◆ ¿Hay bits en la cola de recepción de datos?
 - No, regresar al proceso principal
 - Si, leer los bits que existen en la cola y colocar en un buffer
 - ¿Alguno de los bits es el fin de cadena ($\backslash r$)?
 - No, regresar al proceso principal y esperar más bits
 - Si,
 - ◆ Enviar el Comando de Fin de lectura de canal
 - ◆ Verificar que los datos del buffer cumplan con la estructura “ $F: \%ul, \%u\backslash r T: \%u\backslash r$ ” [donde $\%ul$ significa un numero largo sin signo]
 - ◆ ¿Cumple la Condición?
 - ◆ No, limpiar el buffer y regresar a enviar el Comando de Lectura Nuevamente
 - ◆ Si,
 - ◆ Convertir los datos en microdeformaciones y temperatura
 - ◆ Almacenar los datos en el espacio de memoria asignado en el microcontrolador para el valor actual de la Cuerda n
 - ◆ Hacer $n = n + 1$ para buscar la siguiente cuerda que se desea leer
 - No, regresar al inicio.
- ❖ Fin

2.3.2.2.2 Adquisición Programada

Esta rutina es muy similar a la anterior, solo que esta se ejecutara cuando el plazo de tiempo haya finalizado y guardara los datos en la memoria MicroSD² generando así una lista de archivos por día, una carpeta por mes y una por año que almacenara las carpetas de cada mes.

² Por el momento no está implementada la rutina para almacenar los datos en la memoria microSD debido a los plazos de entrega del proyecto, aunque ya ha sido probada en proyectos previos con éxito permitiendo leer, crear y borrar archivos en memorias MicroSD de diferentes capacidades

Pseudocódigo para la Adquisición Programada

- ❖ Inicio
- ❖ ¿Se activó la lectura Directa de la Única Cuerda?
 - Si
 - ¿Se ha cumplido el plazo de tiempo?
 - No, regresar al flujo principal
 - Si, detener la Adquisición Continua
 - ¿El listado de archivos existe?[Carpeta del año, dentro de esta la carpeta del mes y el archivo del día]
 - ◆ No, crear el listado de carpetas y/o el archivo del día, colocar las cabeceras de datos necesarias
 - ◆ Enviar la configuración del Rango de Frecuencia
 - ◆ Enviar el Comando de Lectura Directa de la Cuerda
 - ◆ ¿Hay bits en la cola de recepción de datos?
 - No, regresar al proceso principal
 - Si, leer los bits que existen en la cola y colocar en un buffer
 - ¿Alguno de los bits es el fin de cadena ($\backslash r$)?
 - No, regresar al proceso principal y esperar más bits
 - Si,
 - ◆ Enviar el Comando de Fin de lectura de canal
 - ◆ Verificar que los datos del buffer cumplan con la estructura “ $F:\%ul,\%u\backslash tT:\%u\backslash r$ ” [donde $\%ul$ significa un número sin signo]
 - ◆ ¿Cumple la condición?
 - ◆ No, limpiar el buffer y pedir nuevamente la lectura del dato
 - ◆ Si,
 - ◆ Convertir los datos en microdeformaciones y temperatura
 - ◆ Almacenar los datos en el espacio de memoria asignado en el microcontrolador
 - ◆ Guardar los Datos en el Archivo del día
 - ◆ Activar la Adquisición Continua
 - ◆ Regresar al Inicio del Algoritmo
 - No, verificar siguiente condición
 - ❖ ¿Se activó la lectura mediante el Multiplexor?
 - Si,
 - ¿Se ha cumplido el plazo de tiempo?
 - No, regresar al flujo principal
 - Si, detener la Adquisición Continua
 - ◆ ¿El listado de archivos existe?[Carpeta del año, dentro de esta la carpeta del mes y el archivo del día]
 - No, crear el listado de carpetas y/o el archivo del día, colocar las cabeceras de datos necesarias
 - ◆ Enviar la configuración del Multiplexor (16 o 32 cuerdas)
 - ◆ Mientras $n \leq$ ultima cuerda configurada
 - Si $n \geq$ ultima cuerda configurada
 - Activar la Adquisición Continua
 - Regresar al Inicio del Algoritmo

- Desde $n=1$ hasta $n \leq$ última cuerda configurada con incrementos $n=n+1$
 - Si la cuerda n está activada
 - Salir del ciclo
 - En otro caso
 - Continuar en el ciclo
 - Enviar la configuración del Rango de Frecuencia de la cuerda n
 - Enviar el Comando de Lectura del Canal n de la Cuerda
 - ¿Hay bits en la cola de recepción de datos?
 - No, regresar al proceso principal
 - Si, leer los bits que existen en la cola y colocar en un buffer
 - ¿Alguno de los bits es el fin de cadena ($\backslash r$)?
 - ◆ No, regresar al proceso principal y esperar más bits
 - ◆ Si,
 - ◆ Enviar el Comando de Fin de lectura de canal
 - ◆ Verificar que los datos del buffer cumplan con la estructura “ $F:\%ul,\%u\backslash tT:\%u\backslash r$ ” [donde $\%ul$ significa un número largo sin signo]
 - ◆ ¿Cumple la Condición?
 - ◆ No, limpiar el buffer y regresar a enviar el Comando de Lectura Nuevamente
 - ◆ Si,
 - ◆ Convertir los datos en microdeformaciones y temperatura
 - ◆ Almacenar los datos en el espacio de memoria asignado en el microcontrolador para el valor actual de la Cuerda n
 - ◆ Guardar los Datos en el Archivo del día
 - ◆ Hacer $n=n+1$ para buscar la siguiente cuerda que se desea leer
- No, regresar al inicio.
- ❖ Fin

2.3.3 Funciones como Servidor Web

2.3.3.1 Comunicación TCP/IP

Para realizar la comunicación con la red Ethernet es necesario seguir los protocolos que definen los estándares de comunicación en la red, para lograr esto se utilizó el “*Stack TCP/IP*” de Microchip, el cual nos permite generar la trama de datos para la intercomunicación de las aplicaciones de una manera más sencilla, por lo cual la programación se enfocó en los algoritmos para la adquisición de datos, administración de las opciones de muestreo y la presentación de los resultados. La programación solo se realiza en la capa de aplicación del modelo TCP/IP, mostrado en la figura 2.17.



Figura 2.17 Modelo TCP/IP

El "Stack TCP/IP" de Microchip solo puede atender a un cliente por vez, pero para simular que puede atender a varios clientes a la vez lo que realiza el "Stack" es que cada sesión de cada cliente es guardada en una estructura de datos estática, para que la rutina que atiende estas peticiones pueda cargar los datos de cada sesión y continuar el envío de información desde el último punto donde se guardaron los datos. Así no mantendrá ocupado el procesador para permitir que las demás rutinas puedan trabajar sin contratiempos. Cada sesión que el microcontrolador almacena un id para la conexión TCP y una localidad de memoria para que se almacene el estado en cual la rutina se detuvo.

Pseudocódigo del microservidor de HTTP

- ❖ Inicio
- ❖ Por cada Conexión programada
 - ¿Se ha desconectado el cliente?
 - ◆ Si, reinicializar los valores de la conexión del cliente desconectado.
 - ¿La máquina de estados que atiende de la conexión actual es distinta de "Inactivo" o existen bytes de información por leer?
 - ◆ Si
 - Sincronizar los datos de la sesión
 - Procesar la trama de datos del paquete, leyendo que tipo de transferencia de datos que se está utilizando, la página web a la que desea acceder el cliente y mandar a llamar las funciones de la capa de aplicación
- ❖ Fin

2.3.3.2 Rutinas para el control, lectura y configuración del Adquisidor de datos

Para realizar la configuración del Adquisidor de datos es necesario que el usuario coloque los datos de la cuerda (o cuerdas) que desea leer y, de ser el caso, colocar la manera en la cual el multiplexor está configurado; ya sean 16 canales los cuales contemplan la lectura de la Cuerda Vibrante y el Termistor que tienen integrados los cables de instrumentación o 32 canales que solo contemplan la lectura de la Cuerda Vibrante. Para esto se ha realizado una página web que se guarda los datos de la muestra en el microcontrolador de 32 bits para realizar las peticiones correctas al Adquisidor de datos. Las siguientes figuras muestran las opciones de la página web y los resultados en la página que presenta los datos recibidos.

Pseudocódigo de las Páginas Web

1. Página: “*index.htm*”

- ❖ Inicio
- ❖ Al Cargar todos los Elementos de la Página Web
 - Verificar el Tipo de Configuración de la Muestra:
 - ◆ Lectura Directa de la Cuerda Vibrante
 - Crear una tabla con 4 columnas y 2 filas
 - ◆ Lectura de 16 Canales
 - Crear una tabla con 4 columnas y 17 filas
 - ◆ Lectura de 32 Canales
 - Crear una tabla con 3 columnas y 33 filas
 - ◆ Sin una muestra Programada
 - Muestra un letrero que indique que no hay ningún tipo de Lectura Programada
 - ◆ En cualquier otro caso
 - Mostrar un mensaje que indique que la opción no es correcta
- ❖ Cada n cantidad de tiempo
 - Solicitar la página XML donde están los almacenados los datos enviados desde el microservidor
 - Verificar el Tipo de Configuración de la Muestra:
 - ◆ Lectura Directa de la Cuerda Vibrante
 - Buscar el dato de la Lectura Directa de la Cuerda en la página XML
 - Buscar el dato de la Temperatura Leída en la página XML
 - Colocar los datos en las Etiquetas creadas en la tabla al cargar la página.
 - ◆ Lectura de 16 Canales
 - Desde $n=1$ hasta $n=16$ con incrementos de $n=n+1$
 - ◆ Buscar el dato de la Lectura del canal n-simo en la página XML
 - ◆ Buscar el dato de la Temperatura del canal n-simo en la página XML
 - ◆ Colocar los datos en la Etiqueta creada para el n-simo canal
 - ◆ Lectura de 32 Canales

- Desde $n=1$ hasta $n=16$ con incrementos de $n=n+1$
 - ◆ Buscar el dato de la Lectura del canal n -simo en la página XML
 - ◆ Colocar el dato en la Etiqueta creada para el n -simo canal
 - ◆ Sin una muestra Programada
 - No realizar ninguna Búsqueda
 - ◆ En cualquier otro caso
 - No realizar Ninguna Búsqueda
- ❖ Fin

2. Página: “*configuracion.html*”

- ❖ Inicio
- ❖ Al Cargar todos los Elementos de la Página Web
 - Crear un Botón de “Cancelar Configuración”
 - Crear un Botón para enviar el formulario con la Nueva Configuración del Usuario hacia el Microcontrolador
 - Crear una lista de selección para Configurar el tipo de Muestra que el Usuario Desea Realizar
 - Verificar el Tipo de Configuración de la Muestra que está Almacenado en el microcontrolador:
 - ◆ Si es: Lectura Directa de la Cuerda Vibrante
 - Crear un formulario
 - Crear una casilla de selección y una lista de opciones para configurar la frecuencia de la Cuerda Vibrante e incluirlo al formulario
 - Crear un Botón para Enviar los Datos del formulario
 - ◆ Si es: Lectura de 16 Canales
 - Crear un formulario
 - Desde $n=1$ hasta $n=16$ con incrementos de $n=n+1$
 - ◆ Crear una casilla de selección y colocarle el id “canal” y concatenarle el numero n
 - ◆ Crear una lista de opciones para configurar la frecuencia de esa cuerda vibrante y colocarle el id “frecuencia” y concatenarle el numero n
 - ◆ Adjuntar al formulario la casilla y la lista de selección
 - Crear un Botón para Enviar los Datos del formulario
 - ◆ Si es: Lectura de 32 Canales
 - Crear un formulario
 - Desde $n=1$ hasta $n=16$ con incrementos de $n=n+1$
 - ◆ Crear una casilla de selección y colocarle el id “canal” y concatenarle el numero n
 - ◆ Crear una lista de opciones para configurar la frecuencia de esa cuerda vibrante y colocarle el id “frecuencia” y concatenarle el numero n
 - ◆ Adjuntar al formulario la casilla y la lista de selección
 - Crear un Botón para Enviar los Datos del formulario
 - ◆ Si es: Sin una muestra Programada
 - Muestra un letrero que indique qué no hay ningún tipo de Lectura Programada

- ◆ En cualquier otro caso
 - Mostrar un mensaje que indique que la opción no es correcta
- ❖ Cada Vez que la lista de selección para Configurar El tipo de Muestra cambia de valor
 - Si el nuevo Valor es:
 - ◆ Lectura Directa de la Cuerda Vibrante
 - Si el formulario contiene las 16 o 32 casillas de selección
 - ◆ Borrar las casillas de selección y las listas de selección de frecuencias
 - Crear una casilla de selección y una lista de opciones para configurar la frecuencia de la Cuerda Vibrante e incluirlo al formulario
 - ◆ Lectura de 16 Canales
 - Si el formulario contiene la casilla de selección para la Lectura Directa
 - ◆ Borrar la casilla de selección y la lista de selección para la frecuencia
 - En otro caso
 - ◆ Si el formulario contiene las 32 casillas de selección
 - ◆ Borrar las casillas de selección y las listas de selección de frecuencias
 - Desde $n=1$ hasta $n=16$ con incrementos de $n=n+1$
 - ◆ Crear una casilla de selección y colocarle el id "canal" y concatenarle el numero n
 - ◆ Crear una lista de opciones para configurar la frecuencia de esa cuerda vibrante y colocarle el id "frecuencia" y concatenarle el numero n
 - ◆ Adjuntar al formulario la casilla y la lista de selección
 - ◆ Lectura de 32 Canales
 - Si el formulario contiene la casilla de selección para la Lectura Directa
 - ◆ Borrar la casilla de selección y la lista de selección para la frecuencia
 - En otro caso
 - ◆ Si el formulario contiene las 32 casillas de selección
 - ◆ Borrar las casillas de selección y las listas de selección de frecuencias
 - Desde $n=1$ hasta $n=32$ con incrementos de $n=n+1$
 - ◆ Crear una casilla de selección y colocarle el id "canal" y concatenarle el numero n
 - ◆ Crear una lista de opciones para configurar la frecuencia de esa cuerda vibrante y colocarle el id "frecuencia" y concatenarle el numero n
 - ◆ Adjuntar al formulario la casilla y la lista de selección
 - ◆ Sin una muestra Programada
 - Muestra un letrero que indique que no hay ningún tipo de Lectura Programada
 - ◆ En cualquier otro caso
 - Mostrar un mensaje que indique que la opción no es correcta

❖ Fin

3 Interfaces Graficas para el Usuario

Las Interfaces que se realizaron para la configuración y muestreo de datos han tratado de ser lo más sencillas posible para el usuario.

3.1 Programa de comunicación a través del puerto USB

El programa que realiza las consultas al dispositivo lector de Cuerda Vibrante tiene 3 funciones principales.

3.1.1 Configuración de la Muestra

Antes de comenzar la generación de archivos, hay que definir cuál será el rango de frecuencias en el cual trabajan las Cuerdas Vibrantes, cuál será el modo de funcionamiento de dispositivo Adquisidor, es decir, solo tomara lecturas de una cuerda o existe un multiplexor con varias cuerdas vibrantes y el modo de operación, si existe, del multiplexor, en modo de 16 canales, el cual contempla la lectura de la Cuerda Vibrante y el Termistor, o de 32 canales, en el cual solo existen lecturas de cuerda vibrante. Además, aquí se configura el puerto de comunicación donde ha sido conectado el adquisidor de datos.

Las siguientes imágenes muestran los modos de configuración de la muestra:

Al iniciar el programa no está configurado algún puerto COM, por lo cual nos muestra un mensaje similar a la figura 3.1

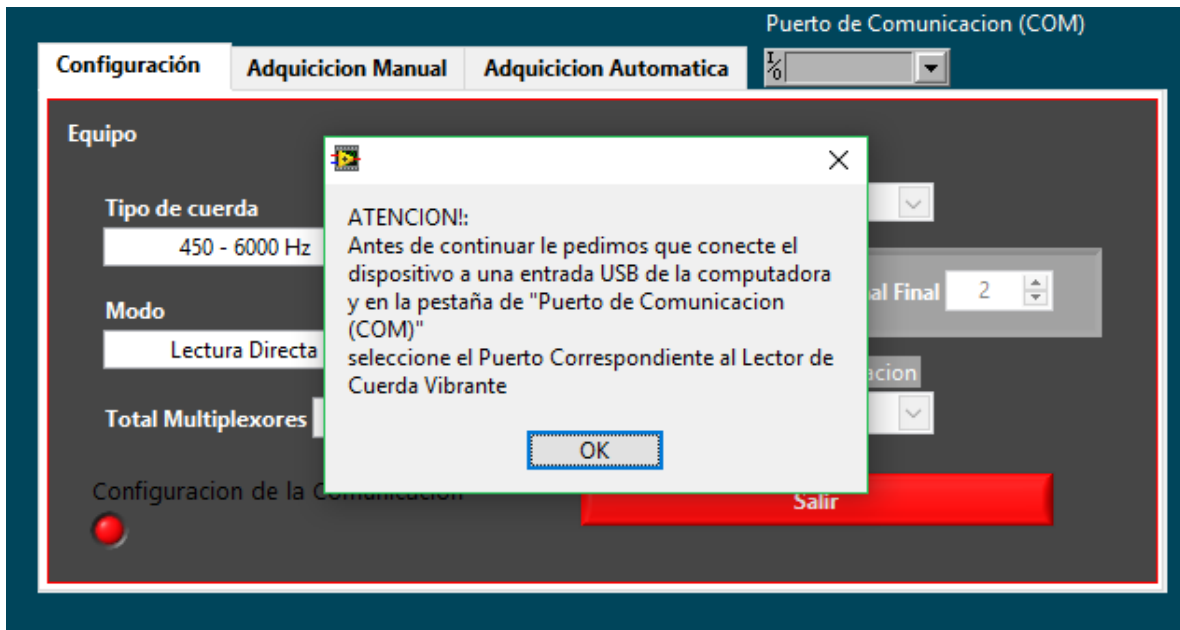


Figura 3.1 Mensaje de Conexión del puerto COM

Posteriormente de presionar el botón “OK”, se procede a seleccionar el puerto COM correcto de la lista que nos arroja la interfaz, la figura 3.2 muestra un ejemplo de los diferentes puertos COM en una PC

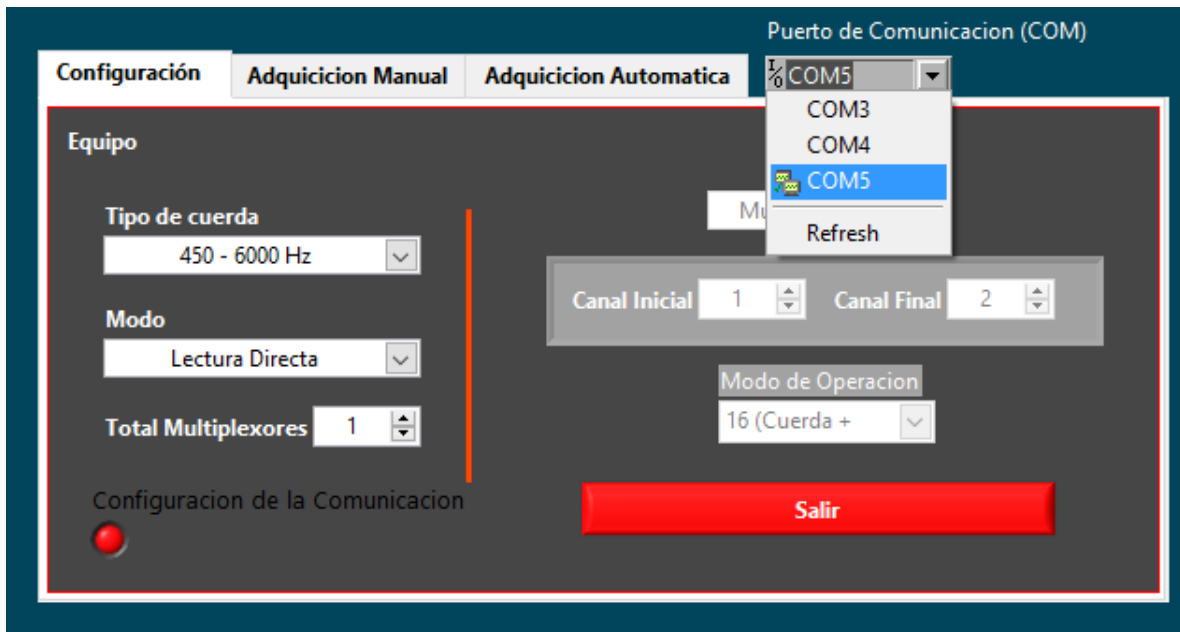


Figura 3.2 Lista de ejemplo de Puertos COM de una PC

En el caso específico de este listado, el “COM3” es el puerto donde está conectado el adquirente, por ello si se selecciona algún otro marcará un error ya que no se pueden configurar correctamente, como se muestra en la figura 3.3

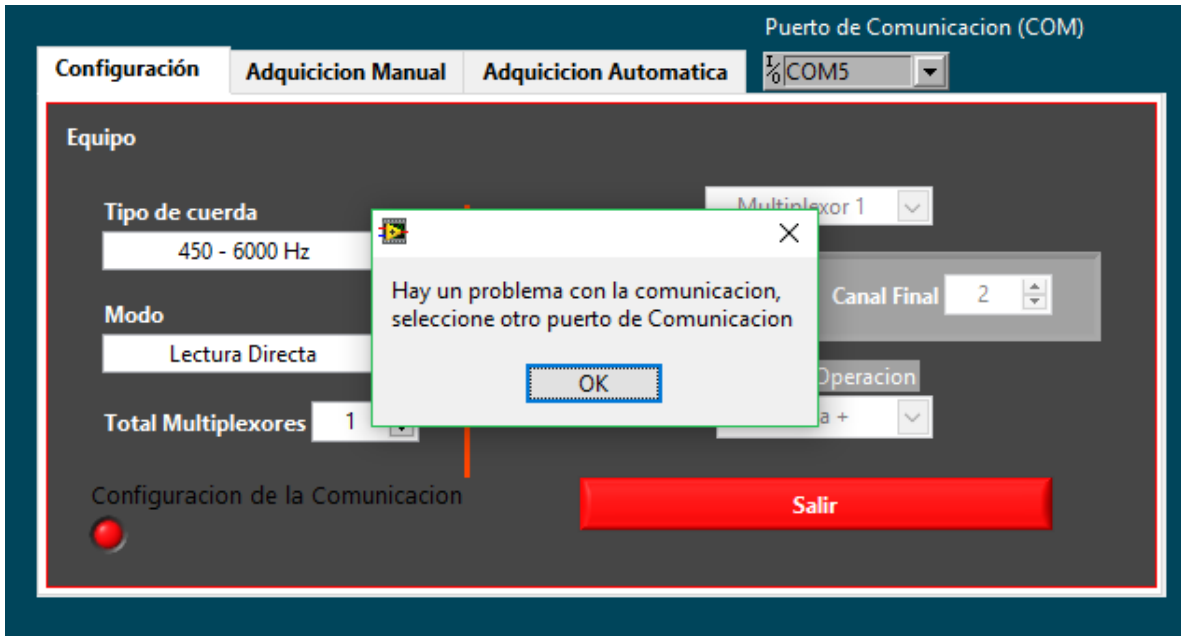


Figura 3.3 Problema con la Comunicación de uno de los puertos COM

Luego de presionar "OK", se procede a seleccionar el puerto COM correcto, lo que hará que el led que indica el estado de la configuración cambie de color a verde como en la figura 3.4

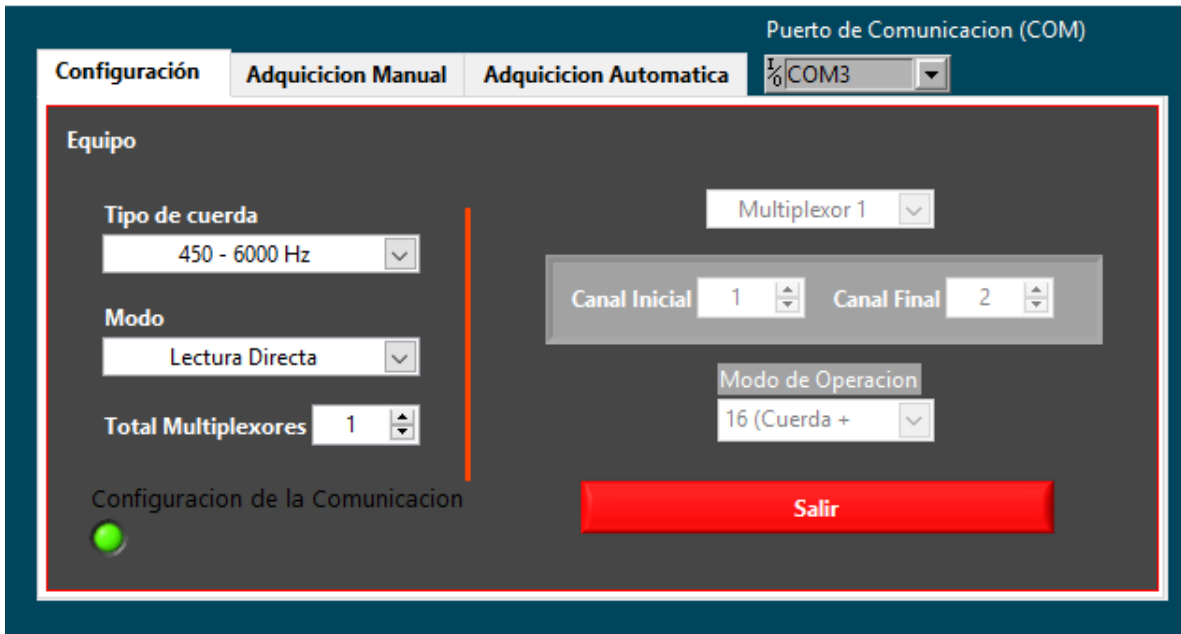


Figura 3.4 Correcta Comunicación con el Puerto COM Seleccionado

Se puede observar ahora los diferentes rangos de frecuencia que es posible seleccionar para las cuerdas vibrantes, similar a la figura 3.5

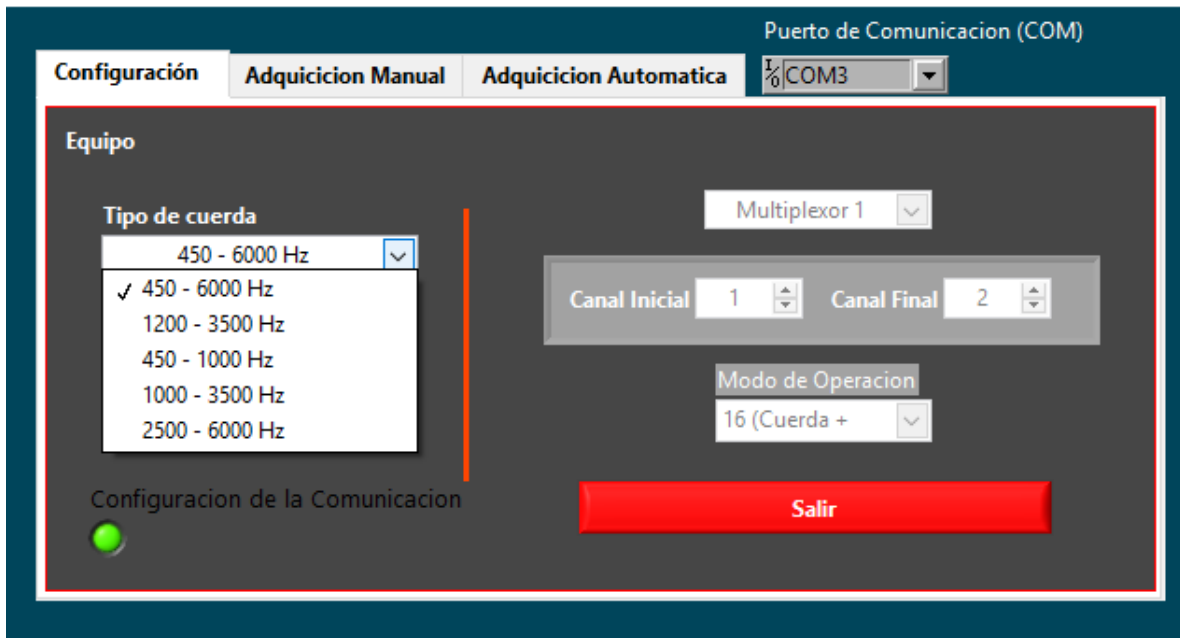


Figura 3.5 Lista de Rangos de Frecuencia de Diferentes Cuerdas Vibrantes

En la pestaña “*Modo*” es posible configurar el tipo de Muestra que se desea realizar, si es en Modo de “*Lectura Directa*”, una única cuerda, se observa que están desactivadas las opciones del lado derecho de la imagen, todas las pestañas están en color gris, como lo muestra la figura 3.6

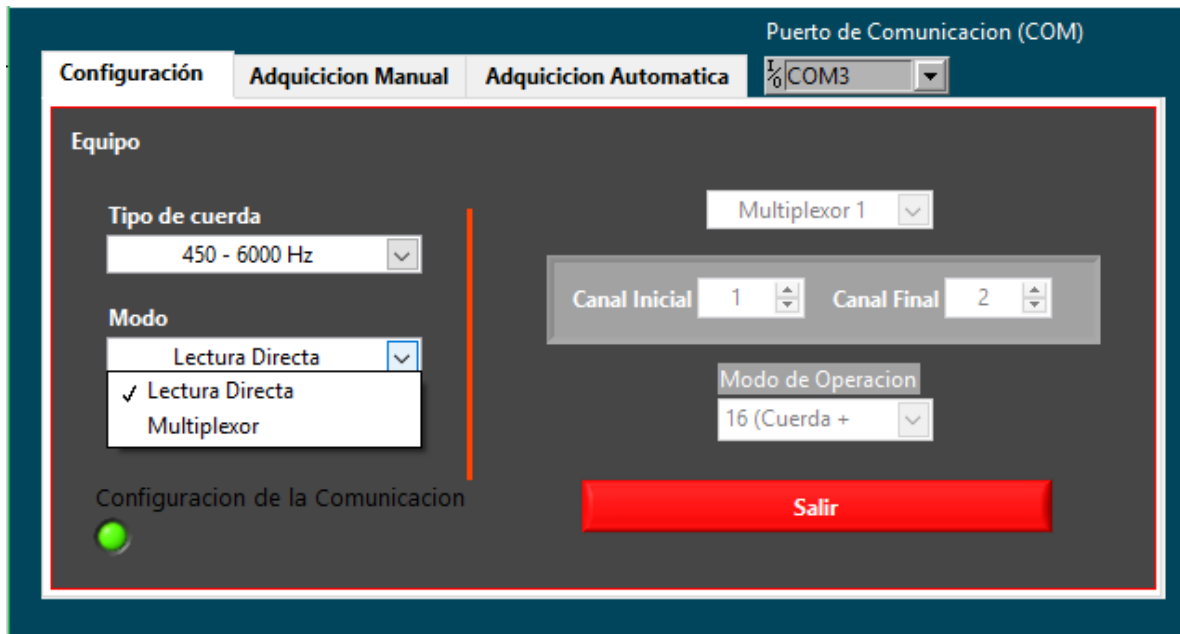


Figura 3.6 Configuración de Lectura de Una sola cuerda

Y al Seleccionar la opción de “*Multiplexor*”, las opciones del lado derecho se activan, la figura 3.7 muestra la activación de las opciones

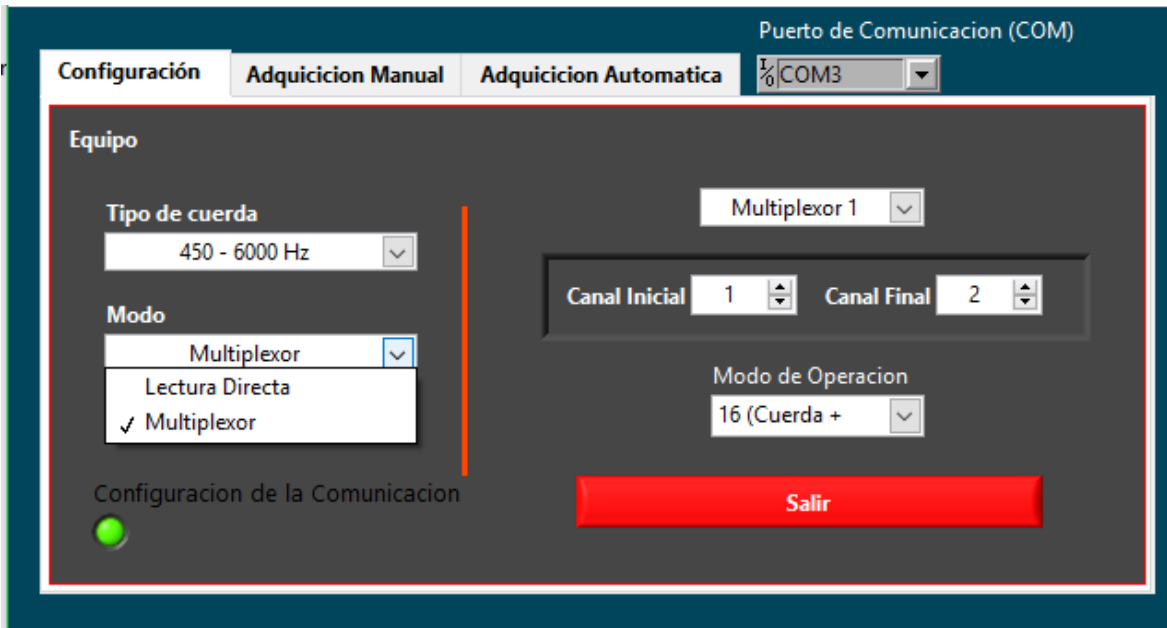


Figura 3.7 Configuración de Lectura a partir de un Multiplexor

En este modo se observa que es posible configurar un “*Canal Inicial*” y un “*Canal Final*” para la muestra secuencial del multiplexor, el “*Canal Final*” tiene como límite el “*Modo de Operación*”, es decir, si se configura en modo de 16 cuerdas, el valor máximo de “*Canal Final*” puede ser 16 y para 32 cuerdas es solo hasta 32. La pestaña de “*Modo de Operación*” muestra las opciones de la configuración del Multiplexor, figura 3.8

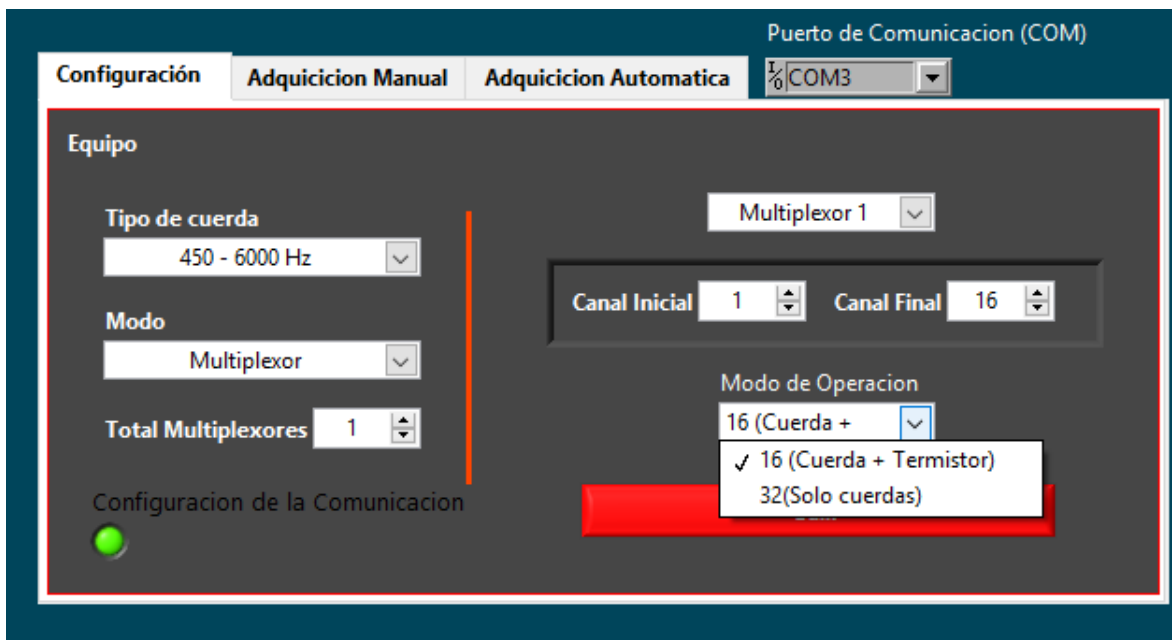


Figura 3.8 Modo de Operación del Multiplexor

Una vez configurado todo esto solo resta pasar a las pestañas de Adquisición de datos, ya sea Manual o Automática

3.1.2 Adquisición manual

La aplicación permite verificar el funcionamiento de la Cuerda (o Cuerdas) Vibrantes que se hayan configurado para que, en caso de tener problemas con alguna de las lecturas o los transductores se puedan tomar las acciones necesarias en el momento correcto.

La ventana presenta un par de pestañas que sirven para seleccionar que canal del multiplexor se desea verificar, dependiendo de la configuración que se le haya colocado en un principio al programa, será la cantidad de canales que aparecerán y a partir de cual iniciara. Las siguientes imágenes ejemplifican la configuración previamente realizada y la lista de canales seleccionados en la sección de “Configuración”, la figura 3.9 muestra la lista de canales.

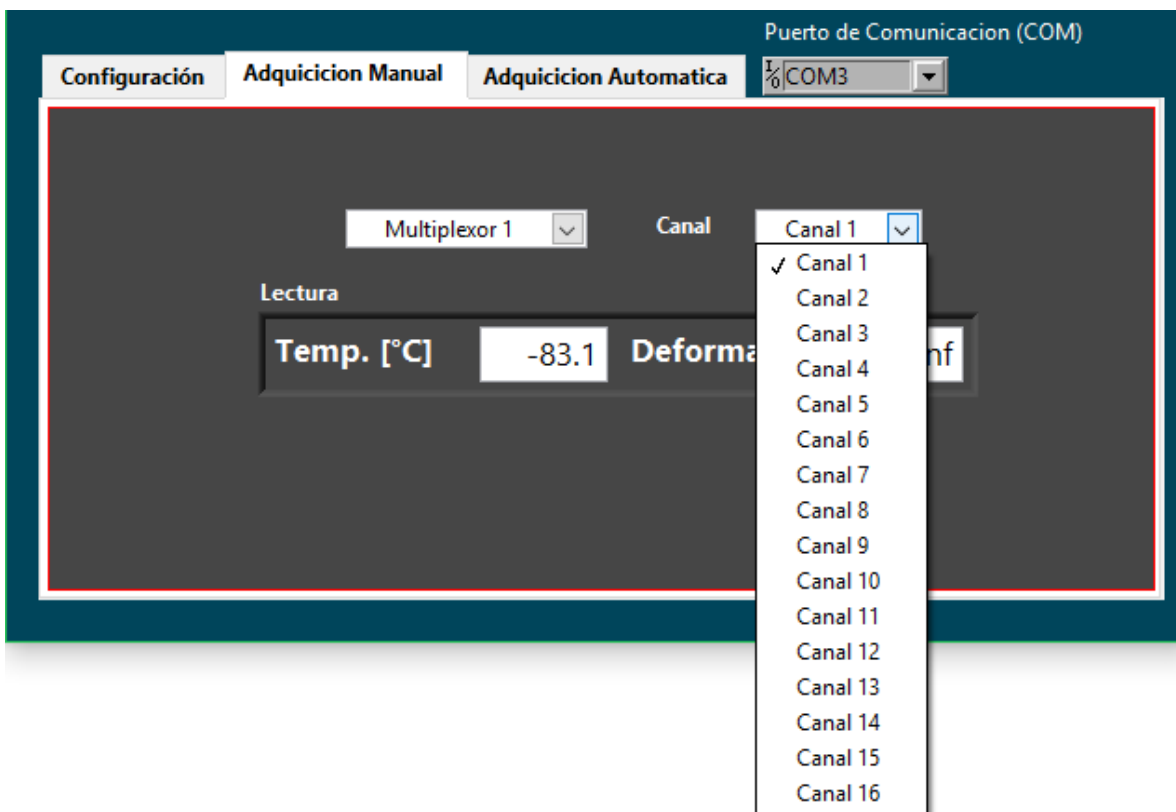


Figura 3.9 Lista de Canales del Multiplexor para una lectura Manual

Al seleccionar cualquier canal de la lista, el multiplexor apuntará a ese canal y el transductor de ese canal será leído.

3.1.3 Adquisición Automática

El programa puede funcionar como un Datalogger, es decir, permite configurar la adquisición de datos periódica y permite guardar las muestras en archivos organizados, la figura 3.10 muestra esta pantalla y sus opciones

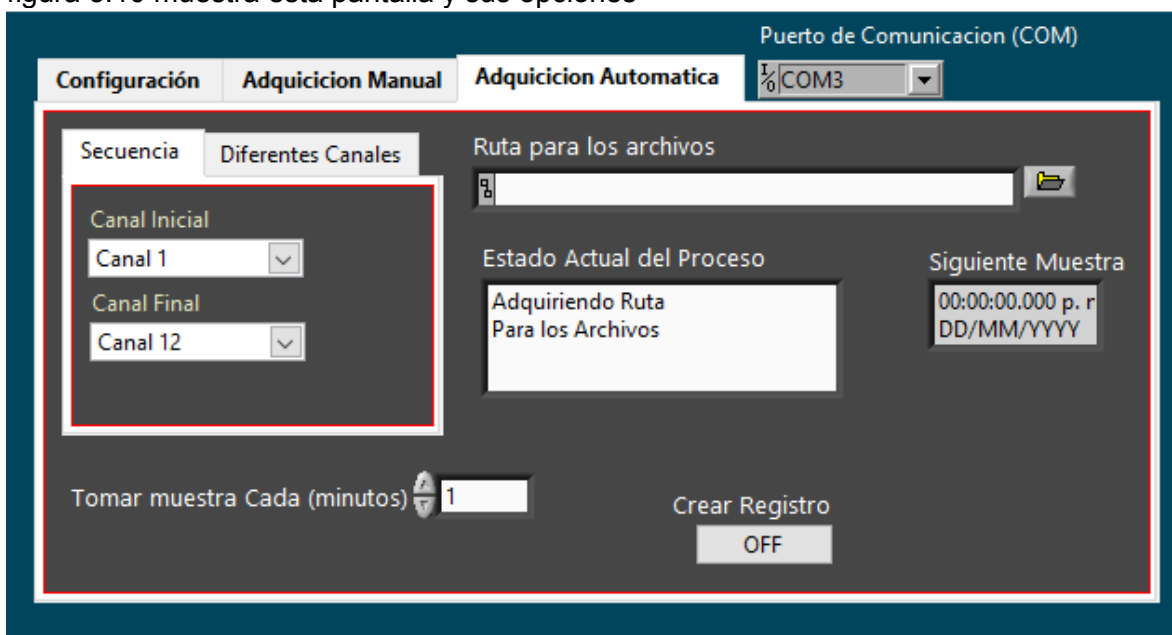


Figura 3.10 Configuración de una Muestra Automatizada y Periódica

Es posible seleccionar una muestra con una serie de canales, en la imagen anterior se muestra la selección del canal 1 como inicial y el canal 12 como final, el tiempo de muestra ha sido seleccionado con 1 minuto.

El “Botón” con una imagen de una carpeta funciona para seleccionar la ruta en la cual se generarán las carpetas y los archivos que se vayan a crear con los datos de las “Muestras Automáticas”, se abrirá una ventana como la mostrada en la figura 3.11

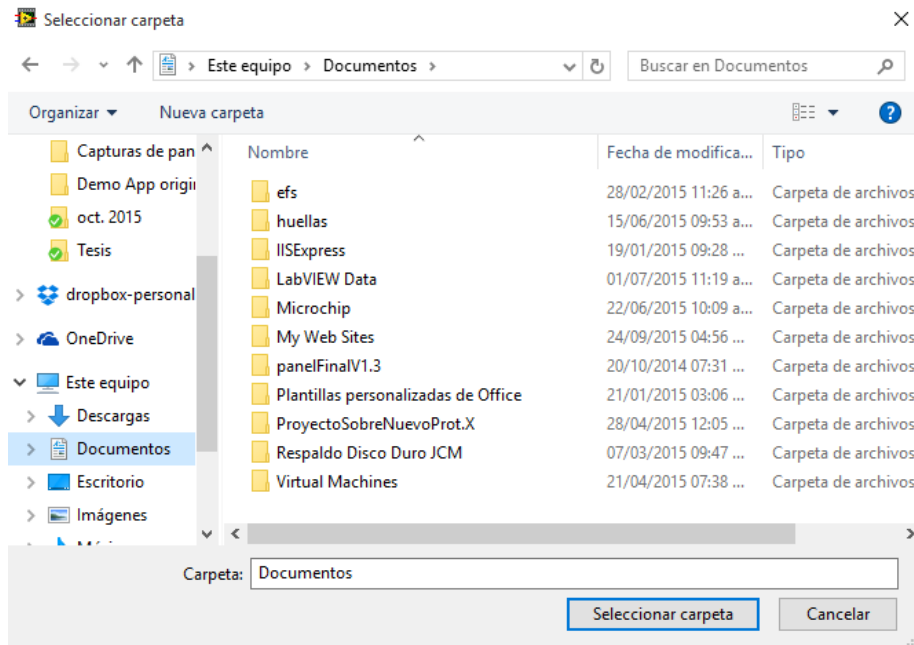


Figura 3.11 Selección de la Ruta para Crear los archivos de las Muestras Automáticas

Una vez seleccionada la ruta, se generan los archivos y carpetas para almacenar los datos. La configuración está completa y para iniciar la captura de datos es necesario presionar el botón “*Crear Registro*”, como se muestra en la figura 3.12

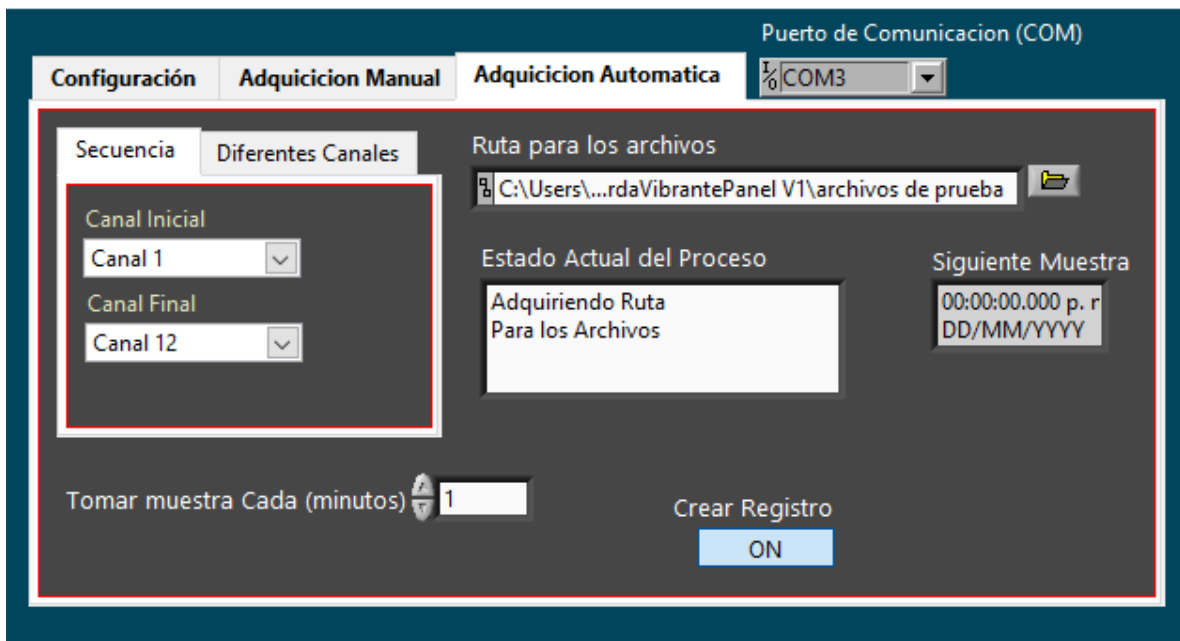


Figura 3.12 Inicio de la Adquisición Automática

Una vez inicializada la captura, será posible cambiar el tiempo de muestra sin afectar al programa, pero en caso de ser necesario cambiar algunos de los canales, ya sea el canal inicial o final, la ruta para almacenar los archivos o el tipo de Muestra, “*Secuencia*” o

“Diferentes Canales”, aparecerán algunas ventanas preguntando si desea cambiar alguno de los datos mencionados, el mensaje es similar al mostrado en la figura 3.13

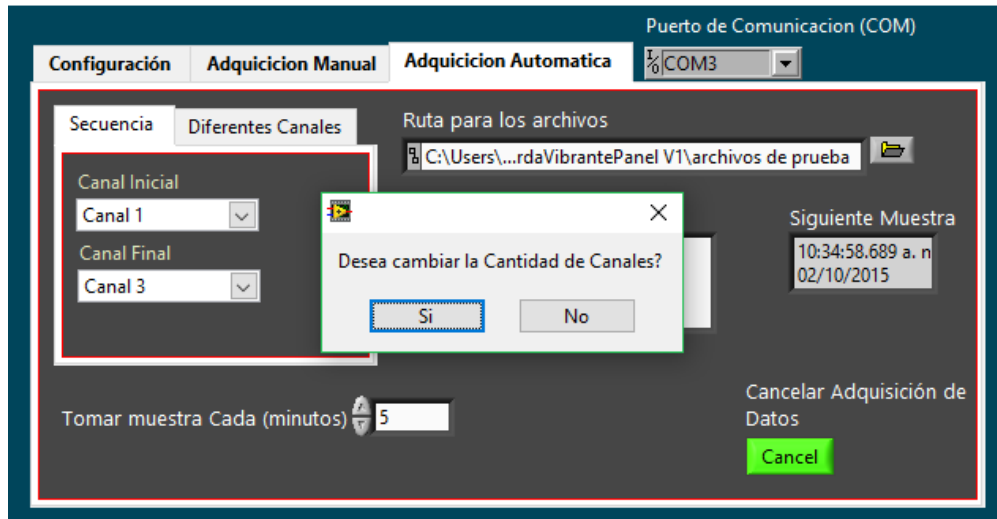


Figura 3.13 Aviso de modificación de datos de la actual muestra

Una vez realizada la muestra, se generará un archivo con el nombre del día en curso, como se muestra a continuación en la figura 3.14

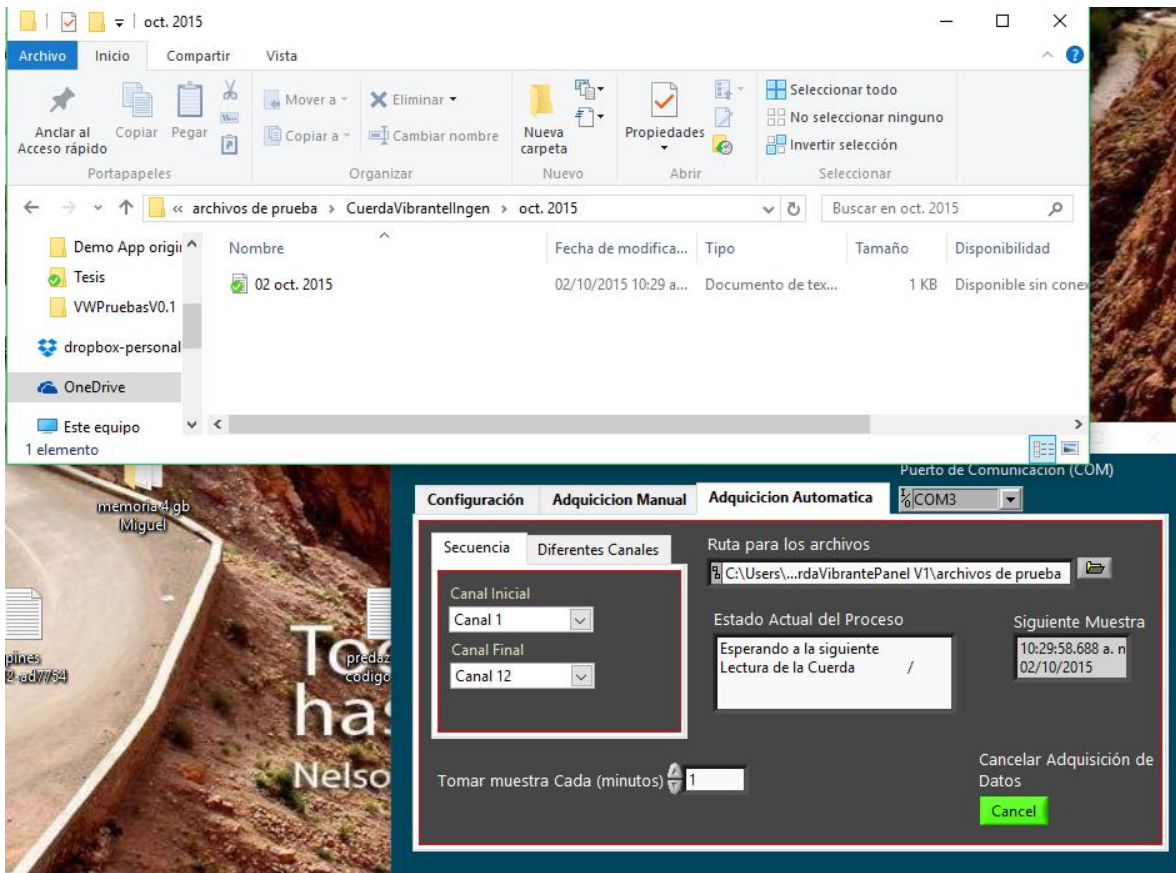
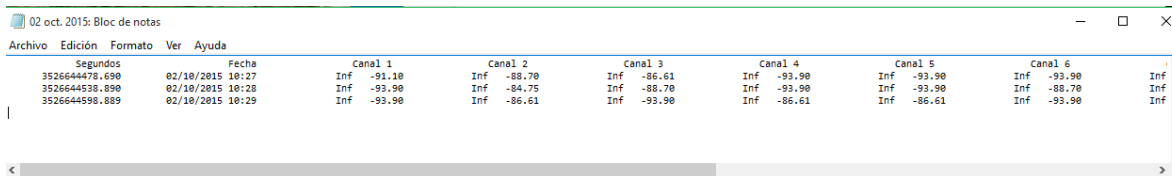


Figura 3.14 Creación del archivo para almacenar los datos de la Muestra Automática

Los datos para este ejemplo están almacenados en las columnas siguientes (figura 3.15)



| Segundos | Fecha | Canal 1 | Canal 2 | Canal 3 | Canal 4 | Canal 5 | Canal 6 |
|----------------|------------------|------------|------------|------------|------------|------------|------------|
| 3526644478.690 | 02/10/2015 10:27 | Inf -91.10 | Inf -88.70 | Inf -86.61 | Inf -93.90 | Inf -93.90 | Inf -93.90 |
| 3526644538.890 | 02/10/2015 10:28 | Inf -93.90 | Inf -84.75 | Inf -88.70 | Inf -93.90 | Inf -93.90 | Inf -88.70 |
| 3526644598.089 | 02/10/2015 10:29 | Inf -93.90 | Inf -86.61 | Inf -93.90 | Inf -86.61 | Inf -86.61 | Inf -93.90 |

Figura 3.15 Ejemplo del archivo creado por el Programa para la Interfaz USB

En el caso que se desee cambiar la cantidad de canales, el programa preguntara si el usuario desea sobrescribir el archivo o crear uno nuevo, tal como se ilustra en la figura 3.16

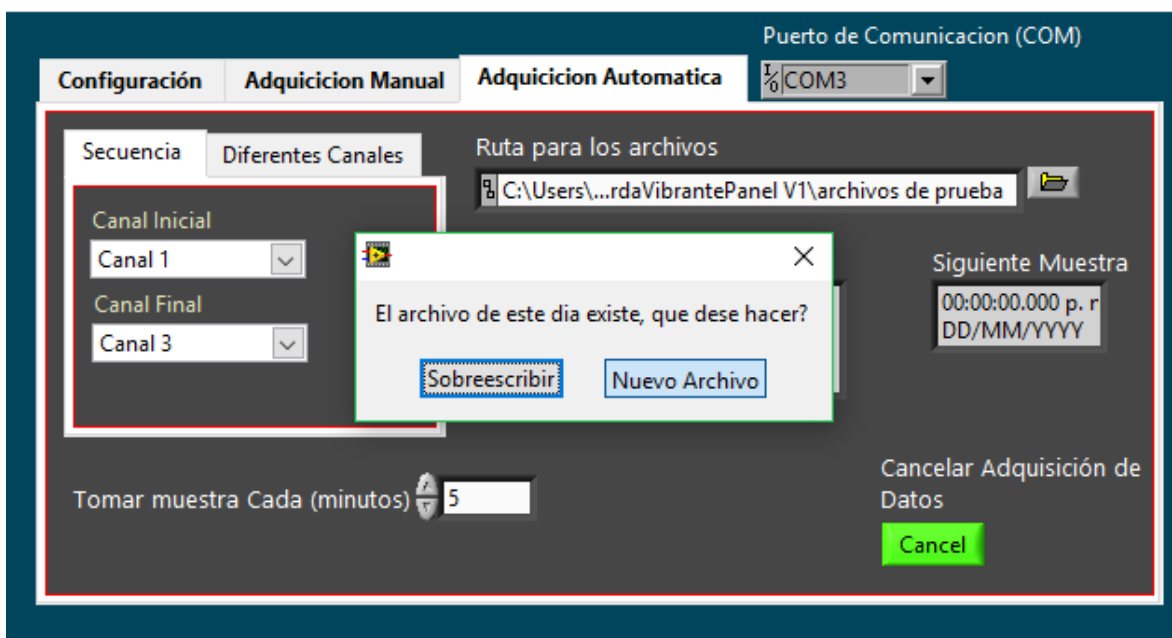


Figura 3.16 Creación ó Sobrescritura de Archivo

Al presionar en el botón de “*Nuevo Archivo*”, se generará un archivo con extensión .txt con el nombre del día, pero con un índice distinto para diferenciarlo, tal como se muestra en la figura 3.17

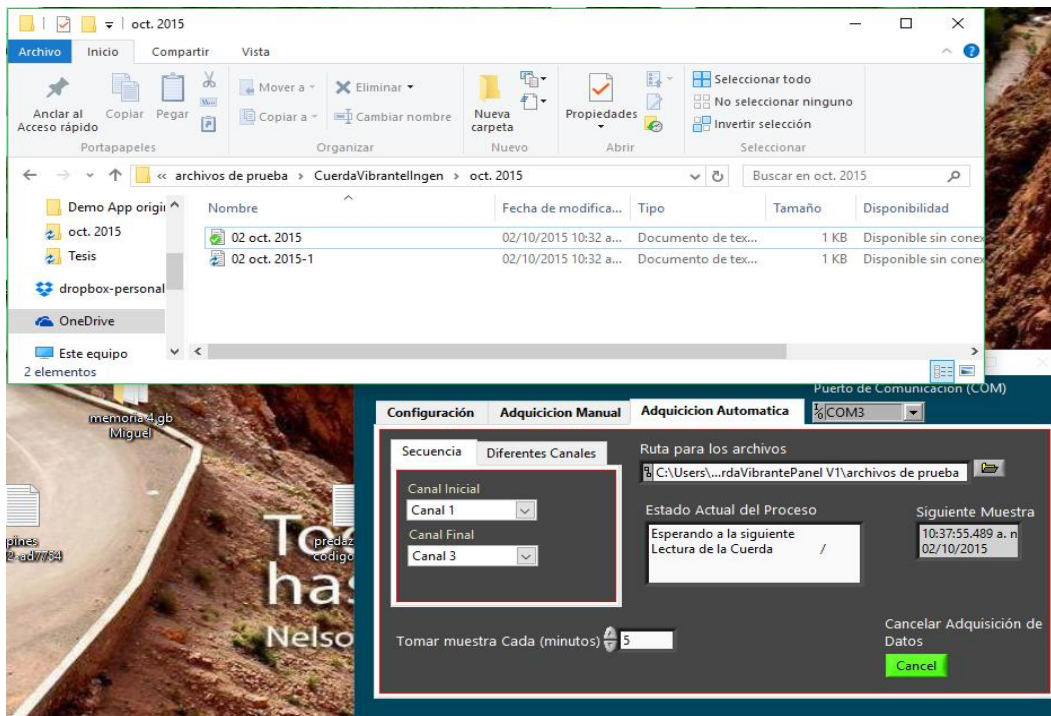


Figura 3.17 Nuevo Archivo para el almacenamiento de Datos de la Muestra

Y el dato que se genera será el siguiente (figura 3.18)

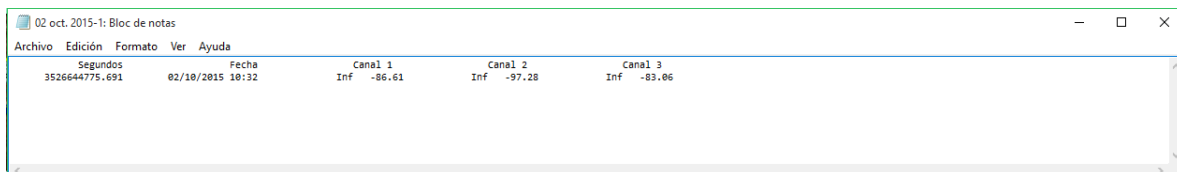


Figura 3.18 Datos De la Nueva Muestra Programada

Y en caso de presionar el botón de “*Sobreescribir*”, la información del archivo sería la siguiente (figura 3.19)

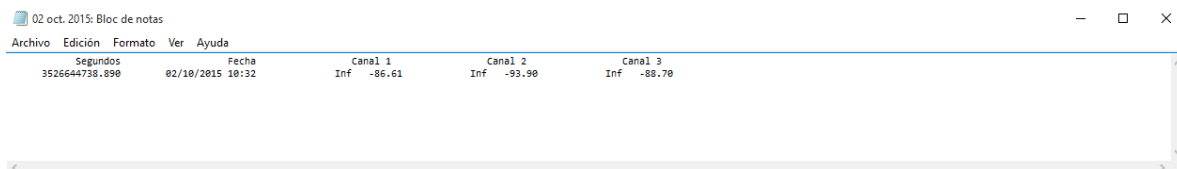


Figura 3.19 Datos Sobrescritos en el archivo

La pestaña de “*Diferentes Canales*” permite realizar un muestreo de canales específicos en un orden dado, como se ilustra en la figura 3.20

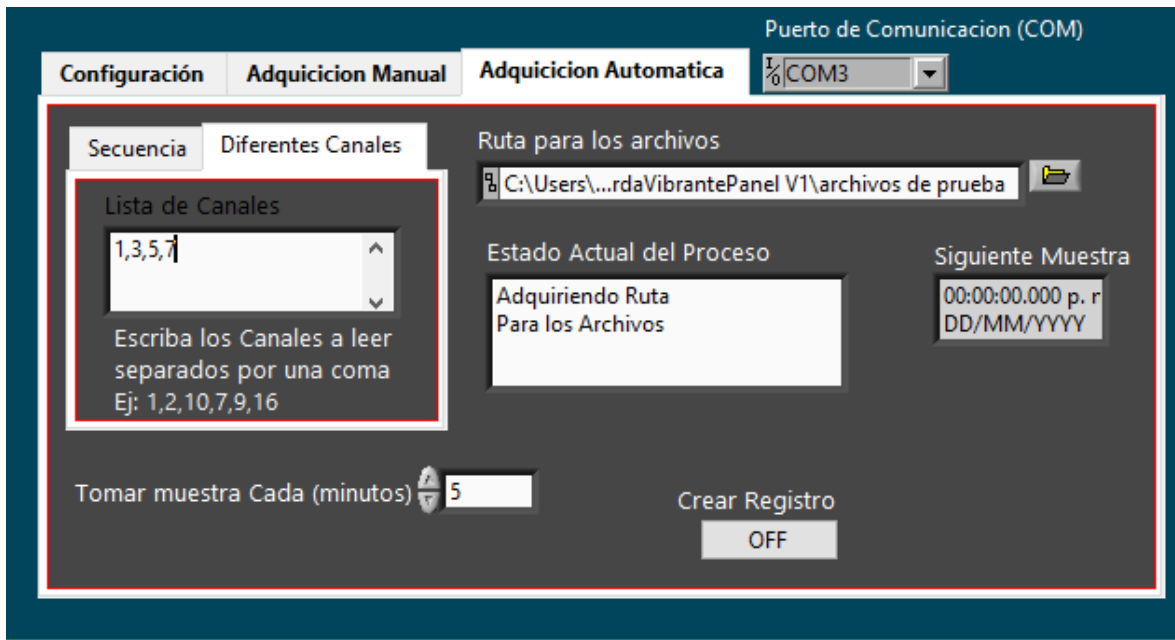


Figura 3.20 Configuración de una Lista de Canales Definida por el Usuario

En la figura 3.20 se muestra que se desea leer el canal “1, 3, 5 y 7” del multiplexor y la opción seleccionada para crear el archivo del día al presionar el botón en la ventana emergente “Crear Registro” fue “Nuevo Archivo”, para lo cual la información almacenada en el nuevo archivo se muestra en la figura 3.21

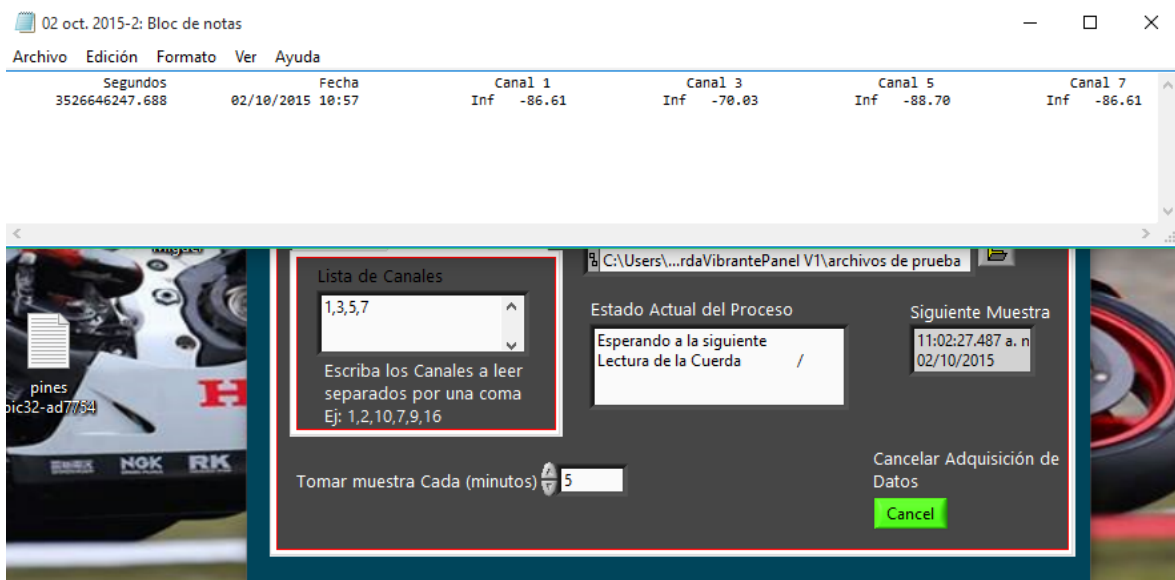


Figura 3.21 Creación del Nuevo Archivo con la lista de Canales Definida por el Usuario

El orden de los canales puede ser el que el usuario desee y ese orden se respetara en el archivo.

3.2 Página Web para mostrar los datos y configurar el Dispositivo

El microservidor alberga 2 páginas web que permiten visualizar y configurar la adquisición del dispositivo de una manera sencilla con opciones similares al Programa de comunicación a través del puerto USB

3.2.1 Página Principal (index.htm)

La página que es mostrada en el navegador cuando se accede a la dirección del microservidor presentara los datos que se estén leyendo en ese momento dependiendo del tipo de adquisición configurado. En el caso de que el dispositivo no tenga configurado ningún tipo de adquisición, se mostrara el mensaje de la figura 3.22



Figura 3.22 Mensaje del Microservidor

En caso contrario se mostrará una tabla con las diferentes configuraciones posibles, como se muestra a continuación:

- Lectura Directa de una cuerda (figura 3.23)



Configuración: Lectura Directa

| Canal | Valor Actual [$\mu\epsilon$] | Valor Anterior [$\mu\epsilon$] | Temperatura [$^{\circ}\text{C}$] |
|-------|--------------------------------|----------------------------------|------------------------------------|
| Unico | 3118.46 | 3119.82 | 19.25 |

[configuración de la muestra](#)

Figura 3.23 Tabla de Datos para la Lectura de una Cuerda Vibrante

- Lectura de 16 Canales (figura 3.24)

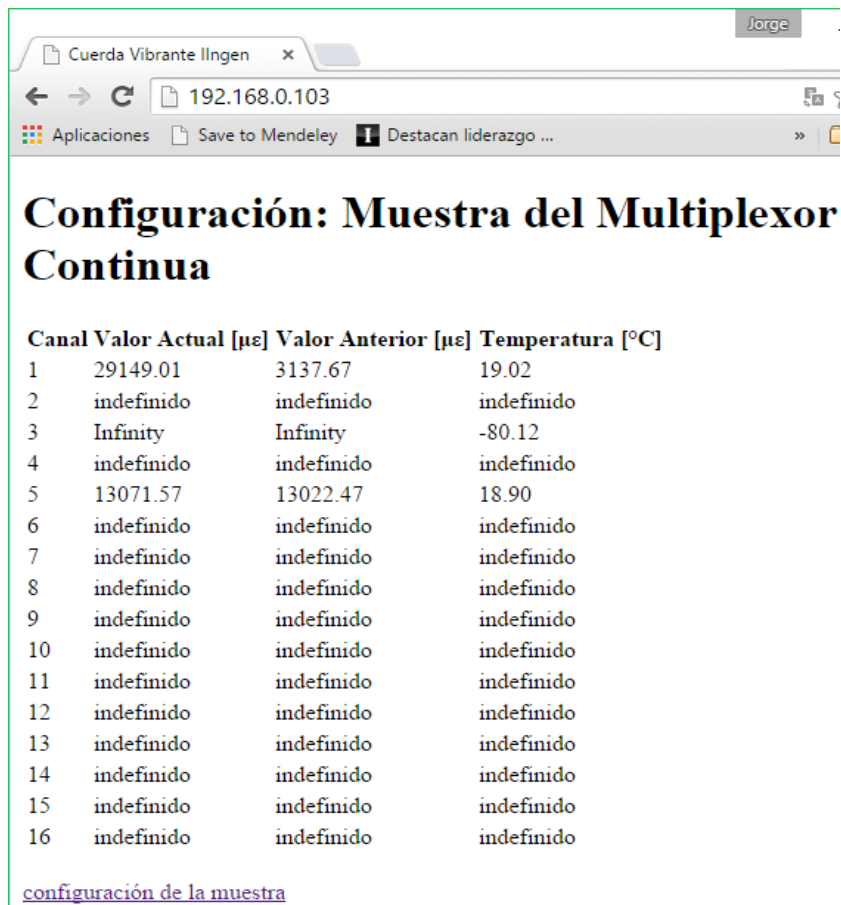


Figura 3.24 Tabla de Datos Para la lectura de 16 canales de un Multiplexor

En la figura 3.24 se observa que solo 3 canales fueron configurados para realizar lecturas en ellos, pero en solo 2 de estos existen transductores, por lo cual la página muestra como valor actual y valor anterior “Infinity” y como temperatura un valor por debajo de los 0 [$^{\circ}\text{C}$].

➤ Lectura de 32 Canales

| Canal | Valor Actual [µe] | Valor Anterior [µe] |
|-------|-------------------|---------------------|
| 1 | 3246.02 | 3259.99 |
| 2 | definido | definido |
| 3 | 3273.17 | 3275.23 |
| 4 | definido | definido |
| 5 | 3273.76 | 3274.35 |
| 6 | definido | definido |
| 7 | definido | definido |
| 8 | definido | definido |
| 9 | definido | definido |
| 10 | definido | definido |
| 11 | definido | definido |
| 12 | definido | definido |
| 13 | definido | definido |
| 14 | definido | definido |
| 15 | definido | definido |
| 16 | definido | definido |
| 17 | definido | definido |
| 18 | definido | definido |
| 19 | definido | definido |
| 20 | definido | definido |
| 21 | definido | definido |
| 22 | definido | definido |
| 23 | definido | definido |
| 24 | definido | definido |
| 25 | definido | definido |
| 26 | definido | definido |
| 27 | 3242.83 | 3274.05 |
| 28 | 3259.11 | 3259.41 |
| 29 | 3251.83 | 3263.50 |
| 30 | definido | definido |
| 31 | definido | definido |
| 32 | definido | definido |

[configuración de la muestra](#)

Figura 3.25 Tabla de Datos para la Lectura de 32 Canales de un Multiplexor

Al configurar el Multiplexor con 32 canales, lo que sucede es que se utilizan los bornes donde se conectan los cables del termistor del Transductor para utilizarlos como un canal extra, por lo que el dato de temperatura no arroja ninguna lectura correcta. Es por ello que para mostrar la tabla de datos de las lecturas se omitió el valor de la temperatura del transductor, como se muestra en la figura 3.25.

3.2.2 Página de Configuración de la adquisición (configuracion.html)

Esta página permite seleccionar el tipo de Muestra que se desea hacer, el tiempo en que se desea almacenar la información en los archivos de la tarjeta MicroSD y el rango de frecuencia de cada cuerda. De manera similar a las diferentes configuraciones mostradas en la sección [Página Principal \(index.htm\)](#), esta página muestra diferentes opciones para realizar la configuración de los canales y los rangos de frecuencia dependiendo del tipo de adquisición, si no se ha configurado ningún tipo de adquisición, la página se mostrara como la figura 3.26:

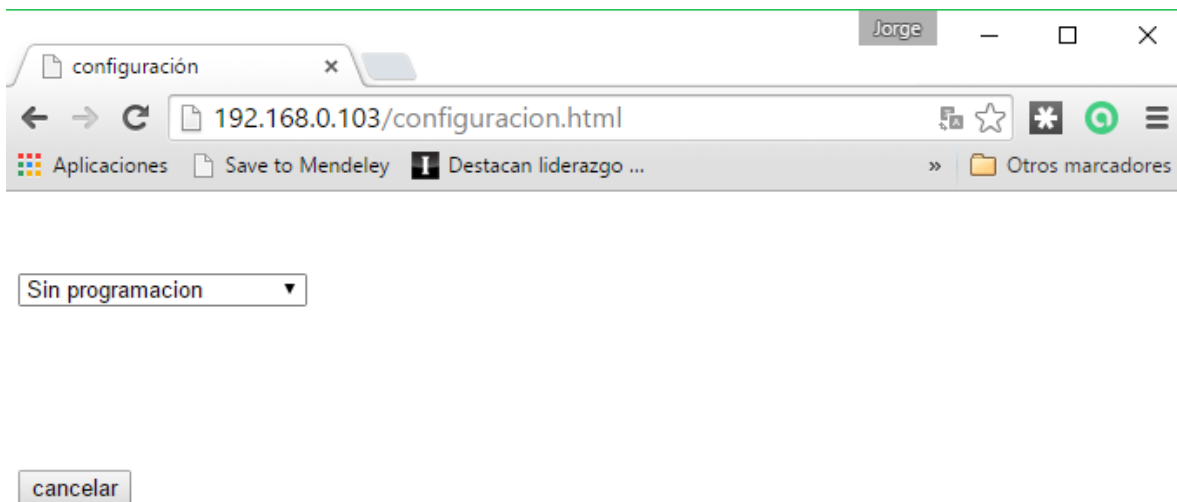


Figura 3.26 Adquisidor de Datos sin Configurar

La pestaña de selección que muestra la frase “*Sin programacion*” es la que alberga las diferentes opciones de configuración, al hacer “*click*” sobre la pestaña se muestran las siguientes opciones (figura 3.27)

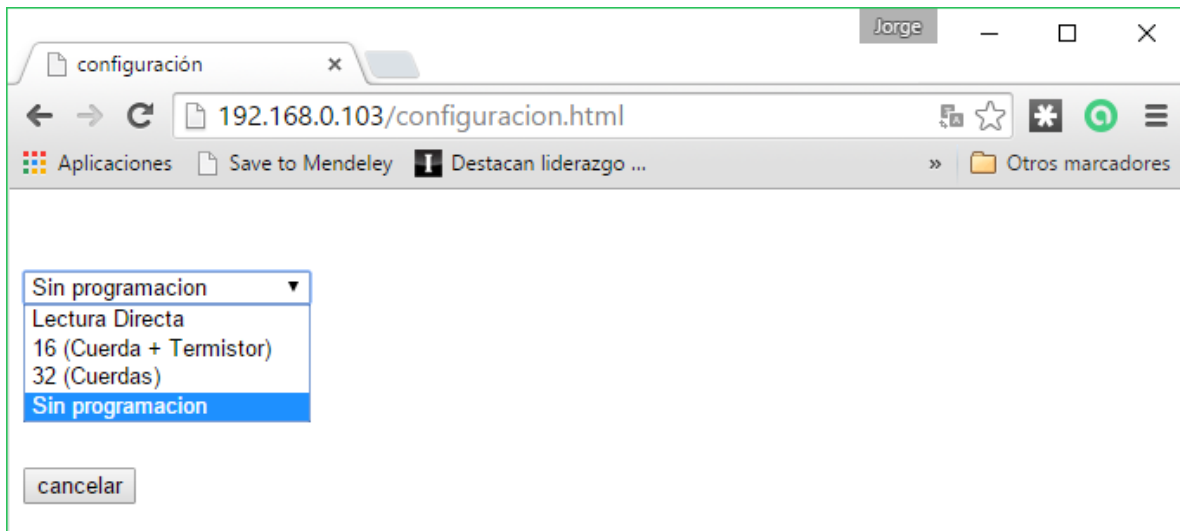


Figura 3.27 Opciones de configuración del Adquisidor de Datos

Al seleccionar alguna de las opciones se generará otra serie de nuevas opciones. A continuación, se muestran las pantallas que se generan con cada opción.

- Lectura Directa (figura 3.28)



Figura 3.28 Creación de los Controles para Configurar la lectura de una Cuerda Vibrante

Las opciones que se generan cuando se selecciona “*Lectura Directa*” se muestran en la figura 3.29)

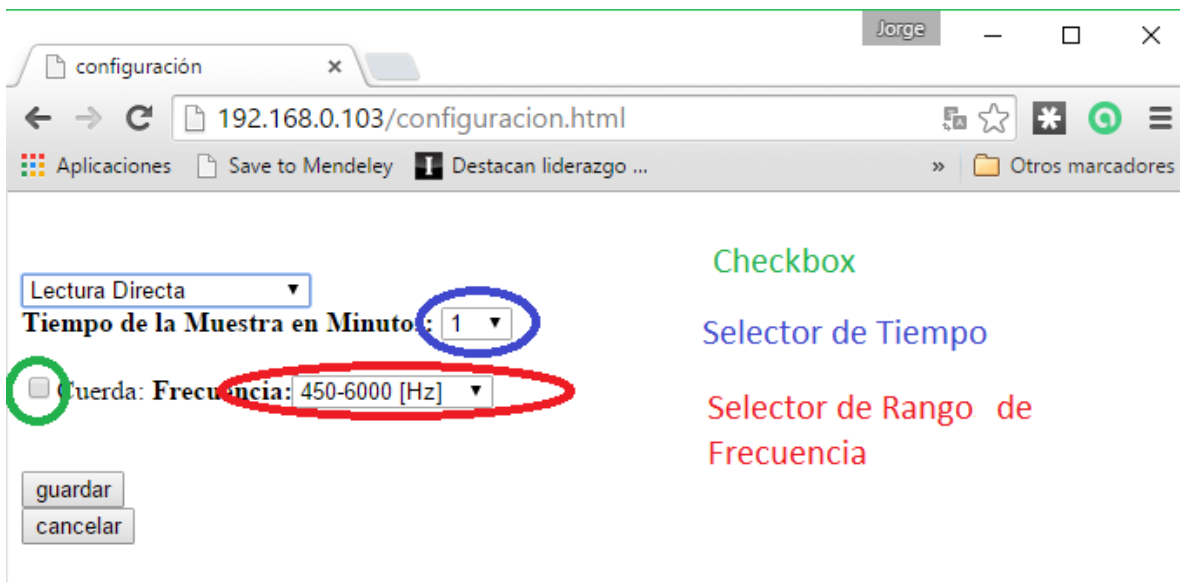


Figura 3.29 Explicación de los Controles Creados para la Lectura de una Cuerda Vibrante

- El “*Checkbox*” permite seleccionar si se desea activar el canal o no, al hacer “*click*” en él se dibuja una paloma para señalar que se ha seleccionado activar el canal.
 - El “*Selector de Tiempo*” es una pestaña con varias opciones que permite seleccionar cuanto tiempo debe transcurrir antes de almacenar los datos en los archivos de la memoria MicroSD.
 - El “*Selector de Rango de Frecuencia*” es similar al “*Selector de Tiempo*”, solo que, en lugar de tener las opciones de tiempo, tiene los diferentes Rangos de Frecuencia de diferentes Transductores de Cuerda Vibrante.
 - El botón “*Guardar*” permite enviar los datos de la configuración previamente realizada.
- 16 (Cuerda + Termistor)

Al seleccionar esta opción se creará una lista de los 16 canales, como muestra la figura 3.30, cada una con un “*Checkbox*” y un “*Selector de Rango de Frecuencia*” que permite tener diferentes tipos de Transductor de tipo Cuerda Vibrante en un mismo multiplexor

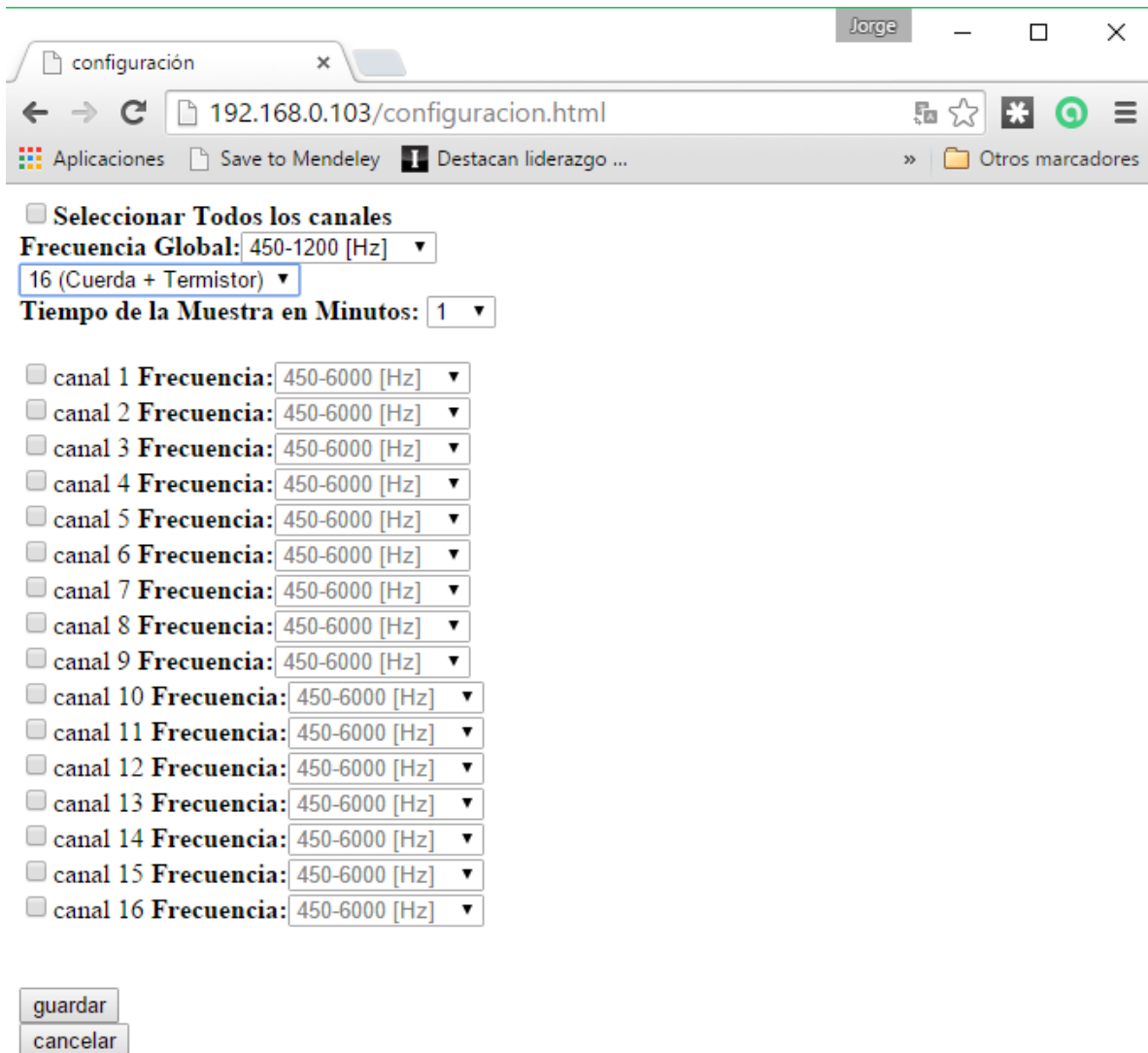


Figura 3.30 Creación de los Controles para Configurar la lectura de los 16 canales del Multiplexor

Para esta pestaña se han agregado 2 elementos extras (figura 3.31)

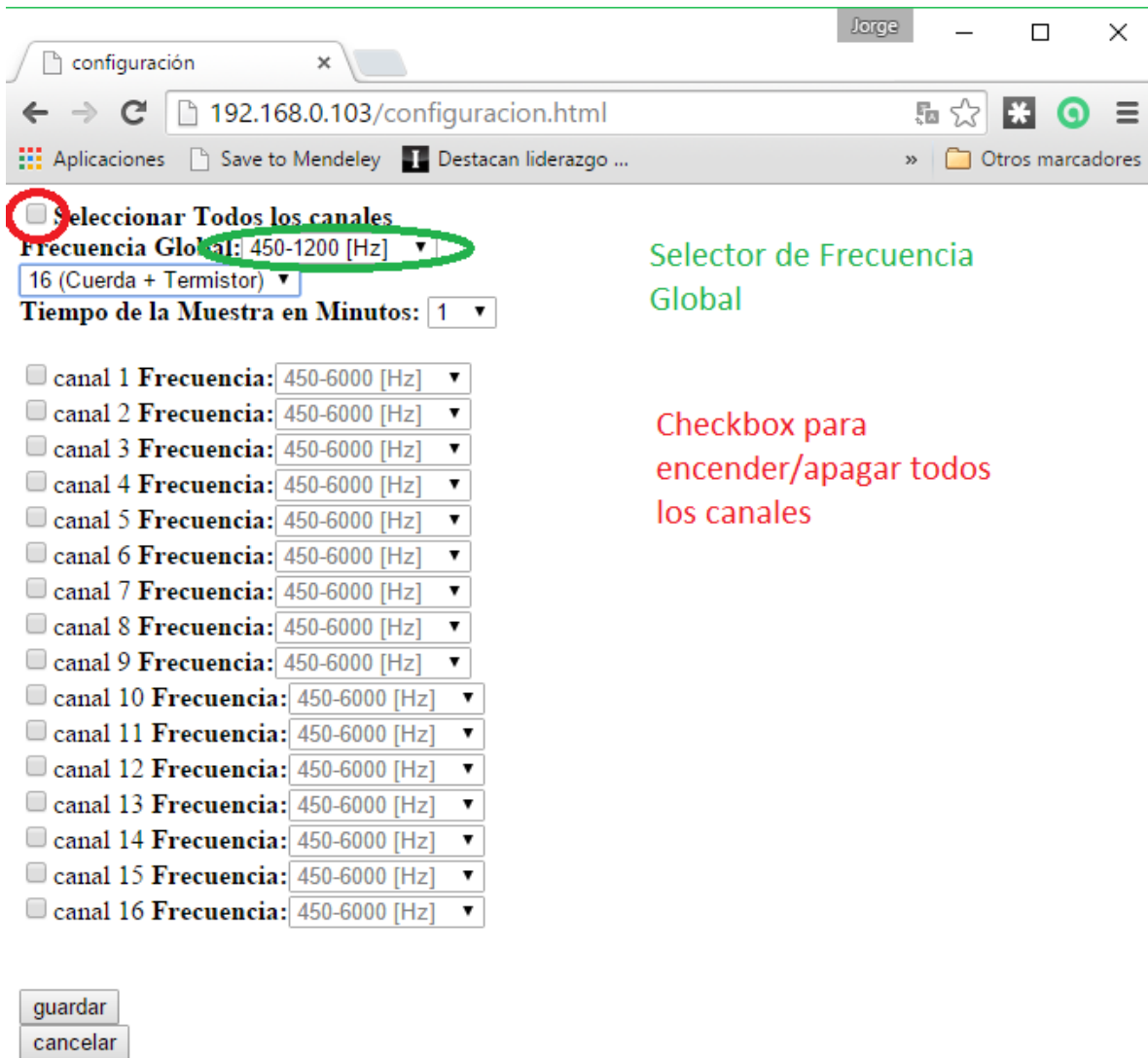


Figura 3.31 Explicación de los Controles Creados para la Lectura los 16 canales del Multiplexor

El “*Checkbox para encender/apagar todos los canales*” realiza la función de colocar una paloma en todos los “checkbox” de cada canal para señalar que ha sido seleccionado, cuando el “checkbox” de un canal no está seleccionado, su pestaña de Frecuencia correspondiente permanecerá deshabilitada y no permitirá seleccionar un rango de frecuencia dado. Las siguientes imágenes muestran el funcionamiento de este “checkbox”

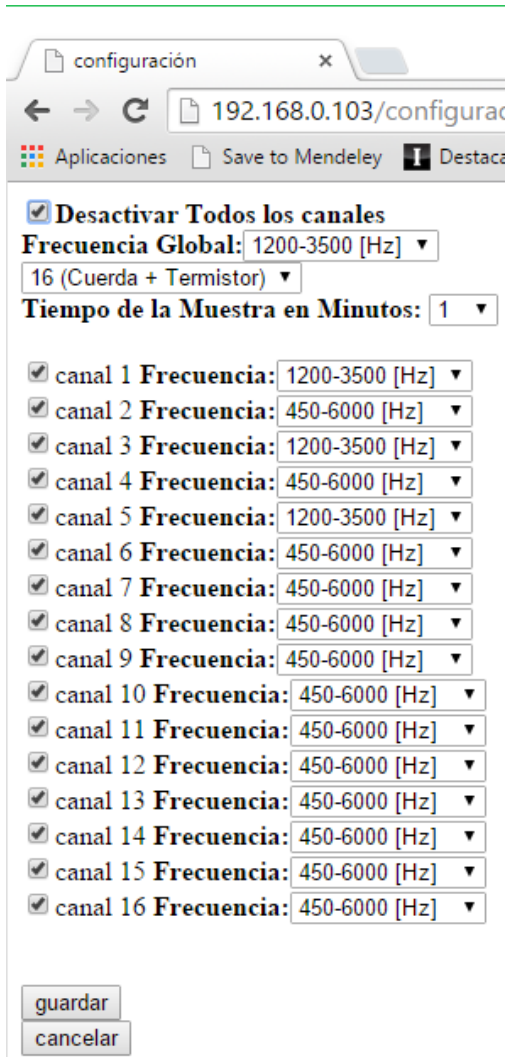


Figura 3.32 Tabla de datos al seleccionar todos los canales

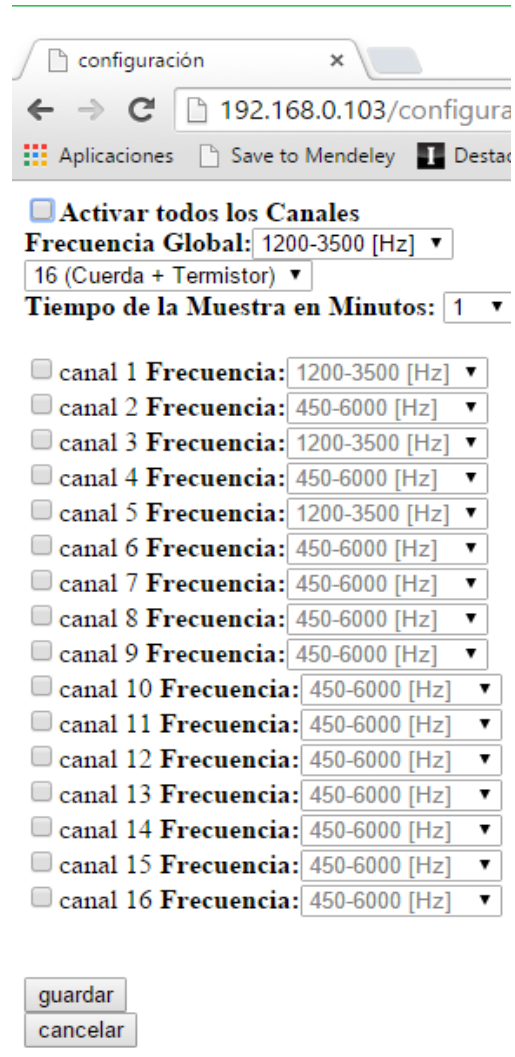


Figura 3.33 Tabla de datos al deseleccionar todos los canales

La figura 3.32 muestra la página al dar “click” sobre el “checkbox”, y la figura 3.33 muestra la misma página después de hacer “click” sobre el mismo “checkbox” para deseleccionar los canales.

El “Selector de Frecuencia Global” es una pestaña con las mismas opciones que el “Selector de Rangos de Frecuencia” de cada canal, permitiendo así colocar cada uno de los selectores de cada canal igual que el selector global, en caso de ser necesario. Las siguientes imágenes muestran el funcionamiento del “Selector Global”.

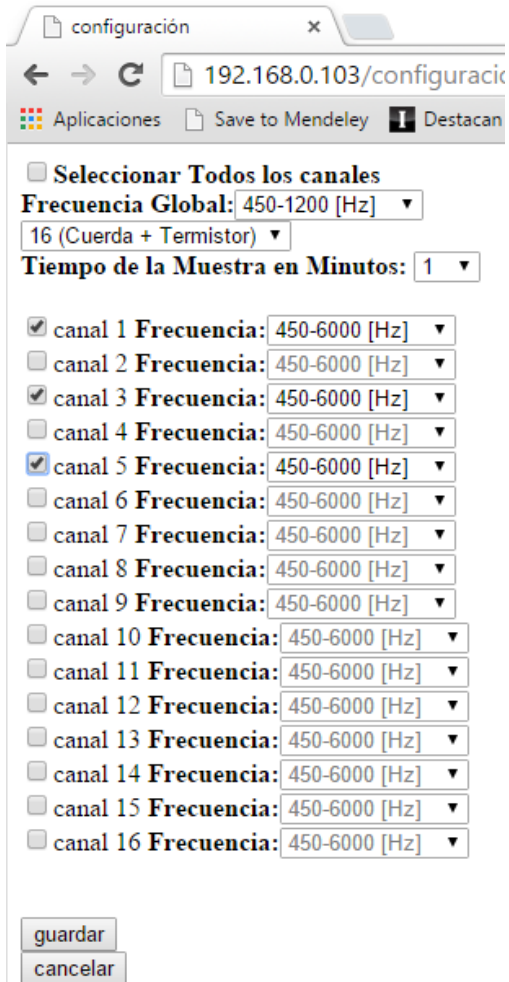


Figura 3.34 Antes de la Selección Global

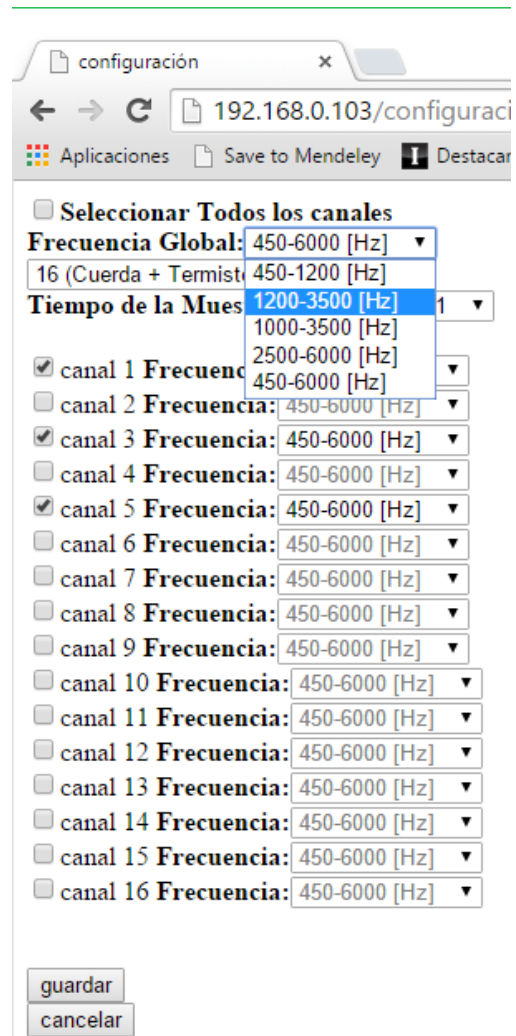


Figura 3.35 Desplegando las Opciones de la Selección Global

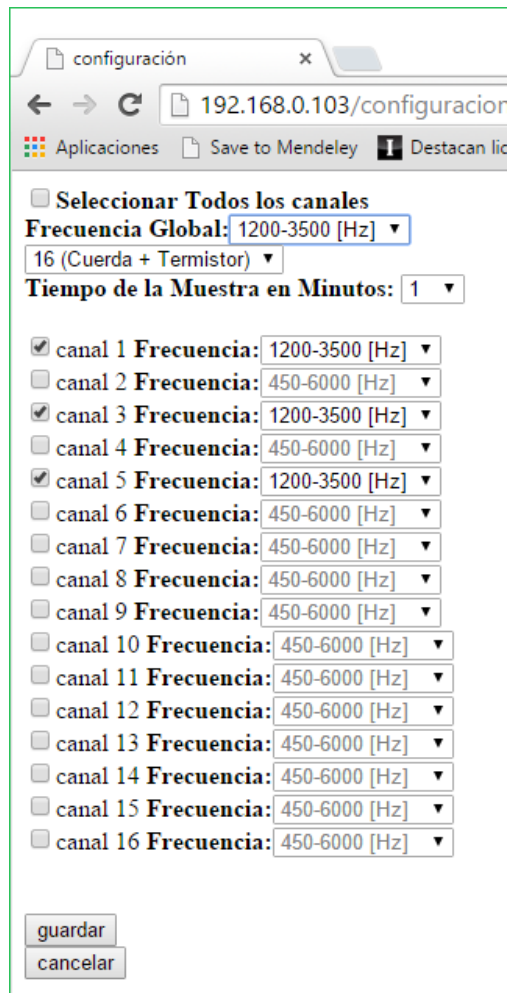


Figura 3.36 Luego de la Selección Global

La figura 3.34 muestra habilitados 3 canales, canal 1, 3 y 5, con una frecuencia y al hacer “click” sobre el “Selector Global”, figura 3.35, se observan las opciones que contiene y al seleccionar alguna de estas se aprecia que todos los canales seleccionados mostrarán la misma frecuencia que se ha seleccionado, todos los canales que no estén seleccionados no presentaran cambios, como se puede observar en la figura 3.36.

- 32 (Cuerdas)

La última de las opciones de adquisición genera el listado de los 32 posibles canales que existen en el multiplexor, de una manera similar a la opción de 16 canales, se puede seleccionar que canales se desean leer, a que rango de frecuencia trabaja el transductor de cada canal. La figura 3.37 muestra la información creada en la página web. Para esta opción no se agregaron nuevas opciones ni elementos a la página.



Figura 3.37 Creación de los Controles para Configurar la lectura de los 32 canales del Multiplexor

4 Resultados

4.1 Lector de Cuerda Vibrante

Para verificar que los datos adquiridos por el dispositivo que se ha construido son correctos, fue necesario utilizar un generador de señal con una onda senoidal fija, similar a la respuesta de una verdadera cuerda vibrante, con el propósito de leer la frecuencia de la onda mediante el dispositivo y comparar la lectura con la señal real. La figura 4.1 muestra el generador de señal con una onda senoidal de 1000 [Hz] y la PC ejecutando un programa que solo recibe los datos enviados por el dispositivo.

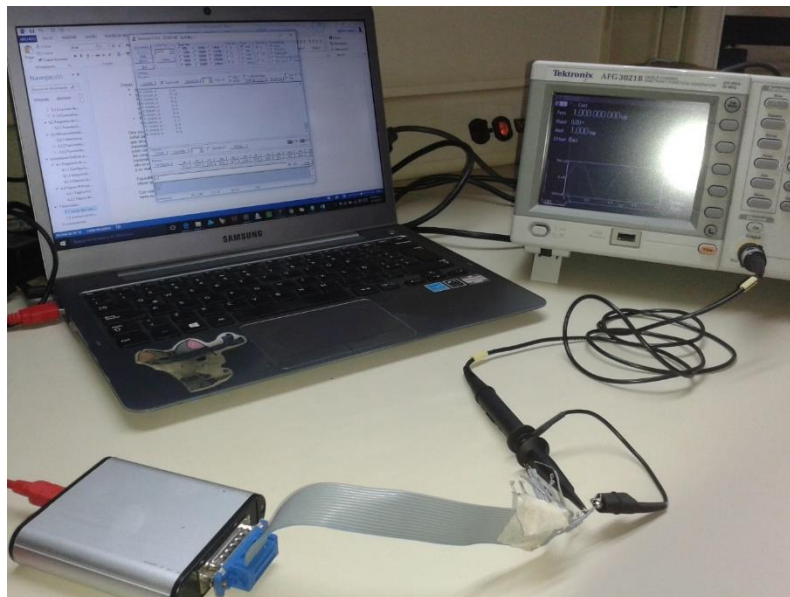


Figura 4.1 Pruebas Con el Patrón de Señal

La fórmula para obtener la frecuencia de la señal leída por el dispositivo es la siguiente:

$$F = \frac{1}{T} = \left(\left(\frac{\text{pulsosLeido}}{\text{cantidadDeFlancos}} \right) \left(\frac{1}{f_{\text{Oscilador16bits}}} \right) \right)^{-1}$$

Donde:

- pulsosLeidos= es la cantidad enviada por el dispositivo
- cantidadDeFlancos = es la cantidad de veces que se lee la señal enviada desde el transductor, esto es debido a que es necesario leer 10 el transductor para obtener una resolución en la lectura de 0.1 [Hz] pues los dispositivos comerciales ofrecen esa resolución. La sección de Lectura del transductor se explica cómo se realiza una sola lectura de un flanco de la señal.
- $f_{\text{Oscilador16bits}}$ = es la velocidad del reloj del contador por hardware de 16 bits para realizar la lectura de la frecuencia del transductor

La figura 4.2 y la figura 4.3 muestran las gráficas correspondientes a 300 lecturas realizadas con el patrón, la figura 4.2 muestra los pulsos leídos mientras la figura 4.3 muestra sus respectivas frecuencias

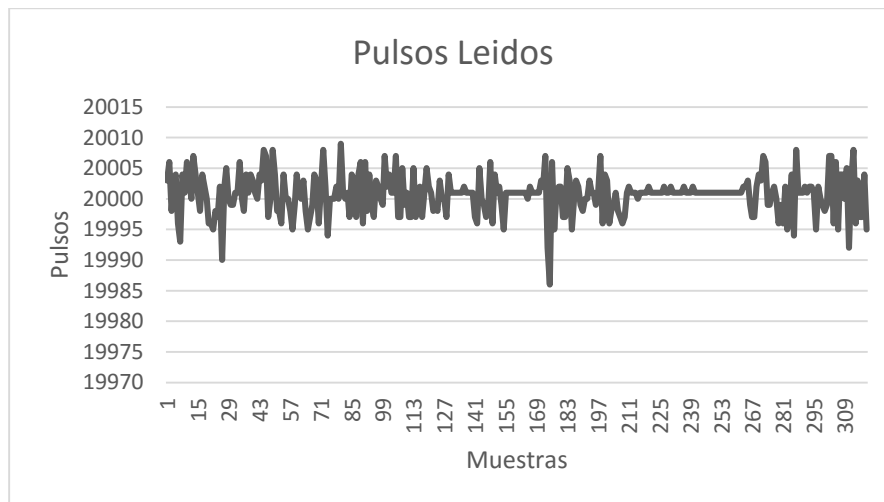


Figura 4.2 Pulsos Adquiridos

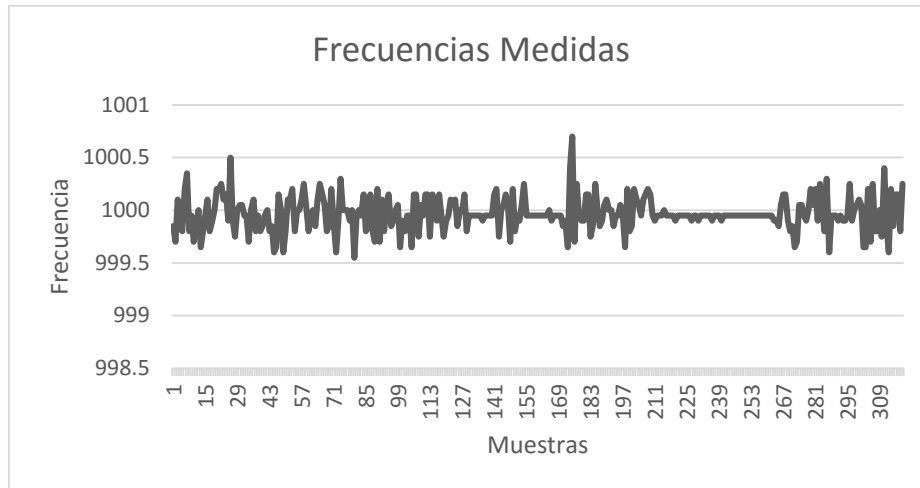


Figura 4.3 Representación en frecuencia de los pulsos

El promedio de las muestras fue de 999.96 [Hz] para una muestra patrón de 1000[Hz], por lo cual se demuestra que las lecturas de la frecuencia corresponden al valor leído. Posteriormente se realizó esta misma prueba con el dispositivo comercial que se estaría utilizando para comprobar que los datos adquiridos por el dispositivo Lector de “Cuerda Vibrante” fueran correctos. Al colocar la señal de 1000[Hz] en el dispositivo comercial Geokon GK-403 (GEOKON, 2009), se mostraba un periodo constante de 999.97 [μs], como se muestra en la figura 4.4, por lo cual la frecuencia que el dispositivo de Geokon estaba adquiriendo sería de 1000.3 [Hz], con esto se concluye que el dispositivo Lector de Cuerda Vibrante que se ha construido está leyendo satisfactoriamente una frecuencia dentro del rango de valores en el cual diferentes tipos de Transductor de “Cuerda Vibrante” están trabajando.



Figura 4.4 Dispositivo Geokon leyendo la Señal Patrón

Una vez verificada la correcta lectura de la frecuencia de la señal patrón, se procedió a comparar los datos adquiridos por el Lector de "Cuerda Vibrante" contra los datos del dispositivo Geokon, teniendo en cuenta que lo importante en este tipo de lectores es que la diferencia entre la lectura " n " y la lectura " $n+1$ " sea la misma, o muy similar, esto es debido a que los componentes electrónicos que utiliza cada dispositivo son distintos, creando diferentes impedancias y por ello la lectura " n " será distinta entre cada uno de estos.

Para esta prueba se midieron las microdeformaciones que presentaba una barra de acero la cual tiene empotrada un Transductor de Cuerda Vibrante cuando se le aplica una fuerza a una distancia conocida y en el momento en el cual la fuerza no está presente, la figura 4.5 muestra la barra de acero antes descrita.



Figura 4.5 Cuerda Vibrante Fijada a una barra de acero

La figura 4.6 muestra el dispositivo comercial conectado al Transductor que esta empotrado en la barra de acero y la figura 4.7 muestra al dispositivo Lector de Cuerda Vibrante.

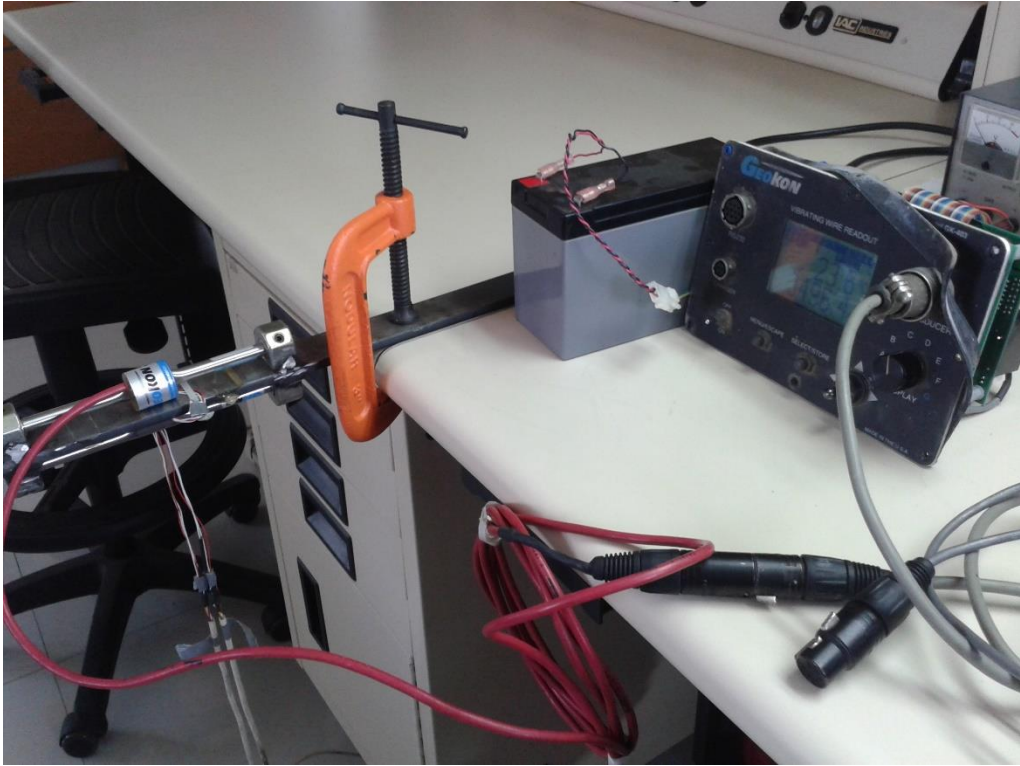


Figura 4.6 Dispositivo Geokon GK-403



Figura 4.7 Dispositivo Lector de Cuerda Vibrante

Se realizaron 20 muestras de las microdeformaciones presentadas antes de aplicar la fuerza y después, para lo cual la tabla 3 muestra la deformación inicial de la cuerda y la deformación final y una columna con la diferencia entre cada muestra que el dispositivo Geokon obtuvo.

Tabla 3 Muestras del Dispositivo Geokon

| Geokon 403 | | | |
|--------------|---------------|-------------|------------|
| # de muestra | Valor Inicial | Valor Final | Diferencia |
| 1 | 3499.8 | 3527.9 | 28.1 |
| 2 | 3500.9 | 3526.5 | 25.6 |
| 3 | 3498.8 | 3525.2 | 26.4 |
| 4 | 3496.5 | 3523.9 | 27.4 |
| 5 | 3502.2 | 3529.7 | 27.5 |
| 6 | 3503.3 | 3529.6 | 26.3 |
| 7 | 3503.4 | 3529.4 | 26 |
| 8 | 3502.9 | 3529.2 | 26.3 |

| | | | |
|----|--------|--------|------|
| 9 | 3501.8 | 3529.3 | 27.5 |
| 10 | 3501.6 | 3529.3 | 27.7 |
| 11 | 3502.7 | 3529.1 | 26.4 |
| 12 | 3501.8 | 3529.7 | 27.9 |
| 13 | 3502.8 | 3529.6 | 26.8 |
| 14 | 3501.9 | 3528.6 | 26.7 |
| 15 | 3502.2 | 3528.7 | 26.5 |
| 16 | 3501.8 | 3528.1 | 26.3 |
| 17 | 3501.4 | 3527.7 | 26.3 |
| 18 | 3501.9 | 3528.5 | 26.6 |
| 19 | 3500.7 | 3527.8 | 27.1 |
| 20 | 3501.4 | 3528.6 | 27.2 |

El promedio de las Diferencias de la tabla 3 es de 26.86 [$\mu\epsilon$].

La Tabla 4 muestra las microdeformaciones que el Dispositivo Lector de Cuerda Vibrante adquirió durante las 20 muestras.

Tabla 4 Muestras del Dispositivo Lector de Cuerda Vibrante

| Dispositivo Lector de Cuerda Vibrante | | | |
|---------------------------------------|---------------|-------------|------------|
| # de muestra | Valor Inicial | Valor Final | Diferencia |
| 1 | 3508.5 | 3535 | 26.5 |
| 2 | 3511.4 | 3534 | 22.6 |
| 3 | 3507.5 | 3534.4 | 26.9 |
| 4 | 3503.6 | 3526.1 | 22.5 |
| 5 | 3512 | 3542.3 | 30.3 |
| 6 | 3516 | 3541.6 | 25.6 |
| 7 | 3513.7 | 3542.6 | 28.9 |
| 8 | 3506.5 | 3537.7 | 31.2 |

| | | | |
|----|--------|--------|------|
| 9 | 3512.6 | 3537.3 | 24.7 |
| 10 | 3508.8 | 3538.3 | 29.5 |
| 11 | 3512.7 | 3539.6 | 26.9 |
| 12 | 3510.7 | 3539.6 | 28.9 |
| 13 | 3511.4 | 3536.6 | 25.2 |
| 14 | 3508.8 | 3537.3 | 28.5 |
| 15 | 3509.5 | 3538.3 | 28.8 |
| 16 | 3511.1 | 3535.3 | 24.2 |
| 17 | 3510.4 | 3536.4 | 26 |
| 18 | 3512 | 3536.3 | 24.3 |
| 19 | 3511.4 | 3537 | 25.6 |
| 20 | 3512.4 | 3536.3 | 23.9 |

El promedio de las diferencias de la tabla 4 es de 26.6 [$\mu\epsilon$].

La diferencia entre el promedio de las diferencias de ambos dispositivos es de 0.26 [$\mu\epsilon$], con este resultado se garantiza que las lecturas obtenidas por dispositivo son fiables y por tanto es un proyecto realizado satisfactoriamente.

4.2 Comunicación a través de la página web

Las pruebas con el prototipo final no pudieron ser realizadas debido a factores ajenos al proyecto, es por esto que las pruebas con el microcontrolador de 32 bits se realizaron con la ayuda de una tarjeta de entrenamiento de Microchip que permite realizar la interconexión “RS232” con voltajes de +12 y -12 volts, un adaptador USB-RS232 con voltajes de -12 +12[V] y el adquisidor de datos que solo permite, por ahora, una conexión USB para realizar la comunicación y muestreo, por lo cual la configuración para la prueba con la interfaz Ethernet fue la siguiente (figura 4.8):

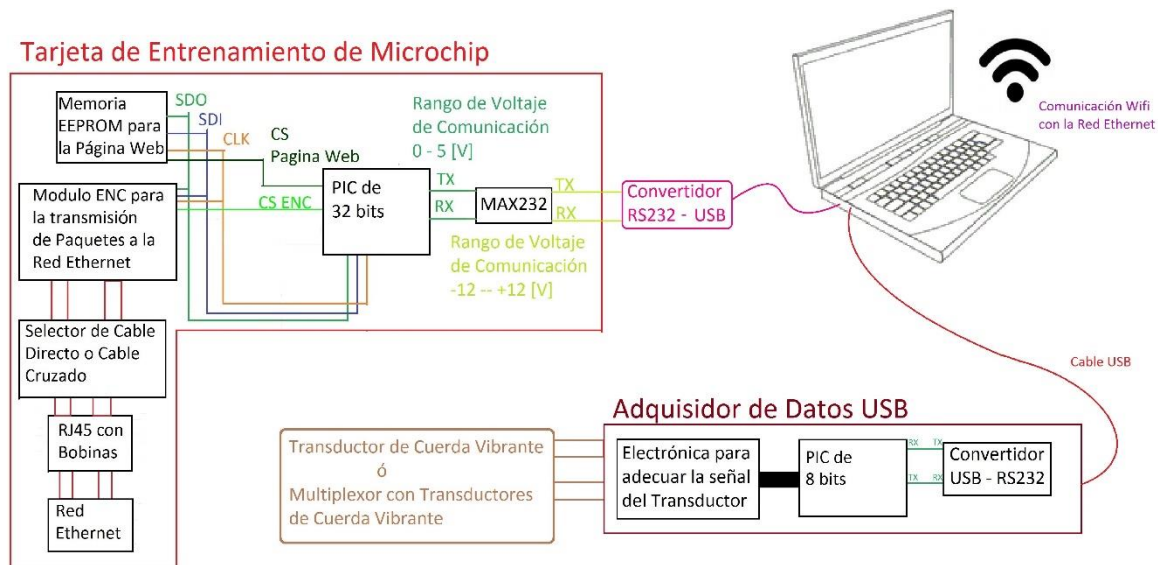


Figura 4.8 Esquema del Dispositivo Adquisidor Vía Ethernet

En la figura 4.8 se observa una laptop a la cual se están conectando los 2 dispositivos, el adquisidor de datos y la tarjeta de entrenamiento que está funcionando como el microservidor, para poder realizar el envío de los datos o comandos, en el siguiente diagrama lógico, figura 4.9, se puede apreciar que la laptop solo está funcionando como adaptador entre los protocolos de los dispositivos:

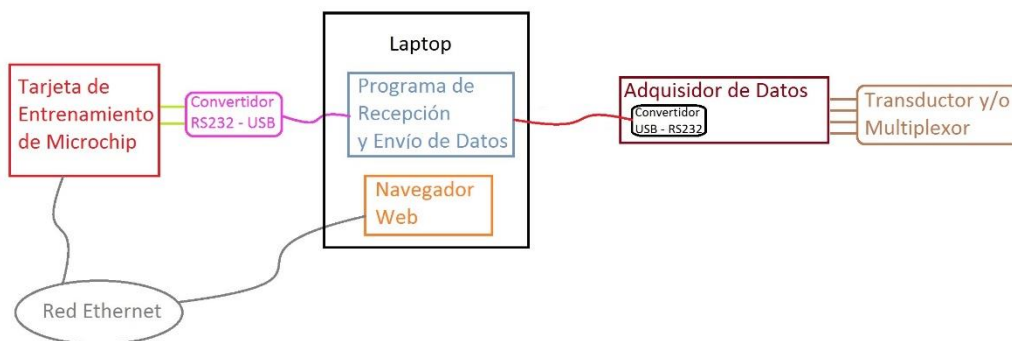


Figura 4.9 Diagrama del Funcionamiento del Adquisidor de Datos Vía Ethernet

La figura 4.9 muestra que la laptop está ejecutando 2 programas independientes, el programa de “Recepción y Envío de Datos” tiene a su cargo adquirir cualquier paquete de

datos enviado desde el puerto USB donde se ha conectado el Convertidor RS232-USB y el Adquisidor de datos, el cual tiene un dispositivo para convertir del protocolo “RS232” a USB, y reenviarlo hacia el otro puerto. En otras palabras, es correcto decir que la laptop está actuando como puente entre los protocolos “RS232” de cada microcontrolador, pues esta recibe los paquetes USB que cada dispositivo Genera y los reenvía hacia el puerto Contrario y cada convertidor está encargado de modificar cada paquete USB en una trama de datos RS232.

El Navegador Web permite realizar una conexión al microservidor donde está configurada la página web que muestra los datos y las posibles configuraciones de los canales del multiplexor. Las siguientes imágenes muestran cada programa siendo ejecutado al mismo tiempo por la PC “Host”.

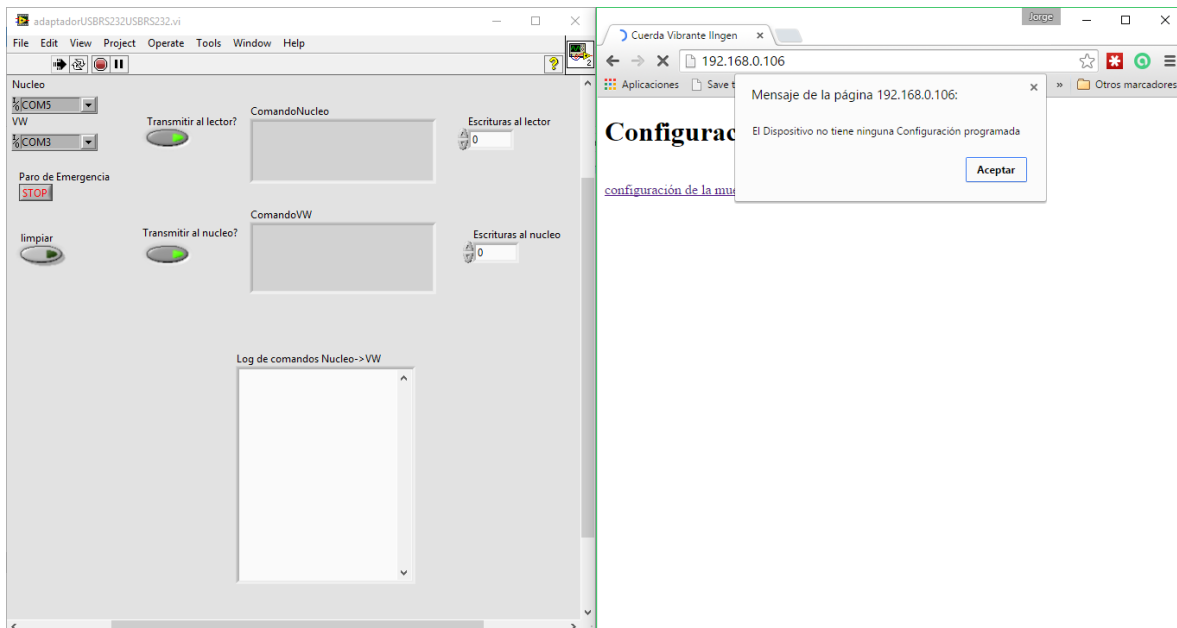


Figura 4.10 Ejecución de los programas Independientes en la PC

La figura 4.10 muestra el navegador web (lado derecho de la imagen) y el programa que se utiliza como puente entre las interfaces USB (lado izquierdo de la imagen) el cual tiene algunas funciones extra. A continuación, se describe de manera general el funcionamiento de cada control del programa de “Recepción y Envío de Datos” ilustrado en la figura 4.11.

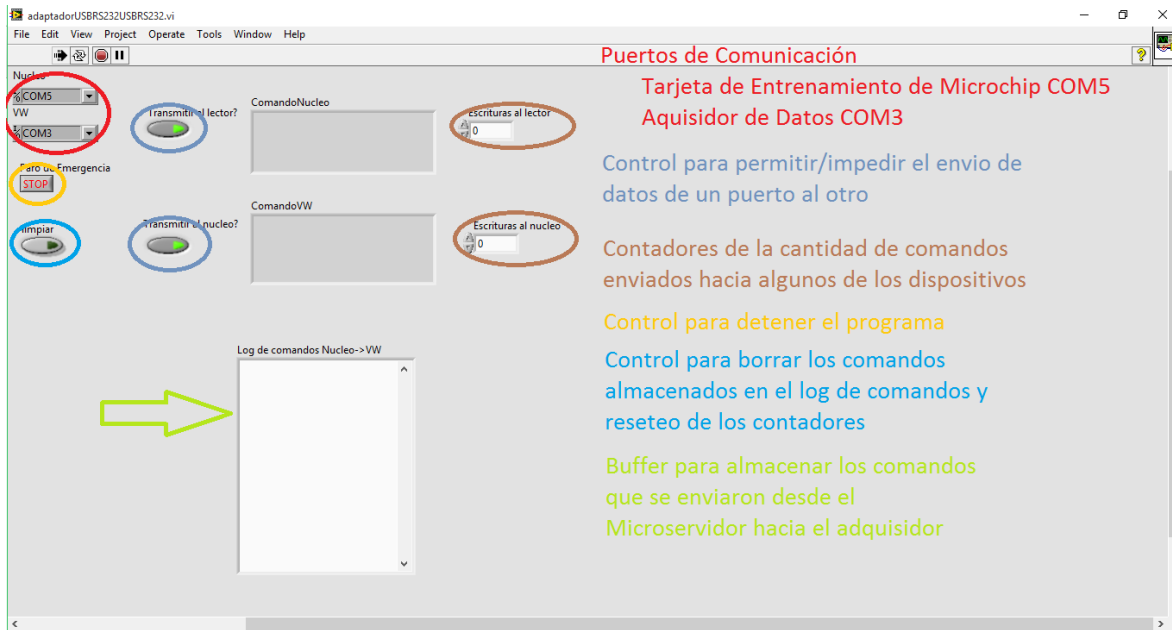


Figura 4.11 Explicación del programa de “Recepción y Envío de Datos”

Los “*Puertos de Comunicación*” son los controles que permiten seleccionar en que puerto USB está conectado cada dispositivo y configurar la velocidad de envío y recepción de datos.

Los “*Controles para permitir/impedir el envío de datos de un puerto al otro*” se utilizaron para realizar las pruebas de comunicación entre la tarjeta de entrenamiento de Microchip con la PC “*Host*” y del adquisidor de datos con la misma PC, para verificar que los comandos enviados por el microcontrolador de 32 bits fueran correctos.

Los “*Contadores de la cantidad de comandos enviados hacia alguno de los dispositivos*” permiten saber si el comando que se envió desde la tarjeta de entrenamiento se envió satisfactoriamente hacia el adquisidor y que la respuesta del mismo adquisidor se enviara correctamente hacia la tarjeta de entrenamiento.

El “*Control para borrar los comandos*” permite reiniciar los contadores y borrar todos los comandos que han pasado por el programa hacia el adquisidor de datos.

El “*Buffer para almacenar los comandos*” permite saber que comandos fueron enviados para verificar si existe algún error al enviar algún comando o algún comando no obtuvo una respuesta o la sincronización no es correcta.

Las siguientes imágenes muestran el comportamiento que tiene este programa dependiendo de las diferentes configuraciones de muestreo del microservidor

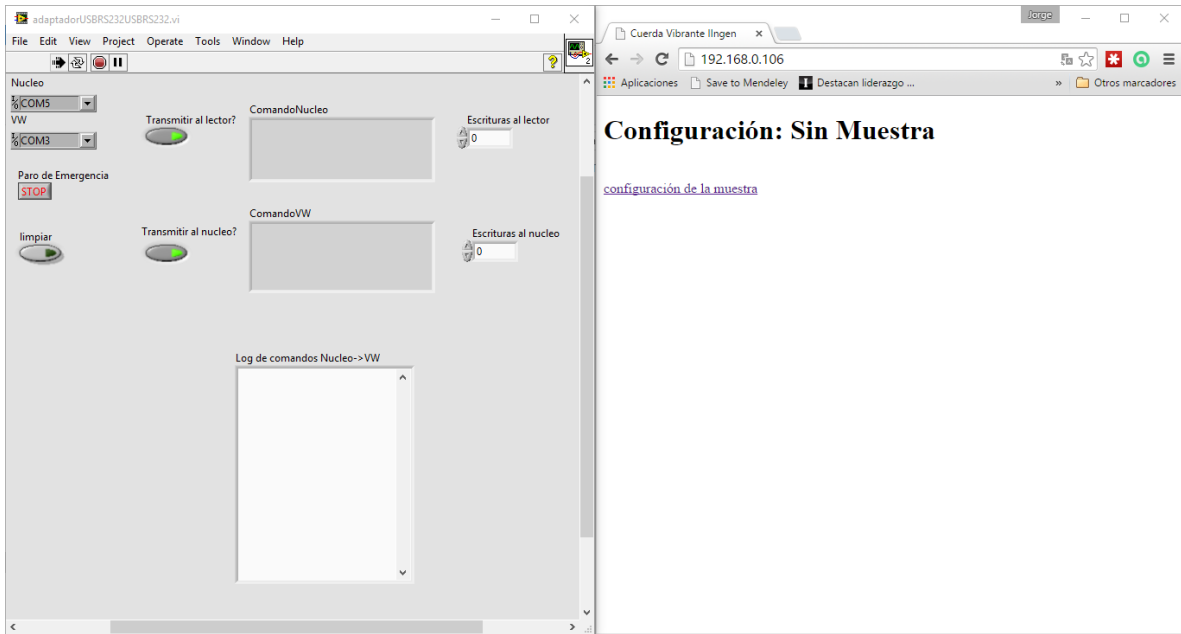


Figura 4.12 Comportamiento de los programas de la PC, Sin ninguna Configuración Programada

El programa no ha recibido ni enviado ninguno comando debido a que el microservidor no tiene configurada alguna muestra, como se muestra en la figura 4.12.

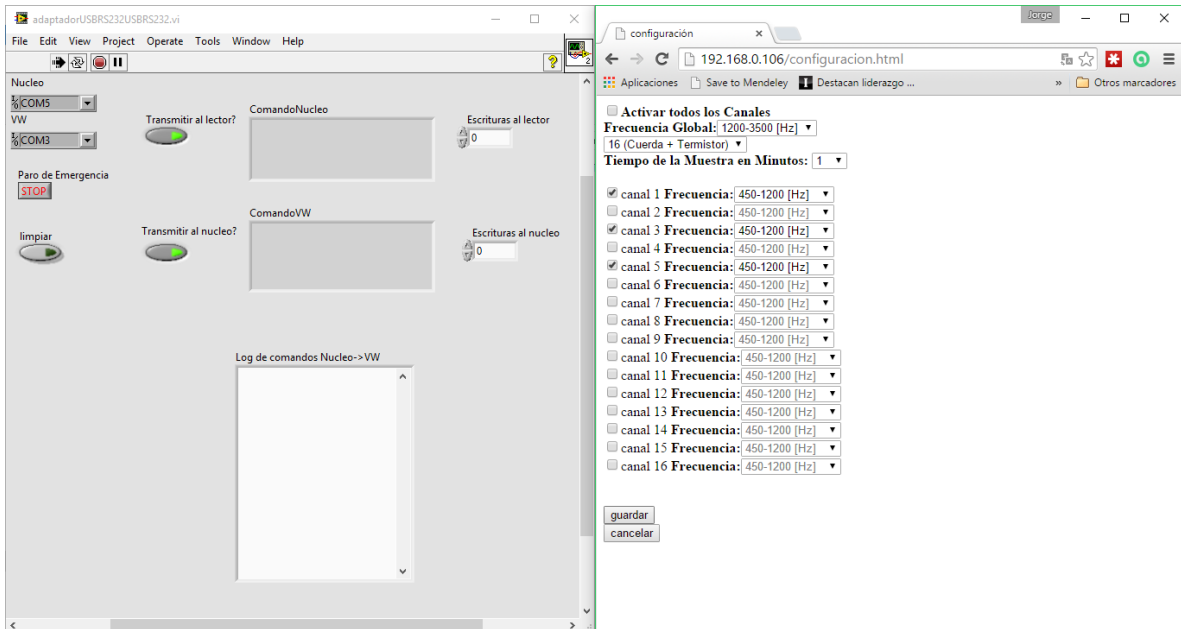


Figura 4.13 Comportamiento de los programas de la PC, Durante la Configuración de la Muestra

Al navegar hacia la página de configuración (configuracion.html) y seleccionar los canales que se desean leer, las frecuencias de la cuerda y el tipo de configuración del multiplexor,

se observa que aún no se han modificado ninguno de los contadores o el log de comandos, figura 4.13

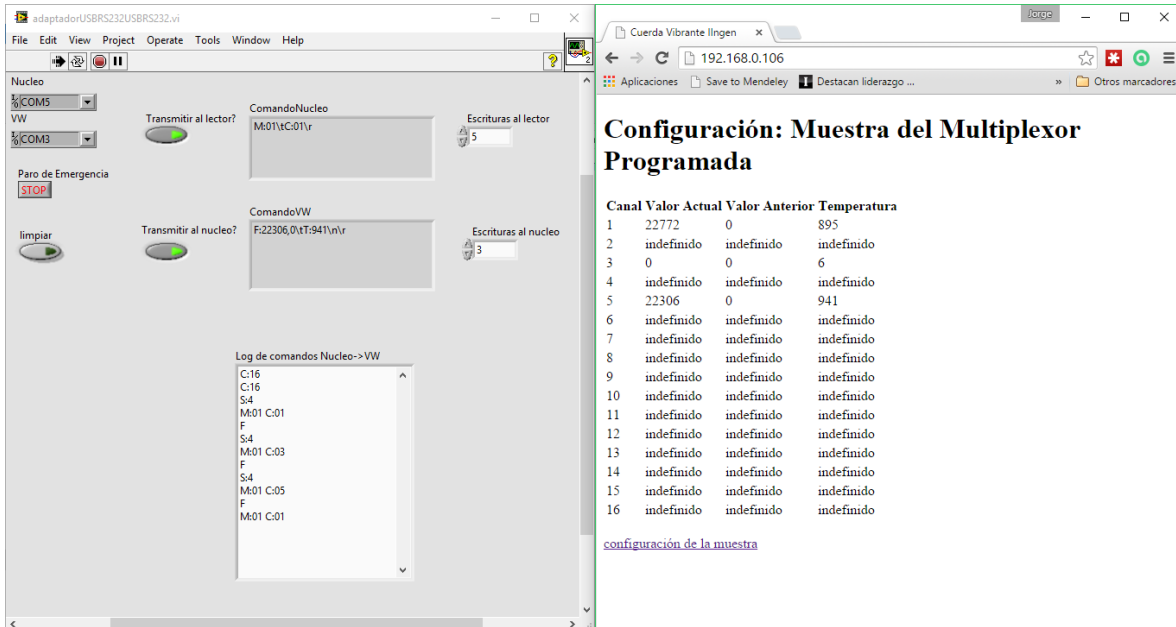


Figura 4.14 Comportamiento de los programas de la PC, Durante la Adquisición de los Datos

Al presionar el botón de “guardar” en la página de “configuracion.html”, se observan reflejados los cambios en ambos programas, se puede observar que los comandos que ha enviado el microcontrolador de 32bits y, en la página web, las respuestas enviadas desde el adquirente. Para fines demostrativos, la página muestra los datos tal cual los está recibiendo del adquirente, sin realizar alguna operación sobre los datos para poder verificar la correcta recepción de los mismos. Todo lo antes descrito se ilustra en la figura 4.14.

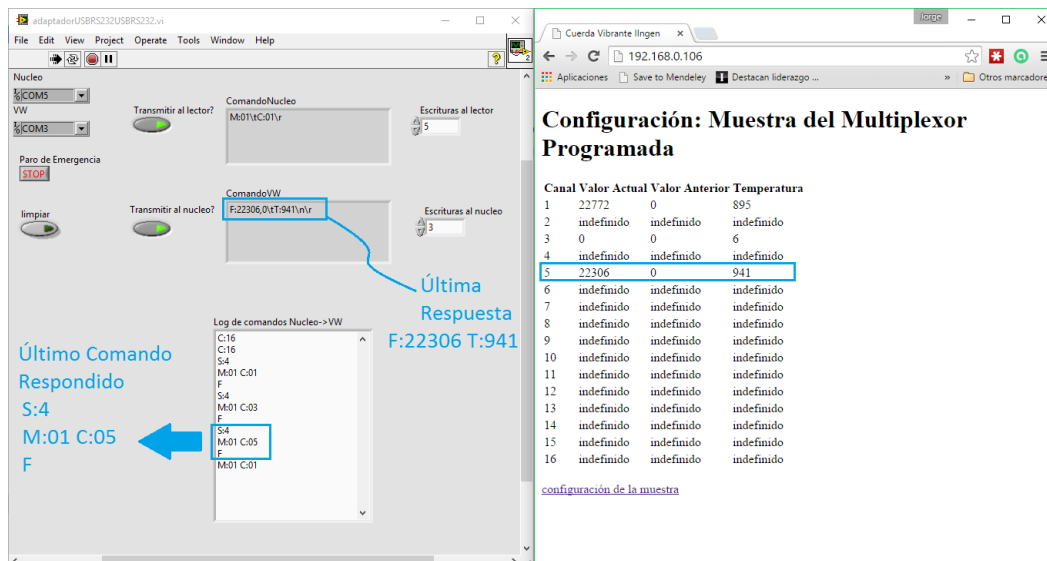


Figura 4.15 Comportamiento de los programas de la PC, Explicación de la Adquisición de los Datos

La figura 4.15 muestra que el último comando enviado fue la lectura del canal 5, a lo cual el adquisidor respondió con 22306 cuentas del contador de 16 bits. Este mismo número está siendo mostrado por la página web. Por lo cual se puede estar seguro de que los datos están siendo correctamente enviados desde el adquisidor hacia el microservidor web. Con esto se demuestra que, aunque no se pudiera trabajar con el prototipo final, el adquisidor de datos puede ser consultado en tiempo real de manera remota sin problemas desde cualquier dispositivo con un navegador web.

Conclusiones

Se logró desarrollar un sistema adquirente de datos programable y que permite ser consultado vía Ethernet o mediante la interfaz USB, el cual puede ser modificado de acuerdo a las necesidades de cada proyecto, ya que ha sido totalmente manufacturado y programado en el Instituto de Ingeniería. Esto da una gran ventaja, debido a que se pueden agregar más elementos o realizar modificaciones si así se requiere, sin la necesidad de violar alguna patente o derecho de autor.

El dispositivo adquirente de datos tiene la ventaja de que no está casado con alguna de las interfaces, tanto USB como Ethernet, por lo que se pueden crear diferentes programas o interfaces para realizar la adquisición de datos, creando así un producto flexible, pues solo se restringe a las instrucciones ya programadas.

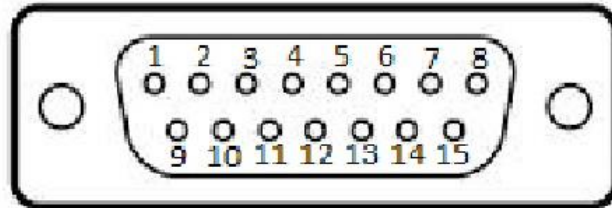
El adquirente de datos puede ser mejorado con hardware de mayor capacidad, tanto en la parte electrónica, que se refiere a los filtros de la señal generada hacia y recibida desde la "Cuerda Vibrante" y al convertidor Analógico/Digital, el cual sería externo al microprocesador y con más bits de resolución, como en la parte lógica, con un microprocesador de mayores capacidades tanto en memoria como en velocidad para la creación de más instrucciones.

En referencia al prototipo de interfaz Ethernet, las mejoras que se podrían hacer son tanto en el hardware, explotando las capacidades del microcontrolador de 32 bits, el cual incorpora la capa de enlace de datos con una dirección MAC propia, pero no incorpora los "transceiver" necesarios para manejar los voltajes y corrientes del medio físico. Al incorporar los "transceiver", se podría incorporar un Sistema Operativo en Tiempo Real, específico para los microcontroladores de Microchip, el cual administra el hardware de una manera más eficiente que el "Stack TCP/IP" que se está utilizando ahora, debido a que este último utiliza máquinas de estado para no acaparar todo el tiempo de procesamiento. Este prototipo se estima que tenga un tiempo de vida de 5 años debido a que el "Stack TCP/IP" ya no tendrá soporte para las nuevas aplicaciones de internet y la capa de encriptación SSL, la cual es importante para evitar que los datos se envíen "en plano" sobre la red.

Con la experiencia obtenida en el presente trabajo, será posible desarrollar un módulo C para el "CompactRIO" de "National Instrument", ya que se podría modificar la interfaz de comunicación serial, de "RS232" a "SPI", para que las lecturas de "Cuerda Vibrante" sean obtenidas a través del programa de Instrumentación Virtual "Labview".

Apéndice A Pines del Lector de Cuerda Vibrante con Conexión USB

Los pines del conector DA-15 para realizar el control del multiplexor y la lectura del Transductor están distribuidos de la siguiente manera, como se muestra en la figura A.1



- | | |
|----------------------------------|-----------------------------|
| 1.-Sin Conexión | Lectura del Transductor |
| 2.-Sin Conexión | |
| | Termistor |
| Lines de Control del Multiplexor | 12.- Parte Positiva |
| 3.-Reset | 13.- Parte Negativa |
| 4.-Reloj (CLK) | |
| | Cuerda Vibrante |
| Alimentación del Multiplexor | 14.-Parte Positiva (COM HI) |
| 5.-Tierra (Power Ground) | 15.-Parte Negativa (COM LO) |
| 9.-+12 [V] | |

Figura A.1 Pines del Lector de Cuerda Vibrante

En los caso en los cuales se conecte directamente el Transductor al Dispositivo Lector de Cuerda Vibrante o el Multiplexor este configurado para leer 16 Canales, se utilizaran los pines del 12 al 15 para realizar las lecturas de la Transductor, pero para el caso en el cual el Multiplexor se configura para 32 Canales, los pines 12 y 13 no estarán físicamente conectados al Multiplexor, las siguientes imágenes muestran el funcionamiento del Multiplexor en modo de 16 (figura A.2) y 32 (figura A.3) Canales

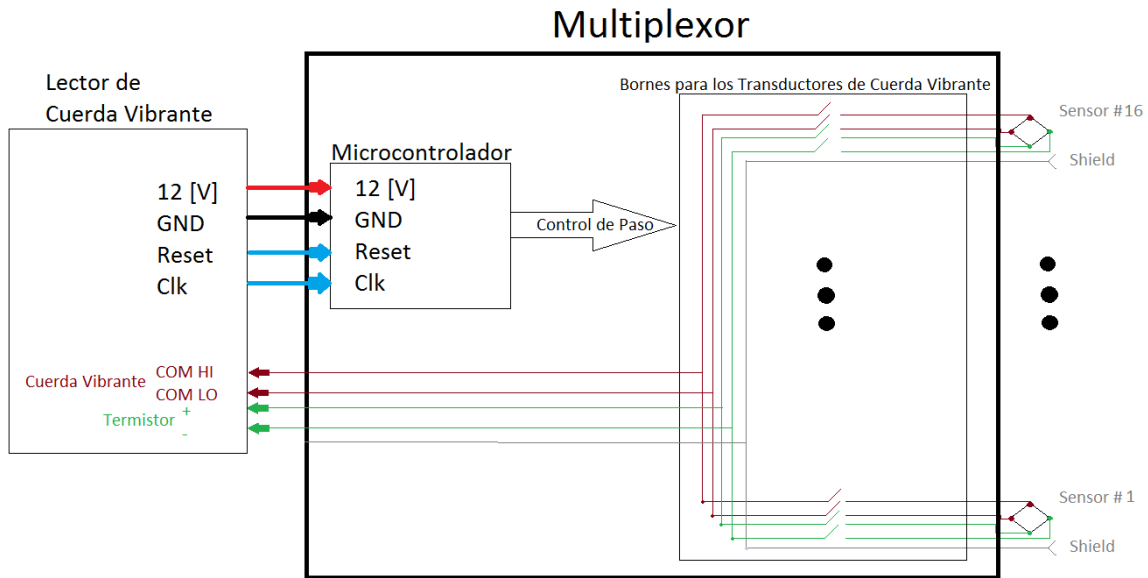


Figura A.2 Multiplexor en Modo de 16 Canales

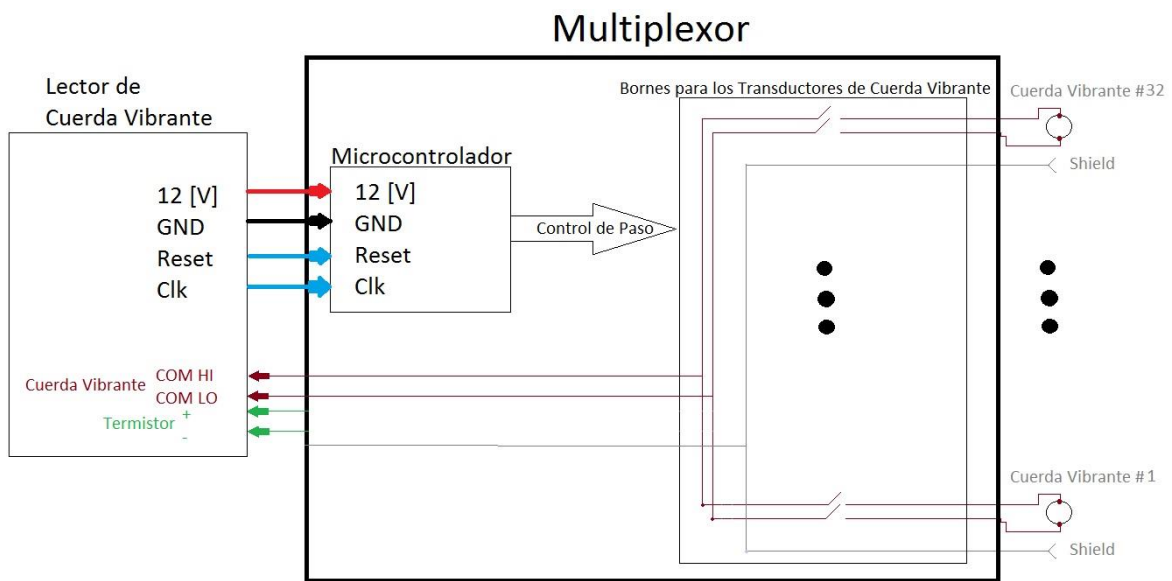


Figura A.3 Multiplexor en Modo de 32 Canales

La figura A.2 muestra que son necesarios 4 "Bornes" del multiplexor para que se pueda leer la Cuerda Vibrante y el Termistor, pero en el caso de la figura A.3 se observa que solo se están utilizando 2 "Bornes" para leer solo la Cuerda Vibrante, por lo que los 2 "Bornes" restantes, que eran utilizados para leer el Termistor, son usados para conectar otra cuerda Vibrante, duplicando el número de Canales en el Multiplexor.

Apéndice B Comandos del Lector de Cuerda Vibrante

El Dispositivo como tal puede ser leído sin la necesidad del Microservidor o el programa de Comunicación USB desarrollado para en este proyecto, por lo que se puede realizar un programa en cualquier otro lenguaje de programación para comunicarse con este dispositivo.

El Lector de Cuerda Vibrante utiliza un dispositivo de comunicación USB el cual trabaja con un controlador Genérico que puede ser reconocido por el Sistema Operativo “Windows” en sus versiones “7, 8, 8.1 y 10” para instalarse automáticamente, por lo cual en “Windows” no es necesario alguna licencia extra. Para este apéndice no se conectó el Transductor al dispositivo.

Los comandos son (Nota. - Todos están presentados en caracteres ASCII de 8 bits):

- LD\n\r
 - Lectura directa del transductor o de un canal del multiplexor que haya sido seleccionado
 - M:##\tC:##\n\r
 - Lectura del multiplexor número ## y del canal ##, donde ## es un número entre 01 hasta 99
 - M:##\tC:##,##\n\r
 - Lectura del multiplexor número ## y de una serie de canales que abarcan del canal ## hasta el canal ##, donde ## es un número entre 01 hasta 16 o 32, dependiendo de la configuración del multiplexor.
 - F\n\r
 - Finalización de la adquisición de datos
 - C:16\n\r ó C:32\n\r
 - Configuración del multiplexor, para lectura de 16 canales o 32 canales
 - S:#\n\r
 - Configuración del rango de frecuencia del Transductor de Cuerda Vibrante, donde # puede ser alguno de los siguientes valores

| Valor | Rango[Hz] |
|-------|--------------------------------|
| 0 | 450 – 1200 |
| 1 | 1200 – 3500 |
| 2 | 1000 – 3500 |
| 3 | 2500 – 6000 |
| 4 | 450 – 6000 (valor por default) |
- H\n\r
 - Detiene la adquisición de datos actual y muestra los comandos programados en el dispositivo

Esta serie de comandos puede ser visualizada mediante el comando “H\n\r” directamente desde el Dispositivo. En este caso, para demostrar que los comandos y los datos “*en bruto*” que envía el Dispositivo, se usó el programa llamado “Terminal”, en su versión “1.9b”, la figura B.1 muestra el programa y el puerto COM donde se ha conectado el dispositivo Lector de Cuerda Vibrante.

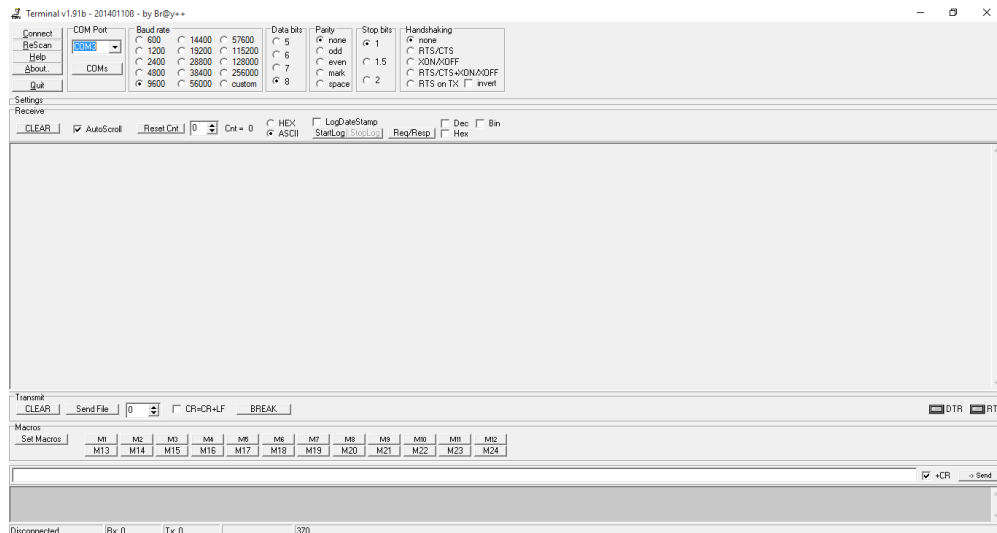


Figura B.1 Programa de Comunicación Genérico

La figura B.2 muestra una breve explicación de las secciones del programa “Terminal” que son utilizadas en esta ocasión

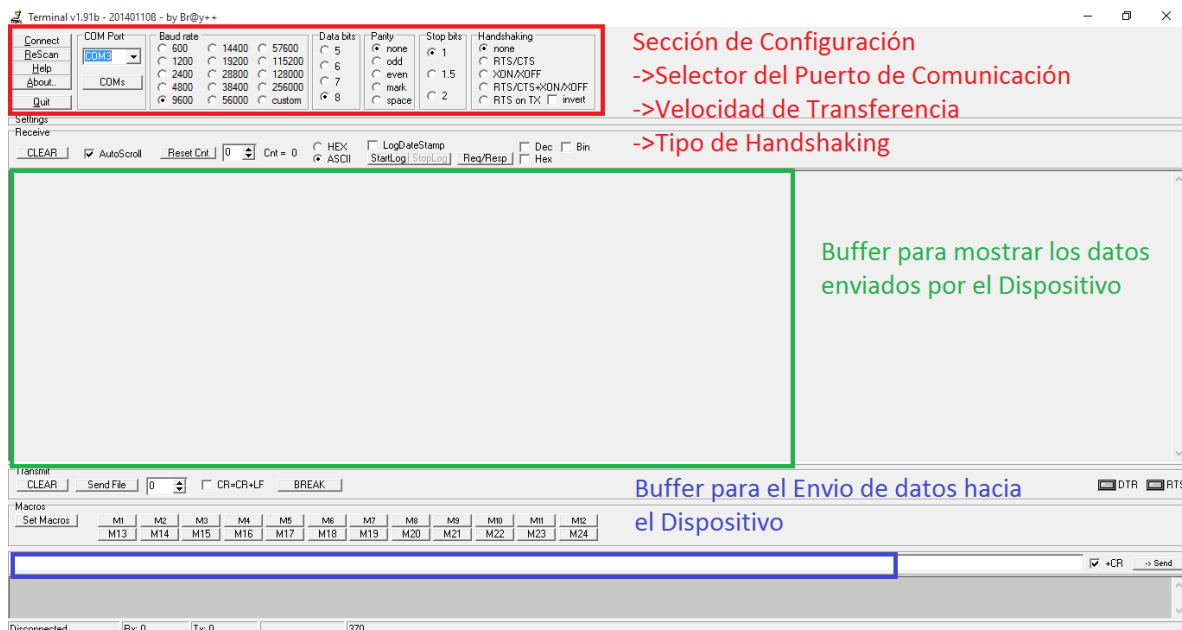


Figura B.2 Breve explicación del programa de Comunicación Genérico

Al presionar el botón de “Connect”, con la configuración mostrada, se realiza la transferencia de los datos que sean colocados en el “Buffer para Envío de datos” luego de que el botón de “->Send” sea presionado.

El dispositivo no tiene implementada una memoria que le permita almacenar la configuración que haya sido seleccionada antes de que el dispositivo se desenergizara, por lo cual, al ser conectado los valores por default que maneja son:

- Multiplexor configurado para 16 Canales
- Rango de frecuencia de 450 a 6000 [Hz]

Al enviar el comando “LDnV” el dispositivo comenzara a leer el Transductor que en ese momento se encuentre conectado o del canal del multiplexor que este configurado de manera indefinida y con la mayor velocidad posible, como se muestra en la figura B.3

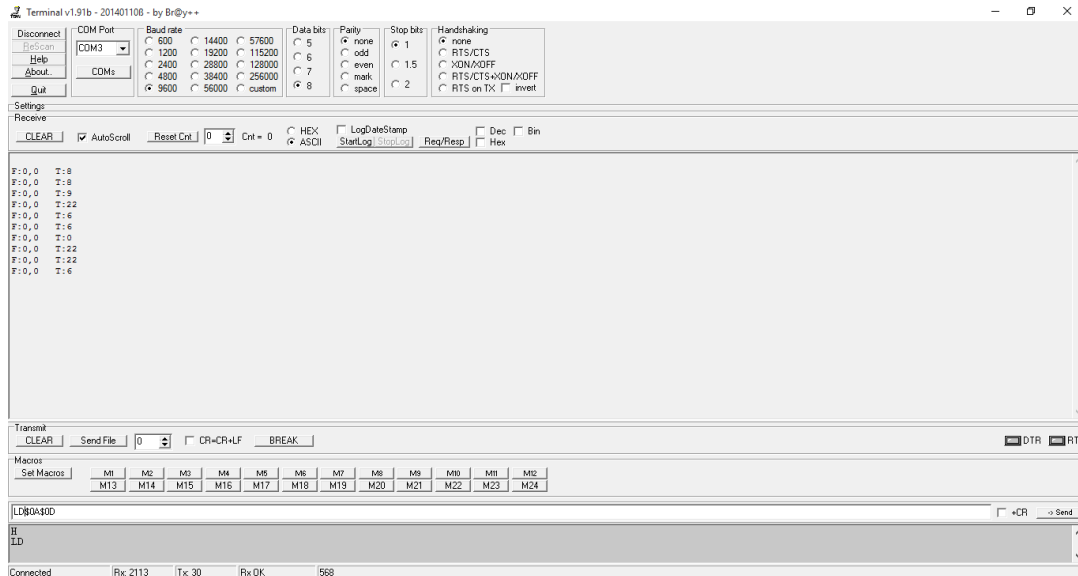


Figura B.3 Comando de Lectura Directa

Hasta que se envié el comando “FVnV” se detendrán las lecturas, como se muestra en la figura B.4

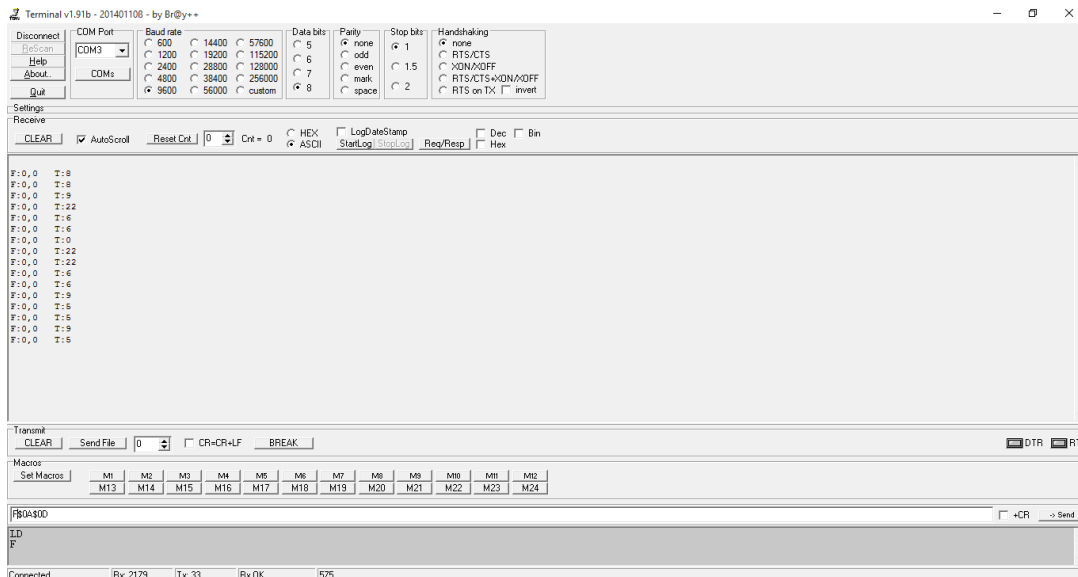


Figura B.4 Comando Fin de Lecturas

Al enviar el comando “M:##tC:##nV”, el dispositivo apuntara al canal especificado por “C:##”, realizara una lectura y la enviara de regreso, el canal no se cambiara, para este ejemplo el comando enviado fue “M:1tC:2nV” aunque podría haberse enviado el comando “M:01tC:02nV” realizando la misma acción, la figura B.5 muestra el comando.

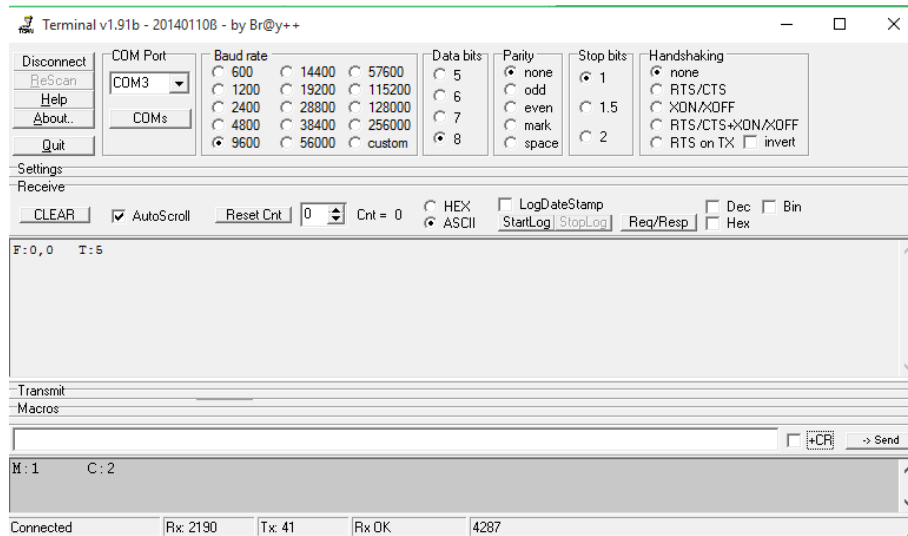


Figura B.5 Comando para Lectura de un Canal del Multiplexor

Al enviar el comando “M:##\tC:##,##\n\r”, el dispositivo apuntara a los canales que se encuentren en el rango de “C:##,##”, el dispositivo solo enviara la lectura de cada uno de los canales, no mostrara alguno otro dato, como se muestra en la figura B.6, para este ejemplo el comando enviado fue “M:1\tC:2,8\n\r”, aunque podría haber sido enviado como “M:1\tC:02,08\n\r”.

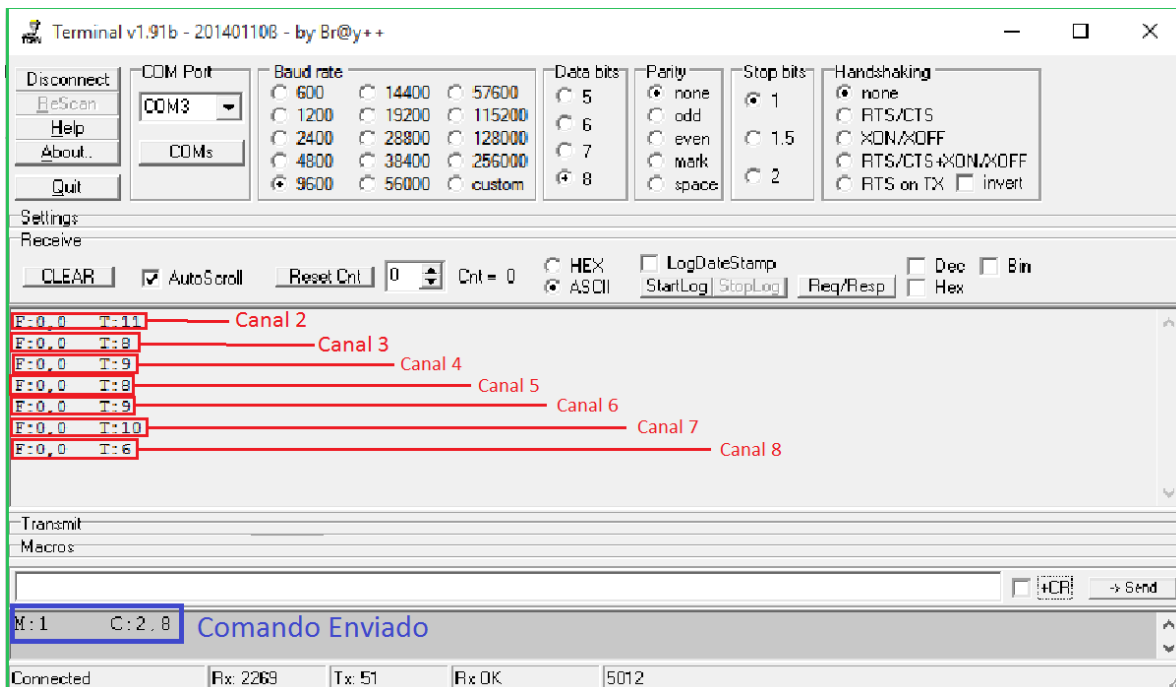


Figura B.6 Comando para Lectura de Varios Canales del Multiplexor

Al enviar los comandos de configuración “C:16\n\r” ó “C:32\n\r” y “S:#” el dispositivo no enviara ningún dato de regreso, en caso de haber sido escritos de manera correcta, como se muestra en la figura B.7

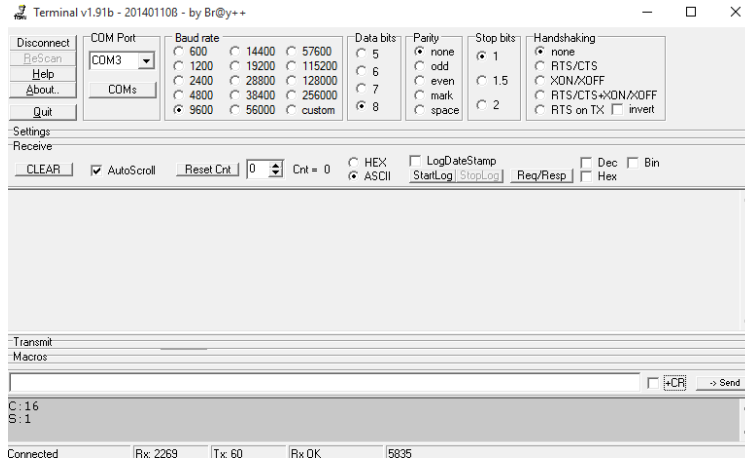


Figura B.7 Comando de Configuración del Multiplexor y Rango de Frecuencia

En caso de que no se envié correctamente algún comando o el dispositivo no reciba correctamente el comando, se enviara un mensaje de alerta como el que se muestra en la figura B.8

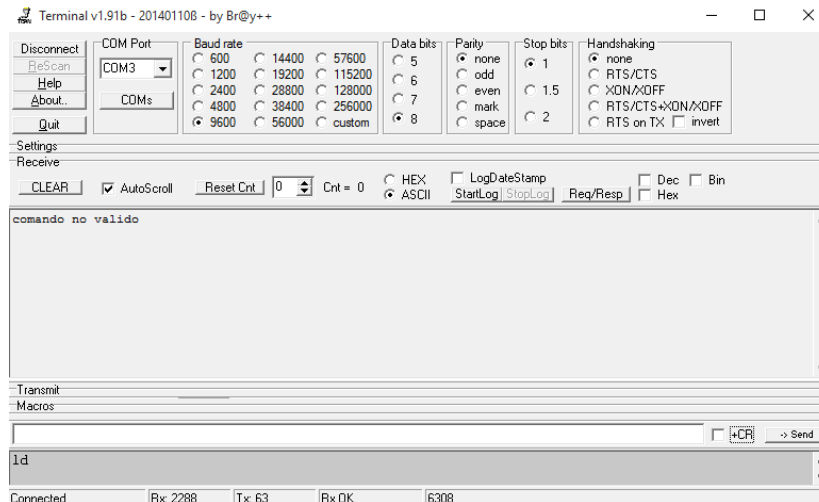


Figura B.8 Comando No Valido

Estos son los comandos básicos que soporta el Lector de Cuerda Vibrante con sus respectivas respuestas.

Glosario

A

adquisidor de dato
Dispositivo especializado para realizar mediciones de uno o más transductores., vi, 2, 5

Algoritmo
Conjunto de Reglas bien definidas, ordenas y finitas que permiten realizar una actividad., i, 2, 23

E

estándar
Conjunto de normas a las que se debe ajustar la información que se transmite para la interoperación de sistemas que deban manejarla., vi, 45

F

Firmware
Bloque de instrucciones en código máquina finito para propósitos específicos, el cual controla las entradas y salidas de los circuitos electrónicos de un dispositivo., 2
Bloque de instrucciones en código máquina para propósitos específicos., i, 23

H

host
Dispositivo que realiza peticiones de información a otros dispositivos., vi, 14, 17, 32

Host
Dispositivo que realiza peticiones de información a otros dispositivos, 13, 14, 15, 29, 80, 81

I

Instrumentación
Acción de colocar transductores que permitan la medición de magnitudes físicas mediante instrumentos especializados., vi, 46

interfaz
Medio físico utilizado para la transferencia de información entre dispositivos mediante un protocolo de comunicación., i, ii, iii, vi, 2, 9, 11, 12, 14, 15, 17, 19, 20, 21, 22, 23, 29, 32, 35, 40, 51, 79, 85

O

onda senoidal
Cuerva que representa gráficamente la función seno., 7, 73

P

periférico
Dispositivo independiente conectado y comunicado a otro dispositivo, 13, 14, 15

protocolo
En Redes, es un conjunto de reglas ordenas para la transferencia de datos entre diferentes equipos a través de la red Ethernet., vi, 11, 16, 17, 18, 23, 29, 40, 80

prototipo
Primer objeto que sirve como modelo para fabricar otros iguales., vi, 16, 20

T

transductor
Es un dispositivo capaz de trasformar una magnitud física en otra magnitud física que un sistema puede aprovechar para realizar la medición y control., i, iii, vi, 2, 3, 4, 5, 6, 8, 20, 21, 23, 26, 27, 28, 29, 30, 31, 32, 33, 55, 63, 64, 67, 71, 74, 75, 76, 86, 88, 90

Referencias

- Bejarano Urrego, L. E. (16 de Mayo de 2014). Tesina. *Instrumentación Geotécnica Del Revestimiento De Concreto*. D.F., Mexico.
- Bit Savers. (12 de Febrero de 2013). *Bit Savers*. Obtenido de SAGE: <http://bitsavers.informatik.uni-stuttgart.de/pdf/ibm/sage/>
- Dr. Navasquillo Hervás, J. (s.f.). *Departamento de Física Aplicada y Electromagnetismo*. Obtenido de Edificio de Investigación: <http://www.uv.es/~navasqui/Tecnologia/Tema4.pdf>
- Facultad de Ciencias Exactas, Física y Naturales, Universidad Nacional de Córdoba. (11 de 10 de 2002). *Universal Serial Bus*. Obtenido de The Easy Way to Plug & Play: <http://www.efn.uncor.edu/etc/liade/informacion/usb.pdf>
- Facultad de Ingeniería. (s.f.). Capitulo 2. *Principios De La Instrumentación De Estructuras*. D.F., México.
- Forero Casallas, J. A. (2006). Strain Gages. *Avances, Investigacion en Ingeniería*, 37-52.
- Fredrick, M. (1997). Introduction. En M. Fredrick, *Microcontrollers and Microcomputers, Principles of Software and Hardware Engineering* (págs. 1-4). New York, Oxford: Oxford University Press, Inc.
- GEOKON. (2009). Instruction Manual. *Model GK-403 Vibrating Wire Readout*. Lebano, United States of America.
- González Santiago, J. d. (2011). Tesis. *Proceso De Instrumentación Geotécnica*. D.F., México.
- Gutierrez, J. J. (27 de Marzo de 2009). *Periféricos Interfaces y Buses*. Obtenido de CTR - Computadores y Tiempo Real: <http://www.ctr.unican.es/asignaturas/pib/pib-tema-iii-b2-2en1.pdf>
- Huang, H.-W. (1996). Introduction to Motorola 68HC11. En H.-W. Huang, *MC 68HC11, An Introduction Software and Hardware Interfacing* (págs. 1-33). United States of America: West Publishing Company.
- Ing. Forero Saboya, N. G. (18 de 05 de 2012). *Normas de Comunicación en Serie: RS-232, RS-422 y RS-485*. Obtenido de Universidad Libre: <http://www.unilibre.edu.co/revistaingeniolibre/revista-11/art13.pdf>
- Ing. López Pérez, E. (s.f.). *Protocolo USB (UNIVERSAL SERIAL BUS)*. Obtenido de Ingeniería en Microcontroladores: <http://www.i-micro.com/pdf/articulos/usb.pdf>
- Ing. Macías Ríos, M. E., Rengería Anguiano, E., & Dr. Rangel Licea, V. (Octubre de 2009). *Facultad de Ingeniería*. Obtenido de Apuntes de la Asignaturas de Redes I y Redes II: <http://profesores.fi->

b.unam.mx/victor/CCNA/Productos/Notas%20de%20Curso/Manual%20de%20la%20Asignatura%20de%20Redes%20de%20Datos%20I%20y%20II%20%20(avance%2050%25).pdf

- Microchip Technology Inc. (2005). Explorer 16 Development Board User's Guide. *DS51589A*. United States of America.
- Microchip Technology Inc. (2009). ENC424J600/624J600 Data Sheet. *DS39935B*. United States of America.
- Microchip Technology Inc. (2009). PIC32MX5XX/6XX/7XX Family Data Sheet. *DS61156G*. United States of America.
- Microchip Technology Inc. (2010). PIC32 Ethernet Starter Kit User's Guide. *DS61166A*. United States of America.
- Microchip Technology Inc. (2011). Microchip TCP/IP Stack Help. United States of America.
- Microchip Technology Inc. (2012). MPLAB XC16 C Compiler User's Guide. *DS50002071D*. United States of America.
- Microsoft. (29 de 11 de 2013). *Microsoft*. Obtenido de Definición de las siete capas del modelo OSI y explicación de las funciones: <https://support.microsoft.com/es-es/kb/103884>
- Montaña, R. (22 de Noviembre de 2007). *Red Iris*. Obtenido de Ethernet: de 2,94 A 1000 Mbps en 25 años: <http://www.rediris.es/difusion/publicaciones/boletin/46-47/ponencia9.html>
- Rahbharti, N. (2002). The Microchip TCP/IP Stack. *AN833*. United States of America.
- Redes Telemáticas. (17 de Septiembre de 2013). *Historia de Internet*. Obtenido de Nacimiento y Evolución: <http://redestelematicas.com/historia-de-internet-nacimiento-y-evolucion/>
- Reen, P., & Mohanswany, N. (2008). Implementing File I/O Functions Using Microchip's Memory Disk Drive File System Library. *AN1045*. United States of America.
- Textos Científicos. (2 de 10 de 2006). *Textos Científicos*. Obtenido de TCP/IP Y EL MODELO OSI: <http://www.textoscientificos.com/redes/tcp-ip/comparacion-modelo-osi>
- Universidad Autónoma de San Luis Potosí. (21 de Octubre de 2015). *Facultad de Ciencias*. Obtenido de Facultad de Ciencias, : <http://galia.fc.uaslp.mx/~cantocar/z8plus/documentos/historia01.pdf>
- Valdez Perez, F. E., & Pallas Areny, R. (2009). Introduction to Microcontrollers. En F. E. Valdez Perez, & R. Pallas Areny, *Microcontrollers, Fundamentals and Applications with PIC* (págs. 1-12). United States of America: CRC Press.