

INSTITUTO TECNOLÓGICO SUPERIOR

AERONÁUTICO

ESCUELA DE TELEMÁTICA

**IMPLEMENTACION DE UN COMPROBADOR DE
CONTINUIDAD DE CABLES PARA TELECOMUNICACIONES Y
MEDICIONES ELECTRÓNICAS**

POR:

Cristián Xavier Espín Beltrán

Christian Roberto Tapia Gaibor

Tesis presentada como requisito para la obtención de título de:

TECNÓLOGO EN TELEMÁTICA

2003

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por los Srs: Cristian Espín y Cristian Tapia, como requisito parcial a la obtención del título de TECNÓLOGO EN TELEMÁTICA.

Latacunga Enero del 2003

Ing. Pablo Pilatasig

DEDICATORIA

El presente trabajo lo dedicamos a todos aquellas personas que hicieron posible la realización del mismo, y de una manera muy especial a nuestros padres, compañeros, educadores e institución que siempre nos brinda el apoyo que se requiere para alcanzar metas como estas.

Esperando que esta tesis sea de gran ayuda para el mejor manejo, organización y superación de la Institución.

**CRISTIAN ESPÍN
CHRISTIAN TAPIA**

AGRADECIMIENTO

Nuestro sincero agradecimiento a Dios, que por intermedio de nuestros padres nos ha guiado con amor hasta esta etapa de nuestra vida con su apoyo moral y decidido en cada día de nuestra vida, poniendo todo su empeño desinteresado, buscando nuestro bienestar personal.

A nuestro rector de escuela que con su comprensión y flexibilidad a permitido que nuestro trabajo no sea tan presionante y lográramos cumplirlo a cabalidad sin sustos de que ya estamos contra el tiempo.

A todo el personal docente y administrativo del INSTITUTO TECNOLÓGICO SUPERIOR AERONÁUTICO y en especial al Ing. Pablo Pilatasig, quienes han hecho posible la realización de este trabajo.

CRISTIAN ESPÍN
CHRISTIAN TAPIA

INDICE

INTRODUCCIÓN	1
CAPITULO I	
1.1.- El problema	2
1.1.1.- Definición del problema	2
1.2.- Objetivos	3
Objetivo General	3
Objetivo Específico	3
1.3.- Justificación	3
CAPITULO II: MARCO TEÓRICO	
2.1.-Conectores RJ11	5
2.1.1.- Uso	5
2.1.2.- Construcción del cable	5
2.1.2.1.- Esquema	5
2.1.2.2.- Materiales	6
2.2.- Conectores RJ45	6
2.2.1.- Uso	6
2.2.2.- Asignación de pines	7
2.2.3.- Esquema de la conexión PC->hub	7
2.2.4.- Esquema cable ‘cruzado’ para conexión directa PC->PC (sin hub)	

2.2.5.- Construcción del cable	8
2.2.5.1.- Materiales	8
2.3.- Puerto serial	9
2.3.1.- Interfaz RS-232	12
2.3.2.- Direcciones	15
2.4.- Labview	17
2.4.1.- Introducción	17
2.4.2.- Características	17
2.4.3.- Aplicaciones	18
2.4.4.- Programación Gráfica con Labview	19
2.4.4.1.- Diseñe la Interfaz de Usuario a partir de su Código.	19
2.4.4.2.- Modularidad	21
2.4.5.- Compilador Gráfico	21
2.4.5.1.- Paleta de Control	22
2.4.5.2.- Paleta de Funciones	23
2.5.- Descripción de las Funciones y Controles Utilizados	24
2.5.1.-Funciones	24
2.5.1.1- Structure	
24 2.5.1.1.1.- Sequence Structure	24
2.5.1.1.2.- Case Structure	25
2.5.1.1.3.- While Loop	25
2.5.1.2.- Numerico	26
2.5.1.2.1.-Numérica Constant (Constante Numerica)	26
2.5.1.3.- Booleano	26
2.5.1.3.1.- True Constant	26

2.5.1.3.2.- False Constant	26
2.5.1.4.– String Cadena	26
2.5.1.4.1.- Search And Replace String	26
2.5.1.4.2.- String Subset	27
2.5.1.4.3.- Concatenate Strings	28
2.5.1.4.4.- Decimal StringTo Number	28
2.5.1.4.5.- String constant	29
2.5.1.4.6.- Empty String	29
2.5.1.4.7.- Tab	29
2.5.1.5.– Arrays	30
2.5.1.5.1.-Build Array	30
2.5.1.6.– Comparación	30
2.5.1.6.1.- Equal? ¿Igual?	30
2.5.1.6.2.- Greater Or Equal? (Mayor O Igual)	31
2.5.1.7.– Tiempo Y Dialogo	31
2.5.1.7.1.- Wait	31
2.5.1.7.2.- Get Date/Time String	32
2.5.1.8.– File I/O	32
2.5.1.8.1.- Open/Create/ReplaceFile	
32 2.5.1.8.2.- Write File	34
2.5.1.8.3.- Close File	35
2.5.1.9.– Instrumentos I/O	36
2.5.1.9.1.- Serial Port Init	36
2.5.1.9.2.- Serial Port Write	37
2.5.1.9.3.- Bytes At Serial Port	37

2.5.1.9.4.- Serial Port Read	38
2.5.1.9.5.- Close Serial Driver	39
2.5.1.9.6.- Property Node	39
2.5.2.-Controles	39
2.5.2.1.- Numéricos	39
2.5.2.1.2.- Controles E Indicadores	39
2.5.2.2.- Boolean	40
2.5.2.2.1.- Creating And Operating Boolean Controls And Indicators	40
2.5.2.3.- Gráficos	40
2.5.2.3.1.- Graph Controls	40
2.5.3.-Paleta De Herramientas	40
2.5.3.1.- Operating	40
2.5.3.2.- Positioning	40
2.5.3.3.- Labeling	41
2.5.3.4.- Wiring	41
2.5.3.5.- Object Shortcut Menu	41
2.5.3.6.- Scroll	41
CAPITULO III: DESARROLLO DE LA INTERFAZ	42
3.1 Analisis de Alternativas	42
3.1.1.- Visual Basic.	42
3.1.1.1.-Introducción.	42
3.1.1.2.- Características de Visual Basic.	43

3.1.1.3.- Ventajas y Desventajas de Visual Basic	44
3.1.2.- Visual C++.	44
3.1.2.1.- ¿Qué Es?	44
3.1.2.2.- Características Principales	45
3.1.3.- Labview	45
3.1.3.1.- Definición	45
3.1.3.2.- Características	45
3.1.3.3.- Aplicaciones	46
3.1.3.4.- Ventajas	46
3.2 Análisis Y Selección De Alternativas	48
3.2.1.- Software	48
3.2.2.- Multímetro	49
3.3.- Requerimientos Técnicos	51
3.3.1.- El Comprobador De Cables	51
3.3.1.1.- Funcionamiento	51
3.3.1.2.- Desarrollo	53
3.3.1.2.1.- Materiales	53
3.3.1.2.2.- Diagrama del circuito	53
3.3.1.2.3.- Circuito Impreso	54
3.3.1.2.4- Montaje de Componentes	54
3.3.1.2.5.- Colocación en La Carcasa	55
3.3.1.3.- Uso del Comprobador	55
3.4 Desarrollo de la Interfaz Grafica	56
3.5 Pruebas y Análisis de Resultados	65

3.5.1.- Pruebas y Análisis de Resultados del Comprobador	65
3.5.2.- Pruebas y Análisis de Resultados el Protek 506	65
3.5.2.1.- Dc Voltaje	65
3.5.2.2.- Ac Voltaje	66
3.5.2.3.- Dc Corriente	66
3.5.2.4.- Ac Corriente	67
3.5.2.5.- Resistencia	67
3.5.2.6.- Continuidad	68
3.5.2.7.- Diodos	68
3.5.2.8.- Frecuencia	68
3.5.2.9.-Temperatura	69
3.5.2.10.- Lógico	69
3.5.2.11.- Señal de Salida	69
3.5.2.12.- Capacitancia	70
3.5.2.13.- Inductancia	72
CAPITULO IV:	72
4.1.- Cronograma	72
4.2.- Presupuesto	73
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES	74
5.1.- Conclusiones	74
5.2.- Recomendaciones	75

BIBLIOGRAFIA

ANEXOS

GLOSARIO

LISTADO DE TABLAS

Numero	Descripción	
Página		
2.1	esquema del conector RJ11	5
2.2	asignación de pines	7
2.3	Asignación de pines PC-HUB	7
2.4	Asignación de pines PC-PC	8
2.5	Transmisión de una dato	12
2.6	Descripción de la interfaz RS-232	13
2.7	Conexiones del RS-232	14
2.8	Direcciones IQR	16
3.1	Descripción del voltaje DC	65
3.2	Descripción del voltaje AC	66
3.3	Descripción de corriente DC	66
3.4	Descripción del corriente AC	67
3.5	Descripción de resistencia	67
3.6	Descripción de continuidad	68
3.7	Descripción de diodos	68
3.8	Descripción de frecuencia	68
3.9	Descripción de temperatura	69
3.10	Descripción de la función lógico	69

3.11	Señal de salida	69
3.12	Descripción de Capacitancia	70
3.13	Comprobación de valores de capacitancias	70
3.14	Descripción de inductancia	71
3.15	Comprobación de valores de inductancias	

LISTADO DE GRÁFICOS.

Numero	Descripción	
Página		
2.1	Conector RJ11	5
2.2	Ponchadora	6
2.3	Ponchar un cable	9
2.4	Conectores DB25 y DB9	16
2.5	Interfaz con el Usuario.	20
2.6	Paleta de controles	22
2.7	Paleta de Funciones	23
2.8	Sequence Structure	24
2.9	Case Structure	25
2.10	While Loop	25
2.11	String Cadena	26
2.12	String Subset	27
2.13	Concatenate String	28
2.14	Decimal String To Number	29
2.15	Build Array	30
2.16	Equal?	31

2.17	Greater or Equal?	31
2.18	Wait (ms)	31
2.19	Get Date/Time String	32
2.20	Open/Create/Replace File	33
2.21	Write File	34
2.22	Close File	35
2.23	Serial Port INIT	36
2.24	Serial Port Write	37
2.25	Bytes at Serial Port	38
2.26	Serial Port Read	38
2.27	Close Serial Driver	39
2.28	Control e Indicador	40
2.29	Paleta de Herramientas	41
3.1	Software	48
3.2	Protek 506	49
3.3	Comprobador	51
3.4	Comprobador seleccionado en RJ11	52
3.5	Comprobador seleccionado en RJ45	52
3.6	Diagrama del circuito	54
3.7	Interfaz gráfica	56
3.8	Inicializar el puerto serial.	56
3.9	Enviar los datos medidos a un documento.	58
3.10	Variable local	58
3.11	Fecha y hora del programa.	59
3.12	Orden de los datos	59

3.13	Enviar los datos a un archivo de texto	60
3.14	Escribir en el puerto serial	60
3.15	Sincronizar el tiempo	60
3.16	Desarrollo del programa	61
3.17	Contador de bytes	62
3.18	Buscar y reemplazar caracteres.	62
3.19	Búsqueda de caracteres	63
3.20	Presentación de la temperatura.	63
3.21	Cerrar el puerto.	64

LISTADO DE ANEXOS

Anexo	Figura	
Página		
A	A.1 Circuito Impreso	1
B	B.1 Montaje de componentes	1
C	C.1 Colocación en la carcasa	1
D	D.1 Formato de datos	1
E	E.1 Conexión del multímetro a la PC	1

CAPITULO I

EL PROBLEMA

1.1.- DEFINICIÓN DEL PROBLEMA

Frecuentemente las redes de los Laboratorios de Informática dejan de funcionar sin razón alguna y siempre perdemos bastante tiempo y en ocasiones dinero en tratar de arreglar este pequeño inconveniente, revisando el software si esta bien instalado o tiene algún defecto, pero con mucha frecuencia el software siempre se encuentra en perfectas condiciones y no se sabe dónde está el error, hasta que se pone a verificar los conectores del cable de la red pin por pin, para ver si tiene continuidad, encontrando el problema en uno de los pines del cable, para evitar esta demora se decidió elaborar el probador de continuidad el mismo que verificará todos los pines del cable al mismo tiempo.

Normalmente cuando se realiza prácticas en los Laboratorios de Electrónica y se necesita un sintonizador para lo cual es necesario un capacitor o un inductor de cierto valor, pero en el laboratorio se tienen pocos inductores marcados con su valor al igual que los capacitores y lo que se necesita no se encuentra marcado, entonces es cuando se utiliza el multímetro Protek 506, en consecuencia los alumnos realizarán con mayor eficiencia sus prácticas y aprenderán de mejor manera, puesto que tendrán todos los materiales a la mano.

1.2.- OBJETIVOS

OBJETIVO GENERAL

- Elaborar un dispositivo con el cual se pueda comprobar cables de redes, e implementar un medidor digital con interfaz a la computadora.

OBJETIVO ESPECIFICO

- Detectar con rapidez el estado de funcionamiento de cables de Telecomunicaciones.
- Implementar en el Laboratorio de comunicaciones un medidor que permitirá medir inductancias y capacitancias de los elementos.
- Visualizar en forma gráfica en un PC las variaciones de las mediciones electrónicas efectuadas por el medidor.

1.3.- JUSTIFICACIÓN

Para lograr superar el problema de los cables de telecomunicaciones se requiere de un dispositivo el cual nos permita realizar este trabajo eficientemente. El dispositivo sería el Comprobador de Cables de Telecomunicaciones. Por este motivo si alguna vez deja de funcionar la red lo primero es revisar el cable en el verificador y luego si está bien, revisar el resto de posibles fallas.

En los laboratorios también existe un gran problema en el área de telecomunicaciones, entonces para corregir este problema lo vamos a realizar mediante un multímetro digital, con interfaz RS-232 que permite enviar esta información a la computadora, este multímetro nos permite medir inductancias, capacitancias, medida de decibeles, valores mínimos, máximos y relativos, tiene memoria que nos permite trabajar en el campo y procesar los datos en la PC, tiene una alta precisión de +- 0.01% y permitirá tanto al profesor como al alumno una mayor facilidad de obtención de datos y aportara a la modernización de los laboratorios de comunicaciones de Telemática y Aviónica.

La misión de la elaboración de estos dispositivos es la de mejorar y ayudar el desarrollo tecnológico del instituto con instrumentos que permitan mejorar la enseñanza y aprendizaje de los alumnos en la asignatura de: Redes de datos, Sistema de Comunicaciones y Laboratorio de Informática en las escuelas de Telemática y Aviónica.

Con la implementación de este comprobador se beneficia los profesores de los laboratorios de comunicaciones y los alumnos debido a la gran facilidad de utilización y exactitud en los datos obtenidos y lo podemos llevarlo a trabajar a lugares aislados y guardar la información y luego procesar los datos en las PC sin ningún riesgo de perderlos y además mejora la tecnología de los laboratorios de comunicaciones.

CAPITULO II

MARCO TEORICO

2.1- CONECTORES RJ11.

Son cables de 4 hilos que se utiliza comúnmente para telefonía y también se lo usa en redes.



Fig 2.1 Connector RJ11

2.1.1.- USO.

Para conexión del módem a la base (roseta) de la red telefónica (RTC)

2.1.2.- CONSTRUCCIÓN DEL CABLE

2.1.2.1.-ESQUEMA

Tabla 2.1 Esquema del conector RJ11

PIN	->	PIN
1		4
2		3
3		2
4		1

2.1.2.2.-MATERIALES.

Cable telefónico plano de cuatro hilos.

Dos conectores RJ11 (cuatro pines).

Ponchadora.



Fig. 2.2 Ponchadora

2.2.- CONECTORES RJ45

Son cables de red de par trenzado UTP Categoría 5 (Cat. 5) de 8 hilos para 100 BaseTX.

Con sus respectivos conectores RJ45, uno para cada punta, que son iguales que los de teléfono pero más grandes.

2.2.1 USO

Los cables se utilizan para conectar los equipos, ya sean conexiones entre Hubs, de PC a Hub, o de PC a PC, para conectar tarjetas ethernet.

2.2.2.-ASIGNACIÓN DE PINES

Tabla 2.2 Asignación de pines

Pin	A	HUB	Señales que transporta el cable y nomenclatura de los dos extremos del mismo.
1	Input Receive Data +	Output Transmit Data +	
2	Input Receive Data -	Output Transmit Data -	
3	Output Transmit Data +	Input Receive Data +	
6	Output Transmit Data -	Input Receive Data -	
4,5,7,8	Not Used	Not Used	

2.2.3.-ESQUEMA DE LA CONEXIÓN PC->HUB

Tabla 2.3 Asignación de pines PC-hub

Orden de colores del conector RJ45		
<i>Numero de Pin</i>	<i>Color</i>	<i>Abreviatura del color</i>
1	Blanco-Naranja	BN
2	Naranja	N
3	Blanco-Verde	BV
4	Azul	A
5	Blanco-Azul	BA
6	Verde	V
7	Blanco-Marrón	BM
8	Marrón	M
Solapa del conector mirando hacia abajo extremo hacia delante		

**2.2.4.-ESQUEMA CABLE 'CRUZADO' PARA CONEXIÓN DIRECTA
PC->PC (SIN HUB)**

Tabla 2.4 Asignación de pines PC-PC

Extremo 1	-	Extremo 2
Naranja y blanco	Pin 1 a Pin 3	Blanco y verde
Naranja	Pin 2 a Pin 6	Verde
Verde y blanco	Pin 3 a Pin 1	Blanco y naranja
Azul	Pin 4 a Pin 4	Azul
Azul y blanco	Pin 5 a Pin 5	Blanco y azul
Verde	Pin 6 a Pin 2	Naranja
Marrón y blanco	Pin 7 a Pin 7	Blanco y marrón
Marrón	Pin 8 a Pin 8	Marrón

2.2.5.- CONSTRUCCIÓN DEL CABLE

2.2.5.1.- MATERIALES

Cable ethernet de par trenzado (4pares).

Dos conectores RJ45 (ocho pines)

Ponchadora.



Fig. 2.3 Ponchar un cable

2.3 PUERTO SERIAL

El ordenador controla el puerto serie mediante un circuito integrado específico, llamado UART (Transmisor-Receptor-Asíncrono Universal). Normalmente se utilizan los siguientes modelos de este chip: 8250 (bastante antiguo, con fallos, solo llega a 9600 baudios), 16450 (versión corregida del 8250, llega hasta 115.200 baudios) y 16550A (con buffers de E/S). A partir de la gama Pentium, la circuiteria UART de la placa base son todas de alta velocidad, es decir UART 16550A. De hecho, la mayoría de los módems conectables a puerto serie necesitan dicho tipo de UART, incluso algunos juegos para jugar en red a través del puerto serie necesitan de este tipo de puerto serie. Por eso hay veces que un 486 no se comunica con la suficiente velocidad con un PC Pentium. Los portátiles suelen llevar otros chips: 82510 (con buffer especial, emula al 16450) o el 8251 (no es compatible).

Para controlar al puerto serie, la CPU emplea direcciones de puertos de E/S y líneas de interrupción (IRQ). En el AT-286 se eligieron las direcciones 3F8h (o 0x3f8) e IRQ 4 para el COM1, y 2F8h e IRQ 3 para el COM2. El estándar del PC llega hasta aquí, por lo que al añadir posteriormente otros puertos serie, se eligieron las direcciones 3E8 y 2E8 para COM3-COM4, pero las IRQ no están especificadas. Cada usuario debe elegir las de acuerdo a las que tenga libres o el uso que vaya a hacer de los puertos serie (por ejemplo, no importa compartir una misma IRQ en dos puertos siempre que no se usen conjuntamente, ya que en caso contrario puede haber problemas). Es por ello que últimamente, con el auge de las comunicaciones, los fabricantes de PCs incluyen un puerto especial PS/2 para el ratón, dejando así libre un puerto serie.

Mediante los puertos de E/S se pueden intercambiar datos, mientras que las IRQ producen una interrupción para indicar a la CPU que ha ocurrido un evento (por ejemplo, que ha llegado un dato, o que ha cambiado el estado de algunas señales de entrada). La CPU debe responder a estas interrupciones lo más rápido posible, para que de tiempo a recoger el dato antes de que el siguiente lo sobrescriba. Sin embargo, las UART 16550A incluyen unos buffers de tipo FIFO, dos de 16 bytes (para recepción y transmisión), donde se pueden guardar varios datos antes de que la CPU los recoja. Esto también disminuye el número de interrupciones por segundo generadas por el puerto serie.

El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a unas velocidades determinadas (normalmente, 9600 bits por segundo o más). Después de la transmisión

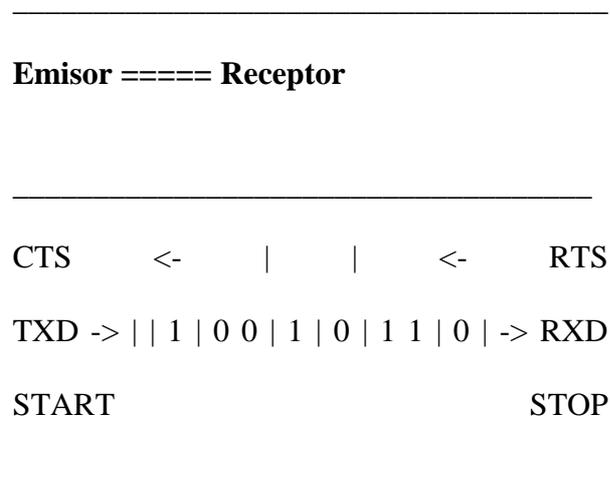
de los datos, le sigue un bit opcional de paridad (indica sí el número de bits transmitidos es par o impar, para detectar fallos), y después 1 o 2 bits de Stop. Normalmente, el protocolo utilizado es 8N1 (que significa, 8 bits de datos, sin paridad y con un bit de stop)

Una vez que ha comenzado la transmisión de un dato, los bits tienen que llegar uno detrás de otro a una velocidad constante y en determinados instantes de tiempo. Por eso se dice que el RS-232 es asíncrono por carácter y sincrónico por bit. Los pines que portan los datos son RXD y TXD. Las demás se encargan de otros trabajos: DTR indica que el ordenador está encendido, DSR que el aparato conectado a dicho puerto está encendido, RTS que el ordenador puede recibir datos (porque no está ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación, o presencia de datos.

Tanto el aparato a conectar como el ordenador (o el programa terminal) tienen que usar el mismo protocolo serie para comunicarse entre sí. Puesto que el estándar RS-232 no permite indicar en qué modo se está trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes. Como ya se ha visto, los parámetros que hay que configurar son: protocolo serie (8N1), velocidad del puerto serie, y protocolo de control de flujo. Este último puede ser por hardware o bien por software (XON/XOFF), el cual no es muy recomendable ya que no se pueden realizar transferencias binarias). La velocidad del puerto serie no tiene por qué ser la misma que la de transmisión de los datos, de hecho debe ser superior. Por ejemplo, para transmisiones de 1200 baudios es recomendable usar 9600, y para 9600 baudios se pueden usar 38400 (o 19200).

Este es el diagrama de transmisión de un dato con formato 8N1. El receptor indica al emisor que puede enviarle datos activando la salida RTS. El emisor envía un bit de START (nivel alto) antes de los datos, y un bit de STOP (nivel bajo) al final de estos.

Tabla 2.5 Transmisión de un dato.



2.3.1.- INTERFAZ RS-232.

El puerto serie RS-232C, presente en todos los ordenadores actuales, es la forma mas comúnmente usada para realizar transmisiones de datos entre ordenadores. El RS-232C es un estándar que constituye la tercera revisión de la antigua norma RS-232, propuesta por la EIA (Asociación de Industrias Electrónicas), realizándose posteriormente una versión internacional por el CCITT, conocida como V.24. Las diferencias entre ambas son mínimas, por lo que a veces se habla indistintamente de V.24 y de RS-232C (incluso sin el sufijo "C"), refiriéndose siempre al mismo estándar.

El RS-232C consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC). En cualquier caso, los PCS no suelen emplear más de 9 pines en el conector DB-25. Las señales con las que trabaja este puerto serie son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V. Dependiendo de la velocidad de transmisión empleada, es posible tener cables de hasta 15 metros.

Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Las más importantes son:

Tabla 2.6 Descripción y función de los pines de la interfaz RS-232

Pin	Función
TXD	(Transmitir Datos)
RXD	(Recibir Datos)
DTR	(Terminal de Datos Listo)
DSR	(Equipo de Datos Listo)
RTS	(Solicitud de Envío)
CTS	(Libre para Envío)
DCD	(Detección de Portadora)

Las señales

TXD, DTR y RTS

son de salida, mientras que RXD, DSR, CTS y DCD son de entrada. La masa de referencia para todas las señales es SG (Tierra de Señal). Finalmente, existen otras señales como RI (Indicador de Llamada).

Tabla 2.7 Conexiones del RS-232

Numero	De Pin	Señal	Descripción	E/S
En DB-25	En DB-9			
1	1	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC(*)	Transmit Clock	S
17	-	RxC(*)	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC(*)	Transmit/Receive Clock	S

(*) = Normalmente no conectados en el DB-25

- **2.3.2.- DIRECCIONES**

El puerto serie utiliza direcciones y una línea de señales, un IRQ para llamar la atención del procesador. Además el software de control debe conocer la dirección.

La mayoría de los puertos series utilizan direcciones standard predefinidas. Éstas están descritas normalmente en base hexadecimal. Cuando se instala un nuevo puerto, normalmente se mueve un jumper o switch para seleccionar un puerto (COM1, COM2, COM3, etc.), con lo que se asigna una dirección y una interrupción usada por la tarjeta del puerto.

Las direcciones e IRQ usadas por los puertos serie fueron definidas al diseñar el ordenador originalmente; sin embargo, las del COM3 y COM4 no se han definido oficialmente, aunque están aceptadas por convenios.

Se pueden añadir gran cantidad de puertos serie a un PC, ya que existe gran flexibilidad a la hora de definir direcciones no standard, siempre que se encuentren entre el rango 100 y 3FF hexadecimal y siempre que no entren en conflicto con otros dispositivos.

Los ordenadores IBM PS/2 usan la Microchannel Architecture, que define las direcciones e IRQs para los puertos desde COM1 a COM8:

Tabla 2.8 Direcciones IRQ

Dirección	Interrupción	(IRQ)	# DE PUERTO
COM1	3F8	4	0
COM2	2F8	3	1
COM3	3E8	4	2
COM4	2E8	3	3
COM5	4220	3	4
COM6	4228	3	5
COM7	5220	3	6
COM8	5228	3	7

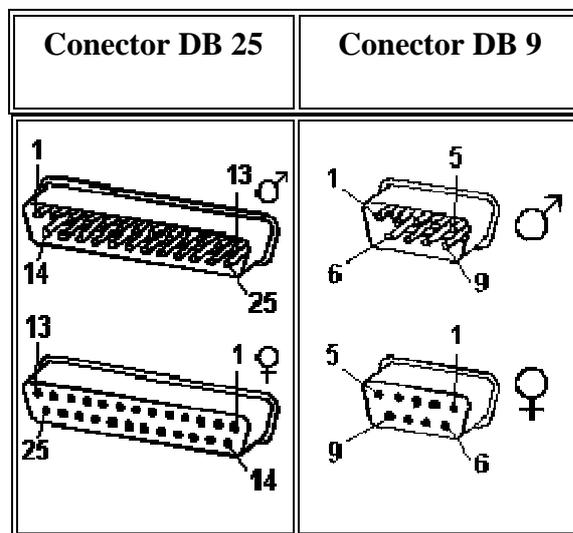


Fig. 2.4 Conectores DB25 y DB9

2.4.- LABVIEW

2.4.1.- INTRODUCCIÓN

Labview es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. Labview permite diseñar interfaces de usuario mediante una consola interactivo basado en software.

Labview es compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación, como por ejemplo Matlab. Tiene la ventaja de que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos (incluyendo adquisición de imágenes), este lenguaje de programación permite desarrollar de una forma más rápida cualquier aplicación, especialmente de instrumentación, en comparación con lenguajes de programación tradicionales basados en texto, sin embargo si usted desea una aplicación sencilla como un programa que sume dos números, definitivamente construirlo bajo Labview es mas demorado y tedioso, seria más sencillo mediante un programador de texto donde usted simplemente incluirá una línea. Pero para un programa más complejo usted puede diseñar un prototipo y modificarlo de una manera más rápida con Labview gracias a que es un lenguaje programación gráfico.

2.4.2 CARACTERÍSTICAS DE LABVIEW

Una de las principales características de Labview es su modularidad, es decir, la capacidad de utilizar bloques funcionales para la definición de la especificación.

Labview permite conectarse a otras aplicaciones mediante un intercambio de datos como Active X, librerías dinámicas, bases de datos, Excel y/o a protocolos de comunicación como DataSocket, TCP/IP, UDP, RS-232, entre otras.

Una característica de cada aplicación o función consiste en que se puede utilizar en cualquier parte de otro programa, dándole a Labview una estructura jerárquica.

Otra característica se encuentra en el flujo de datos, que muestra la ejecución secuencial del programa, es decir, una tarea no se inicia hasta no tener en todos sus variables de entrada información o que las tareas predecesoras hayan terminado de ejecutarse. Debido al lenguaje gráfico el compilador con que cuenta Labview es más versátil ya que sobre el mismo código de programación se puede ver fácilmente el flujo de datos, así como su contenido. Labview también puede ser un programa en tiempo real donde la aplicación trabaja sin la necesidad de otro sistema operativo, este programa denominado Labview RT viene con su propio Kernel que se encarga de la administración de las tareas. Mediante un constructor de aplicaciones también es posible generar un archivo que puede ejecutarse fuera de Labview.

2.4.3.- APLICACIONES DE LABVIEW

Labview tiene su mayor aplicación en sistema de medición, como monitoreo de procesos y aplicaciones de control, un ejemplo de esto pueden ser sistemas de

monitoreo en transportación, Laboratorios para clases en universidades, procesos de control industrial.

Labview es muy utilizado en procesamiento digital de señales (wavelets, FFT, Total Distorsion Harmonic TDH), procesamiento en tiempo real de aplicaciones biomédicas, manipulación de imágenes y audio, automatización, diseño de filtros digitales, generación de señales, entre otras, etc.

2.4.4.- PROGRAMACIÓN GRÁFICA CON LABVIEW

Cuando usted diseña programas con Labview está trabajando siempre bajo algo denominado VI, es decir, un instrumento virtual, se pueden crear VI a partir de especificaciones funcionales que usted diseñe. Este VI puede utilizarse en cualquier otra aplicación como una subfunción dentro de un programa general. Los VI's se caracterizan por: ser un cuadrado con su respectivo símbolo relacionado con su funcionalidad, tener una interfaz con el usuario, tener entradas con su color de identificación de dato, tener una o varias salidas y por su puesto ser reutilizables.

2.4.4.1- DISEÑE LA INTERFAZ DE USUARIO A PARTIR DE SU CÓDIGO.

En el ambiente de trabajo de Labview existen dos paneles, el panel frontal y el panel de programación (o diagrama de bloques); en el panel frontal se diseña la interfaz con el usuario y en el panel de programación se relacionan los elementos utilizados en la interfaz mediante operaciones

que determinan en sí como funciona el programa o el sistema, exactamente es la parte donde se realizan las especificaciones funcionales.

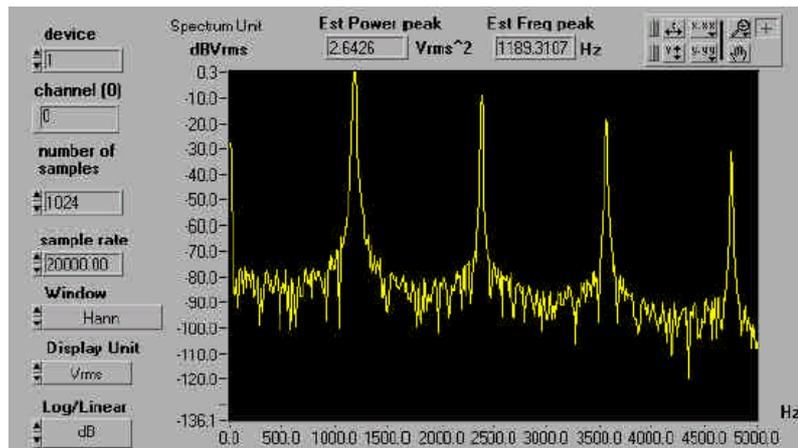


Fig. 2.5 Interfaz con el Usuario.

En el panel de programación usted puede diseñar de manera gráfica y como si fuera un diagrama de bloques el funcionamiento de su sistema. La programación gráfica se basa en la realización de operaciones mediante la asignación de íconos que representen los datos numéricos e íconos que representan los procedimientos que se deben realizar (VI's), con estos íconos y mediante una conexión simple como lo es una línea recta se enlazan para determinar una operación y/o una función.

Al diseñar el programa de forma gráfica, se hace visible una programación orientada al flujo de datos, donde se tiene una interpretación de los datos también de forma gráfica, por ejemplo un dato

booléano se caracteriza por ser una conexión verde, cada tipo de dato se identifica con un color diferente dentro de Labview; también es necesario tener en cuenta que cuando se realiza una conexión a un VI esta conexión se identifica por un tipo de dato específico, que debe coincidir con el tipo de dato de la entrada del VI (aunque esto no necesariamente es cierto ya que puede haber varios tipos de datos conectados de VI a VI, además de que un arreglo de datos ``cluster`` puede albergar varios tipos de variables) permitiendo una concordancia en el flujo de datos; no siempre el tipo de dato de la entrada del VI es el mismo que el de la salida, pero sin embargo para la mayoría de los casos si se cumple. El flujo de datos va de izquierda a derecha en el panel de programación y está determinado por las operaciones o funciones que procesan los datos. Es fácil observar en el panel de programación como se computan los datos en cada parte del programa cuando se realiza una ejecución del programa paso a paso. En Labview las variables se representan mediante un figura tanto en el panel frontal como en el panel de programación, de esta forma se puede observar su respuesta en la interfaz del usuario y en el flujo de datos del código del programa. Otros objetos como gráficas y accesos directos a paginas web cumplen estas mismas condiciones.

2.4.4.2.- MODULARIDAD

Cuando se ha diseñado la aplicación usted puede definirla como un VI, de esta forma se puede reutilizar en un nuevo programa, esto se hace mediante la selección del diagrama funcional y la opción crear VI, usted también puede diseñar el símbolo que represente su aplicación y definir

las entradas y las salidas, esto consiste en la asignación de cada variable de entrada a un sub-bloque dentro del bloque general que representa el símbolo.

2.4.5.- COMPILADOR GRÁFICO

En Labview se realiza la compilación bajo el principio básico de programación de forma gráfica. Debido a que se trabaja con flujo de datos es fácil ver el comportamiento de estos a través del programa, teniendo la posibilidad de ver el depurador ejecutarse paso a paso, se puede observar como cambian los datos en cualquier parte del programa, y como van pasando de una función a otra dentro del diagrama de bloques. El compilador optimiza internamente el código manteniendo una buena velocidad en la ejecución del programa.

2.4.5.1 PALETA DE CONTROL

La programación G (gráfica) de Labview consta de un panel frontal y un panel de código como se mencionó antes. En el panel frontal es donde se diseña la interfase de usuario y se ubican los controles e indicadores. En el panel de código se encuentran las funciones. Cada control que se utiliza en la interfaz tiene una representación en el panel de código, igualmente los indicadores necesarios para entregar la información procesada al usuario tienen un icono que los identifica en el panel de código o de programación.



Figura 2.6 Paleta de controles

Los controles pueden ser booleanos, numéricos, strings, un arreglo matricial de estos o una combinación de los anteriores; y los indicadores pueden ser como para el caso de controles pero pudiéndolos visualizar como tablas, gráficos en 2D o 3D, browser, entre otros.

2.4.5.2.- PALETA DE FUNCIONES

Las funciones pueden ser VIs prediseñados y que pueden ser reutilizados en cualquier aplicación, estos bloques funcionales constan de entradas y salidas, igual que en un lenguaje de programación estándar las funciones procesan las entradas y entregan una o varias salidas, estos VI pueden también estar conformados de otros sub-VIs y así sucesivamente, de esta forma se pueden representar como un árbol genealógico donde un VI se relaciona o depende de varios Sub-VIs.

Labview tiene VIs de adquisición de datos e imágenes, de comunicaciones, de procesamiento digital de señales, de funciones

matemáticas simples, hasta funciones que utilizan otros programas como Matlab o HiQ para resolver problemas, otras más complejas como "nodos de fórmula" que se utilizan para la resolución de ecuaciones editando directamente estas como en lenguajes de programación tradicionales y definiendo las entradas y las salidas.



Figura.2.7. Paleta de Funciones

Labview también se puede utilizar para graficar en tres dimensiones, en coordenadas polares y cartesianas, tiene disponibles herramientas para análisis de circuitos RF como la Carta de Smith, tiene aplicaciones en manejo de audio y se puede comunicar con la tarjeta de sonido del computador para trabajar conjuntamente.

Entre sus muchas funciones especiales se encuentran las de procesamiento de imágenes, como capturar una imagen a través de una

tarjeta de adquisición como la PCI-1408 (monocromática), analizarla y entregar respuestas que difícilmente otros sistemas realizarían.

2.5.-DESCRIPCIÓN DE LAS FUNCIONES Y CONTROLES UTILIZADOS

2.5.1-FUNCIONES

2.5.1.1-STRUCTURE

2.1.5.1.1.-Sequence Structure (Estructura de Secuencia)

Consiste en uno o más subdiagramas que se ejecutan secuencialmente.

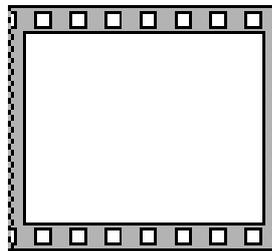


Figura 2.8.- Sequence Structure

2.5.1.1.2.-Case Structure (Estructura Case)

Tiene uno o más subdiagramas que se ejecuta dependiendo del valor de la entrada pasada a la estructura.

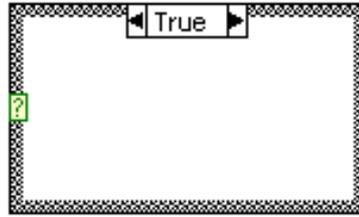


Figura 2.9.- Case Structure

2.5.1.1.3.-While Loop (lazo While)

La secuencia While Loop repite el subdiagrama dentro de él hasta que el terminal condicional (un término de la entrada) recibe un valor de Boolean FALSO o VERDADERO.



Figura 2.10.- While Loop

2.5.1.2.- NUMERICO

2.5.1.2.1.- Numérica Constant (Constante numérica)

Se usa para pasar un valor numérico a cualquier función que se encuentre en el diagrama de bloque.

2.5.1.3.– BOOLEANO

2.5.1.3.1.- True Constant (Constante Verdadera)

Se usa esta constante para proporcionar un valor constante VERDADERO a una función del diagrama de bloque.

2.5.1.3.2.- False Constant (La Constante falsa)

Se usa esta constante para proporcionar un valor FALSO a una función del diagrama de bloque.

2.5.1.4.– STRING CADENA

2.5.1.4.1.- Search and Replace String (Buscar y Reemplazar cadenas)

Reemplaza uno o todos los casos de un subcadena con otro subcadena.

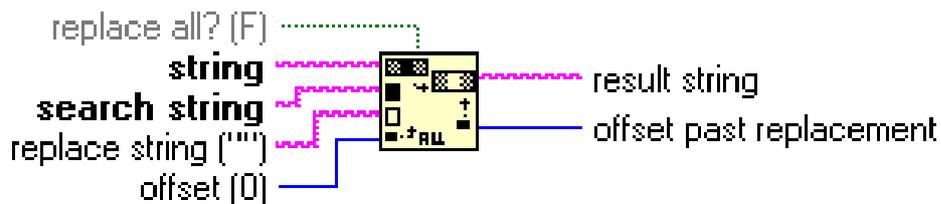


Figura 2.11.- String Cadena

String contiene la cadena que usted quiere investigar.

Saerch string es la cadena que usted quiere buscar y reemplazar.

Replace string reemplaza la cadena en la cadena de la búsqueda.

Result string contiene la cadena con uno o todas las ocurrencias de la cadena de la búsqueda reemplazada con replace string.

2.5.1.4.2.- String Subset (El Subconjunto de cadena)

Retorna la subcadena de la cadena original empezando al desplazamiento y conteniendo número de longitud de caracteres.

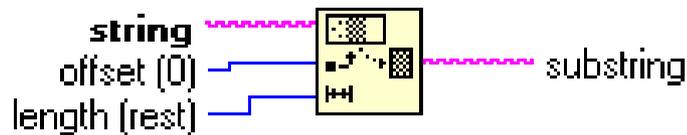


Figura 2.12.-String Subset

Length debe ser scalar. Muestra hasta que carácter debe desplazarse la cadena.

Offset debe ser scalar. Aquí es donde empieza el desplazamiento.

String es la cadena de la entrada las búsquedas de la función. La longitud de la cadena menos el desplazamiento es la longitud de substring.

Substring es el resultado del desplazamiento y conteniendo que se obtiene del string mediante el offset y el length.

2.5.1.4.3.- Concatenate Strings (Concatenación de Cadena)

Une varias cadenas en una sola.

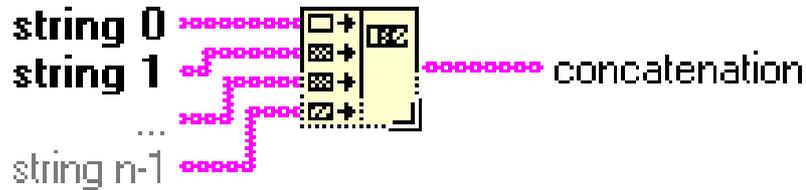


Figura 2.13.- Concatenate String

String0 y **String1** son los términos de la entrada predefinidos. Usted puede agregar tantos términos de la entrada como usted necesita seleccionar.

Concatenation contiene la entrada encadenada dentro de la cadena en el orden que usted los encadena.

2.5.1.4.4.- Decimal String To Number (Cadena decimal a Números)

Convierte los caracteres a números en cadenas, empezando donde muestra el desplazamiento.

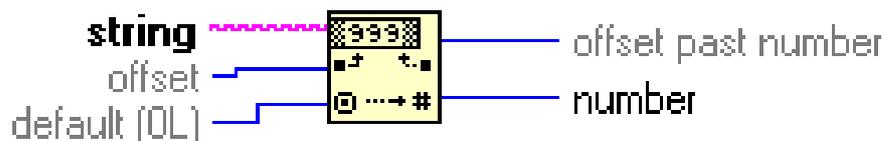


Figura 2.14.- Decimal String To Number

String puede ser una cadena, un racimo, una serie de cadenas, o una serie de racimos.

Offset debe ser scalar. Aquí es donde empieza el desplazamiento.

Number puede ser un número, un racimo, o una serie de números, mientras dependiendo de la estructura de la cadena y el desplazamiento

2.5.1.4.5.- String constant (Constante de cadena)

Se usa esta constante para proporcionar un valor ASCII constante al diagrama de bloque.

2.5.1.4.6.- Empty String (Cadena Vacía)

Consiste en una cadena constante que está vacío. La longitud es el cero.

2.5.1.4.7.- Tab (La etiqueta horizontal)

Consiste en una cadena constante que contiene el valor ASCII HT (Tab).

2.5.1.5.- ARRAYS

2.5.1.5.1.-Build Array (Constructor de Array)

Sirve para construir un array bidimensional de cualesquier dato numérico o string.

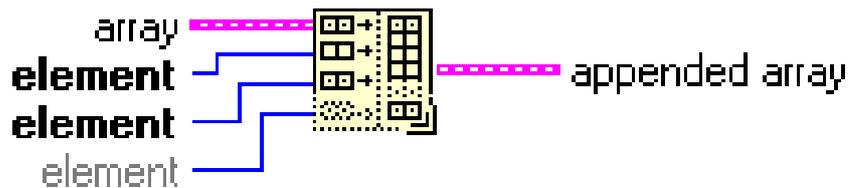


Figura 2.15.- Build Array

Array o Input element es la serie que se ingresa para su modificación.

Appended array u output array es la serie resultante.

2.5.1.6.- COMPARACION

2.5.1.6.1.-Equal? ¿Igual?

Regresa VERDADERO si x es igual a y . Por otra parte, esta función vuelve FALSO. Esta función acepta los números complejos.

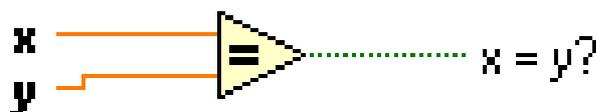


Figura 2.16.- Equal?

X e y deben ser del mismo tipo. Estos son los elementos que se van a comparar.

2.5.1.6.2.-Greater Or Equal? (Mayor O Igual)

Ingresa VERDADERO si x es mayor que o igual a y. Por otra parte, esta función vuelve FALSO.

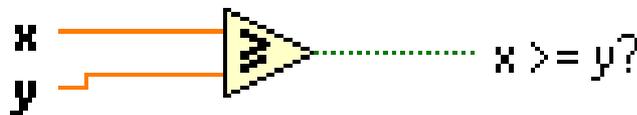


Figura 2.17.- Greater or Equal?

X e y deben ser del mismo tipo. Estos son los elementos que se van a comparar.

2.5.1.7.- TIEMPO Y DIALOGO

2.5.1.7.1.-Wait

Espera el número especificado en milisegundos. Por consiguiente, no completa la ejecución hasta que el tiempo especificado haya pasado.

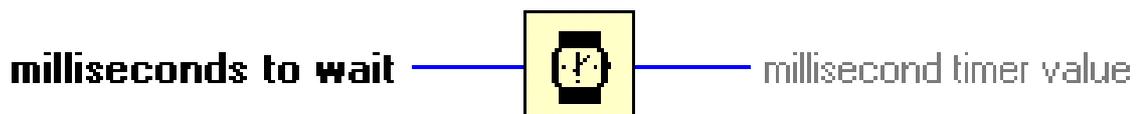


Figura 2.18.- Wait

Milliseconds es la entrada especificada para ver cuantos milisegundos debe esperar.

2.5.1.7.2.-Get Date/Time String (consiga la cadena dato/tiempo)

Time Universal, da una fecha y cadena de tiempo en la zona de tiempo configurada para su computadora.

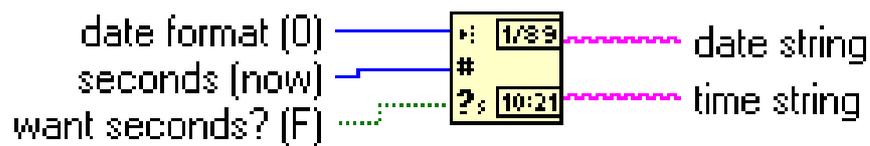


Figura 2.19.- Get Date/Time String

Want seconds? Los mandos del despliegue de segundos en la cadena de tiempo.

Date string es la cadena de la fecha

Time string la cadena del tiempo

2.5.1.8.- FILE I/O

2.5.1.8.1.-Open/Create/Replace File (Abrir/crear/Reemplazar archivo)

Abre un archivo existente, crea un nuevo archivo, o reemplaza un archivo existente, mientras usa una caja de diálogo de archivo interactivamente.

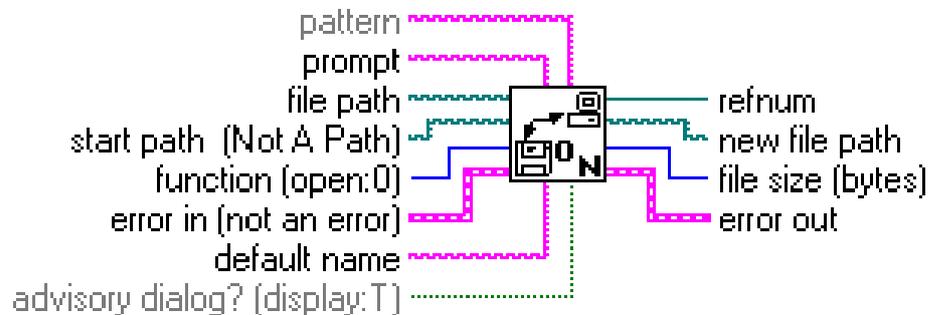


Figura 2.20.- Open/Create/Replace File

File path es el nombre del camino del archivo.

Function es el funcionamiento para realizar.

0.- Abren un archivo existente. Error 7 ocurre si el archivo no puede encontrarse (el valor predefinido).

1.- Abre un archivo existente o crea un nuevo archivo si uno no existe.

2.- Crean un nuevo archivo o reemplazan un archivo si existe y usted da el permiso.

3.- Crean un nuevo archivo. Error 8 ocurre si el archivo ya existe.

4.- Abren un archivo existente para sólo lectura. Error 7 ocurre si el archivo no puede encontrarse.

Refnum es el número de la referencia del archivo abierto. El valor no es un Refnum si el archivo no puede abrirse.

2.5.1.8.2.-Write File (Escriba el Archivo)

Escribe los datos al archivo especificado por el refnum.

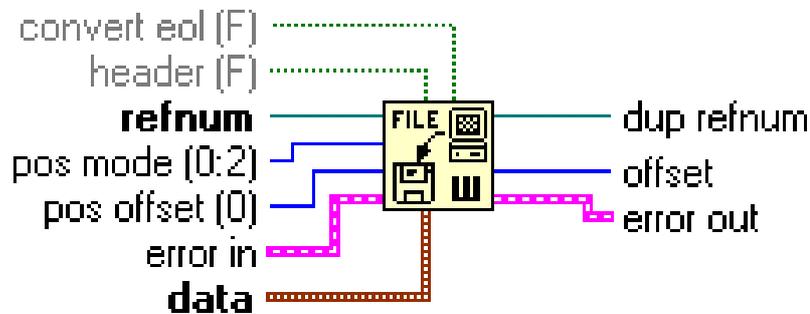


Figura 2.21.- Write File

Refnum es el refnum del archivo asociado con el archivo que usted quiere cerca.

Pos mode junto con **pos offset**, especifica donde empieza a escribir funcionamiento.

0.- Pos mode indica que él empiece a escribir el funcionamiento al principio del archivo más el desplazamiento del pos.

1.- Pos mode indica que él empiece a escribir el funcionamiento al final del archivo más el desplazamiento del pos.

2.- Pos mode indica que él empieza a escribir el funcionamiento a la situación actual de la marca del archivo más el desplazamiento del pos.

Data contains contiene los datos leídos del archivo. Puede ser cualquier tipo de los datos.

Dup refnum es un flujo a través del parámetro con el mismo valor como el refnum.

2.5.1.8.3.-Close File (Cerrar Archivo)

Cierra el archivo especificado por refnum y retorna el camino asociado al archivo con el refnum.

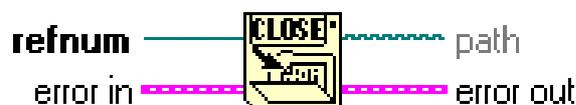


Figura 2.22.- Close File

Refnum es el refnum del archivo asociado con el archivo a que usted quiere cerca.

2.5.1.9.- INSTRUMENTOS I/O

2.5.1.9.1.-Serial Port INIT (EL PUERTO DE SERIE INIT)

Inicializa el puerto de serie seleccionado a las escenas especificadas.

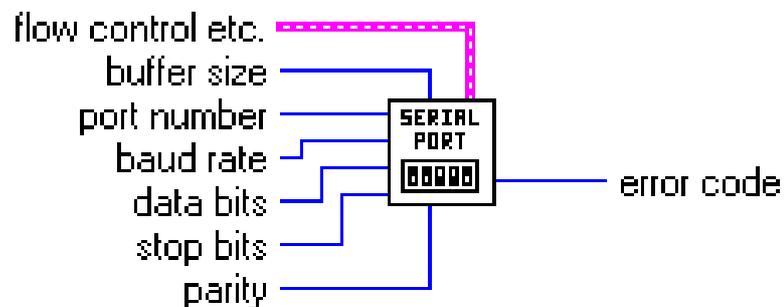


Figura 2.23.- Serial Port INIT

Buffer Size indica el tamaño de la entrada y pulidores del rendimiento que los asignan para la comunicación a través del puerto especificado. El buffer size está en los bytes.

El valor predefinido: 0

Port Number Los parámetros para los números del puerto de serie dependen de la plataforma que usted usa: Windows, Macintosh, o UNIX.

El valor predefinido: 0

Baud rate es la velocidad de transmisión.

El valor predefinido: 9600

Data bits son el número de datos entrantes. El valor de los datos está entre cinco y ocho.

El valor predefinido: 8

Stop bits Especifica el número de bits de parada.

2.5.1.9.2.-Serial Port Write (Escritura de Dato)

Escribe los datos en la cadena escrita al puerto serie indicando el número del puerto.

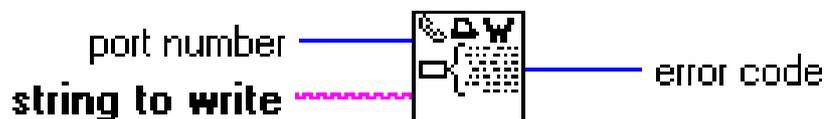


Figura 2.24.- Serial Port Write

Port number Los parámetros para los números del puerto de serie dependen de la plataforma que usted usa: Windows, Macintosh, o UNIX.

El valor predefinido: 0

String to write son los datos a ser escritos al puerto serie.

2.5.1.9.3.-Bytes At Serial Port (Contador de Bytes).

Indica el número bytes leídos por el puerto.

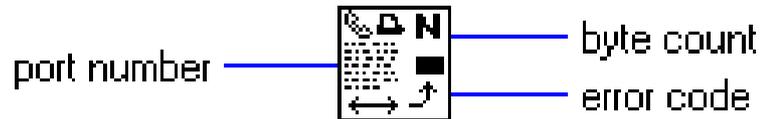


Figura 2.25.- Bytes at Serial Port

Port number Los parámetros para los números del puerto de serie dependen de la plataforma que usted usa: Windows, Macintosh, o UNIX.

El valor predefinido: 0

Byte count es actualmente el número de bytes que ingresan al puerto de serie.

2.5.1.9.4.-Serial Port Read (Lectura de Datos)

Lee el número de caracteres especificado por el contador de bytes requeridos para el puerto de serie indicado en el número del puerto.



Figura 2.26.- Serial Port Read

Port number Los parámetros para los números del puerto de serie dependen de la plataforma que usted usa: Windows, Macintosh, o UNIX.

El valor predefinido: 0

Requested byte count especifica el número de caracteres para ser leído.

String read los ingresos que los bytes leyeron en la cadena leída.

2.5.1.9.5.-Close Serial Driver (Cierra del Puerto)

Cierra el puerto especificado.

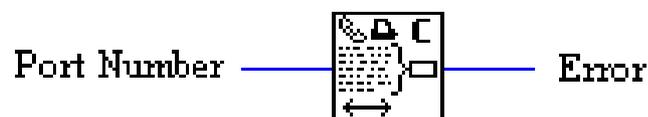


Figura 2.27.- Close Serial Driver

Port number Los parámetros para los números del puerto de serie dependen de la plataforma que usted usa: Windows, Macintosh, o UNIX.

El valor predefinido: 0

2.5.1.9.6.-Property Node (Propiedad del nodo)

Proporciona en forma gráfica las propiedades de un control e indicador.

2.5.2.-CONTROLES

2.5.2.1.-NUMERICOS

2.5.2.1.2.-Digital Controls and Indicators (Controles e Indicadores Digitales)

Los controles digitales e indicadores son la manera más simple para ingresar y desplegar los datos numéricos.



Figura 2.28.- Control e Indicador

2.5.2.2.-BOOLEAN

2.5.2.2.1.- Boolean Controls and Indicators (Controles e Indicadores Booleanos)

Nos sirve para controlar estados en verdadero y falso.

2.5.2.3.-GRAFICOS

2.5.2.3.1.- Graph Controls (Los controles de gráfico)

Use controladores e indicadores de gráfico, localizados en la paleta de Controls»Graph, para trazar los datos numéricos en mapa o formulario del gráfico.

2.5.3.-PALETA DE HERRAMIENTAS

2.5.3.1.- Operating (Operación)

Los cambios que opera el valor de un mando o selecciona el texto dentro de un mando.

2.5.3.2.- Positioning (Posicionamiento)

Posiciona y selecciona los objetos.

2.5.3.3.- Labeling (Etiquetando)

Revisa el texto y crea las etiquetas libres.

2.5.3.4.- Wiring (Alambrando)

Refiérase a los alambres usados para unir Bloque Diagrama Objetos.

2.5.3.5.- Object Shortcut Menu (Menu de Atajo)

Abre el menú del atajo de un objeto.

2.5.3.6.- Scroll

Mueve los Bloques, Diagramas, Objetos a la posición deseada.



Figura 2.29.- Paleta de Herramientas

CAPITULO 3

DESARROLLO DE LA INTERFAZ

3.1 ANALISIS DE ALTERNATIVAS

En este punto vamos a exponer algunos de los lenguajes de programación que podemos utilizar para el desarrollo de la interfaz grafica con el multímetro y trataremos sus características, ventajas y desventajas de cada uno.

3.1.1.- VISUAL BASIC.

3.1.1.1.-INTRODUCCIÓN.

Visual Basic es uno de los tantos lenguajes de programación que podemos encontrar hoy en día. Dicho lenguaje nace del BASIC (Beginner's All-purpose Symbolic Instruction Code) que fue creado en su versión original en el Dartmouth College, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación. Luego de sufrir varias modificaciones, en el año 1978 se estableció el BASIC estándar. La sencillez del lenguaje ganó el desprecio de los programadores avanzados por considerarlo "un lenguaje para principiantes".

Primero fue GW-BASIC, luego se transformó en QuickBASIC y actualmente se lo conoce como Visual Basic y la versión más reciente es la 6 que se incluye en el paquete Visual Studio 6 de Microsoft. Esta versión combina la sencillez del BASIC con un poderoso lenguaje de programación Visual que juntos permiten desarrollar robustos programas

de 32 bits para Windows. Esta fusión de sencillez y la estética permitió ampliar mucho más el monopolio de Microsoft, ya que el lenguaje sólo es compatible con Windows, un sistema operativo de la misma empresa.

Visual Basic ya no es más "un lenguaje para principiantes" sino que es una perfecta alternativa para los programadores de cualquier nivel que deseen desarrollar aplicaciones compatibles con Windows.

3.1.1.2.- CARACTERÍSTICAS DE VISUAL BASIC.

Diseñador de entorno de datos: Es posible generar, de manera automática, conectividad entre controles y datos mediante la acción de arrastrar y colocar sobre formularios o informes.

Los Objetos Activos son una nueva tecnología de acceso a datos mediante la acción de arrastrar y colocar sobre formularios o informes.

Asistente para formularios: Sirve para generar de manera automática formularios que administran registros de tablas o consultas pertenecientes a una base de datos, hoja de calculo u objeto (ADO-ACTIVE DATA OBJECT)

Asistente para barras de herramientas donde el usuario selecciona los botones que desea visualizar durante la ejecución.

En las aplicaciones HTML: Se combinan instrucciones de Visual Basic con código HTML para controlar los eventos que se realizan con frecuencia en una página web.

La Ventana de Vista de datos proporciona acceso a la estructura de una base de datos.

-

- **3.1.1.3.- VENTAJAS E INCONVENIENTES DE VISUAL BASIC**

La ventaja principal de este lenguaje de programación es su sencillez para programar aplicaciones de cierta complejidad para Windows, y sus desventajas son la necesidad de archivos adicionales además del ejecutable y cierta lentitud en comparación con otros lenguajes. Hoy en día este último factor es cada vez menos determinante debido a la gran potencia de los ordenadores de última generación.

Aunque Visual Basic 6 se anunció como un lenguaje orientado a objetos la realidad es que solamente soporta la encapsulación (y a medias) y no soporta la herencia, el polimorfismo, la creación de interfaces ni la sobrecarga.

-
- **3.1.2.- VISUAL C++.**

- **3.1.2.1.- ¿QUÉ ES?**

Es un compilador de C/C++ que permite la creación de proyectos estructurando de forma sencilla, programas que incluyen muchos ficheros.

Además, incluye las MFC(Microsoft Foundation Classes), equivalentes a las OWL (Object Window Library) de Borland. Estas clases facilitan la programación Windows, sin tener que utilizar directamente el API de Windows, ya que agrupan las librerías de Windows en clases C++.

MFC es una jerarquía de clases C++ para programar en Windows, entre las cuales hay algunas de alto nivel que proporcionan funcionalidad general (por ejemplo la clase CWnd) y otras que implementan funciones más específicas.

Además de esta jerarquía de clases, MFC nos da un modelo de desarrollo de aplicaciones llamado modelo documento / vista, que nos permite diseñar aplicaciones de forma que los datos de la aplicación vayan separados de los elementos que componen la interfaz de usuario. Esto permite modificar de forma independiente las dos partes del programa.

- **3.1.2.2.- LAS CARACTERÍSTICAS PRINCIPALES SON:**

- Arquitectura documento / vista.
- Interfaz de documentos múltiples (MDI).
- Soporte para bases de datos.
- Programación para Internet.
- Controles comunes para Windows 96/98 y NT.
- Soporte multithread.

3.1.3.- LABVIEW

3.1.3.1.- DEFINICIÓN

Labview es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. Labview permite diseñar interfaces de usuario mediante una consola interactivo basado en software.

3.1.3.2.- CARACTERÍSTICAS DE LABVIEW

Una de las principales características de Labview es su modularidad, es decir, la capacidad de utilizar bloques funcionales para la definición de la especificación. Labview permite conectarse a otras aplicaciones mediante un intercambio de datos como Active X, librerías dinámicas, bases de datos, Excel y/o a protocolos de comunicación como DataSocket, TCP/IP, UDP, RS-232, entre otras.

Una característica de cada aplicación o función consiste en que se puede utilizar en cualquier parte de otro programa, dándole a Labview una estructura jerárquica.

Otra característica se encuentra en el flujo de datos, que muestra la ejecución secuencial del programa, es decir, una tarea no se inicia hasta no tener en todos sus variables de entrada información o que las tareas predecesoras hayan terminado de ejecutarse.

Debido al lenguaje gráfico el compilador con que cuenta Labview es más versátil ya que sobre el mismo código de programación se puede ver fácilmente el flujo de datos, así como su contenido. Labview también puede ser un programa en tiempo real donde la aplicación trabaja sin la necesidad de otro sistema operativo, este programa denominado Labview RT viene con su propio Kernel que se encarga de la administración de las tareas. Mediante un constructor de aplicaciones también es posible generar un archivo que puede ejecutarse fuera de Labview.

3.1.3.3.- APLICACIONES DE LABVIEW

Labview tiene su mayor aplicación en sistema de medición, como monitoreo de procesos y aplicaciones de control, un ejemplo de esto pueden ser sistemas de monitoreo en transportación, Laboratorios para clases en universidades, procesos de control industrial.

Labview es muy utilizado en procesamiento digital de señales (wavelets, FFT, Total Distorsión Harmónica TDH), procesamiento en tiempo real de aplicaciones biomédicas, manipulación de imágenes y audio, automatización, diseño de filtros digitales, generación de señales, entre otras, etc.

Funcionalidad, tener una interfaz con el usuario, tener entradas con su color de identificación de dato, tener una o varias salidas y por su puesto ser reutilizables.

-

- **3.1.3.4.- VENTAJAS**

LabVIEW permite construir sus propias soluciones a científicos e ingenieros en sistemas y electrónica. LabVIEW provee de flexibilidad y de un poderoso lenguaje de programación que permite generar aplicaciones sin dificultades ni complicaciones.

LabVIEW viene dando satisfacciones a miles de usuarios como un exitoso programa de instrumentación y sistemas de adquisición de datos. Al usar LabVIEW como prototipo de diseño, prueba e implementación de sus sistemas de instrumentación, se pueden desarrollar sistemas en poco tiempo e incrementar la productividad en un factor de 4 a 10.

LabVIEW y National Instruments provee de varios medios para dar soporte a su gran cantidad de usuarios a través de sus centros de servicio

al cliente y el Internet que aseguran al usuario la implementación exitosa de sus aplicaciones.

3.2 ANALISIS Y SELECCIÓN DE ALTERNATIVAS

3.2.1.- SOFTWARE

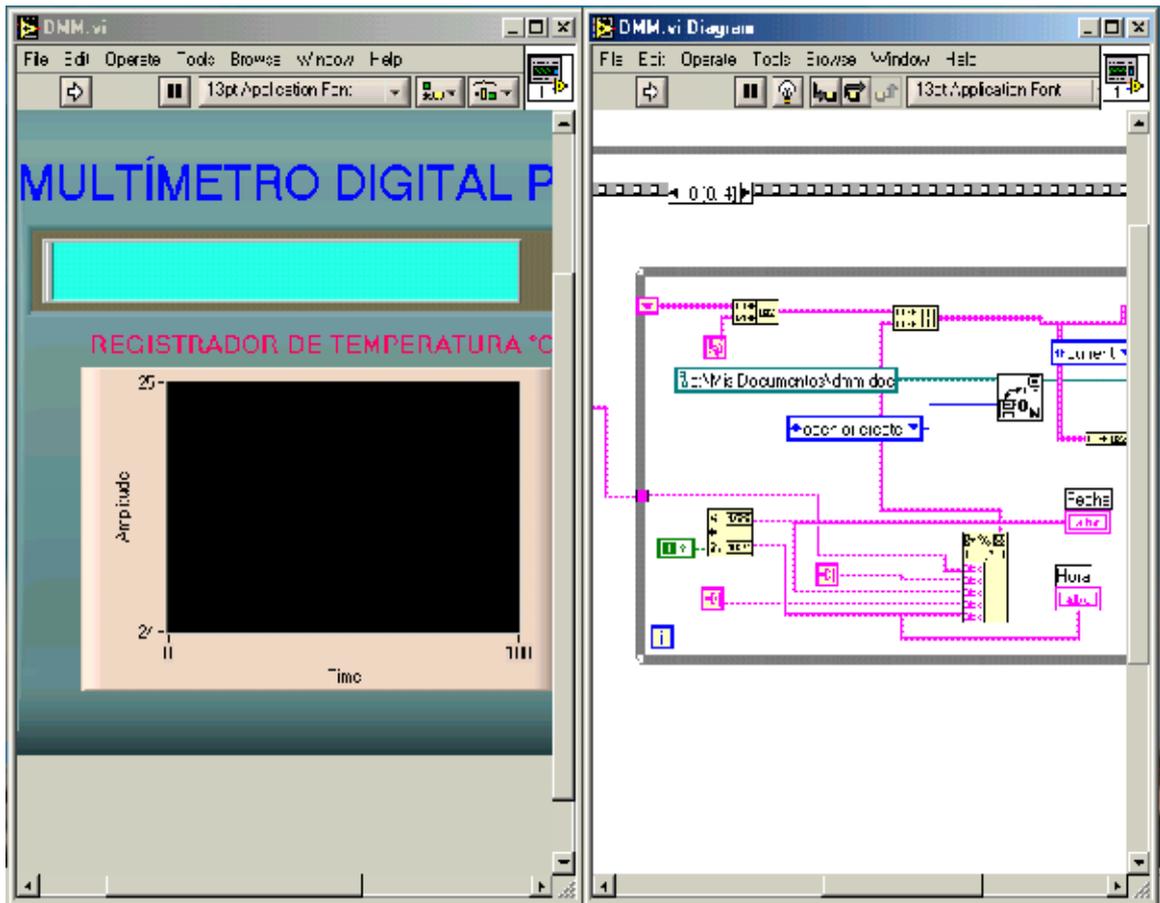


Fig. 3.1 Software

Después de haber analizado y estudiado cada uno de los lenguajes de programación propuestos anteriormente con detenida atención y por las características y ventajas que proporciona se ha decidido realizar el desarrollo de la interfaz gráfica en LABVIEW 6.0 por las siguientes razones.

- Labview es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control.
- Labview permite diseñar interfaces de usuario mediante una consola interactivo basado en software.

- Labview es a la vez compatible con herramientas de desarrollo similares
- Tiene la ventaja de que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos.
- Labview también puede ser un programa en tiempo real donde la aplicación trabaja sin la necesidad de otro sistema operativo.
- Labview tiene su mayor aplicación en sistema de medición, como monitoreo de procesos y aplicaciones de control.
- Labview es muy utilizado en procesamiento digital de señales.
- Mejor presentación.

3.2.2.- MULTÍMETRO



Fig. 3.2 Protek 506

Debido a las necesidades del instituto y la recomendación de estos se ha decidido adquirir un multímetro digital con interfaz serial para la computadora llamado Protek 506 que presenta las siguientes características:

- Puerto RS232C para la interfaz con la computadora personal
- Mediciones verdaderas de RMS
- Despliegue de la luz interna en el multímetro
- Despliegue dual, por °C/°F, Hz/ACV, etc.
- 10 localizaciones de memorias
- Funciones de tiempo, para el reloj de la alarma o el pare
- Medición de decibeles
- Medición de capacitancias e inductancias
- Medición de la temperatura en (°C/°F) usando “K” tipo thermocouple
- Pieza de temperatura (°C/°F) sin cualquier sonda de temperatura
- Señal de salida para señales lógicas para el chequeo de inyección o audio del circuito
- Chequeo de la función del nivel lógico
- Medición de la frecuencia hasta 10Mhz
- Protección contra cargas excesivas para todas las funciones
- Modo MAX/MIN/AVG
- Modo relativo
- Auto encendido, apagado y mantener en modo
- Datos estáticos y modo de funcionamiento
- Indicación de batería baja

3.3.- REQUERIMIENTOS TÉCNICOS

3.3.1.- EL COMPROBADOR DE CABLES

Este comprobador ha sido diseñado para poder verificar líneas de transmisión con conectores RJ11 y RJ45 los cuales son los más usados en las telecomunicaciones.

En el instituto se utiliza este tipo de líneas de transmisión en sus laboratorios a diario, en las diferentes conexiones de área local, por lo que un daño de estas líneas causa retrasos en la emisión de información y con ellos molestias a los usuarios de la red.

Debido a la cantidad de líneas utilizadas es molesto y demoroso saber que línea es la defectuosa y chequearla si hay continuidad en cada pin tornándose así una molestia y pérdida de tiempo en solucionar estos inconvenientes y poder mejorar el funcionamiento de la red. Es por ello que hemos decidido realizar un comprobador de líneas de transmisión para conectores RJ45 y RJ11.

3.3.1.1.- FUNCIONAMIENTO

Este comprobador funciona primeramente seleccionando el tipo de conector, el switch es de tres estados por lo cual hacia arriba selecciona el conector RJ11, en el centro el circuito esta apagado, y hacia abajo selecciona el conector RJ45, luego conectamos la línea de transmisión.



Figura. 3.3 Comprobador

Antes de elegir el tipo de conector el circuito se encuentra apagado, una vez seleccionado el tipo de conector el circuito se encuentra en estado abierto. Una vez colocada la línea de transmisión con el conector respectivo el circuito se cierra en el cual la línea de transmisión es la que completa el circuito que se encontraba abierto y entra en funcionamiento, en el caso del RJ11 se encienden los 4 LEDs primeros indicando que todos los pines están bien, o si algún LED no se prende indica que algún pin está abierto y de igual manera cuando seleccionamos al RJ45.



Figura. 3.4 Comprobador seleccionado RJ11

Para realizar más compacto y práctico el comprobador de cables hemos decidido utilizar 8 LEDs en total, los 4 primeros para el conector RJ11 y

los 4 LEDs restantes incluyendo los 4 LEDs del RJ11 utilizamos para el RJ45.



Figura. 3.5 Comprobador seleccionado RJ45

3.3.1.2.- DESARROLLO

3.3.1.2.1.- Materiales

Para la realización de este circuito se utilizó:

- 8 resistencias de 220Ω
- 8 LEDs
- 1 placa de baquelita
- 1 ácidos sulfúricos
- 1 marcador indeleble
- 1 interruptor de tres posiciones
- 2 conectores RJ11
- 2 conectores RJ45
- 1m de estaño

- 1 pomada para soldar
- 1 caja de aluminio
- 1 batería de 9V
- 1 broche para batería

3.3.1.2.2.- Diagrama del circuito

Según los problemas mencionados anteriormente con las líneas de transmisión, hemos acordado hacer el siguiente circuito que nos permita realizar las comprobaciones de las líneas de transmisión de una manera más rápida y eficaz.

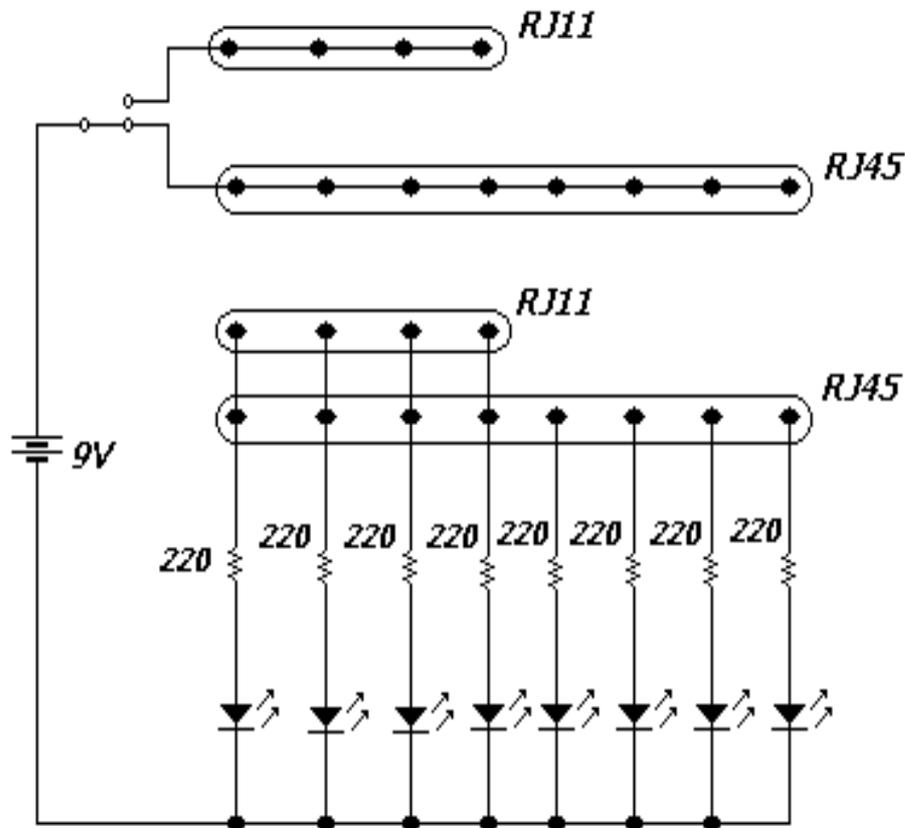


Fig. 3.6 Diagrama del circuito

Según el diagrama del circuito se procede a realizar su comprobación de funcionamiento armando el circuito en el protobar, obteniendo un funcionamiento adecuado del circuito armado, y por ende los resultados deseados.

3.3.1.2.3.- CIRCUITO IMPRESO

Una vez probado el circuito en el protobar, se procede al desarrollo del diagrama en la baquelita. Obteniendo como resultado el circuito que muestra el anexo A.

3.3.1.2.4- MONTAJE DE COMPONENTES

Una vez finalizado el circuito impreso, realizamos las perforaciones respectivas y la inserción de los componentes correspondientes como se muestra en el en el anexo B.

3.3.1.2.5.- COLOCACIÓN EN LA CARCASA

Una vez terminado el circuito impreso con la colocación de sus componentes, colocamos en una carcasa de aluminio para su presentación final como se muestra en el en el anexo C.

3.3.1.3.- USO DEL COMPROBADOR

Este comprobador es de fácil uso. Como muestra en el anexo C, tiene un interruptor con el cual se selecciona el tipo de conector de la línea de

transmisión, ya sea el RJ45 o RJ11. Con este mismo interruptor se apaga el dispositivo .

3.4.- DESARROLLO DE LA INTERFAZ GRAFICA

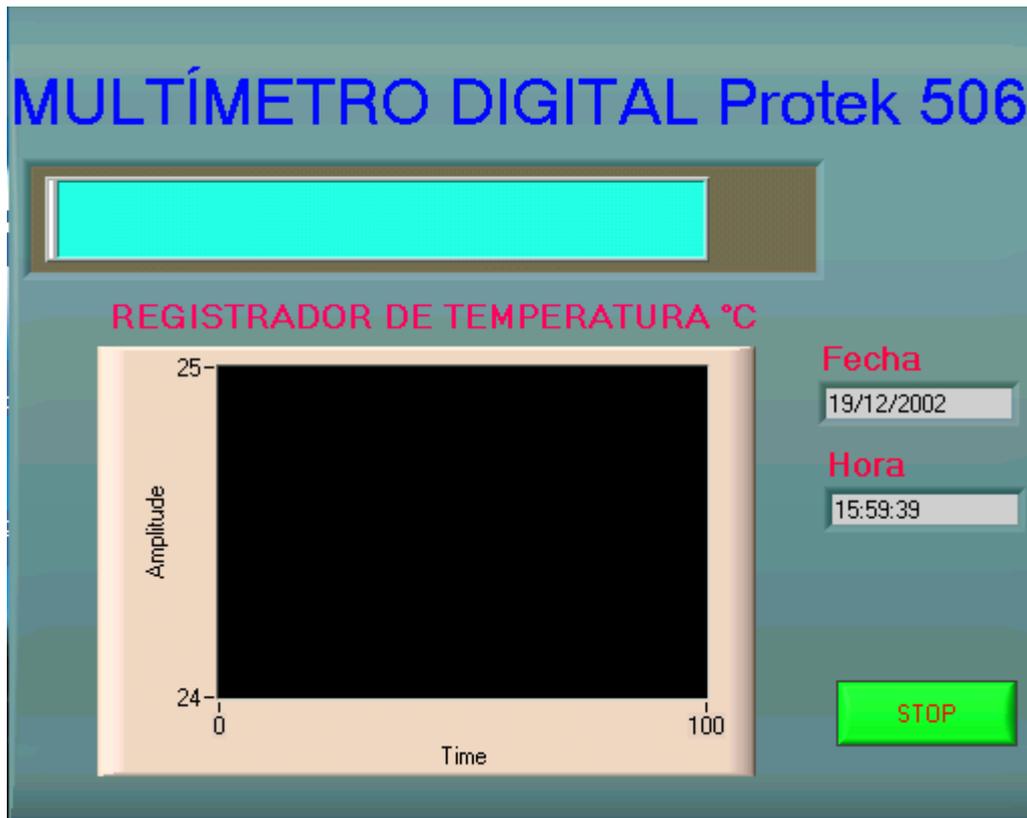


Fig. 3.7 Interfaz gráfica

Una vez seleccionado Labview 6.0 para el desarrollo la interfaz gráfica se procedió a realizar la misma.

Para el desarrollo se realizó un estudio del puerto serial y sus componentes de Labview para su mejor desarrollo y funcionalidad del programa.

Se utilizó la estructura Sequence, la misma que se utiliza de la siguiente manera:

- En la Sequence 0 inicializamos el puerto serial.

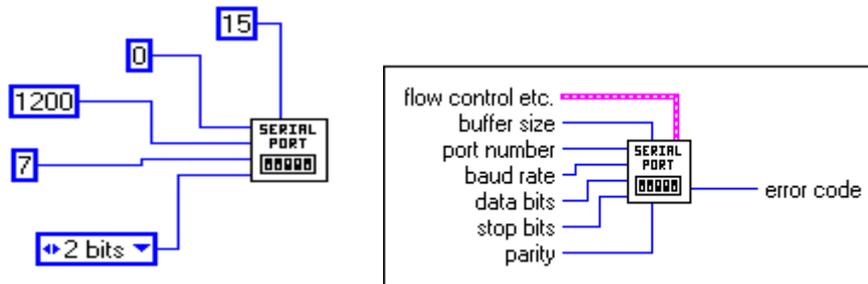


Fig. 3.8 Inicializar el puerto serial.

En el buffer size colocamos 15 por que es el número máximo de bits que va a circular por el puerto (ver anexo D).

En el port number colocamos el numero de puerto que vamos a utilizar en este caso el COM1 que su código es 0 (Véase Tabla 2.8, Pág. 16).

En el baud rate colocamos 1200 que es la velocidad de transmisión utilizada por el multímetro.

En el data bits se coloca 7 debido a que esos son los bits que se utilizarán.

En el stop bits se coloca 2 de acuerdo al manual de comunicación serial del Protek 506.

Para mostrar la información obtenida en la interfaz a un documento de texto, se puede realizarlo en la Séquense 0 o Séquense 3 para mayor comodidad se lo realiza en esta secuencia por espacio y para sincronizar el tiempo a un segundo.

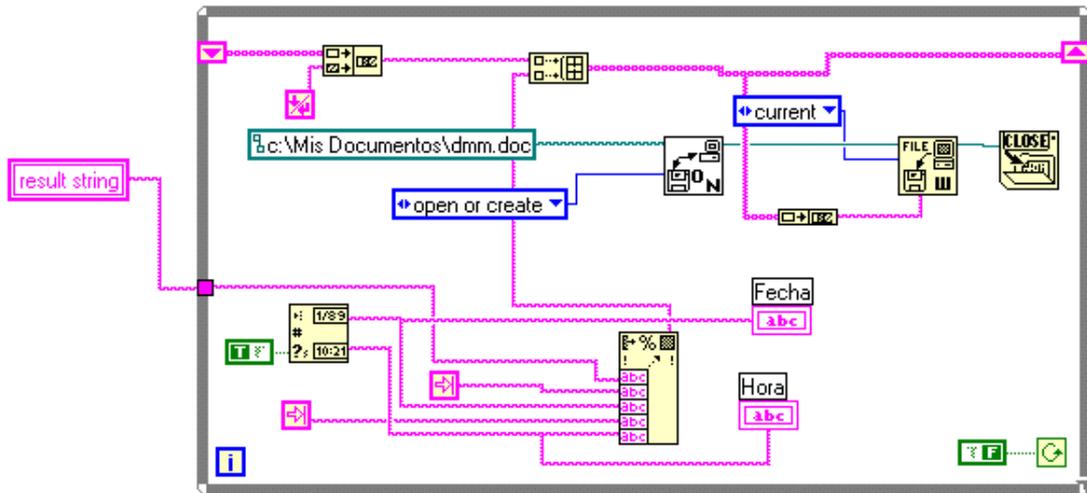


Fig. 3.9 Enviar los datos medidos a un documento.

Se utiliza el result string donde se muestra una variable local que permite llevar información de un lugar a otro sin alterarla.

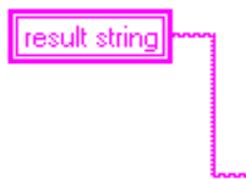


Fig. 3.10 Variable local

Esta información pasa al while loop, este se encarga de repetir un subdiagrama hasta una condición de falso, pasa al Get Date/Time String donde se coloca la fecha y la

hora. Luego pasando al Format Into String que nos sirve para arreglar los datos para su presentación posterior, se utiliza las flechas para darle una separación de un tab tanto a la fecha como a la hora y luego estas salen a dos indicadores.

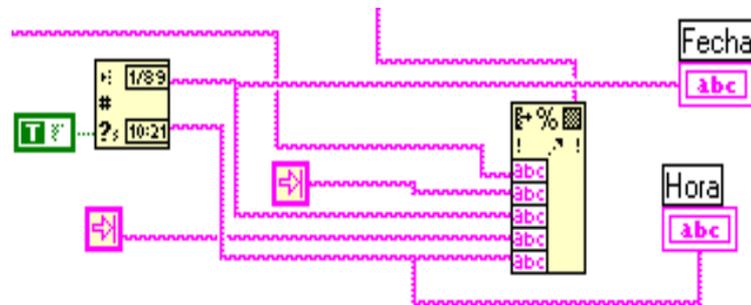


Fig. 3.11 Fecha y hora del programa.

Luego pasa al Concatenate Strings donde se indica que cada dato se vaya poniendo al final como una memoria lifo dándole un enter .

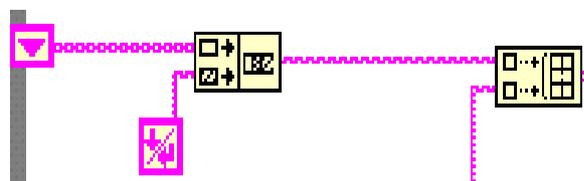


Fig. 3.12 Orden de los datos

Después de tener arreglados los datos, pasa al Open/Create/Replace File, donde se selecciona abrir o crear archivo y se da la dirección C:/MIS DOCUMENTOS/DMM.DOC , pasando esta información al Write File donde ingresa los datos medidos y se actualiza cada vez que ingrese un dato, terminando en el Close File donde se cierra el archivo.

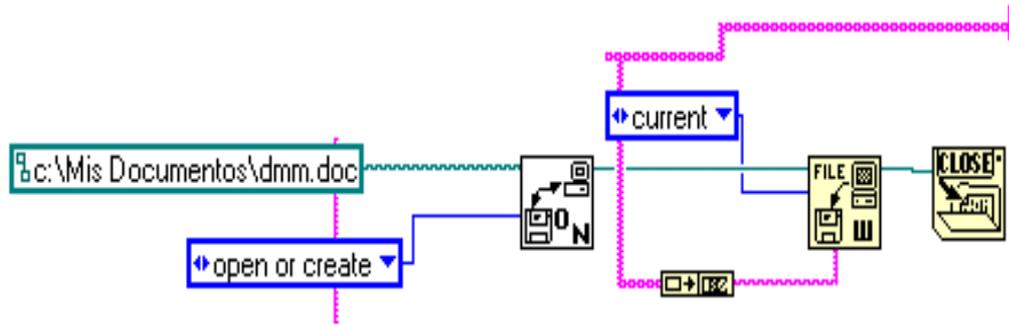


Fig. 3.13 Enviar los datos a un archivo de texto

- En la Sequence 1 se escribe en el puerto serial, el dato 0

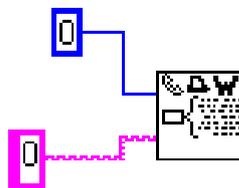


Fig. 3.14 Escribir en el puerto serial

En el port number se coloca el número de puerto a utilizar que es el COM1 o sea el código 0.

En el string to write especifica que es el dato 0.

- En la Sequence 2 retardo de tiempo tiempo.

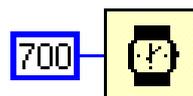


Fig. 3.15 Sincronizar el tiempo

En este paso sincronizamos el tiempo para la transmisión y recepción de datos entre el Multímetro y la computadora, este tiempo está en milisegundos.

- En la Sequence 3 desarrollo del programa.

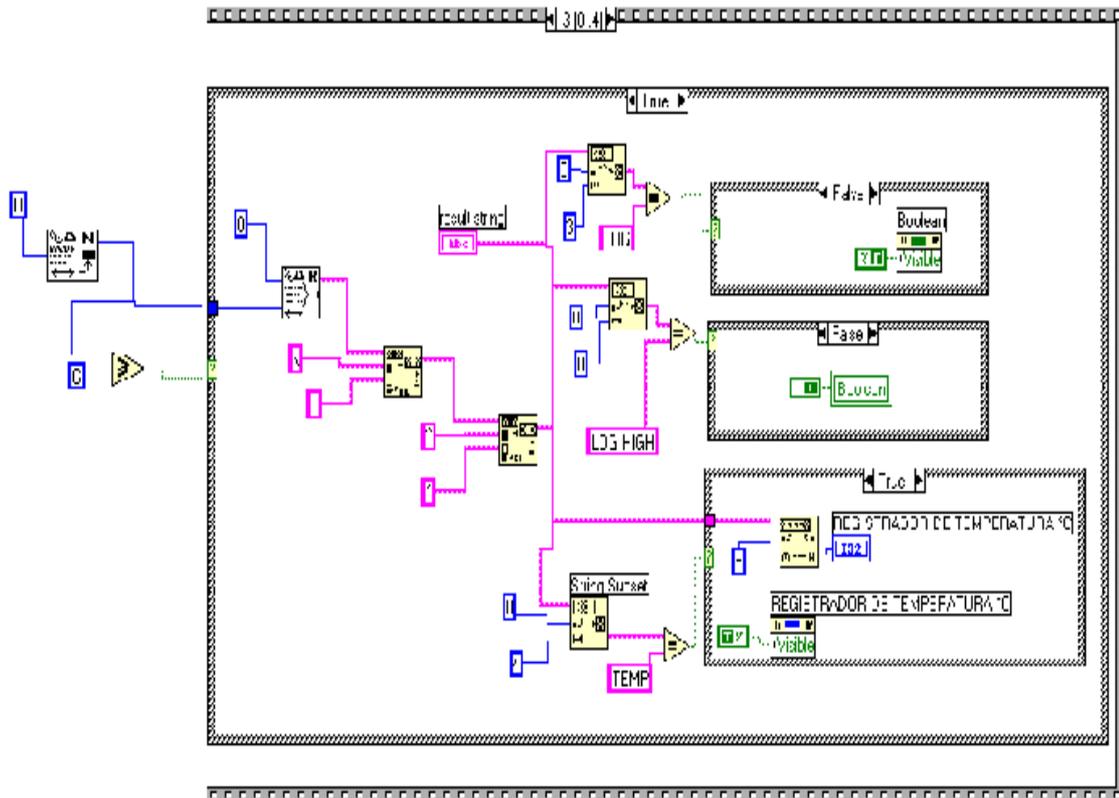


Fig. 3.16 Desarrollo del programa

El Bytes At Serial Port cuenta el número de bits existentes en el puerto, luego pasa este número a un comparador donde este número es comparado con el número de bits a utilizarse, si es menor o igual a 6 pasa a la estructura case o sea si es verdadero.

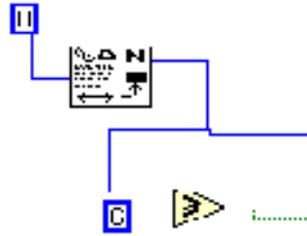


Fig. 3.17 Contador de bytes

Luego de haber pasado a la estructura case, pasa al Serial Port Read donde hay que dar el número de puerto que es 0 y pasa al número de bits contados de paso anterior, la cadena leída pasa al Search and Relace String donde el Protek 506 manda por defecto un / después de cada dato, lo que se hace aquí es buscar los / y se lo substituye por espacios en blanco, se vuelve a repetir este paso para buscar este símbolo ^ se lo reemplaza por el símbolo de grados ° para un mejor entendimiento de los datos y saber que es grados.

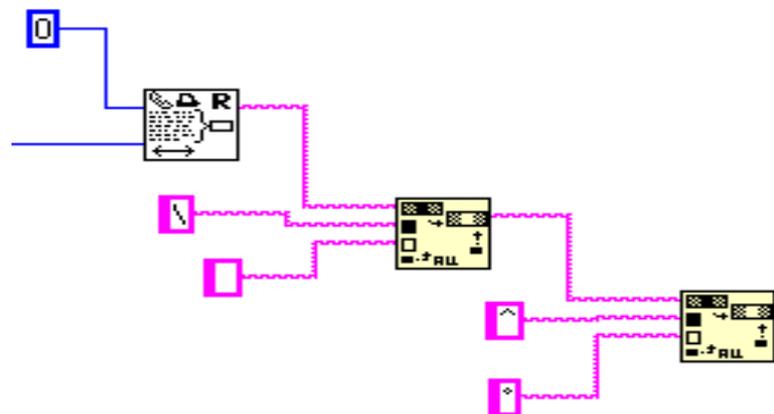


Fig. 3.18 Buscar y reemplazar caracteres.

Luego de haber eliminado los datos no deseados se pasan a un String Subset, donde para poder presentar el estado de una compuerta y saber si es alto o bajo.

Se busca palabras con tres caracteres y se lo compara con la palabra LOG, si es igual pasa al laso case donde si es verdadero se enciende el indicador, que es un led.

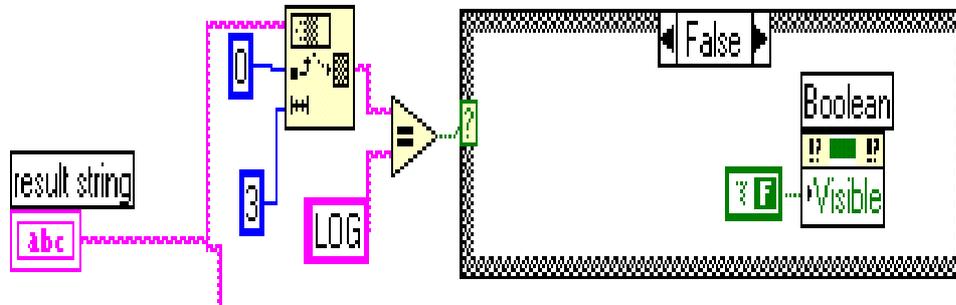


Fig. 3.19 Búsqueda de caracteres

De igual manera para la temperatura se utiliza el String Subset, primero se busca la palabra con 4 caracteres que es el indicador de estado y luego se comprueba, lo encontrado con la palabra TEMP si es igual pasa a un laso case, si es verdadero pasa a un indicador donde aparece el valor medido en grados y si es falso no indica. Para una mejor visualización se decidió graficar la temperatura para lo cual se utiliza el Decimal String To Number, se utiliza para transformar de una cadena de caracteres a numéricos ya que solo se gráfica con datos numéricos únicamente y luego pasa a un indicador gráfico (Wave form Chart)

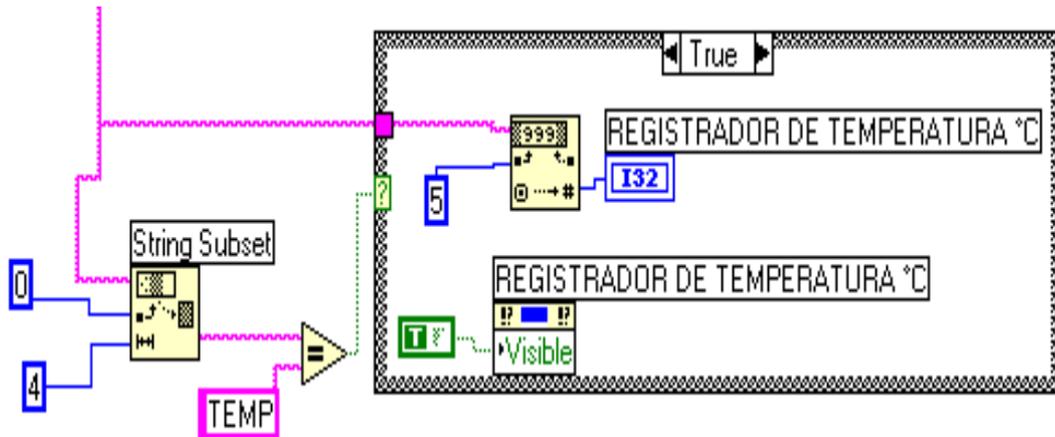


Fig. 3.20 Presentación de la temperatura.

- En la Sequence 4 cierra el puerto.

En este paso solo se procede a cerrar el puerto indicando el número de puerto utilizado.

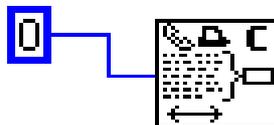


Fig. 3.21 Cerrar el puerto.

3.5 PRUEBAS Y ANÁLISIS DE RESULTADOS

3.5.1.- PRUEBAS Y ANÁLISIS DE RESULTADOS DEL COMPROBADOR

Probando el comprobador de cables funciona de manera adecuada para los cables RJ11 y RJ45, dando los resultados que se esperaba en este tipo de cables y sin ninguna inconveniente de funcionamiento.

3.5.2.- PRUEBAS Y ANÁLISIS DE RESULTADOS DEL PROTEK 506

Realizadas las pruebas en el Protek 506 de medidas, se especifica los rangos de medición de cada una de las funciones sin tener ningún problema en las medidas realizadas.

3.5.2.1.- DC VOLTAJE

Tabla. 3.1 Descripción de voltaje dc

Función	Rango	Resolución	Exactitud	Impedancia
DCmV	400mV	0,1mV	0,3%+2d	>1G ohms
DCV	4V	0,001V	0,5%+2d	10M ohms
	40V	0,01V		
	400V	0,1V		
	1000V	1V		

3.5.2.2.- AC VOLTAJE

Tabla. 3.2 Descripción de voltaje ac

Función	Rango	Resolución	Exactitud	Frecuencia
ACmV	400mV	0,1mV	1%+3d	50Hz~ 1KHz
ACV	4V	0,001V	1,5%+5d	50Hz~ 100Hz
	40V	0,01V		50Hz
	400V	0,1V		500Hz
	750V	1V		

3.5.2.3.- DC CORRIENTE

Tabla. 3.3 Descripción de corriente dc

Función	Rango	Resolución	Exactitud	Carga de voltaje
UA	400uA	0,1uA	1,0%+2d	1mV/uA
MA	400mA	0,1mA		1mV/mA
20 A	20 ^a	0,01A		10mV/A

3.5.2.4.- AC CORRIENTE

Tabla.3.4 Descripción de corriente ac

Función	Rango	Resolución	Exactitud	Frecuencia
UA	400uA	0,1uA	1,5%+3d(3,0%+5d)	1mV/uA
MA	400mA	0,1mA		50Hz- 100Hz (100Hz- 1KHz)
20 A	20 ^a	0,01 ^a		10mV/A

3.5.2.5.- RESISTENCIA

Tabla 3.5 Descripción de resistencias

Rango	Resolución	Exactitud	Circuito abierto
400 ohm	0,1 ohm	0,5%+2d	2,5V
4K ohm	0,001K ohm		
40K ohm	0,01K ohm		
400K ohm	0,1K ohm		
4M ohm	0,001M ohm	1%+2d	1,2V
40M ohm	0,01M ohm		

3.5.2.6.- CONTINUIDAD

Tabla 3.6 Descripción de continuidad

Rango	Pruebas	Pito	Digito principal	Sub digito
400 ohm	Abajo 100 ohm +/- 0,5%	Tono audible	Shrt	Valor de las resistencias desplegadas
	Encima 100 ohm +/- 0,5%	Sin tono	OPEn	

3.5.2.7.- DIODOS

Tabla 3.7 Descripción de diodos

Rango	Pruebas	Pant. Princ.	Pant. Secun.
4V	Abajo 05V	Shrt	Voltaje de los diodos desplegados
	Encima 1,0V	OPEn	
	0,5 a 1,0V	Good	

3.5.2.8.- FRECUENCIA

Tabla 3.8 Descripción de frecuencia

Rango	Resolución	Exactitud	Impedancia	Sensibilidad
10KHZ	1Hz	0,01% +/- 2d	10M ohm//<1nF	1.5V rms
100KHz	10Hz			
1MHz	100Hz			
10MHz	1KHz			

3.5.2.9.- TEMPERATURA

Tabla 3.9 Descripción de temperatura

Función	Rango	Resolución	Exactitud
°C	-20 °C a 1200 °C	1 °C	3%+5d desde -20 °C hasta 10 °C
			3%+3d hasta 350 °C
			5%+3d hasta 1200 °C
°F	°F En el despliegue del sub digito en los LCD es calculado y desplegado Por la formula $^{\circ}\text{F} = 32 + (9/5 \times ^{\circ}\text{C})$		

3.5.2.10.- LÓGICO

Tabla 3.10 Descripción de la función lógica

Rango	Pruebas	Pant. Princ..	Pant. Secun.	Observaciones
40 V	Abajo 0,8V	Lo	Valor DCV	Basado en TTL los niveles lógicos
	Encima 2,0V	Hi		
	0,8 a 2,0	-----		

3.5.2.11.- SEÑAL DE SALIDA

Tabla 3.11 Señal de salida

Función	Exactitud	Forma de onda	Nivel de salida
2048Hz	+/- 0,10%	Onda cuadrada	4,0Vp-p Min. sin cargar
4096Hz			
8192Hz			

3.5.2.12.- CAPACITANCIA

Tabla 3.12 Descripción de capacitancia

Rango	Resolución	Exactitud	Circuito abierto
100 uF	0,01uF	3%+5d	3,2V MAX

Debido que le Multímetro fue adquirido especialmente para la medición de capacitancias e inductancias se realiza pruebas más detalladas, como es la medición de capacitancias en el rango que especifica y con ella su comprobación.

El rango de la medición de las capacitancias es de 0.01uF hasta 100uF.

Tabla 3.13 Comprobacion de valores de capacitancias

Valor en uF especificado	Valor medido en uF
0.01uF	0.01uF
0.27uF	0.30uF
0.40uF	0.40uF
01.00uF	01.12uF
04.50uf	4.68uF
10.00uF	10.81uF
100.00uF	99.99uF
470.00uf	99.99uf

Esto son algunos valores medidos y comprobado que mide hasta el rango especificado de 0.01uF hasta 100uF.

Si se intenta medir valores fuera de este rango la pantalla muestra 99.99uF que es el máximo valor que alcanza a medir.

3.5.2.13.- INDUCTANCIA

Tabla 3.14 Descripción de inductancia

Rango	Resolución	Exactitud	Circuito abierto
100H	0,01H	3%+5d up to 20H	3,2V MAX
		3%+5d up to 50H	
		10%+5d up to 100H	

Debido que el Multímetro fue adquirido especialmente para la medición de capacitancias e inductancias se realiza pruebas más detalladas, como es la medición de inductancias en el rango que especifica y con ella su comprobación.

Tabla 3.15 Comprobación de valores de inductancias

Valor medido en H
4.17H
16.18H
4.05H
3.81H
2.37H

Estos valores fueron medidos en transformadores, obteniendo unas medidas adecuadas pero en escala de Henrios.

CAPITULO IV

4.1.- CRONOGRAMA

ACTIVIDADES	FECHAS																			
	MAYO / AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DICIEMBRE			
	M	J	J	A	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Preparación y entrega del perfil	X																			
Recolección de información		X																		
Clasificación de la información			X																	
Importación del multímetro		X	X	X	X															
Elaboración del capítulo I y II					X	X														
Entrega del primer borrador						X														
Elaboración del capítulo V							X													
Compra de elementos							X													
Estudio de Labview a utilizarse								X	X											
Construcción del circuito para el módulo										X	X	X								
Pruebas y análisis del circuito												X	X							
Elaboración de la interfaz													X	X	X	X	X	X	X	
Elaboración del capítulo III y IV															X	X	X			
Entrega del segundo borrador																	X			
Pruebas del medidor completo																			X	
Elaboración del capítulo V y anexos																				X
Entrega del borrador final																				X

4.2.- PRESUPUESTO

Multímetro PROTECK 506	\$526
Elementos electrónicos	\$20
Armazón	\$10
Cables	\$5
Recursos humanos	\$170
Conversor de DB25 a DB9	\$15
TOTAL	\$746

CAPITULO V

5.1.- CONCLUSIONES

- Se ha logrado concluir el comprobador de cables con los resultados esperados.
- Determinar con eficacia y rapidez los problemas en la línea de transmisión
- Mide con eficacia los valores de capacitancia comprendidos entre 0.01 a 100 uf,
- Mediante la interfaz se ha logrado visualizar y almacenar los datos en la computadora.
- Mediante este proyecto se aumento el desarrollo tecnológico en los laboratorios de telecomunicaciones del instituto.
- Se ha logrado una comunicación perfecta entre la PC y el multímetro.

5.2.- RECOMENDACIONES

- Mantener el interruptor del comprobador en el centro, que indica que se encuentra apagado.
- Tener cuidado al conectar las líneas de transmisión para evitar daños en los conectores.
- En las mediciones con el multímetro seleccionar correctamente la función a medir y su escala.
- Al momento de transmitir, verificar si el indicador “RS232” se encuentre parpadeando, esto significa que está transmitiendo.
- Una vez leído los datos en el archivo, se recomienda borrarlos para evitar confusiones en lecturas posteriores.
- Conectar correctamente el cable de la interfaz.

BIBLIOGRAFIA:

WAYNE TOMASI

SISTEMAS DE COMUNICACIONES ELECTRONICAS

EDICION : SEGUNDA

JUAN CARLOS RIVERA G.

CABLEADO EXTRUCTURADO

INTERNET EN LAS SIGUIENTES DIRECCIONES.

HTTP:// WWW.GOOGLE.COM.

HTTP:// WWW.MONOGRAFIAS.COM

HTTP://WWW.PROTEK.COM

ANEXOS

ANEXO A

CIRCUITO IMPRESO



Fig. A.1. Circuito Impreso

ANEXO B

MONTAJE DE COMPONENTES



Fig. B.1. Montaje de Componentes

ANEXO C

COLOCACIÓN EN LA CARCASA



Fig. C.1. Colocación en la carcasa

ANEXO D

No	Función	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Bytes
1	DC V	D	C		3	.	9	9	9		V	↵					11
2	DC V	D	C		-	3	.	9	9	9		V	↵				12
3	DC V	D	C		0	L	↵										6
4	AC V	A	C		3	.	9	9	9		V	↵					11
5	AC V	A	C		0	L	↵										6
6	DC mV	D	C		3	9	9	.	9		m	V	↵				12
7	DC mV	D	C		-	3	9	9	.	9		m	V	↵			13
8	AC mV	A	C		3	9	9	.	9		m	V	↵				12
9	DC uA	D	C		3	9	9	.	9		u	A	↵				12
10	DC uA	D	C		-	3	9	9	.	9		u	A	↵			13
11	DC uA	A	C		3	9	9	.	9		u	A	↵				12
12	DC mA	D	C		3	9	9	.	9		m	A	↵				12
13	DC mA	D	C		-	3	9	9	.	9		m	A	↵			13
14	AC mA	A	C		3	9	9	.	9		m	A	↵				12
15	DC 20A	D	C		3	9	.	9	9		A	↵					11
16	DC 20A	D	C		-	3	9	.	9	9		A	↵				12
17	Ac 20A	A	C		3	9	.	9	9		A	↵					11
18	Resistencia	R	E	S		3	.	9	9	9		M	O	H	M	↵	15
19	Resistencia	R	E	S		3	9	9	.	9		O	H	M	↵		14
20	Resistencia	R	E	S		0	L	↵									7
21	Continuidad	B	U	Z		S	H	O	R	T	↵						10
22	Continuidad	B	U	Z		O	P	E	N	↵	↵						9
23	Diodo	D	I	O		O	P	E	N	↵	↵						9
24	Diodo	D	I	O		S	H	O	R	T	↵						10
25	Diodo	D	I	O		G	O	O	D	↵	↵						9
26	Lógico	L	O	G		L	O	W	↵								8
27	Lógico	L	O	G		H	I	G	H	↵							9
28	Lógico	L	O	G		U	N	D	E	T	↵	↵					10
29	Frecuencia	F	R		9	.	9	9	9		M	H	z	↵			13
30	Frecuencia	F	R		9	.	9	9	9		K	H	z	↵			13
31	Capacitancia	C	A	P		9	9	.	9	9		u	F	↵			13
32	Capacitancia	C	A	P		O	L	↵									7
33	Inductancia	I	N	D		9	9	.	9	9		H	↵				12
34	Inductancia	I	N	D		0	L	↵									7
35	Temperatura	T	E	M	P		0	0	2	5		^	C	↵			13
36	Temperatura	T	E	M	P		-	0	0	2	5		^	C	↵		14

FORMATO DE DATOS

- Este es un formato de los datos enviados desde el multímetro a la computadora personal.
- Cada función de datos consiste de 6 a 15 bytes como muestra en el cuadro.

- Cada byte tiene códigos ASCII correspondientemente a cada carácter o número.

ANEXO E

MANUAL DEL USUARIO PARA CONECTAR EL MULTÍMETRO A LA PC



Fig. E.1. Conexión del Multímetro con la PC

- Retiramos la tapa en la parte superior del multímetro que cubre la interfaz.
- Conectamos el cable tanto en el multímetro como en la PC.
- Ingresamos al programa Labview, escogemos la opción open VI y seleccionamos tesis, luego aparece un cuadro de dialogo donde escogemos (DMM) y estamos dentro del programa.

- Corremos el programa.
- En el multímetro aplastamos la tecla MENU, y la mantenemos presionada hasta que parpadee en la pantalla RS232.
- Presionamos enter (↵) y el RS232 deja de parpadear, y debajo de este aparece Tx, Rx indistintamente indicando que esta transmitiendo correctamente.
- Para desactivar la conexión presionamos MENU y luego enter (↵) y desaparece de la pantalla el indicador RS232

GLOSARIO

Roseta: Se compone usualmente de dos salida aunque existe de una salida, posee un circuito impreso que soporta al RJ45 y conectores IDC.

RTC: Solicitud de datos de envió en el puerto serial.

UTP categoría 5: Esta denominación se aplica a cables UTP cuyas características de transmisión permiten un ancho de banda de hasta 100Mhz.

100 base TX: Son cables de red de par trenzado UTP de categoría 5 de 8 hilos.

Hubs: Es un dispositivo que centraliza la conexión de los cables de cada estación de trabajo. El concentrador (Hub) organiza los cables y transmite las señales de entrada a los restantes segmentos del medio de transmisión.

DB-9: Conectores de bus de datos de 9 pines.

DB-25: Conectores de bus de datos de 25 pines.

DATOS PERSONALES

APELLIDOS: TAPIA GAIBOR
NOMBRES: CHRISTIAN ROBERTO
EDAD: 21
CEDULA DE IDNT: 050267785-9
DIRECCIÓN DOMICILIARIA: CDLA.
TELEFONO: 803-733

ESTUDIOS REALIZADOS:

PRIMARIA: ESCUELA FISCAL "Dr. ISIDRO
AYORA"
SECUNDARIA: HERMANO MIGUEL
SUPERIORES: INSTITUTO TECNOLÓGICO SUPERIOR
AERONÁUTICO "ITSA"

DATOS PERSONALES

APELLIDOS: ESPIN BELTRAN
NOMBRES: CRISTIAN XAVIER
EDAD: 21
CEDULA DE IDNT: 050226936-8
DIRECCIÓN DOMICILIARIA: CDLA. LOS NEVADOS 1-49
TELEFONO: 800-128

ESTUDIOS REALIZADOS:

PRIMARIA: ESCUELA FISCAL "Dr. ISIDRO
AYORA"
SECUNDARIA: INSTITUTO TECNOLÓGICO SUPERIOR
"VICENTE LEON"
SUPERIORES: INSTITUTO TECNOLÓGICO SUPERIOR
AERONÁUTICO "ITSA"

HOJA DE LEGALIZACION DE FIRMAS

ELABORADO POR

Cristian Espín

Christian Tapia

DIRECTOR DE ESCUELAS DEL ITSA

Ing. Eduardo Castillo

Mayo. Tec. Avc.

Latacunga, 29 de Enero del 2003

