

T
621.462
EME



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Visualización de señales eléctricas para las prácticas de Laboratorio de Maquinaria Eléctrica utilizando el software LabView”

TESIS DE GRADO

Previo a la obtención del Título de:

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA INDUSTRIAL



CIB -ESPOL

Presentada por:

Julio Andrés Emén Sánchez



CIB -ESPOL

GUAYAQUIL-ECUADOR

Año: 2004

AGRADECIMIENTO

A todas las personas que colaboraron de alguna forma para la realización de esta tesis y especialmente al Sr. Manuel Rufino Asán por su invaluable ayuda.

DEDICATORIA



CIB -ESPOL

**A MIS PADRES
A MIS HERMANOS
A JOYCE**

TRIBUNAL DE GRADUACIÓN



Ing. Miguel Yapur A.
SUBDECANO DE LA FIEC
PRESIDENTE



Ing. Wilmer Naranjo R.
DIRECTOR DE TESIS



Ing. Jorge Chiriboga V.
VOCAL




Ing. Holger Cevallos U.
VOCAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)



Julio Andrés Emén Sánchez

RESUMEN

La presente tesis diseña e implementa la visualización de las señales eléctricas que se encuentran en el interior de los grupos motor-generador KATO del Laboratorio de Maquinaria Eléctrica de la FIEC, con el objetivo de brindar a los estudiantes una herramienta pedagógica para realizar las prácticas en el mencionado laboratorio.

Para el desarrollo de la presente se utilizaron dispositivos de hardware de National Instruments (SCXI-1327, SCXI-1000, SCXI-1120 y PCI6024E) para la adquisición, adecuación, filtrado y procesamiento de las señales y se utilizó el software LabView 7, también de National Instruments, para la programación de las diferentes pantallas de monitoreo y control; se utilizó también un microcontrolador de la familia Microchip (PIC 16F877A) para realizar las diferentes conexiones en las que pueden funcionar las máquinas KATO y controlar el accionamiento de los contactores del sistema.

En los primeros cuatro capítulos se explicarán los fundamentos básicos utilizados tanto de LabView 7 como de la programación del PIC 16F877A y la configuración de los dispositivos de hardware para el procesamiento de las señales. En el capítulo 5 se detalla el diseño de la aplicación incluyendo la tarjeta de control y las aplicaciones de LabView para el monitoreo de las señales y el control del puerto serial COM1. El capítulo 6 corresponde a los manuales del usuario y en el resto de capítulos se explican las ecuaciones utilizadas, algunas aplicaciones especiales y por último los programas en Ensamblador para el PIC y las aplicaciones de LabView..

INDICE GENERAL

	Pág.
RESUMEN.....	VI
INDICE GENERAL.....	VII
INDICE DE FIGURAS.....	XII
INDICE DE TABLAS.....	XIV
INDICE DE ESQUEMÁTICOS.....	XV
INTRODUCCIÓN.....	1
CAPITULO 1	
1. LENGUAJE DE PROGRAMACIÓN LABVIEW.....	3
1.1. Descripción de LabView.....	3
1.1.1. ¿Qué es LabView?.....	3
1.1.2. Funcionamiento de LabView.....	5
1.2. Recursos de LabView.....	7
1.3 ActiveX.....	11
1.4 Adquisición de datos.....	15

CAPITULO 2

2. MICROCONTROLADOR PIC 16F877A.....	19
2.1. Características del PIC 16F877A.....	19
2.2. Función del PIC en la aplicación.....	39
2.3. Recursos del PIC 16F877A.....	41
2.4. Detalles importantes del diseño con el PIC.....	52
2.5. Programación del PIC.....	53

CAPITULO 3

3. HARDWARE DE ADQUISICIÓN DE DATOS.....	55
3.1. Descripción del sistema de adquisición de datos.....	55
3.2. Características y configuración del conector SCXI-1327.....	57
3.3. Características y Configuración del Chasis SCXI-1000.....	60
3.4. Características y Configuración del módulo SCXI-1120.....	61
3.5. Características de la Tarjeta de adquisición de datos PCI-6024E.....	63

CAPITULO 4

4. INSTALACIÓN DE LOS CONTROLADORES (DRIVERS) Y CONFIGURACIÓN DEL SISTEMA.....	65
4.1. Instalación de los controladores.....	65
4.2. Generalidades de los controladores.....	67

4.3. Canales Virtuales (Virtual Channels).....	69
4.3.1. Generalidades.....	69
4.3.2 Configuración.....	70
CAPITULO 5	
5. DESARROLLO DE LA APLICACIÓN.....	74
5.1 Tarjeta de Control del Grupo Motor-Generador KATO.....	74
5.1.1. Descripción y objetivo de la tarjeta.....	74
5.1.2. Esquemático de la tarjeta.....	80
5.1.3. Descripción de los componentes.....	87
5.1.4 Costo de la tarjeta.....	90
5.2. Comunicación PC-PIC.....	91
5.2.1. Interfase RS-232.....	91
5.2.2. Protocolo de comunicación.....	96
5.2.2.1. Comandos de configuración.....	97
5.2.2.2. Comandos de operación.....	99
5.3. Programa en LabView.....	103
5.3.1. Explicación de las variables usadas.....	103
5.3.2. Explicación del programa.....	106
CAPITULO 6	
6. MANUALES DEL USUARIO.....	109



6.1. Manual de usuario para la tarjeta de control.....	109
6.2. Manual de usuario para las aplicaciones de LabView.....	119

CAPITULO 7

7. CIRCUITOS EQUIVALENTES DE LAS MAQUINAS ELECTRICAS.....	140
7.1. Ecuaciones para calcular los circuitos equivalentes.....	140
7.1.1. Generalidades de los circuitos equivalentes.....	140
7.1.2. Circuito equivalente del motor AC trifásico KATO.....	141
7.2. Pantallas de LabView para las diferentes pruebas.....	147

CAPITULO 8

8. MONITOREO DE SEÑALES.....	154
8.1. Pantallas de adquisición de datos.....	154
8.2. Análisis de las señales adquiridas.....	158

CAPITULO 9

9. BASE DE DATOS.....	162
9.1. Objetivos.....	162
9.2. Estructura de la base de datos.....	164

CAPITULO 10

10. SOFTWARE DE LA APLICACIÓN.....	167
10.1. Programa en Assembler para el PIC.....	167
10.2. Programa en LabView.....	218

CONCLUSIONES Y RECOMENDACIONES.....	388
-------------------------------------	-----

ANEXOS

BIBLIOGRAFIA

INDICE DE FIGURAS

	Pag.
Figura 1.1 Funciones del puerto serial.....	9
Figura 1.2 Funciones de ActiveX.....	13
Figura 1.3 Funciones de adquisición de datos.....	16
Figura 2.1 Empaquetados de los microcontroladores.....	21
Figura 2.2 Distribución de los pines del PIC16F877A.....	27
Figura 2.3 Diagrama de bloques del PIC.....	28
Figura 2.4 Ciclo de instrucción.....	30
Figura 2.5 Registro STATUS.....	32
Figura 2.6 Memoria de programa.....	33
Figura 2.7 Memoria de datos.....	38
Figura 2.8 Diagrama de bloques del TMR0.....	44
Figura 2.9 Registro OPTION_REG.....	45
Figura 2.10 Diagrama de bloques del transmisor del USART.....	47
Figura 2.11 Diagrama de bloques de receptor del USART.....	47
Figura 2.12 Registro RCSTA.....	49
Figura 2.13 Registro TXSTA.....	50
Figura 2.14 Instrucciones para la programación del PIC.....	54
Figura 3.1 Estructura del sistema.....	57
Figura 3.2 Conector SCXI-1327.....	59
Figura 4.1 NI-DAQ driver.....	68
Figura 5.1 Sistema de control y fuerza.....	78
Figura 6.1 Mensaje inicial.....	110
Figura 6.2 Función de las botoneras.....	110
Figura 6.3 Menú modo de operación.....	111
Figura 6.4 Tipo de arranque.....	113
Figura 6.5 Tipo de conexión.....	114
Figura 6.6 Conexión de las bobinas.....	114
Figura 6.7 Tiempo para la resistencia de cambio de giro.....	114
Figura 6.8 Sentido de giro.....	115
Figura 6.9 Mensaje final.....	115
Figura 6.10 Desconexión de R1.....	116
Figura 6.11 Cambio Ye-Delta.....	117
Figura 6.12 Mensaje de error.....	117
Figura 6.13 Pruebas para el motor AC KATO.....	120
Figura 6.14 Resistencia del estator.....	121

Figura 6.15	Prueba de rotor bloqueado.....	122
Figura 6.16	Prueba de vacío.....	124
Figura 6.17	Circuito equivalente por fase.....	125
Figura 6.18	Configuración de la máquina AC.....	126
Figura 6.19	Definición de tiempos.....	128
Figura 6.20	Pantalla de contactores.....	131
Figura 6.21	Visualización de señales.....	134
Figura 6.22	Análisis de los voltajes de línea.....	136
Figura 6.23	Análisis de potencia.....	137
Figura 6.24	Record de ondas.....	138
Figura 7.1	Circuito equivalente final.....	141
Figura 7.2	Circuito equivalente del motor con el rotor bloqueado.....	143
Figura 7.3	Circuito equivalente del motor en vacío.....	146
Figura 7.4	Pruebas para el motor AC.....	148
Figura 7.5	Resistencia del estator.....	149
Figura 7.6	Valor de resistencia inválido.....	149
Figura 8.1	Diagrama de bloques para la adquisición de datos.....	155
Figura 8.2	Al Acquire Waveforms.vi.....	156

INDICE DE TABLAS

	Pag.
Tabla 1 Capacitores recomendados.....	23
Tabla 2 Selección de bancos.....	36
Tabla 3 Puertos del PIC.....	41
Tabla 4 Configuración de los pines del PIC.....	42
Tabla 5 Configuración de los puertos de Entrada/Salida.....	42
Tabla 6 Asignación de señales.....	58
Tabla 7 Ganancia de los canales.....	62
Tabla 8 Configuración de los canales virtuales.....	73
Tabla 9 Comandos de configuración.....	98
Tabla 10 Comandos de operación.....	100

INDICE DE ESQUEMÁTICOS

	Pag.
Esquemático 1 Interfaz de entrada.....	80
Esquemático 2 Borneras.....	81
Esquemático 3 Interfaz de salida (1).....	82
Esquemático 4 Microcontrolador.....	83
Esquemático 5 Interfaz de salida (2).....	84
Esquemático 6 Circuito de fuerza.....	85
Esquemático 7 Bloque Terminal SCXI-1327.....	86

INTRODUCCIÓN

El presente trabajo titulado “Visualización de señales eléctricas para las prácticas del Laboratorio de Maquinaria Eléctrica utilizando el software LabView” ha sido implementado para ofrecer a los profesores y estudiantes una alternativa para el desarrollo de las prácticas que se realizan en el mencionado laboratorio de la Facultad de Ingeniería Eléctrica y Computación.

Para el desarrollo de este proyecto de tesis se utilizaron dispositivos de National Instruments (NI) y el software “LabView 7” que pertenece también a NI. Se eligieron los dispositivos de NI debido a que el Laboratorio de Maquinaria Eléctrica de la FIEC ya contaba con ellos y eran adecuados para la implementación del proyecto; junto a los dispositivos de NI se utilizaron otros elementos como son: un tacogenerador, tres shunts, un cable trifásico, dos tomas trifásicas, dos enchufes trifásicos, entre otros, los cuales en su mayoría se encontraban en la bodega del laboratorio. Todo lo mencionado forma parte del sistema de monitoreo del proyecto, para el sistema de control se utilizaron: doce contactores con un contacto auxiliar cada uno, una tarjeta de control (diseñada para el proyecto), doce resistencias de alta potencia y un cable de comunicación serial (RS-232). Se diseñó una tarjeta de control

para darle independencia al sistema de control del sistema de monitoreo, por lo tanto, el sistema de control puede funcionar por sí solo con ciertas limitaciones.

Todo los elementos utilizados tienen como objetivo realizar las prácticas del laboratorio pero con mayor precisión y profundidad, debido a que se podrán observar los transientes que ocurren en el arranque y en cualquier otra situación que los produzca.

Otro objetivo del proyecto es analizar la máquina KATO para obtener su circuito equivalente y predecir su comportamiento mediante las curvas que se pueden generar con los datos obtenidos de las señales de voltaje, corriente y velocidad de la misma.

Cabe indicar que el proyecto deja espacio para que el alumno aporte con sus conocimientos o investigue, el proyecto no lo hace ni lo muestra todo.

CAPITULO 1

1. LENGUAJE DE PROGRAMACIÓN LABVIEW

1.1 Descripción de LabView.

1.1.1 ¿Qué es LabView?

LabView es un lenguaje de programación gráfico que usa íconos en lugar de líneas de texto para crear aplicaciones. En contraste con los lenguajes basados en texto, donde comandos o instrucciones determinan la ejecución del programa, LabView utiliza programación por flujo de información, en donde los datos determinan la ejecución del programa. En LabView el programador diseña una interfase para el usuario usando un conjunto de herramientas y objetos. La interfase de usuario se conoce como panel frontal (o Front Panel); luego se añade el código usando representaciones gráficas de funciones para

controlar los objetos del panel frontal. El código está en el diagrama de bloques (o Block Diagram). Si se diseña apropiadamente el diagrama de bloques se asemeja a un diagrama de flujo.

LabView está listo completamente para la comunicación con otros hardwares como son GPIB, VXI, PXI, RS-232, RS-485 y dispositivos de adquisición de datos, también dispone de características internas para conectar una aplicación a internet usando el servidor web de LabView y estándares de software como TCP/IP y ActiveX.

LabView permite crear aplicaciones de 32 bits que dan una ejecución más rápida para realizar adquisición de datos, pruebas, mediciones y controlar procesos.

Algunos de los recursos ya nombrados se explicarán detalladamente más adelante.

1.1.2 Funcionamiento de LabView.

Los programas en LabView son llamados Instrumentos Virtuales (Virtual Instruments o VIs), porque su apariencia y operación imita a instrumentos reales como osciloscopios y multímetros. Cada VI usa funciones que manipulan entradas desde la interfase del usuario u otras fuentes y muestran la información o la mueven a otros archivos u otras computadoras.

Un VI contiene los siguientes componentes:

- ✓ **Panel frontal:** Sirve de interfase para el usuario.

- ✓ **Diagrama de bloques:** Contiene código fuente gráfico que determina el funcionamiento del VI.

- ✓ **Icono y Conector:** identifica al VI de tal forma que puede ser usado por otros VI, es decir para ser usado como subrutina o SubVI.

Panel Frontal (front panel).- Como se dijo anteriormente es la interfase entre el código y el usuario, se lo construye o diseña con controles e indicadores , que son las entradas y salidas de los terminales del VI respectivamente. Los controles consisten en perillas, botones, diales, entre otros; los indicadores son gráficos, leds y otros tipos de displays. Los controles simulan las entradas de los dispositivos los cuales suministran datos al diagrama de bloques o código fuente. Los indicadores simulan las salidas de los dispositivos y muestran los datos que generan el diagrama de bloques.

Diagrama de bloques (Block diagram).- Después de que se construye el panel frontal, se añade el código usando representaciones gráficas de funciones para controlar los objetos del panel frontal. El diagrama de bloques contiene el código fuente gráfico. Los objetos del panel frontal aparecen en el diagrama de bloques como terminales. Cada control o indicador en el panel frontal tiene su correspondiente terminal en el diagrama de bloques. Cables o líneas conectan cada uno de los nodos en el diagrama de bloques incluyendo los

terminales de los controles, indicadores, funciones y estructuras.

1.2 Recursos de LabView.

A continuación se hará una descripción y explicación de los recursos de LabView que se usaron en la implementación del proyecto.

Los recursos que se usaron son los siguientes:

- ✓ Comunicación serial (RS-232).

- ✓ Adquisición de datos por medio de una tarjeta de adquisición de datos.

- ✓ Comunicación con Microsoft Excel por medio de ActiveX.



CIB -ESPOL

Comunicación Serial (RS-232).- Este recurso LabView es utilizado para comunicar al computador con la tarjeta de control del grupo motor-generador Kato. Mediante este puerto se envían datos a la tarjeta y también se reciben datos desde ella. Los datos que se envían a la tarjeta pueden ser acciones de conexión del motor o datos de configuración de las diferentes variables de temporización que se utilizan.

Para que LabView ofrezca este recurso se debe haber incluido en la instalación de los controladores de National Instruments la opción que permite a LabView acceder a este puerto. En la instalación se la encuentra como VISA Serial.

La figura 1.1 muestra cómo encontrar y cuáles son las funciones que permiten manejar el puerto serial del computador utilizando LabView.

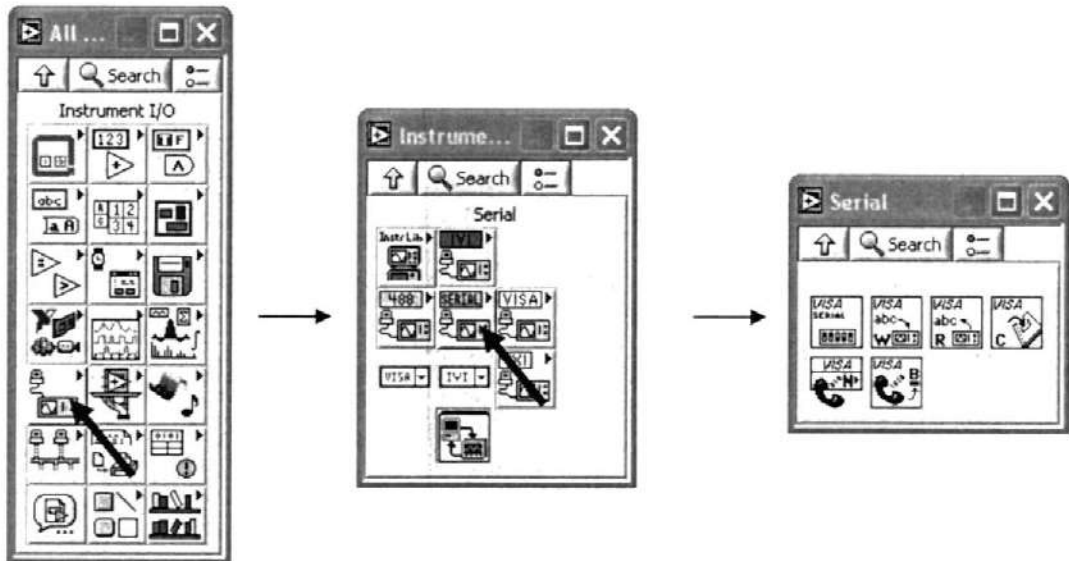
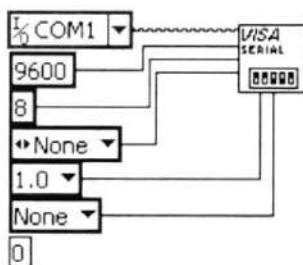


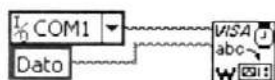
Figura 1.1, Funciones del puerto serial.

A continuación se muestran las funciones y la explicación de cada una de ellas con un ejemplo:



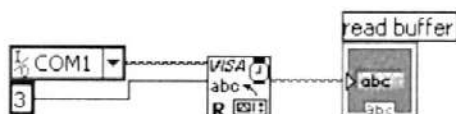
VISA Serial: Esta función es la primera que se debe utilizar ya que sirve para configurar el puerto serial. Se debe especificar el nombre del puerto (COM1), la velocidad en baudios (9600), los bits de datos (8), la paridad (None=nula), bits de parada (1) y control de flujo (None).

VISA Write: Esta función escribe datos en el puerto serial.



Se especifica el nombre del puerto (COM1), y el dato que se quiere escribir (Dato).

VISA Read: Esta función lee datos desde el puerto serial. Se debe especificar el nombre del puerto (COM1) y el número de bytes que se van a recibir. Como salida la función devuelve los datos leídos como una cadena de caracteres.



VISA Close: Cierra la sesión VISA correspondiente al puerto especificado (COM1), esto permite que otros programas puedan utilizar el puerto sin ningún problema.



1.3 ActiveX.

ActiveX es el nombre general para un conjunto de tecnologías de Microsoft que permite a los usuarios re-usar y re-llamar programas individuales para satisfacer sus necesidades de software. Basado en la tecnología COM (Component Object Model o Modelo de componentes de objetos), ActiveX es una continuación o mejora de una tecnología previa llamada OLE (Object Linking and Embedding). El principio de esta tecnología consiste en que los componentes no necesitan ser re-escritos para cada programa sino que son re-usados dándole al usuario el poder de combinar varias aplicaciones a la vez.

En LabView encontramos algunas definiciones que nos ayudan a usar ActiveX, entre ellas se encuentran:

- ✓ **ActiveX Automation:** Se refiere al proceso de controlar un programa desde otro por medio de ActiveX. Como en las redes de computadoras, un programa actúa como cliente y el otro como servidor. Con LabView se pueden hacer aplicaciones tanto como cliente como servidor. Cada programa existe de forma

independiente el uno del otro, pero ellos son capaces de compartir información entre ellos. Los objetos tienen métodos y propiedades que pueden ser accedidos por el Automation Client. Las propiedades de los objetos son simples atributos de un objeto cuyos valores pueden ser cambiados o recuperados por otros programas. De forma similar los métodos son funciones que realizan los objetos y ellos pueden ser invocados o llamados por otros programas. En el caso de esta aplicación LabView se comunicará con Microsoft Excel usando ActiveX (esto será explicado en el capítulo 9).

- ✓ **Controles y Contenedores ActiveX:** El uso más común de ActiveX es el de usar los controles ActiveX, los cuales son componentes embebidos que existen dentro de los contenedores ActiveX. Cualquier programa que puede funcionar como un contenedor ActiveX dejará que otros usuarios (programas) inserten controles ActiveX en ellos. Desde estos contenedores los controles ActiveX tienen sus propias propiedades y funcionalidad. Una vez más se controla todo por medio de propiedades y métodos.

- ✓ **Eventos ActiveX:** ActiveX es una tecnología que se maneja por eventos. Esto significa que los programas reportan cuando ciertos eventos han sucedido para que otros programas o el usuario pueda reaccionar de forma acorde.

La figura 1.2 muestra las funciones de LabView que permiten la comunicación por medio de ActiveX y se explican los más usados.

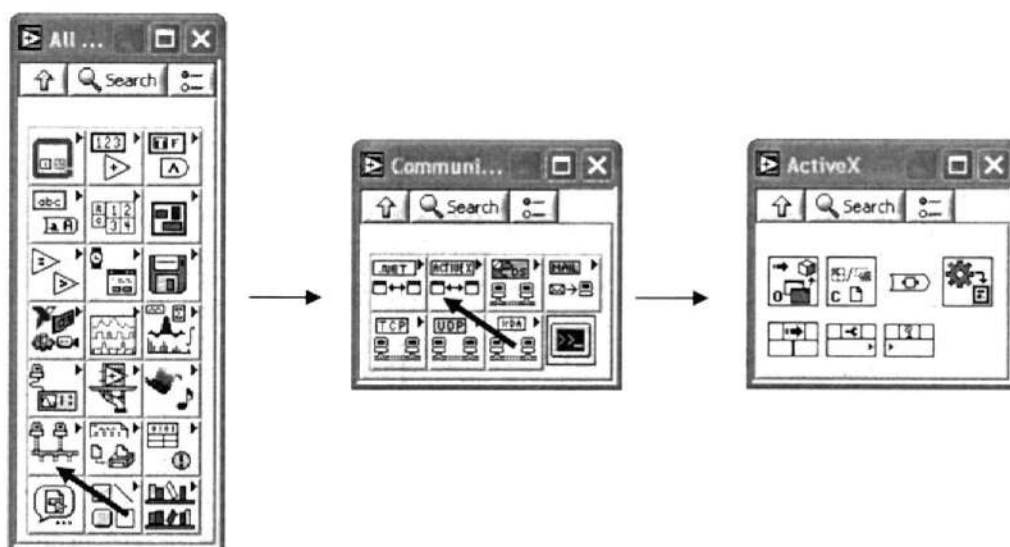
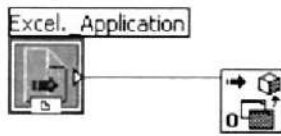
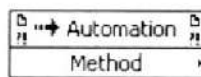


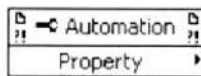
Figura 1.2, Funciones de ActiveX



Automation Open: Sirve para seleccionar con qué aplicación se va a comunicar LabView. Hay una gran variedad de librerías de objetos disponibles.



Invoke Node: Sirve para llamar a un método o una función del otro programa.



Property Node: Sirve para llamar o recuperar el valor de una propiedad del objeto.



Variant to Data: Convierte un item del otro programa en un dato válido para LabView.



To Variant: Convierte un dato de LabView en un dato o item válido para el otro programa.

1.4 Adquisición de datos.

Otro de los recursos muy importantes de LabView que se utilizó en la aplicación es la facilidad que da LabView para conectarse con tarjetas de adquisición de datos y obtener datos a una velocidad elevada. En el caso de la aplicación diseñada la tarjeta de adquisición de datos que se utilizó es la PCI-6024E, la cual pertenece a National Instruments, como introducción se puede decir que esta tarjeta funciona en bus PCI y que dispone de entradas analógicas, entradas y salidas digitales y una salida analógica (esta tarjeta se explicará más detalladamente en el capítulo 3).

Para que LabView pueda obtener información desde la tarjeta PCI-6024E, se debe instalar el respectivo controlador (o driver) de la tarjeta, el cual se encuentra en el CD de instalación de los dispositivos National Instruments.

Otro aspecto importante que vale resaltar es que la forma más fácil de acceder a la información de la tarjeta es asignándole nombres a cada entrada de la tarjeta, esto se lo realiza desde un programa de National

Instruments que sirve de complemento a LabView, este programa se llama Measurement and Automation Explorer.

Asignarle un nombre a una entrada (o canal) recibe el nombre de Canal virtual (Virtual Channel), es decir que se hace un canal virtual para cada entrada que se quiera utilizar en la aplicación.

Después de que se asignaron todos los canales virtuales que se necesitan se debe hacer uso de las funciones de LabView que se comunican con la tarjeta para recoger los datos que han sido adquiridos, en la figura 1.3 se muestran las funciones más utilizadas y su explicación.

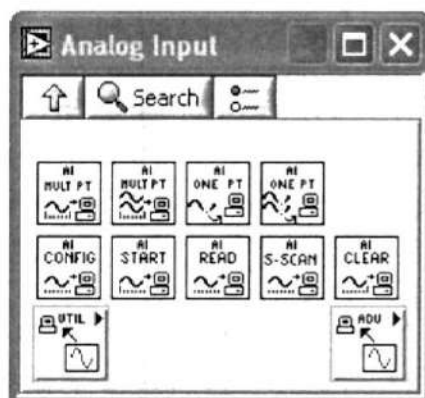
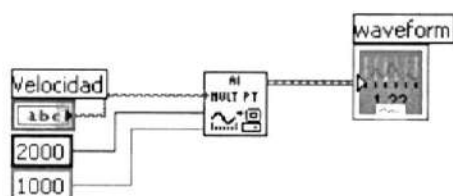
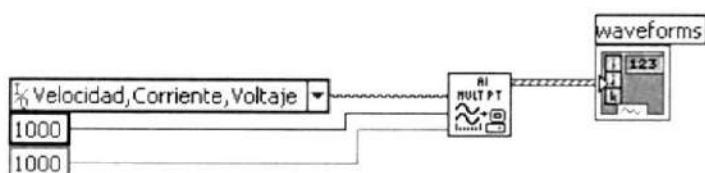


Figura 1.3, Funciones de adquisición de datos.

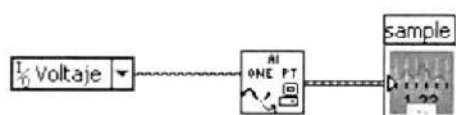
AI Acquire Waveform (Adquirir forma de onda de entrada analógica): Adquiere una forma onda o una sucesión de mediciones y las muestra en una gráfica. Se especifica el nombre del canal (Velocidad), el número de muestras que se tomarán (2000) y las muestras por segundo (1000).



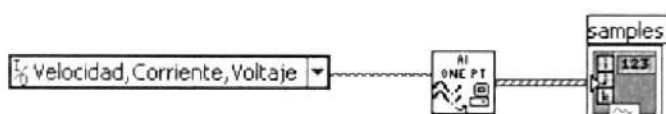
AI Acquire Waveforms: Hace lo mismo que la función



anterior pero adquiere mediciones de varios canales a la vez (Velocidad, Corriente, Voltaje) y guarda las mediciones en una tabla. Se debe especificar los canales a los que se quiere hacer un muestreo, el número de muestras y el número de muestras por segundo.



AI Sample Channel (Muestra de un canal de entrada analógica): Toma solo una muestra del canal especificado y lo muestra en algún indicador.



AI Sample Channels: Hace lo mismo que la anterior pero toma una muestra de cada canal especificado a la vez.

CAPITULO 2

2. MICROCONTROLADOR PIC 16F877A.

2.1 Características del PIC16F877A.

El microcontrolador PIC16F877A de Microchip pertenece a una gran familia de microcontroladores de 8 bits (bus de datos) que tienen las siguientes características generales que los distinguen de otras familias:

- Arquitectura Harvard
- Tecnología RISC
- Tecnología CMOS

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa y por lo tanto en la velocidad de ejecución.

Microchip ha dividido sus microcontroladores en tres grandes subfamilias de acuerdo al número de bits de su bus de instrucciones:

Base – Line: 12 PIC12XXX y PIC14XXX

Mid – Range: 14 PIC16XXX

High – End: 16 PIC17XXX y PIC18XXX

Variantes principales.

Los microcontroladores que produce Microchip cubren una amplio rango de dispositivos cuyas características pueden variar como sigue:

- Empaquetado (desde 8 patitas hasta 68 patitas)
- Tecnología de la memoria incluida (EPROM, ROM, Flash)
- Voltajes de operación (desde 2.5 v. Hasta 6v)
- Frecuencia de operación (Hasta 20 Mhz)

Empaquetados

Aunque cada empaquetado tiene variantes, especialmente en lo relativo a las dimensiones del espesor del paquete, en general se pueden encontrar paquetes tipo PDIP (Plastic Dual In Line Package), PLCC (Plastic Leaded Chip Carrier) y QFP (Quad Flat Package), los cuales se muestran en la figura 2.1.

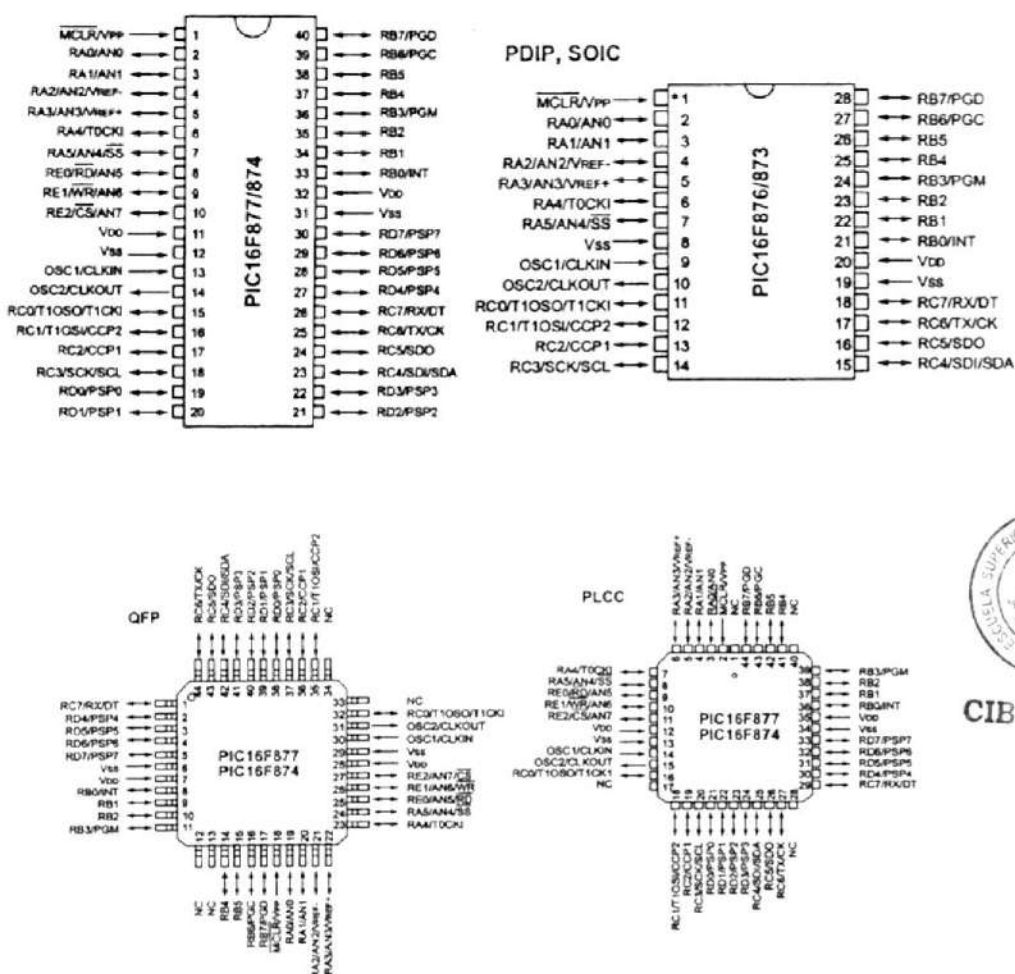


Figura 2.1, Empaquetados de los microcontroladores.



Oscilador.

Los PIC de rango medio permiten hasta 8 diferentes modos para el oscilador. El usuario puede seleccionar alguno de estos 8 modos programando 3 bits de configuración del dispositivo denominados: FOSC2, FOSC1 y FOSC0. En algunos de estos modos el usuario puede indicar que se genere o no una salida del oscilador (CLKOUT) a través de una patita de Entrada / salida. Los modos de operación se muestran en la siguiente lista:

- **LP:** Baja frecuencia (y bajo consumo de potencia)
- **XT:** Cristal / Resonador cerámico externos, (Media frecuencia)
- **HS:** Alta velocidad (y alta potencia) Cristal / resonador
- **RC:** Resistencia / capacitor externos (mismo que EXTRC con CLKOUT)
- **EXTRC:** Resistencia / capacitor externos
- **EXTRC:** Resistencia / Capacitor externos con CLCKOUT
- **INTRC:** Resistencia / Capacitor internos para 4 MHz
- **INTRC:** Resistencia / Capacitor internos para 4 MHz con CLKOUT

Los tres modos LP, XT y HS usan un cristal o resonador externo, la diferencia sin embargo es la ganancia de los drivers internos, lo cual se ve reflejado en el rango de frecuencia admitido y la potencia consumida.

En la tabla 1 se muestran los rangos de frecuencia así como los capacitores recomendados para un oscilador en base a cristal.

Modo	Frecuencia típica	Capacitores recomendados	
		C1	C2
LP	32 khz	68 a 100 pf	68 a 100 pf
	200 khz	15 a 30 pf	15 a 30 pf
XT	100 khz	68 a 150 pf	150 a 200 pf
	2 Mhz	15 a 30 pf	15 a 30 pf
	4 Mhz	15 a 30 pf	15 a 30 pf
HS	8 Mhz	15 a 30 pf	15 a 30 pf
	10 Mhz	15 a 30 pf	15 a 30 pf
	20 Mhz	15 a 30 pf	15 a 30 pf

Tabla 1, Capacitores recomendados.

Cristal externo: En los tres modos mostrados en la tabla anterior se puede usar un cristal o resonador cerámico externo.

Circuito RC externo: En los modos RC y EXTRC el PIC puede generar su señal oscilatoria basado en un arreglo RC externo conectado a la patita OSC1.

En cuánto a las características del funcionamiento del PIC 16F877A podemos mencionar las siguientes:

- CPU RISC
- Sólo 35 instrucciones que aprender
- Todas las instrucciones se ejecutan en un ciclo de reloj, excepto los saltos que requieren dos
- Frecuencia de operación de 0 a 20 MHz (DC a 200 nseg de ciclo de instrucción)
- Hasta 8k x 14 bits de memoria Flash de programa
- Hasta 368 bytes de memoria de datos (RAM)
- Hasta 256 bytes de memoria de datos EEPROM
- Hasta 4 fuentes de interrupción
- Pila (stack) de hardware de 8 niveles
- Reset de encendido (POR)
- Timer de encendido (PWRT)
- Timer de arranque del oscilador (OST)
- Sistema de vigilancia Watchdog timer.

- Protección programable de código
- Modo SEP de bajo consumo de energía
- Opciones de selección del oscilador
- Programación y depuración serie "In-Circuit" (ICSP) a través de dos patitas
- Lectura / escritura de la CPU a la memoria flash de programa
- Rango de voltaje de operación de 2.0 a 5.5 voltios.
- Alta disipación de corriente de la fuente: 25mA.
- Rangos de temperatura: Comercial, Industrial y Extendido
- Bajo consumo de potencia:
 - * Menos de 0.6mA a 3V, 4 Mhz
 - * 20 μ A a 3V, 32 Khz
 - * menos de 1 μ A corriente de standby.

Periféricos:

- **Timer0:** Contador/Temporizador de 8 bits con pre-escalador de 8 bits.
- **Timer1:** Contador / temporizador de 16 bits con pre-escalador.
- **Timer2:** Contador/Temporizador de 8 bits con pre-escalador y post-escalador de 8 bits y registro de periodo.
- **Dos módulos de Captura, Comparación y PWM.**
- **Convertidor Analógico/Digital:** de 10 bits, hasta 8 canales.
- **Puerto Serie Síncrono (SSP).**
- **Puerto Serie Universal (USART/SCI).**
- **Puerto Paralelo Esclavo (PSP):** de 8 bits con líneas de protocolo.

Distribución de los Pines del PIC 16F877A:

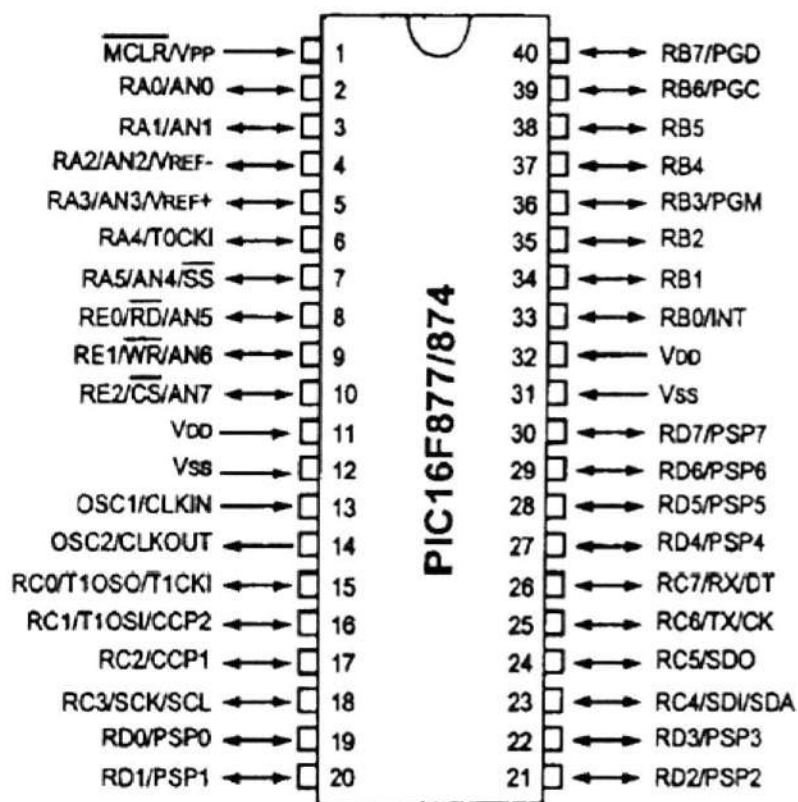


Figura 2.2, Distribución de pines del PIC16F877A

Diagrama de Bloques interno del PIC 16F877A:

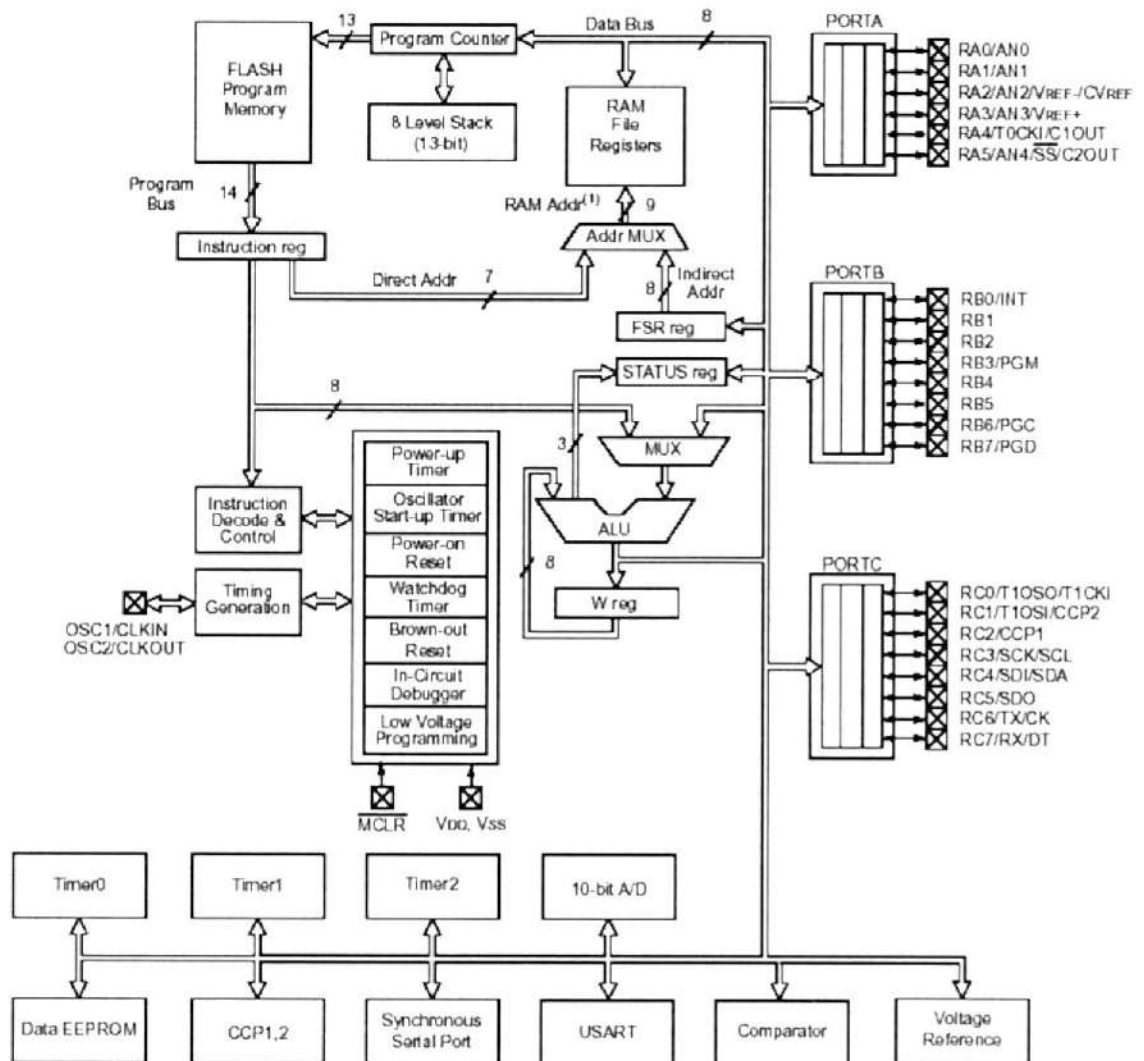


Figura 2.3, Diagrama de bloques del PIC16F877A.

Descripción de la CPU

La CPU es la responsable de la interpretación y ejecución de la información (instrucciones) guardada en la memoria de programa. Muchas de estas instrucciones operan sobre la memoria de datos. Para operar sobre la memoria de datos además, si se van a realizar operaciones lógicas o aritméticas, requieren usar la Unidad de Lógica y Aritmética (ALU). La ALU controla los bits de estado (Registro STATUS), los bits de este registro se alteran dependiendo del resultado de algunas instrucciones.

Ciclo de instrucción: El registro Program Counter (PC) es gobernado por el ciclo de instrucción como se muestra en la figura 2. Cada ciclo de instrucción la CPU lee (ciclo Fetch) la instrucción guardada en la memoria de programa apuntada por PC y al mismo tiempo ejecuta la instrucción anterior, esto debido a una cola de instrucciones que le permite ejecutar una instrucción mientras lee la próxima.

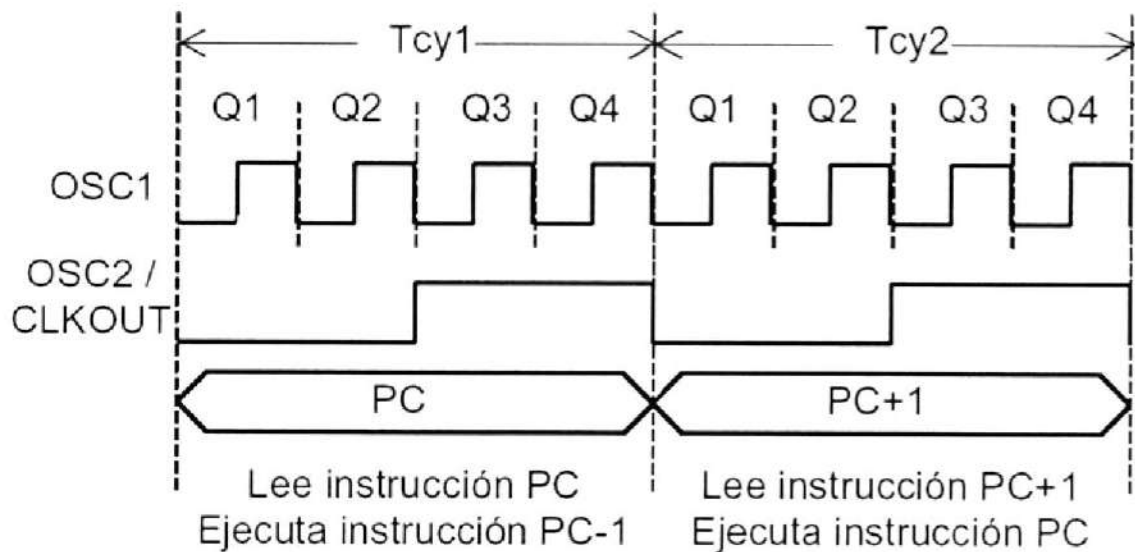


Figura 2.4, Ciclo de instrucción.

Como puede verse, cada ciclo de instrucción (Tcy) se compone a su vez de cuatro ciclos del oscilador (Tosc). Cada ciclo Q provee la sincronización para los siguientes eventos:

- Q1: Decodificación de la instrucción
- Q2: Lectura del dato (si lo hay)
- Q3: Procesa el dato
- Q4: Escribe el dato

Registros de la CPU.

Registro PC.- Registro de 13 bits que siempre apunta a la siguiente instrucción a ejecutarse. En la siguiente sección se dan mayores detalles en el manejo de este registro.

Registro de Instrucción.- Registro de 14 bits. Todas las instrucciones se colocan en el para ser decodificadas por la CPU antes de ejecutarlas.

Registro W.- Registro de 8 bits que guarda resultados temporales de las operaciones realizadas por la ALU.

Registro STATUS.- Registro de 8 bits, cada uno de sus bits (denominados Banderas) es un indicador de estado de la CPU o del resultado de la última operación como se indica en la figura 2.5.



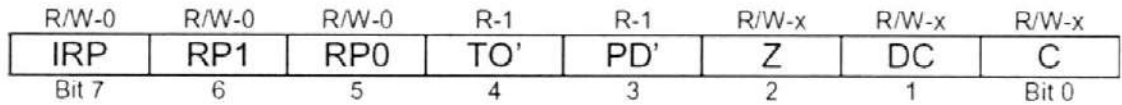


Figura 2.5, Registro STATUS

Z.- Este bit se coloca en uno (1) para indicar que el resultado de la última operación fue cero, de lo contrario se coloca en cero (0).

C.- Bit de acarreo o préstamo de la última operación aritmética (en el caso de resta, se guarda el préstamo invertido)

CD.- Acarreo / préstamo proveniente del cuarto bit menos significativo. Funciona igual que el bit C, pero para operaciones de 4 bits.

Descripción de la Memoria de Programa

Los PIC de rango medio poseen un registro Contador del Programa (PC) de 13 bits, capaz de direccionar un espacio de 8K x 14, como todas las instrucciones son de 14 bits, esto significa un bloque de 8k instrucciones. El bloque total de 8K x 14 de memoria de programa está subdividido en 4 páginas de 2K x 14. En la figura 2.6 se muestra esta organización.

Dirección	
0000h	Vector de Reset
...	...
0004h	Vector de interrupción
0005h	Página 0
...	
07FFh	
0800h	Página 1
...	
0FFFh	
1000h	Página 2
...	
17FFh	
1800h	Página 3
...	
1FFFh	

Figura 2.6, Memoria de programa.

Vector de Reset.- Cuando ocurre un reset el contenido del PC es forzado a cero, ésta es la dirección donde la ejecución del programa continuará después del reset, por ello se le llama “dirección del vector de reset”.

Vector de interrupción.- Cuando la CPU acepta una solicitud de interrupción ejecuta un salto a la dirección 0004h, por lo cual a esta se le conoce como "dirección del vector de interrupción". El registro PCLATH no es modificado en esta circunstancia, por lo cual habrá que tener cuidado al manipular el PC dentro de la Rutina de Atención a la Interrupción (Interrupt Service Routine o ISR).

Memoria de Stack

La memoria de stack es una área de memoria completamente separada de la memoria de datos y la memoria de programa. La pila (stack) consta de 8 niveles de 13 bits cada uno. Esta memoria es usada por la CPU para almacenar las direcciones de retorno de subrutinas. El apuntador de stack no es ni legible ni escribible. Cuando se ejecuta una instrucción CALL o es reconocida una interrupción el PC es guardado en el stack y el apuntador de stack es incrementado en 1 para apuntar a la siguiente posición vacía. A la inversa, cuando se ejecuta una instrucción RETURN, RETLW o RETFIE el contenido de la posición actual del stack es colocado en el PC.

La Memoria de Datos.

La memoria de datos consta de dos áreas mezcladas y destinadas a funciones distintas:

- Registros de Propósito Especial (SFR)
- Registro de Propósito General (GPR) (o variables)

Los SFR son localidades asociadas específicamente a los diferentes periféricos y funciones de configuración del PIC y tienen un nombre específico asociado con su función. Mientras que los GPR son memoria RAM de uso general.

Bancos de memoria.

Toda la memoria de datos está organizada en 4 bancos (o páginas) enumerados 0, 1, 2 y 3. Para seleccionar un banco se debe hacer uso de los bits del registro STATUS<7:5> denominados IRP, RP1 y RP0.

Hay dos maneras de acceder a la memoria de datos:

Direccionamiento directo e indirecto: La selección de bancos se basa en la tabla 2.

Direccionamiento Indirecto (IRP)	RP1:RP0	Banco
0	00	0
	01	1
1	10	2
	11	3

Tabla 2, Selección de bancos.

Direccionamiento Directo.

Para acceder una posición de memoria mediante direccionamiento directo, la CPU simplemente usa la dirección indicada en los 7 bits menos significativos del código de operación y la selección de banco de los bits RP1:RP0.

Direccionamiento indirecto

Este modo de direccionamiento permite acceder una localidad de memoria de datos usando una dirección de memoria variable a diferencia del direccionamiento directo, en que la dirección es

fija. Esto puede ser útil para el manejo de tablas de datos. Para este tipo de direccionamiento se utiliza el registro INDF y el registro FSR.

***El registro INDF:** Para hacer posible el direccionamiento indirecto se debe usar el registro INDF. Cualquier instrucción que haga un acceso al registro INDF en realidad accesa a la dirección apuntada por el registro FSR (File Select Register). La selección de banco en el caso de direccionamiento indirecto se realiza mediante los bits IRP (STATUS<7>) y el bit 7 del registro FSR. El registro INDF mismo al leerse de manera indirecta (con FSR=0) producirá un cero. Y al escribirse de manera indirecta no es afectado.

La memoria de Datos (o registros).

En ella se guardan los registros de propósito especial y los de propósito general. Es una memoria de tipo RAM y esta distribuida como se muestra en la figura 2.7.

File Address	File Address	File Address	File Address
Indirect addr. ^(*) 00h	Indirect addr. ^(*) 80h	Indirect addr. ^(*) 100h	Indirect addr. ^(*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	105h	185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	107h	187h
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h	108h	188h
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h	109h	189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEADATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh	8Fh	EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h	90h	110h	190h
TMR2 11h	SSPCON2 91h	111h	191h
T2CON 12h	PR2 92h	112h	192h
SSPBUF 13h	SSPADD 93h	113h	193h
SSPCON 14h	SSPSTAT 94h	114h	194h
CCPR1L 15h	95h	115h	195h
CCPR1H 16h	96h	116h	196h
CCP1CON 17h	97h	General Purpose Register 117h	General Purpose Register 197h
RCSTA 18h	TXSTA 98h	16 Bytes 118h	16 Bytes 198h
TXREG 19h	SPBRG 99h	119h	199h
RCREG 1Ah	9Ah	11Ah	19Ah
CCPR2L 1Bh	9Bh	11Bh	19Bh
CCPR2H 1Ch	9Ch	11Ch	19Ch
CCP2CON 1Dh	9Dh	11Dh	19Dh
ADRESH 1Eh	ADRESL 9Eh	11Eh	19Eh
ADCON0 1Fh	ADCON1 9Fh	11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h - 7Fh
7Fh	EFh	16Fh	1EFh
	F0h	170h	1F0h
Bank 0	Bank 1	Bank 2	Bank 3
	FFh	17Fh	1FFh

Figura 5, Memoria de datos.

2.2 Función del PIC en la aplicación.

El PIC 16F877A cumple un papel muy importante en el diseño del proyecto, su principal objetivo es controlar al grupo motor-generator KATO por medio de una interfase compuesta de opto acopladores, relés y contactores. A esto se le suma que nos brinda la posibilidad de comunicarnos con el computador por medio de su puerto USART.

Cuando se enciende la tarjeta que contiene al PIC, se visualizan una serie de mensajes en la pantalla LCD, la cual nos servirá para configurar la tarjeta. Si se elige que se no se quiere usar la comunicación con el computador el PIC procede a preguntar cómo será el arranque del motor, las conexiones de las bobinas del motor, los tiempos para las resistencias (en caso de arranque por resistencias) y el sentido de giro.

Si se elige la comunicación con el computador, el PIC se queda en espera hasta que llegue un dato a través de su puerto de comunicación serial asincrónica (USART) y procederá a ejecutar la acción ordenada.

Otra función importante del PIC es la de verificar la conexión que se ordenó, esto lo hace comparando qué contactores se ordenó que se energicen y cuáles realmente se energizaron, esto lo hace por medio de los contactos auxiliares de los contactores y de una interfase con opto acopladores. Si detecta un error en esta comparación el PIC desenergiza todos los contactores y muestra un mensaje de error en la pantalla LCD o en el computador (en el computador muestra cuál fue el error, es decir qué contactor no se energizó o se energizó de más).

La configuración de tiempos también la realiza el PIC, es decir que estamos usando tiempo real, cuando se ordena un arranque por resistencia o un cambio de giro, el tiempo que cada resistencia esta conectada es configurado y controlado por el PIC.

Para realizar todas las operaciones antes descritas el PIC se vale de un conjunto de recursos que serán explicados a continuación.

2.3 Recursos del PIC 16F877A.

A continuación se explicarán los recursos del PIC que fueron utilizados en el desarrollo e implementación del proyecto:

- ✓ **Puertos de entrada / salida digitales:** El PIC dispone de los siguientes puertos de entrada/salida digitales:

Nombre del Puerto	Pines del Puerto
PORTA (Puerto A)	6 pines (A0 a A5)
PORTB (Puerto B)	8 pines (B0 a B7)
PORTC (Puerto C)	8 pines (C0 a C7)
PORTD (Puerto D)	8 pines (D0 a D7)
PORTE (Puerto E)	3 pines (E0 a E2)

Tabla 3, Puertos del PIC 16F877A

Cada pin puede ser configurado como entrada o salida en cualquiera de los puertos. La configuración de cada puerto para la implementación del proyecto se muestra tabla 4.



CIB -ESPOL

Puerto	Pines	Configuración
PORTA	A0 a A3	Salidas
PORTA	A4 y A5	Entradas
PORTB	B0 a B7	Salidas
PORTC	C0 a C3 y C6 a C7	Entradas
PORTC	C4 y C5	Salidas
PORTD	D0 a D7	Entradas
PORTE	E0 y E1	Entradas
PORTE	E2	Salida

Tabla 4, Configuración de los pines.

Para lograr la configuración que se mostró en la tabla anterior se debe guardar el valor correspondiente en los registros de configuración de los puertos. Cuando se quiere configurar un pin como entrada se coloca "1" y para configurarlo como salida se coloca "0". De esta manera se determinaron los valores para los registros de configuración de cada puerto y se los muestra en la tabla 5.

Registro de Configuración	Valor a cargar (Hex)
TRISA (puerto A)	0x30
TRISB (puerto B)	0x00
TRISC (puerto C)	0xCF
TRISD (puerto D)	0xFF
TRISE (puerto E)	0x03

Tabla 5, Configuración de los puertos de entrada / salida.

La función de cada puerto se muestra a continuación:

Puerto A: Se utilizaron los pines desde A0 hasta A3 como entradas de control desde los contactores para saber si el estado de los contactores es el correcto, A4 es la tecla con la etiqueta "1" en el teclado y A5 es la tecla "2".

Puerto B: Todo el puerto B se utiliza como salida hacia los contactores, sirve para activar los contactores.

Puerto C: Los pines desde C0 hasta C3 se utilizaron como entradas de control desde los contactores al igual que los pines A0 hasta A3 del puerto A. Los pines C4 y C5 modifican el bus de datos o el flujo de los datos hacia la pantalla LCD o hacia la interfase de salida. Por último C6 y C7 se utilizan para la comunicación serial, C6 es para transmisión y C7 para recepción.

Puerto D: Todos los pines se utilizaron como entradas de control desde los contactores.

Puerto E: El pin E0 corresponde a la tecla "3", el pin E1 corresponde a la tecla de cambio de giro y E2 esta libre para salida.

- ✓ **Temporizador:** El PIC 16F877A dispone de tres temporizadores que pueden funcionar en diferentes modos, pero para la aplicación se uso solo un temporizador (TMR0). El TMR0 puede funcionar como contador o temporizador de 8 bits con pre-escalador. En la figura 6 se muestra el diagrama de bloques del módulo temporizador TMR0.

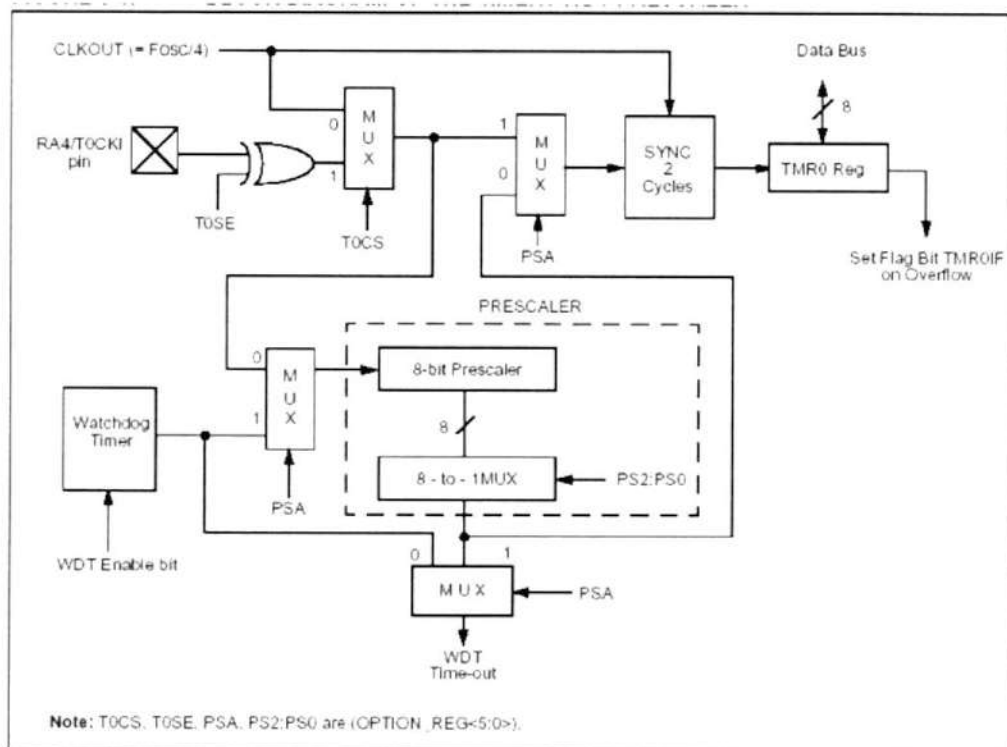


Figura 2.8, Diagrama de bloques de TMR0

Para indicar cómo debe funcionar el temporizador TMR0 se debe configurar el registro OPTION_REG:

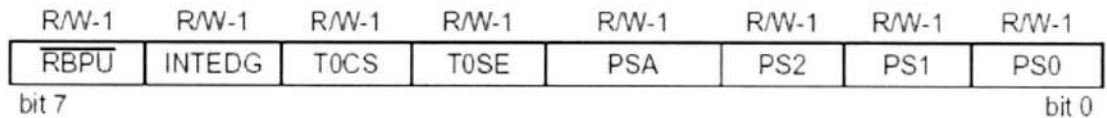


Figura 2.9, Registro OPTION_REG

Los bits PS0, PS1 y PS2 determinan el valor del pre escalador, en la aplicación se les dio el valor "1" a los tres para que el pre escalador sea de 1:128. El bit PSA se lo puso en "0" para que el pre escalador funcione con TMR0 y no con el perro guardián. El bit TOSE determina que flanco de reloj se usará para la temporización, se seleccionó "0" para usar el flanco de subida. El bit TOCS se colocó en "0" para que use el cristal que se conectó al PIC, los bits restantes no tienen importancia y se los colocó en cero. Por lo tanto el valor que se cargó en este registro es: 0x07 Hex.

- ✓ **Comunicación serial:** El PIC ofrece un módulo de comunicación serial llamado USART (Universal Synchronous Asynchronous Receiver Transmitter), también se lo conoce con el nombre de SCI (Serial Communication interface). Este módulo puede funcionar a diferentes frecuencias o baudios. Para la aplicación se definió que el

módulo funcionará a 9600 baudios, 8 bits de datos, 1 bit de parada, sin paridad ni control de flujo.

A través de el USART se comunicará al PIC con el computador, pero para que esto ocurra se necesita añadir un convertidor de TTL a RS-232 que es la interfase que usa el computador. Se utilizó el convertidor MAX-232, que será explicado más adelante.

Para configurar al USART se deben tomar en cuenta algunos registros que lo manejan y determinan su funcionamiento:

- INTCON (si se desea trabajar con interrupciones)
- PIR1 (para habilitar la interrupción asociada al USART)
- RCSTA (configuración del receptor del USART)
- TXSTA (configuración del transmisor del USART)
- PIE1 (indica las interrupciones ocurridas)
- SPBRG (generador de baudios)
- RCREG (búfer de recepción)

El diagrama de bloques del transmisor del USART se muestra en la figura 2.10.

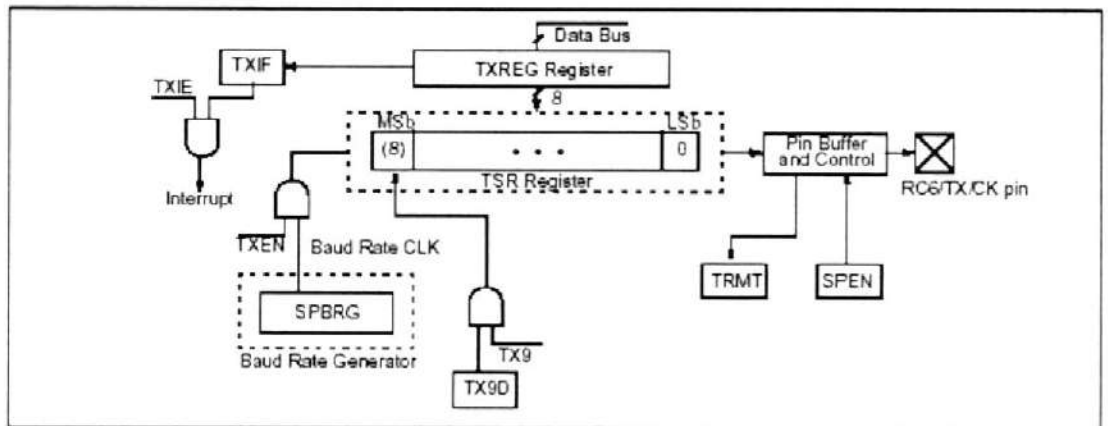


Figura 2.10, Diagrama de bloques del trasmisor del USART

El receptor del USART esta conformado como se muestra en la figura 2.11.

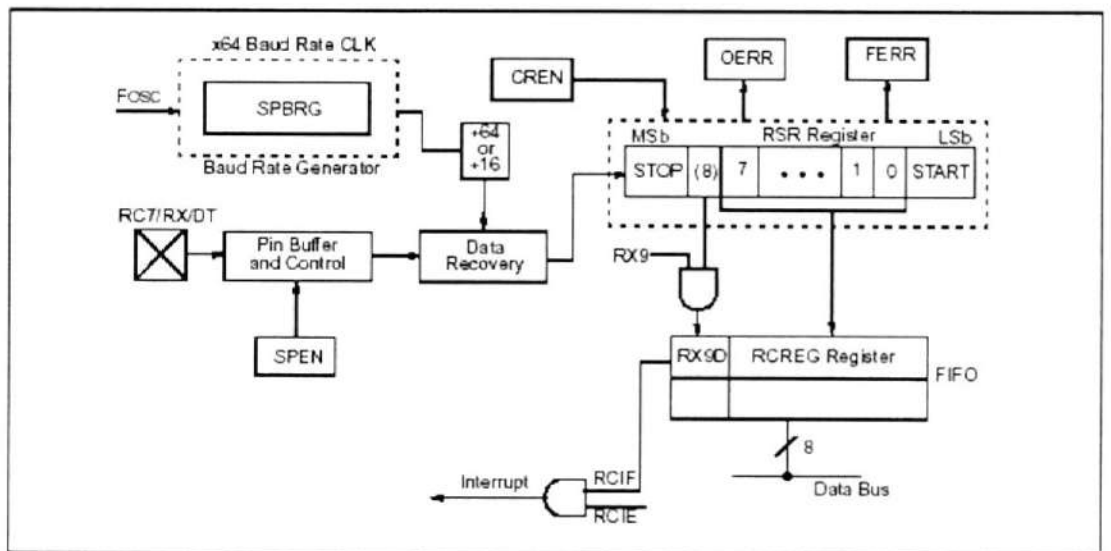


Figura 2.11, Diagrama de bloques del receptor del USART.

El funcionamiento del USART depende de los registros antes mencionados, a continuación se explica la configuración de cada uno de ellos de acuerdo a las necesidades de la aplicación:

INTCON: Este registro se lo explicará más adelante porque corresponde a otro recurso del PIC que son las interrupciones.

PIE1: La función de este registro es la de enmascarar o desenmascarar las interrupciones de algunos de los módulos del PIC; en la aplicación se decidió que la recepción de información por medio del USART sería por interrupción, por lo tanto se desenmascaró la interrupción correspondiente a la recepción del USART (el bit 5 se lo coloca en "1"); ya que no se usarán más interrupciones el resto se enmascararon colocando los bits restantes en cero. Por lo tanto el valor cargado a este registro es: 20 Hex.

PIR1: Cada vez que ocurre una interrupción, la bandera o indicador correspondiente al módulo que causó la interrupción

se pone en "1" en este registro. Por lo tanto se puede identificar al módulo interrumpió leyendo el este registro, el bit asociado a la interrupción por recepción del USART es el 5, cuando se lee el dato recibido esta bandera se coloca en cero automáticamente.

RCSTA: Configura la recepción del USART.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Figura 2.12, Registro RCSTA

El bit SPEN habilita el puerto serial, el bit RX9 habilita la recepción de 9 bits de datos, el bit SREN sirve para recibir en cualquier momento o de forma continua y se lo colocó en uno para usar la recepción continua, el bit ADDEN sirve para darle dirección al USART en caso de ser usado como esclavo. El resto de bits no se utilizaron en la aplicación y se colocaron en cero.

RCREG: Cuando al receptor del USART llega un dato, el dato se guarda en éste registro. Este registro debe ser leído y guardar el valor en otro registro (o variable) para salvarlo y dejar listo al USART para recibir nuevamente.

TXSTA: Configura la transmisión del USART.

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

Figura 2.13, Registro TXSTA.

Los bits que fueron usados son TXEN para habilitar la transmisión, el bit SYNC para indicar que es comunicación asincrónica (se lo colocó en “0”) y el bit BRGH en “1” para indicar velocidad de transmisión alta.

SPBRGH: Este registro sirve para generar la frecuencia de baudios deseada para la comunicación. El valor a cargar en este registro depende del cristal que se está usando (10 MHz en la aplicación) y de la frecuencia de baudios que se desea, los valores que puede tomar este registro están tabulados y el

valor indicado para la comunicación a 9600 baudios con un cristal de 10 MHz es 64 (decimal).

- ✓ **Interrupciones:** El PIC 16F877A tiene 15 diferentes fuentes de interrupción. Para configurar las interrupciones se manipulan los registros INTCON, PIE1 y PIE2. Para esta aplicación se utilizará solo una interrupción, la que le corresponde a la recepción del USART, es decir que cuando el búfer de recepción del USART está lleno se produce una interrupción, esto causa que el PIC busque el vector de interrupción (0004H) y salte a la rutina de procesamiento de la interrupción, que no es más que una parte del programa que se encarga de procesar la interrupción para después regresar al programa principal. Cuando se produce la interrupción de la recepción del USART, la bandera RXIF se pone en "1" en el registro PIR1 (bit 5). Para desenmascarar la interrupción de la recepción del USART se debe colocar en "1" el bit 7 de INTCON (GIE, Global interrupt enable) que permite desenmascarar las interrupciones de cada módulo. Luego se debe poner en "1" el bit 5 de PIE1 para desenmascarar la interrupción asociada a la recepción del USART.

2.4 Detalles importantes del diseño con el PIC.

Durante el desarrollo de la aplicación, específicamente en el diseño del circuito con el PIC se presentaron algunos problemas que me llevaron a incluir este comentario. A continuación expongo algunas sugerencias o explico las soluciones que hallé a los diferentes inconvenientes que se presentaron:

- ☆ **Ruido:** El PIC maneja en el circuito de control de las máquinas KATO doce contactores por medio de una interfase opto acoplada, al desconectarse los contactores para ejecutar una acción diferente se producía un transiente que originaba que el PIC ejecute un Reset y pierda la información o la secuencia que tenía que ejecutar. Para resolver este problema se colocaron capacitores de 0.01 uF cerca de la alimentación del PIC para absorber dichos transientes; por lo tanto se recomienda colocar capacitores cerca de la alimentación del PIC cuando se encuentre en ambientes ruidosos.

- ☆ **Circuitos Impresos:** Recomiendo que el diseño en Proto Board sea probado por un tiempo no muy prolongado y luego hacer circuitos impresos, ya que el PIC es muy sensible a el pobre contacto que hacen los cables en el proto board, lo cual ocasiona muchos problemas.

- ☆ **Cable correcto:** Se recomienda prestar atención al cable que se usa en la comunicación serial, ya que no todos los cables que se encuentran son iguales pin a pin. En los anexos se encontrará una descripción del cable adecuado.

2.5 Programación del PIC.

En este punto se mostrarán las instrucciones o comandos utilizados en la programación del PIC 16F877A, el cual pertenece a la familia de los microcontroladores RISC, por lo tanto solo posee 35 instrucciones para su programación, las cuales se muestran en la figura 2.14.

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111	dfff ffff	C,DC,Z 1,2
ANDWF	f, d	AND W with f	1	00 0101	dfff ffff	Z 1,2
CLRF	f	Clear f	1	00 0001	1fff ffff	Z 2
CLRW	-	Clear W	1	00 0001	0xxx xxxx	Z
COMF	f, d	Complement f	1	00 1001	dfff ffff	Z 1,2
DECf	f, d	Decrement f	1	00 0011	dfff ffff	Z 1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011	dfff ffff	1,2,3
INCF	f, d	Increment f	1	00 1010	dfff ffff	Z 1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111	dfff ffff	1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100	dfff ffff	Z 1,2
MOVF	f, d	Move f	1	00 1000	dfff ffff	Z 1,2
MOVWF	f	Move W to f	1	00 0000	1fff ffff	
NOF	-	No Operation	1	00 0000	0xxx 0000	
RLF	f, d	Rotate Left f through Carry	1	00 1101	dfff ffff	C 1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100	dfff ffff	C 1,2
SUBWF	f, d	Subtract W from f	1	00 0010	dfff ffff	C,DC,Z 1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110	dfff ffff	1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110	dfff ffff	Z 1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb	bfff ffff	1,2
BSF	f, b	Bit Set f	1	01 01bb	bfff ffff	1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01 10bb	bfff ffff	3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01 11bb	bfff ffff	3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x	kkkk kkkk	C,DC,Z
ANDLW	k	AND literal with W	1	11 1001	kkkk kkkk	Z
CALL	k	Call subroutine	2	10 0kkk	kkkk kkkk	
CLRWDf	-	Clear Watchdog Timer	1	00 0000	0110 0100	TO,PD
GOTO	k	Go to address	2	10 1kkk	kkkk kkkk	
IORLW	k	Inclusive OR literal with W	1	11 1000	kkkk kkkk	Z
MOVLW	k	Move literal to W	1	11 00xx	kkkk kkkk	
RETFIE	-	Return from interrupt	2	00 0000	0000 1001	
RETLW	k	Return with literal in W	2	11 01xx	kkkk kkkk	
RETURN	-	Return from Subroutine	2	00 0000	0000 1000	
SLEEP	-	Go into Standby mode	1	00 0000	0110 0011	TO,PD
SUBLW	k	Subtract W from literal	1	11 110x	kkkk kkkk	C,DC,Z
XORLW	k	Exclusive OR literal with W	1	11 1010	kkkk kkkk	Z

Figura 2.14, Instrucciones para la programación del PIC.

Otro aspecto importante en la programación del PIC es que se usó el software MPLAB IDE v.6.40 de Microchip, el cual es un compilador que permite el uso de macros. En el desarrollo del programa se utilizaron algunas macros las cuales son pequeñas subrutinas que se colocan fuera del programa principal, las cuales mostraré más adelante.

CAPITULO 3

3. HARDWARE DE ADQUISICIÓN DE DATOS.

3.1 Descripción del sistema de adquisición de datos.-

El sistema de adquisición de datos tiene como principal objetivo adquirir, acondicionar y convertir las señales eléctricas en valores numéricos que sirvan para el análisis de dichas señales y el cálculo de parámetros basados en dichas mediciones.

El sistema de adquisición de datos está formado por varios elementos:

- 1.- Sensores o Transductores.
- 2.- Acondicionador de Señal.
- 3.- Tarjeta de Adquisición de datos.



CIB -ESPOL

Sensores.- Se utilizaron pocos sensores, uno para la velocidad (tacogenerador) y uno para cada corriente de línea (shunts).

Acondicionador de Señal.- Está formado por un conector SCXI-1327, un chasis SCXI-1000 y un módulo de entradas análogas SCXI-1120. La función del acondicionador de señal es filtrar el ruido y convertir las señales eléctricas a voltajes bajos que puedan ser aceptados por la tarjeta de adquisición de datos. Cada parte del acondicionador de señal tiene una función diferente que serán explicadas más adelante.

Tarjeta de Adquisición de datos.- Es el elemento encargado de muestrear las ondas y convertirlas a digital para su posterior análisis. La tarjeta de adquisición de datos que se usa en la aplicación es la PCI-6024E. En la figura 3.1 se muestra la estructura del sistema de adquisición de datos.

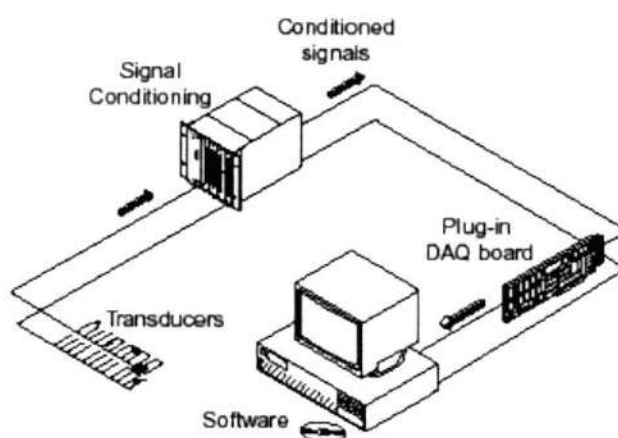


Figura 3.1, Estructura del sistema.

3.2 Características y Configuración del Conector SCXI-1327.-

Características:

- 1.- Es un conector de 8 canales.
- 2.- Posee atenuadores de voltaje de 100:1
- 3.- Posee un sensor de temperatura incorporado.
- 4.- Voltaje máximo entre canales 300 Vrms.

Configuración: Cada canal posee un atenuador de voltaje de 100:1, los cuales se pueden habilitar o deshabilitar moviendo los switches que se encuentran a la entrada de las señales. En la aplicación se

habilitaron los atenuadores de los canales correspondientes a los voltajes de línea (canal 0, canal 1 y canal 7) y el resto de atenuadores se deshabilitaron de manera que en los demás canales las señales pasan de 1:1, o sea sin atenuación.

Las señales que se asignaron a cada canal se muestran la tabla 6.

CANAL	SEÑAL	ATENUACIÓN
Cero	Voltaje AB	100 a 1
Uno	Voltaje BC	100 a 1
Dos	Corriente A	1 a 1
Tres	Corriente B	1 a 1
Cuatro	Corriente C	1 a 1
Cinco	Velocidad	1 a 1
Seis	Voltaje DC	100 a 1
Siete	Voltaje CA	100 a 1

Tabla 6, Asignación de señales.

Este conector o bloque terminal es el enlace entre los sensores y el módulo de acondicionamiento de señales SCXI-1120.

La figura 3.2 ilustra el conector SCXI-1327 por dentro.

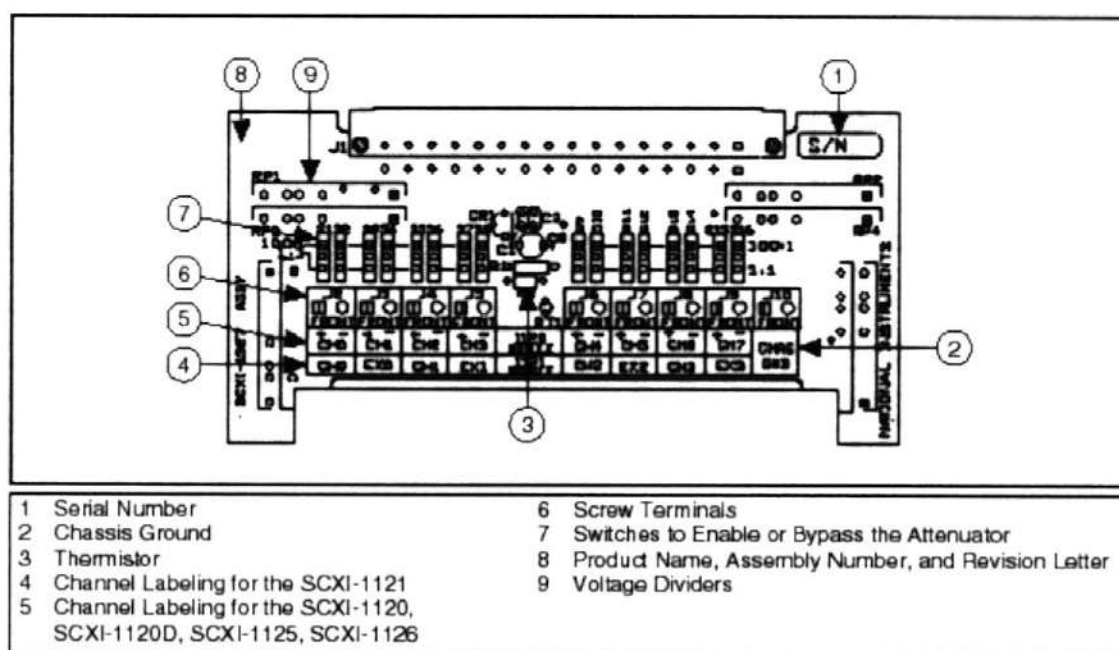


Figura 3.2, Conector SCXI-1327

3.3 Características y Configuración del Chasis SCXI-1000.-

Características:

- 1.- Es un chasis de 4 ranuras.
- 2.- Se alimentan con 120 Vac.
- 3.- Suministra un ambiente bajo en ruido para el acondicionamiento de señales.
- 4.- Sirve para los módulos antiguos y futuros.

Configuración: Este chasis sirve para la conexión de hasta 4 módulos acondicionadores de señal, lo cual nos permite ampliar la cantidad y variedad de entradas que tiene la tarjeta de adquisición de datos. El chasis posee una dirección que puede ser cambiada (es configurable) para que el sistema pueda identificarlo, se pueden conectar varios chasis con varios módulos acondicionadores para estructurar un sistema más grande y complejo donde cada chasis tiene una dirección y cada módulo de cada chasis también posee una dirección, por ejemplo: chasis 3 módulo 2. Para la aplicación sólo se usa un chasis con un módulo, por lo tanto no se necesitan asignar direcciones específicas.

3.4 Características y Configuración del módulo SCXI-1120.-

Características:

- 1.- Posee 8 canales de entrada aisladas.
- 2.- Acondiciona señales de diferentes tipos:
 - a.- Termocuplas.
 - b.- Fuentes de voltios y milivoltios.
- 3.- Dos modos de trabajo: Paralelo y multiplexado.
- 4.- Posee filtros configurables para cada canal.
- 5.- Ganancias configurables para cada canal (desde 1 hasta 2000).

Configuración: El módulo acondicionador SCXI-1120 tiene varios jumpers de configuración, algunos son para configurar las ganancias de cada canal, otros para la frecuencia del filtro en cada canal y otros para la operación del módulo en conjunto con otros dispositivos. En la tabla 7 se muestra la ganancia de cada canal y la posición correcta de los jumpers para lograr dicha ganancia.

CANAL	POSICIÓN JUMPER 1	POSICIÓN JUMPER 2	GANANCIA TOTAL
Vab (0)	D	B	2
Vbc (1)	D	B	2
Vca (7)	D	B	2
Ia (3)	C	A	10
Ib (4)	C	A	10
Ic (5)	C	A	10
Sp (2)	D	A	1
VDC (6)	D	C	5

Tabla 7, Ganancias de los canales.

Todos los canales tienen la misma configuración de los filtros, todos los jumpers de configuración de los filtros están en la posición B.

3.5 Características de la Tarjeta de adquisición de datos PCI-6024E.-

La tarjeta de adquisición de datos PCI-6024E fabricada por National Instrument trabaja en slot PCI, posee 8 canales de entradas análogas (configuradas en modo diferencial), 2 canales de salidas análogas y 8 líneas de entrada/salida digital.

Para la aplicación que se desarrolló se utilizaron todas las entradas análogas, algunas líneas digitales (solo para la comunicación con el chasis y el módulo acondicionador de señales) y ninguna salida análoga.

Debido a que el sistema contiene un acondicionador de señales, los controladores de National Instrument detectan el chasis y el módulo acondicionador y configuran automáticamente la tarjeta de adquisición de datos basándose en el tipo de chasis y módulo que está presente en el sistema.

La tarjeta de adquisición de datos es capaz de muestrear varias señales analógicas al mismo tiempo, con lo que se gana sincronismo. Para configurar las muestras que se toman se debe tomar en cuenta el criterio de Nyquist que será expuesto más adelante, en base a éste

criterio se configuran cuántas muestras se quieren tomar y cuántas muestras por segundo se quieren tomar.

La tarjeta de adquisición de datos posee en su parte posterior un conector de 68 pines, en el caso de ésta aplicación en particular se conecta la tarjeta PCI-6024E con el chasis SCXI-1000 a través de éste conector.

Si no se hubiesen usado el conector SCXI-1327, el chasis SCXI-1000 y el módulo SCXI-1120, la tarjeta PCI-6024E podría recibir directamente señales analógicas, pero en un rango de +10 a -10 Voltios. Gracias a los otros elementos mencionados ganamos simplicidad en el diseño para la adquisición de las señales.

CAPITULO 4

4. INSTALACIÓN DE LOS CONTROLADORES (DRIVERS) Y CONFIGURACIÓN DEL SISTEMA

4.1 Instalación de los controladores.-

Para el correcto funcionamiento del sistema se debe hacer instalar y configurar el sistema de forma apropiada. Se incluyó este capítulo por la necesidad de saber qué variables se deben configurar y cómo se debe instalar la tarjeta de adquisición de datos en caso de una nueva instalación del sistema en otras máquinas.

Como punto de partida menciono algo de suma importancia: Se debe instalar primero LabView, luego los controladores (drivers) del hardware de National Instruments y al final se instalan la tarjeta de adquisición de datos, el chasis y el módulo de acondicionamiento de señales; si se altera éste orden el sistema puede funcionar pero el computador se

tornará más lento y ciertas características de programación de LabView se pueden perder.

El software de monitoreo y control utilizado para la implementación es el LabView 7 Express de National Instruments, la instalación se la realiza desde el CD de instalación, no hay opciones en la instalación de LabView 7 Express; simplemente se inserta el CD y se autoriza la instalación.

Los controladores del hardware (drivers) de National Instrument se encuentran en otro CD, esta instalación tiene muchas opciones y a continuación se mostrarán las opciones mínimas que se deben elegir para obtener el funcionamiento adecuado del sistema:

1.- En el menú de Data Acquisition se deben instalar:

1.1.- Traditional NI-DAQ 7.0

1.1.1.- LabView 7 Support

1.2.- NI-DAQmx 7.0

1.2.1.- LabView 7 Support

2.- En el menú de Instrument Control:

2.1.- NI-VISA 3.0

2.2.- IVI Compliance Package 2.0

2.2.1.- IVI Class Drivers for LabView 7

2.3.- NI Instrument I/O Assistant 1.0

3.- NI Measurement and Automation Explorer 3.0

4.2 Generalidades de los controladores.-

Los controladores o drivers son programas que sirven de enlace entre el computador y el hardware conectado a éste. Los drivers le informan al sistema operativo cómo puede acceder al hardware y cómo utilizar los recursos del mismo.

Los drivers para la tarjeta de adquisición de datos PCI-6024E se encuentran en una librería de National Instruments que se llama NI-DAQ.

NI-DAQ es una extensa librería de funciones que se pueden utilizar desde el lenguaje de programación que se está usando (LabView 7 en este caso), estas funciones permiten utilizar todos los recursos de la tarjeta.

NI-DAQ direcciona o configura parámetros complejos que existen entre el computador y la tarjeta (por ejemplo las interrupciones). En la figura 4.1 se ilustra la relación del driver NI-DAQ con el resto del computador, se observa claramente que si no existiera el NI-DAQ no habría comunicación con la tarjeta de adquisición de datos.

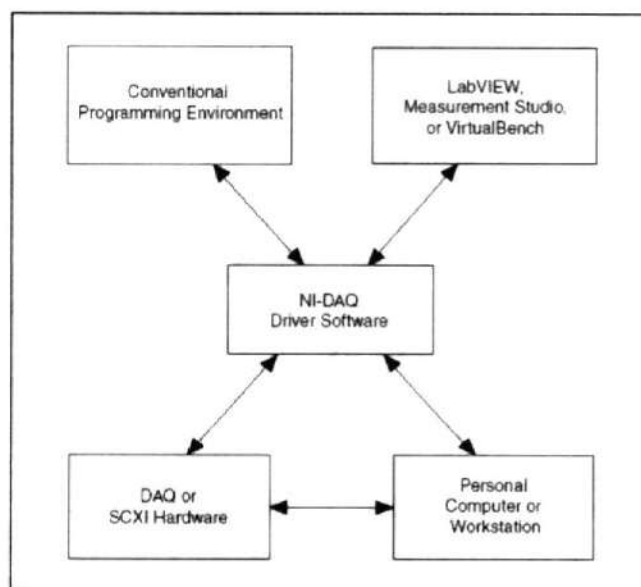


Figura 4.1, NI-DAQ driver.

4.3 Canales Virtuales (Virtual Channels).-

4.3.1 Generalidades.

Los Canales Virtuales son la forma más sencilla de tomar mediciones en los diferentes canales de la tarjeta de adquisición de datos, estos canales virtuales establecen la comunicación entre LabView y la tarjeta de adquisición de datos de tal forma que con sólo asignarle un nombre a una entrada se puede tomar mediciones de ella escribiendo su nombre en una función en LabView.

Los canales virtuales (o virtual channels) se los crea, edita y configura en el programa Measurement and Automation Explorer (MAX), el cual se instala junto con los controladores de hardware de National Instrument como se mencionó anteriormente.

Cuando se configuran los canales virtuales se incluye en la configuración varios parámetros como son el tipo de señal a adquirirse (corriente, voltaje, etc.), el rango de la señal y la escala que se quiere, el dispositivo mediante el cual se hará la medición y algunas otras, por lo tanto cuando se ingresa el nombre de un

canal en una función de LabView no se deben ingresar los parámetros antes mencionados porque ya están dentro de la configuración del canal, de esta manera se hace más fácil y práctica la programación.

IMPORTANTE: SE DEBEN CREAR Y CONFIGURAR LOS CANALES VIRTUALES ANTES DE EJECUTAR LA APLICACIÓN.



CIB -ESPOL

4.3.2 Configuración.

Para crear los canales virtuales que necesita la aplicación para adquirir las señales de voltaje, corriente y velocidad, se ejecuta el Measurement and Automation Explorer (MAX) y se siguen los siguientes pasos:

- 1.- Presionar el botón derecho del ratón sobre la etiqueta "Data Neighborhood" y se selecciona la opción "Create New", luego se selecciona "Traditional NI-DAQ Virtual Channel" y luego se presiona "Finish".

2.- Se abre una ventana y se selecciona "Analog Input" como tipo de entrada (todas las entradas que se necesitan son de este tipo), luego se presiona "Siguiente".

3.- Se le da un nombre al canal (o a la entrada analógica) y se hace un comentario sobre ella si se desea, luego se presiona "Siguiente".

4.- Se debe seleccionar qué tipo de señal se va a recibir, para la aplicación todas las señales son de voltaje, por lo tanto se selecciona "Voltage", y se presiona "Siguiente".

5.- Se debe ingresar la unidad de medición y el rango de la señal, para la aplicación se ingresa "Voltios" como la unidad de medición y los valores que se deben ingresar como rango se muestran en la tabla 7, luego se presiona "Siguiente".

6.- La siguiente ventana pide información sobre qué escala se le aplicará a la señal que se está adquiriendo, para la

aplicación se selecciona "Map Ranges" y luego se ingresan los valores de la escala para la señal correspondiente al canal que se está configurando, los valores para cada entrada se mostrarán en la tabla 8, luego se presiona "Siguiente".

7.- Se debe ingresar el dispositivo que se usará para la adquisición de datos (PCI-6024E en este caso), luego se selecciona el número de canal que se quiere asignar a la señal que se está configurando y se selecciona "Differential" como la forma en que la señal entra a la tarjeta, la asignación de los canales se mostrará más adelante, luego se presiona "Finalizar". El canal virtual se crea automáticamente con la configuración que se ha ingresado.

<u>Nombre del Canal</u>	<u>Rango de Voltaje</u>	<u>Escala de la señal</u>	<u>Canal Número:</u>
Vab	-3.6 a 3.6 V	-7.2, 7.2	0
Vbc	-3.6 a 3.6 V	-7.2, 7.2	1
Vca	-3.6 a 3.6 V	-7.2, 7.2	7
Ia	-5 a 5 V	-10, 10	3
Ib	-5 a 5 V	-10, 10	4
Ic	-5 a 5 V	-10, 10	5
Sp	-6 a 6 V	-1800,1800	2
Vdc	-1.25 a 1.25 V	-125, 125	6

Tabla 8, Configuración de los Canales Virtuales

CAPITULO 5

5. DESARROLLO DE LA APLICACIÓN.

5.1 Tarjeta de Control del Grupo Motor-Generador KATO.

5.1.1. Descripción y objetivo de la tarjeta.

La tarjeta de control tiene como principal objetivo tomar las decisiones o seguir las rutinas necesarias para realizar las diferentes conexiones del motor.

Las decisiones o rutinas las toma un microcontrolador PIC-16F877A, el cual fue explicado detalladamente en el capítulo 2 y la interfase de fuerza está compuesta por optoacopladores,

relés y contactores. El microcontrolador también tiene como tarea detectar cualquier error en la conexión del motor debido a alguna falla en los contactores, para esto utiliza una interfase de entrada al microcontrolador compuesta de optoacopladores y resistencias, de esta forma se evita que ingrese ruido al PIC.

La interfase de fuerza como se dijo está compuesta de optoacopladores, relés y contactores con bobinas de 220 Voltios; para lograr las diferentes conexiones del motor trifásico KATO se necesitaron 11 contactores, con los cuales se puede conectar al motor en Ye serie, Ye paralelo, Delta serie y Delta Paralelo, también se pueden conectar 2 resistencias en cada fase para un rotor devanado, también se pueden conectar 2 resistencias por fase en el estator para los arranques con resistencia o para los cambios de giro y parada. Las resistencias utilizadas son de 12 ohmios y soportan hasta 10 Amperios cada una y se colocaron en boquillas. Los contactos auxiliares de cada contactor se utilizan para indicarle al PIC que el contactor está energizado o no, esto se logra con un voltaje de control de 12 Voltios que después de pasar por la interfase de entrada al PIC se convierte en 5 Voltios o 0 voltios

dependiendo del estado del contactor. Se utilizó el voltaje de 12 Voltios porque experimentalmente se notó que era más inmune al ruido que 5 Voltios.

La tarjeta de control tiene 2 modos de operación: sin conectarse a la computadora y conectándose a la computadora. Cuando la tarjeta funciona por sí sola (no se conecta al computador) solo sirve para controlar al motor, no sirve para monitoreo alguno. En este modo de operación se pueden realizar los arranques en cualquier configuración de conexión con resistencias conectadas al estator para disminuir el voltaje de alimentación y cambios de giro.

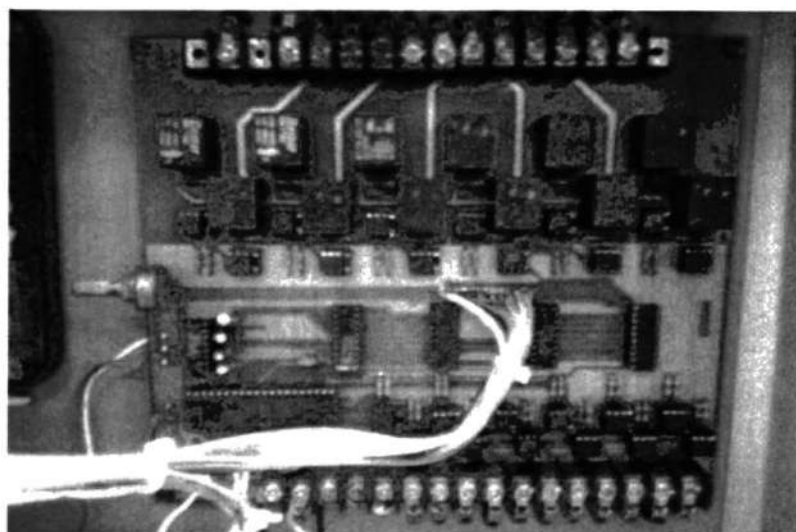
Si se elige el modo de operación con la computadora, la tarjeta espera que el computador le mande la información necesaria para funcionar. En este modo se pueden hacer los arranques antes mencionados, cambios de giro, cambios de conexión del motor mientras está funcionando, parada, los ensayos para determinar el circuito equivalente, conectar resistencias al rotor devanado y monitorear las señales en las pantallas de

LabView. Si ocurre algún error en la conexión de los contactores el PIC enviará un reporte de errores al computador en el que se indica el o los errores encontrados.

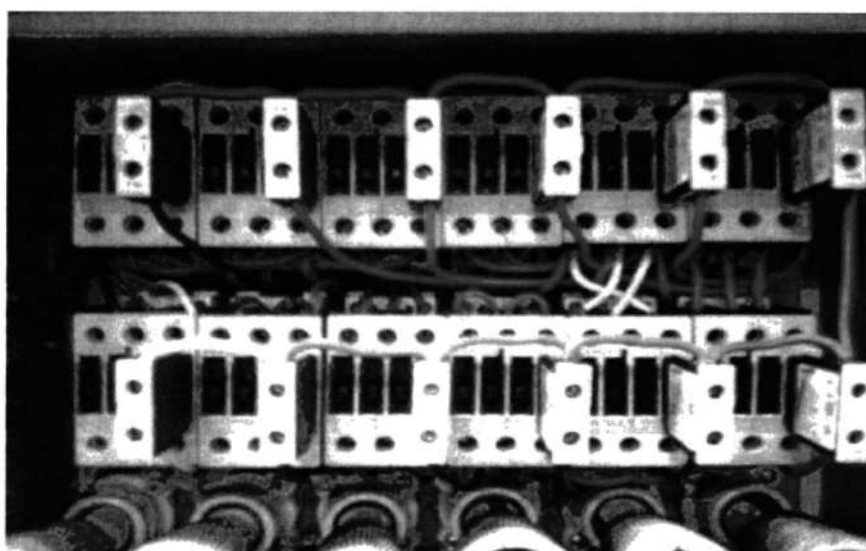
La tarjeta de control se comunica con el computador por comunicación serial asincrónica con norma RS-232. Por medio de esta norma el PIC recibe y envía información al computador. Desde el computador se pueden configurar los tiempos para conectar las resistencias en los arranques y el tiempo para conectar la resistencia de cambio de giro, también se conectan o se desconectan en cualquier momento las resistencias conectadas al rotor devanado.

Cuando la tarjeta se utiliza sin la conexión al computador se configuran todas las opciones por medio de 5 teclas y una pantalla de cristal líquido (LCD) ubicada en un costado del panel. Las instrucciones para la configuración de la tarjeta se explicarán en el capítulo 6.

A continuación se muestra una foto del sistema de control y fuerza:



Sistema de control

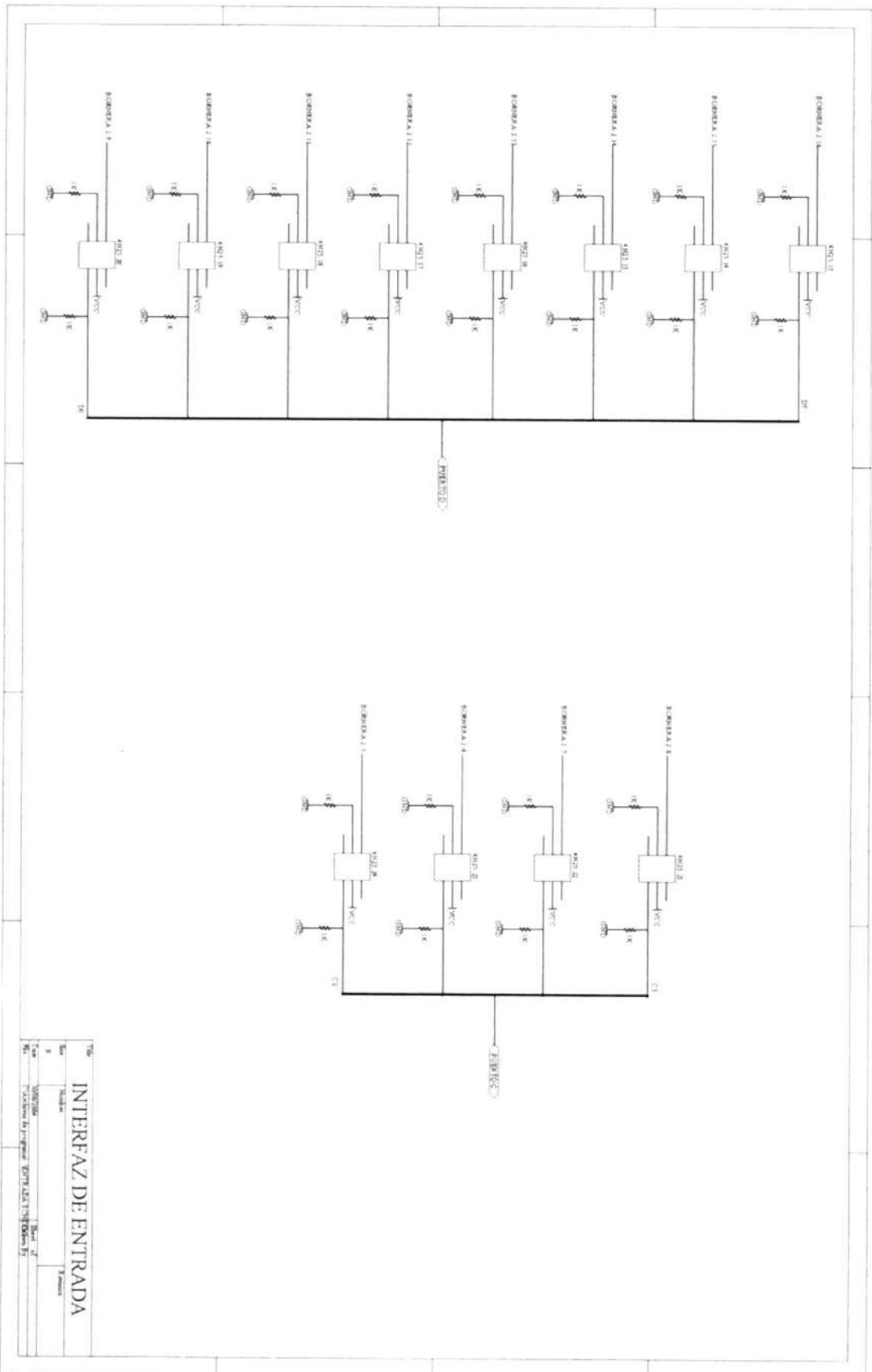


Sistema de fuerza

Figura 5.1, Sistema de Control y Fuerza.

5.1.2. Esquemático de la tarjeta.

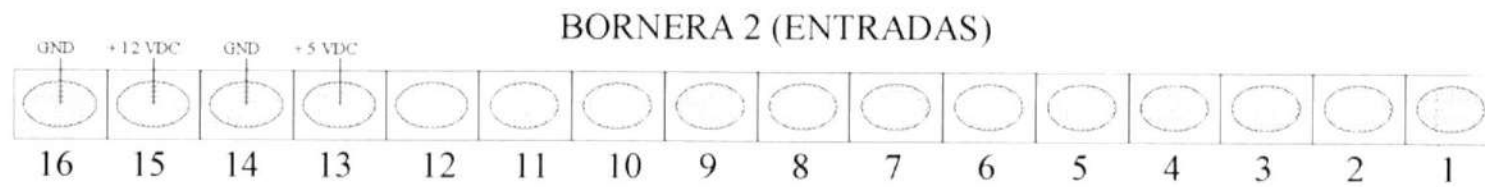
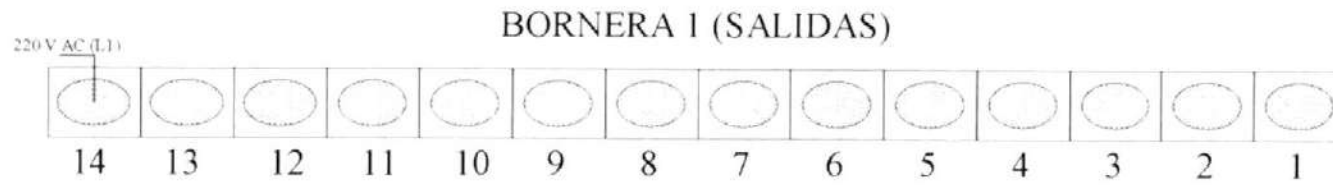
A continuación se muestra el esquemático de la tarjeta de control y de la interfase de fuerza.



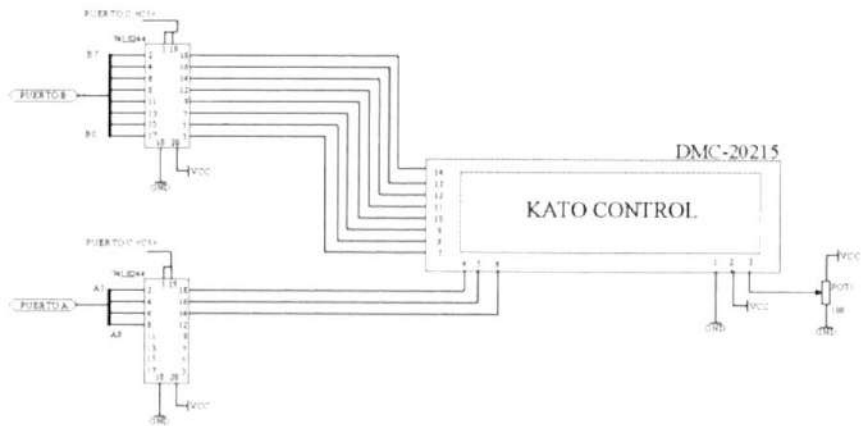
Título: INTERFAZ DE ENTRADA
Autor: [Blank]
Fecha: [Blank]
Tipo: [Blank]
Nº: [Blank]
Código de Proyecto: [Blank]
Código de Documento: [Blank]



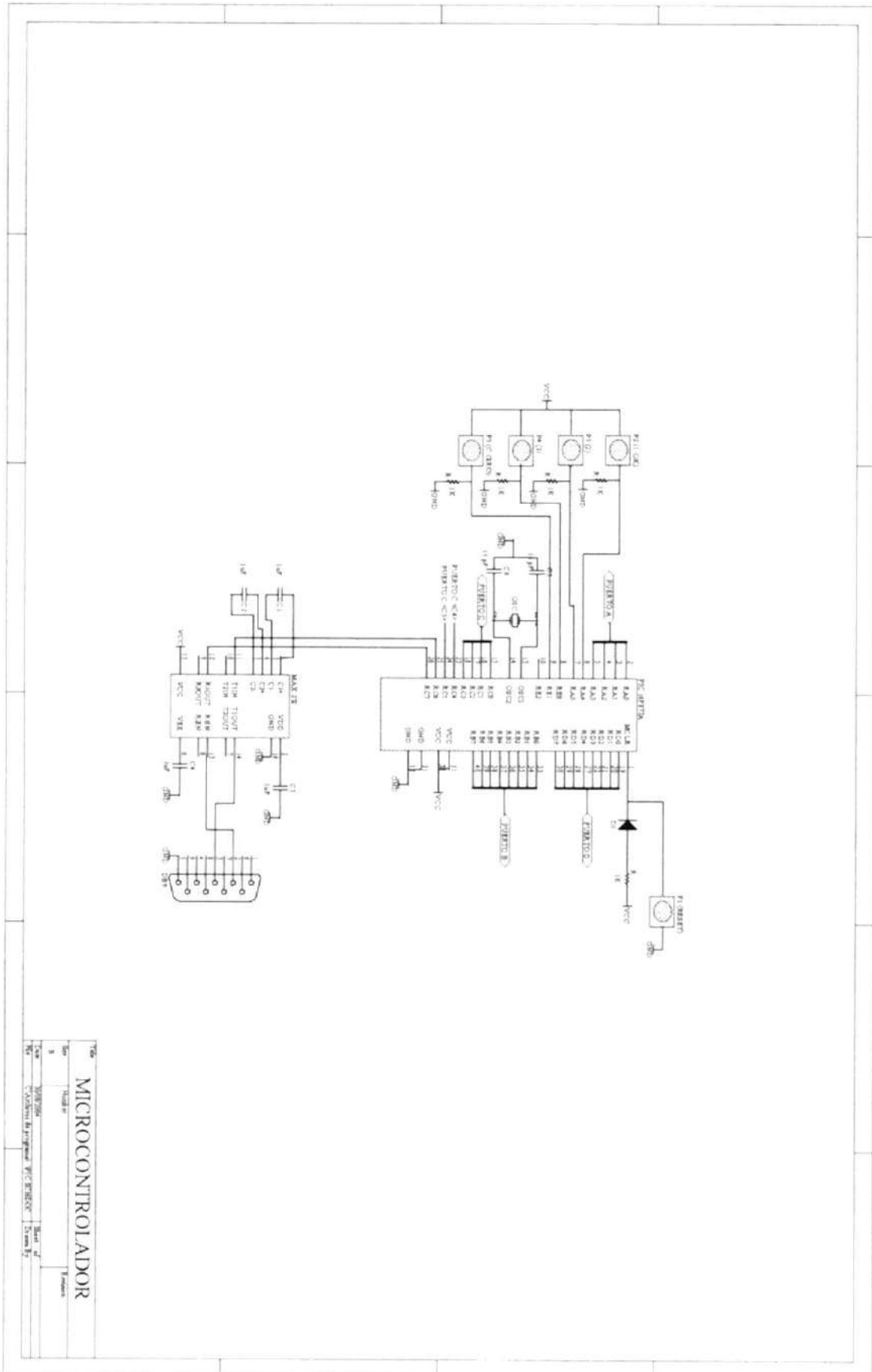
CIB-ESPOL



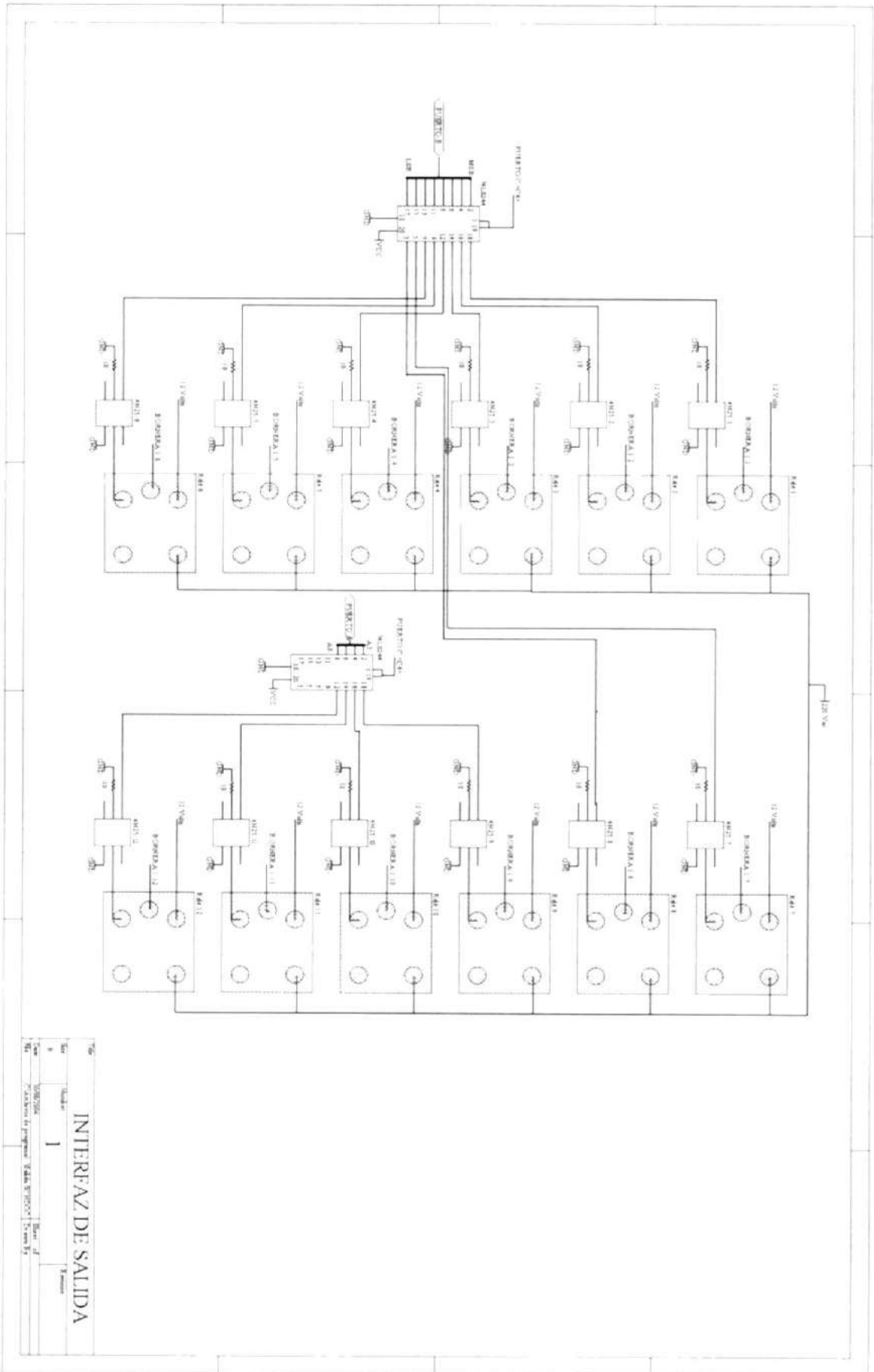
BORNERAS			
Esc	Hoja	Rev	
	1		
Esc	10/01/2004	Rev	
Elab	Elaborado por: INGENIERO EN SISTEMAS DE COMPUTACIÓN		Revisado por:



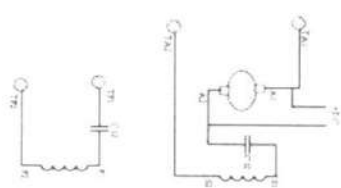
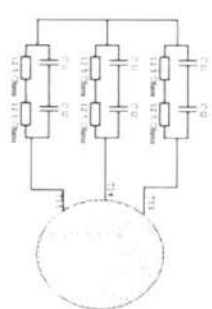
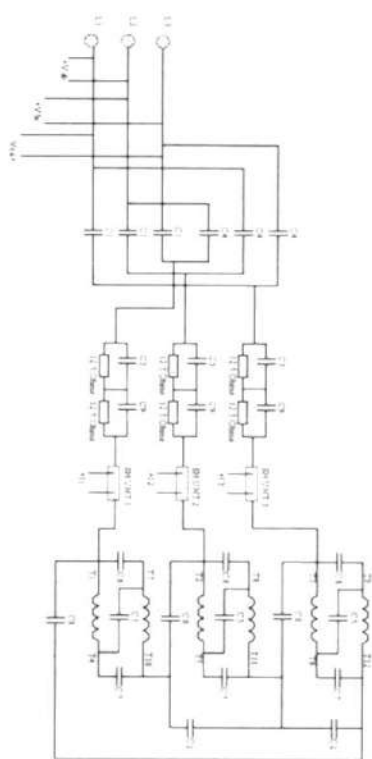
T6		
INTERFAZ DE SALIDA		
Sr	Modelo	Revision
1	2	
Esc	MR/2004	Blow 27
Re	Elaborado en programa: EPC ENDEDOR	
		Tramite



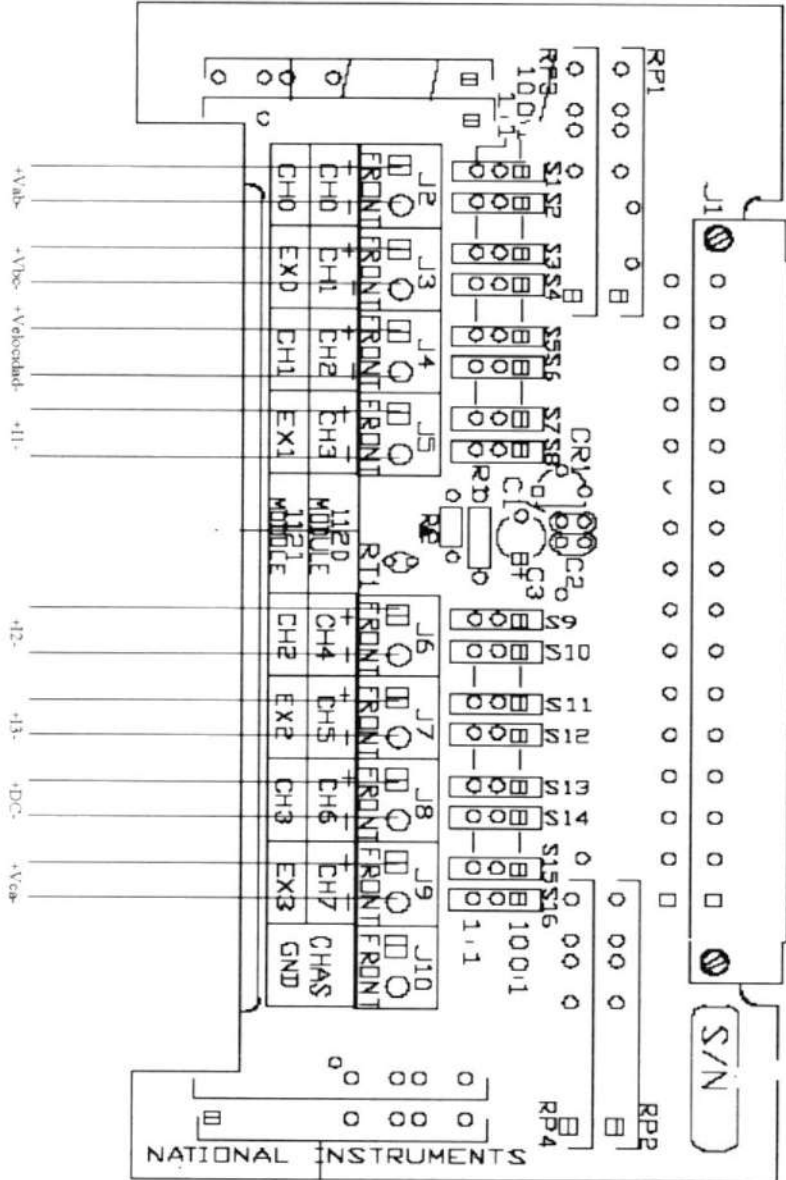
Título		MICROCONTROLADOR
Autores		
Curso	SISTEMAS DE COMPUTADORES I	
Fecha de impresión: 10/10/2023 10:54:37 AM		
Página 83 de 100		



INTERFAZ DE SALIDA	
Fig. No.	1
Proyecto	Zona de Ingeniería "ELECTRÓNICA"
Nombre	[Blank]



76		CIRCUITO DE FUERZA	
Nº	1	1	
Tip	Diagrama	1	
Nº	1	1	
Circuito de fuerza para el motor de 1000W			



750 BLOQUE TERMINAL SCXI-1327
 Rev. 1
 1
 "National Instruments" "SCXI-1327" "Rev. 1"

5.1.3. Descripción de los componentes.

Para el diseño y la implementación de la tarjeta se utilizaron diversos circuitos integrados, resistencias, relés, capacitores y borneras, entre otros.

A continuación se muestra una lista y una descripción de cada uno de los componentes utilizados:

PIC-16F877A: Es un microcontrolador de 8 bits, de 40 pines y su tarea es la de controlar las rutinas de conexión del motor y detectar errores.



CIB -ESPOL

MAX-232: Circuito integrado de 16 pines cuya función consiste en convertir los voltajes de TTL a la norma RS-232 para la comunicación serial con el computador.

Buffer 74LS244: Circuito integrado que se utiliza para direccionar el bus de datos hacia la pantalla de cristal líquido o

hacia la interfaz de salida a los contactores. Se utilizaron 4, con esto se ampliaron los recursos del PIC.

LTV-4N35: Circuitos integrados optoacopladores, se los utiliza en las interfases de entrada y salida para aislar al PIC de ruido.

Relés: Componentes electromecánicos que sirven de interfase entre el control y fuerza en la tarjeta, utilizan 12 voltios para energizarse y en su contacto de salida suministran 220 voltios a las bobinas de los contactores. Se utilizaron 12.

Cristal de 10 Mhz: Cristal de cuarzo que sirve para la temporización interna del microcontrolador antes mencionado, indispensable para el funcionamiento del mismo.

Capacitores de 0.01uF: Utilizados para filtrar transientes de voltaje para que no afecten al sistema, sin ellos el sistema se reinicializa (reset) y se pierde la información.

Capacitores de 1uF: Utilizados para la configuración del circuito integrado encargado de la conversión a norma RS-232.

Capacitores de 15pF: Utilizados para mantener la frecuencia del cristal de cuarzo estable.

Resistencias: De diversos valores para lograr el funcionamiento deseado de las interfases de entrada y salida.

Potenciómetro: De 100 ohmios para controlar el contraste de la pantalla de cristal líquido.

Borneras: Para facilitar las conexiones.

Cable Serial: Para la conexión entre la PC y la tarjeta de control.

5.1.4. Costo de la tarjeta.

A continuación se muestran los valores correspondientes a los elementos utilizados para la elaboración de la tarjeta de control:

Elemento	Cantidad	Valor Unitario	Subtotal
PIC16F877A	1	\$ 10	\$ 10
MAX232A	1	\$8	\$ 8
74LS244	4	\$ 0.80	\$ 3.20
4N35	24	\$ 0.70	\$ 18.90
1N4148	1	\$ 0.05	\$ 0.05
Potenciómetro	1	\$ 0.50	\$ 0.50
Resistencias (1/4 W)	53	\$ 0.02	\$ 1.06
Relés (12 V)	12	\$ 1	\$ 12
Capacitores electrolíticos	4	\$ 0.05	\$ 0.20
Capacitores cerámicos	4	\$ 0.10	\$ 0.40
Circuito impreso	1	\$ 50	\$ 50
LCD	1	\$ 20	\$ 20
Borneras	4	\$ 5.25	\$ 21
Zócalos	26	\$ 0.05	\$ 1.30
Fuente	1	\$ 10	\$ 10
		Total	\$ 156.51



CIB-ESPOL

5.2.- Comunicación PC-PIC.

5.2.1.- Interfase RS-232.

El puerto RS232, existente en todos los ordenadores actualmente es el sistema más común para la transmisión de datos entre computadoras o entre periféricos y computadoras.

El RS232 es un estándar de comunicaciones propuesto por la Asociación de Industrias Electrónicas (EIA) y es la última de varias versiones anteriores.

Lo más importante del estándar de comunicaciones es la funcionalidad específica de cada pin de entrada y salida de datos porque nos encontramos básicamente con dos tipos de conectores: los de 25 pines y los de 9 pines, siendo la más usada la versión de 9 pines, aunque la versión de 25 permite muchas más información en la transferencia de datos.

Las señales con la que actúa el puerto son digitales (0 - 1) y la tensión a la que trabaja es de 12 Voltios bipolares, es decir:

+ 12Voltios = 0 lógico

-12 Voltios = 1 lógico

Las características de los pines y su nombre son:

TXD	Transmitir Datos	Señal de salida	
RXD	Recibir Datos	Señal de entrada	
RTS	Solicitud de envío	Señal de salida	
DTR	Terminal de datos listo	Señal de salida	
CTS	Libre para envío	Señal de entrada	
DSR	Equipo de datos listo	Señal de entrada	
DCD	Detección de portadora	Señal de entrada	
SG	Tierra	Referencia	para
			señales
RI	Indicador de llamada	Señal de entrada	

Los pines que portan los datos son RxD y TxD los demás se encargan de otros trabajos, el DTR indica que el ordenador esta encendido, DSR que el dispositivo conectado al puerto esta encendido, RTS que el ordenador al no estar ocupado puede recibir datos, al revés de CTS que lo que informa es que es el dispositivo el que puede recibir datos, DCD detecta que existen presencia de datos, etc.

Con los puertos de E/S se pueden intercambiar datos mientras que las IRQ producen las interrupciones para indicar a la CPU que ha ocurrido un acontecimiento, como la llegada de datos por el puerto serial.



CIB -ESPOL

Cuando ocurre un evento en un puerto serie se activa la IRQ que avisa a la CPU que debe recoger el dato lo antes posible, pues se puede anular con un nuevo dato, para eso esta la UART 16550A que incluye 2 buffers o almacenes de información de tipo FIFO (First In Firsts Out) uno para entrada y otro para salida de 16 bits guardan los datos antes de que la CPU los recoja.

El RS232 puede hacer transmisión de datos en grupos de 5, 6, 7, u 8 bits a determinada velocidad (normalmente 9600 bits por segundo o más). Después de los datos, le sigue un bit opcional de paridad (indica que el número de bits transmitidos es par o impar) y luego 1 o 2 bits de parada.

La transmisión debe de ser constante y a una velocidad predeterminada (baudios). Los bits deben de llegar uno detrás de otro y en determinados instantes de tiempo .

Antes de iniciar cualquier comunicación con el puerto RS232 se debe de determinar el protocolo a seguir dado que el estándar del protocolo no permite indicar en que modo se esta trabajando, es la persona que utiliza el protocolo el que debe decidir y configurar ambas partes antes de iniciar la transmisión de datos.

Siendo los parámetros a configurar los siguientes:

Protocolo serie (numero bits-paridad-bits stop)

Velocidad de puerto

Protocolo de control de flujo.

La configuración de la comunicación que se utiliza en la aplicación consiste en 8 bits de información, 1 bit de parada, 1 bit de inicio, sin paridad, sin control de flujo y a 9600 baudios.

Para la comunicación entre la PC y el microcontrolador el primer problema que hay que superar es que los niveles lógicos TTL que salen del PIC no son compatibles con los niveles lógicos del puerto, para ello debemos introducir en el circuito un puente que nos traduzca los datos del PIC al puerto y viceversa, este puente es el micro MAX232.

5.2.2. Protocolo de comunicación.

El protocolo de comunicación es el conjunto de códigos que tienen una función específica para el sistema. En la aplicación existen códigos de configuración de la tarjeta, códigos de operación de la tarjeta y también existen códigos que envía la tarjeta de control a las aplicaciones en LabView para reportar errores.

Cuando la tarjeta de control funciona con la computadora se habilita la comunicación serial por medio de la interfase RS-232 y se coloca en un estado de espera. Cada vez que le llega un dato al PIC (un byte) se produce una interrupción en el programa del PIC y se almacena el dato que llegó. Para que el PIC realice una configuración en el sistema o ejecute una operación sobre el motor debe recibir 3 bytes. Es indispensable que lleguen 3 bytes, ya que si llega solo 1 o 2 el programa del PIC sigue en el estado de espera y no configura o ejecuta acción alguna.

5.2.2.1. Comandos de configuración.

Los comandos de configuración sirven para configurar la tarjeta de control. La configuración de la tarjeta de control incluye la configuración de los tiempos para la desconexión de cada resistencia en los arranques con resistencia, el tiempo para la desconexión de la resistencia para cambio de giro, cuántas resistencias externas se conectan al rotor devanado (pueden conectarse o desconectarse en cualquier momento), re-inicialización del sistema y determinar el estado de la tarjeta de control.

Para indicarle al PIC que se enviará un comando de configuración el primer byte que se envía es 204 (decimal) siempre, luego se envían los otros 2 bytes que especifican qué es lo que se configura.

En la tabla 9 se ilustran los comandos (o códigos) de configuración y su función.

<u>Primer Byte</u>	<u>Segundo Byte</u>	<u>Tercer Byte</u>	<u>Función</u>
204	47	XXX	Pregunta estado de la tarjeta.
204	55	XXX	Indica que se hará una parada.
204	63	Xseg	Setea el tiempo de desconexión de R1 en Xseg.
204	62	Xseg	Setea el tiempo de desconexión de R2 en Xseg.
204	61	Xseg	Setea el tiempo de transición de Ye a Delta en Xseg.
204	58	1	Conecta RR1 al rotor devanado.
204	58	2	Conecta RR2 al rotor devanado.
204	58	3	Conecta RR1 y RR2 al rotor devanado.
204	60	XXX	Re-inicializa el sistema (RESET).

Tabla 9, Comandos de configuración

Nota 1: XXX se refiere a cualquier valor.

Nota 2: Todos los valores están en decimal.

5.2.2.2. Comandos de operación.

Los comandos de operación indican a la tarjeta qué tipo de arranque tendrá el motor, qué conexión tendrán sus bobinas y la configuración de ellas (Ye o delta), también le indican cuándo ejecutar cambios de giro y parada.

A diferencia de los comandos de configuración, en los comandos de operación solo es importante el primer byte, el segundo y el tercer byte pueden ser cualquier valor.

En la tabla 10 se muestran los códigos de operación y su función.

<u>Primer Byte</u>	<u>Función</u>
17	Arranque directo, conexión Ye, serie a la derecha.
1	Arranque directo, conexión Ye, serie a la izquierda.
50	Arranque directo, conexión Delta, paralelo a la derecha.
34	Arranque directo, conexión Delta, paralelo a la izquierda.
18	Arranque directo, conexión Delta, serie a la derecha.
2	Arranque directo, conexión Delte, serie a la izquierda.
49	Arranque directo, conexión Ye, paralelo a la derecha.
33	Arranque directo, conexión Ye, paralelo a al izquierda.
20	Arranque con resistencias en Ye, serie a la derecha.
4	Arranque con resistencias en Ye, serie a la izquierda.
52	Arranque con resistencias en Ye, paralelo a la derecha.
36	Arranque con resistencias en Ye, paralelo a la izquierda.
21	Arranque con resistencias en Delta, serie a la derecha.
5	Arranque con resistencias en Delta, serie a la izquierda.
53	Arranque con resistencias en Delta, paralelo a la derecha.
37	Arranque con resistencias en Delta, paralelo a la izquierda.
19	Arranque Ye-Delta, en serie a la derecha.
3	Arranque Ye-Delta, en serie a la izquierda.
51	Arranque Ye-Delta, en paralelo a la derecha.
35	Arranque Ye-Delta, en paralelo a la izquierda.

Tabla 10, Comandos de operación

Los valores de los comandos de configuración se eligieron sin ninguna razón en particular, pero los comandos de operación se originaron de la programación de la aplicación en LabView ya que se creó un Cluster (conjunto de algún tipo de objeto) de Interruptores, cada interruptor representa una función, por ejemplo la posición apagado de un interruptor significa giro a la derecha, pero la posición encendido significa giro a la izquierda. A continuación se muestra el Cluster de interruptores que se creó en LabView para generar los comandos de operación:

Cluster



Figura 5.2, Cluster de interruptores.

I0, I1 e I2: Determinan el tipo de arranque según la siguiente tabla:

<u>I2</u>	<u>I1</u>	<u>I0</u>	<u>Tipo de arranque</u>
0	0	1	Directo en Ye
0	1	0	Directo en Delta
0	1	1	Ye – Delta
1	0	0	Ye con resistencias
1	0	1	Delta con resistencias

Tabla 11, Tipos de arranque.

I3: No se utiliza.



CIB -ESPOL

I4: Determina el sentido del giro (derecha o izquierda)

I5: Determina la conexión de las bobinas (serie o paralelo)

5.3 Programa en Lab View.

5.3.1 Explicación de las variables usadas.

En este capítulo se explicarán la función de las variables más importantes que se utilizaron en la programación de la aplicación en LabView.

SERIAL: Variable que guarda la configuración del puerto serial para la comunicación con la PC. Se la encuentra en la mayoría de las pantallas de LabView.

ESTADO DE LOS CONTACTORES: Almacena información sobre los contactores desde C1 hasta C8. Guarda y muestra el estado (conectado o desconectado) de los contactores mencionados.

ESTADO DE LOS CONTACTORES 2: Igual que la anterior pero con los contactores C9 hasta C12.

VOLTCHECK: Almacena el valor máximo de voltaje permitido para la alimentación del motor.

IMAX: Almacena el valor máximo de corriente permitida en los embobinados del estator del motor.

YE / DELTA: Variable de tipo boolean que almacena si el motor debe conectarse en ye o en delta.

SERIE / PARALELO: Variable de tipo boolean que almacena si las bobinas del estator del motor están conectadas en serie o en paralelo.

DER / IZQ: Variable que almacena el sentido de giro del motor.

Vab-RMS, Vbc-RMS, Vca-RMS: Almacenan los valores RMS de los voltajes línea-línea.

Ia-RMS, Ib-RMS, Ic-RMS: Almacenan los valores RMS de las corrientes de línea.

TETA, TETA 2, TETA 3: Almacenan los ángulos de desfase entre los voltajes y las corrientes de cada fase.

FACTOR DE POTENCIA, FACTOR DE POTENCIA 2, FACTOR DE POTENCIA 3: Almacenan los valores del factor de potencia por fase.

POTENCIA ACTIVA TRIFÁSICA, POTENCIA REACTIVA TRIFÁSICA, POTENCIA APARENTE TRIFÁSICA: Almacenan los resultados obtenidos de los cálculos de las potencias mencionadas.

DATA CLUSTER: Variable que representa un conjunto de interruptores que forman una orden de un byte, el cual sirve para cambiar las conexiones o el sentido de giro del motor.

OPCODE: Variable que almacena el byte que está almacenado en "DATA CLUSTER" en base decimal.

VRB, IRB, FRB: Almacenan los valores de voltaje, corriente y factor de potencia de la prueba de rotor bloqueado.

5.3.2 Explicación del programa.

La aplicación diseñada está compuesta de 2 partes: una aplicación diseñada en LabView 7 y otra aplicación diseñada en Assembler para el microcontrolador PIC16F877A. Estas dos aplicaciones se comunican entre sí (cuándo se está trabajando en modo con PC) pero realizan operaciones independientes, es decir, la aplicación en LabView 7 se encarga de monitorear las señales de voltaje, corriente y velocidad del motor y calcula diversos parámetros a partir de ellos; mientras que la aplicación en Assembler del PIC se encarga de controlar al motor y

controlar el funcionamiento correcto de los contactores. Las dos aplicaciones interactúan sólo cuando se desea hacer un cambio de conexión al motor, entonces se presiona un botón en la pantalla y LabView envía un código por el puerto serial (COM 1) al PIC el cual lo decodifica y cambia la conexión del motor y verifica que los contactores funcionen correctamente.

Con relación a la aplicación en LabView 7, ésta aplicación está estructurada con varias pantallas llamadas Vis (virtual instruments), cada VI se encarga de un procedimiento o una subrutina.

Existen 2 formas de ejecutar la aplicación: incluyendo las pruebas para hallar el circuito equivalente y hacer funcionar el motor directamente sin hacer las pruebas. Para no realizar las pruebas se ejecuta "FLUJO.vi", y para realizar las pruebas se ejecuta "PRUEBAS PARA EL MOTOR AC.vi".

El funcionamiento de la aplicación en LabView es simple, el usuario primero ingresa la conexión que desea en el motor y la ejecuta, en general la aplicación muestrea las señales continuamente, por lo tanto, las señales que se muestran en la pantalla tienen un pequeño retraso con la realidad. Aparte de muestrear las señales la aplicación está pendiente o escuchando al puerto serial (COM 1) para leer información de errores que el PIC le puede enviar, también está pendiente de los botones e interruptores que existen los paneles frontales de modo que si el usuario desea cambiar una conexión ésta se envíe al PIC por el COM1.

Si existe un error en la conexión el PIC le enviará un informe a la aplicación en LabView, la cual mostrará cuál fue el contactor que falló.

CAPITULO 6

6. MANUALES DEL USUARIO.

6.1 Manual de usuario para la tarjeta de control.

La tarjeta de control está conformada por varios elementos, pero los únicos que funcionan como interfase con el usuario son 5 botoneras, una pantalla LCD y un cable de comunicación serial con el computador.

Para encender la tarjeta de control existe un interruptor que se encuentra a la derecha de la pantalla LCD en la parte frontal de la puerta del tablero. Después de ser encendida aparece un mensaje inicial como se observa en la figura 6.1.



Figura 6.1, Mensaje inicial

Las 5 botoneras que se encuentran en la parte frontal del tablero tienen funciones específicas y cada una tiene un número y/o una función escrita debajo de ella, como se muestra en la figura 6.2.

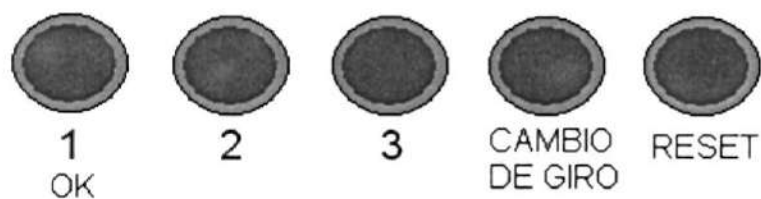


Figura 6.2, Función de cada botonera

A parte de las funciones escritas debajo de las botoneras tienen otras funciones que se mencionarán más adelante.

Los números que se encuentran debajo de las primeras 3 botoneras corresponden a las opciones que se muestran en la pantalla LCD, por ejemplo, para elegir el modo de operación de la tarjeta en la LCD se muestra el mensaje que se observa en la figura 6.3. Se ve en la pantalla que en la segunda línea se enumeran las posibles respuestas: 1) SI y 2) NO, entonces si se presiona la botonera con el número 1 se está respondiendo SI a la pregunta y con la número 2 se responde NO, cualquier otra botonera que se pulse no causará ningún efecto, excepto la botonera de RESET que reinicializa la tarjeta.



Figura 6.3, Menú de modo de operación.

Para continuar con la configuración de la tarjeta, después del mensaje de la figura 6.1 se debe presionar y soltar la botonera 1; luego aparece el mensaje de la figura 6.3.

La tarjeta de control tiene dos modos de operación como se dijo anteriormente:

- a) Sin conexión al computador.
- b) Con conexión al computador.

a) Sin conexión al computador: Cuando se trabaja en éste modo de operación no se necesita de una computadora para configurar la tarjeta, se la configura respondiendo unas cuantas preguntas sobre la forma en que se quiere hacer funcionar al motor. Este modo de operación tiene algunas restricciones con respecto al modo de operación con computador, ésta son:

- ❖ La máquina AC sólo funciona como motor, no como generador.
- ❖ No hay control sobre la máquina DC, por defecto estará conectada siempre como generador DC.
- ❖ No se pueden hacer cambios de conexión sobre la marcha.
- ❖ No se puede cambiar el tiempo para la desconexión de la resistencia de cambio de giro sobre la marcha, se lo configura una sola vez al inicio.

- ❖ Si ocurre alguna falla en la conexión de un contactor no se puede saber cuál falló por medio de la pantalla LCD.

Luego de seleccionar el modo de operación sin computador aparece al mensaje de la figura 6.4.



Figura 6.4, Tipo de arranque.

Si se selecciona la opción 1 se ejecuta un arranque directo, con la opción 2 un arranque con resistencias conectadas a las bobinas del estator para reducir el voltaje en el arranque y la opción 3 ejecuta un arranque y-delta.

Si se eligió la opción 1 en el mensaje de la figura 6.4 aparece el mensaje de la figura 6.5.



CIB-ESPOL



Figura 6.5, Tipo de conexión.

Luego se debe especificar la conexión interna de las bobinas de cada fase, como se muestra en la figura 6.6.



Figura 6.6, Conexión de las bobinas

A continuación se debe especificar el tiempo (en segundos) que una resistencia permanecerá conectada en cada fase para disminuir el voltaje cuando se ejecuta un cambio de giro, como se muestra en la figura 6.7.



Figura 6.7, Tiempo para la resistencia de cambio de giro.

Para aumentar el tiempo se presiona y se libera la botonera 3; cuando se tenga en pantalla el tiempo deseado se presiona "OK" (la botonera 1).

Para completar la configuración del arranque directo se especifica el sentido de giro del motor, como se muestra en la figura 6.8.



Figura 6.8, Sentido de giro.

De ésta forma se completa la configuración del arranque directo (por ejemplo). Cuando se termina la configuración el motor se pone en marcha con el tipo de arranque seleccionado y las conexiones especificadas y en la pantalla LCD aparece el mensaje de la figura 6.9.



Figura 6.9, Mensaje final.

Si se hubiese elegido el arranque con resistencias (opción 2 en la figura 6.4) no hubiese aparecido el mensaje de la figura 6.5 sino el de la figura 6.10.



Figura 6.10, Desconexión de R1.

Se presiona la botonera 3 hasta llegar al tiempo que se desea para la desconexión de la resistencia 1 en el arranque con resistencias, luego se presiona "OK" y aparece el mismo mensaje pero para la resistencia 2 del arranque; se procede de la misma forma: la botonera 3 para aumentar el tiempo y luego "OK". Los tiempos que se definen corresponden a los segundos que estarán conectadas cada resistencia en el arranque, cuando se cumple el tiempo definido, las resistencias se desconectan. Después de definir el tiempo de la resistencia 2 aparece el mensaje de la figura 6.5 y se sigue la secuencia de mensajes que se explicó para el arranque directo.

Si se hubiese elegido el arranque Ye-Delta (opción 3 en la figura 6.4), hubiese aparecido el mensaje de la figura 6.11.



Figura 6.11, Cambio Ye-Delta.

De igual manera la botonera 3 aumentará el tiempo y luego "OK" para continuar con el mensaje de la figura 6.5 y la secuencia antes mencionada.

Si ocurriese un error en la conexión de los contactores, el PIC 16F877A procederá a desconectar todos los contactores y mostrará el mensaje de la figura 6.12 intermitentemente hasta presionar "OK".



Figura 6.12, Mensaje de error.

Cuando el motor se encuentra funcionando se puede presionar la botonera "CAMBIO DE GIRO" y el motor cambiará su sentido de giro

obedeciendo al tiempo definido para la resistencia de cambio de giro. Se pueden realizar los cambios de giro que se desee.

Si se estuviese usando un rotor devanado también se pueden conectar resistencias externas (dos resistencias). Cuando el motor está funcionando si se presiona la botonera 1 se desconectará R1 del rotor devanado y permanecerá conectada R2 del rotor devanado; si se presiona la botonera 2 se desconectará R2 del rotor devanado y permanecerá conectada R1 del rotor devanado y por último si se presiona la botonera 3 se desconectarán las dos resistencias. Cuando el motor arranca están conectadas las dos resistencias del rotor devanado por defecto.

En caso de que se quiera desconectar al motor se presiona la botonera "RESET" la cual desconectará el motor y limpiará todos los registros de configuración de la tarjeta. La botonera "RESET" funciona igual que un pulsador de emergencia, se la puede presionar en cualquier momento y el motor se desconectará completamente.

6.2 Manual de usuario para las aplicaciones de LabView.

En las siguientes páginas se explicará cómo funcionan desde el punto de vista del usuario las pantallas de LabView que conforman la aplicación desarrollada.

Para lograr una comprensión total de cómo funcionan las pantallas se mostrará una foto de la pantalla y a continuación se explicarán los detalles de funcionamiento de la misma.

Como se mencionó anteriormente se puede ejecutar la aplicación de 2 formas: realizando las pruebas al motor o simplemente poner en marcha el motor y cambiar sus conexiones mientras se monitorean las señales presentes. Se iniciará la explicación asumiendo que se realizarán las pruebas al motor ya que ésta forma contiene a todas las pantallas, para esto se debe ejecutar la aplicación "PRUEBAS PARA EL MOTOR AC.vi"; la cual se muestra en la figura 6.13.

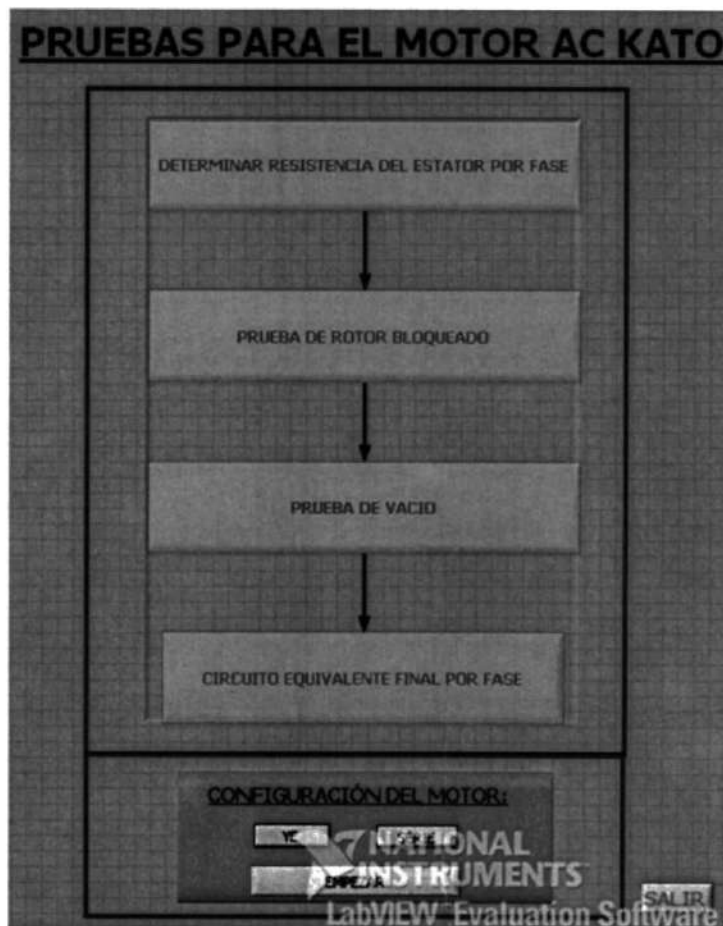


Figura 6.13, Pruebas para el motor AC KATO.

Al iniciarse esta aplicación se debe especificar la configuración para la conexión del motor, esto se lo realiza con los botones que se encuentran en la parte inferior dentro del cuadro rojo que tiene las leyendas "YE" y "SERIE". Si se presiona el botón que dice "YE" aparecerá la leyenda "DELTA", de igual forma si se presiona el que dice "SERIE" aparecerá "PARALELO"; de esta manera se coloca la

configuración del motor, esto servirá para que al realizarse las pruebas de vacío y de rotor bloqueado la tarjeta de control conecte al motor. Cabe mencionar que los circuitos equivalentes que su muestren durante las pruebas corresponden a la conexión que se especifique en ésta aplicación. Cuando se haya colocado la conexión deseada se presiona "EMPEZAR". Los 4 recuadros que aparecen en la parte central de la pantalla indicarán qué prueba se realizará, al iniciarse la aplicación todos los recuadros estarán opacos (señal de no funcionamiento) y cada uno obtendrá brillo a medida que se vayan realizando las pruebas. El primer recuadro pide determinar la resistencia del estator por fase, por lo tanto luego de presionar "EMPEZAR", el primer recuadro adquiere brillo y aparece la aplicación de la figura 6.14.

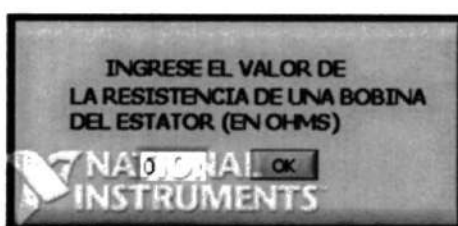


Figura 6.14, Resistencia del estator.

Se ingresa el valor de una bobina del estator y luego se presiona "OK", si se ingresa el valor de cero la aplicación mostrará un error e insistirá en un valor diferente de cero.

Luego de presionar "OK" desaparece la aplicación de la figura 6.14 y el segundo recuadro de la figura 6.13 adquiere brillo indicando que la prueba de rotor bloqueado empezará, cuando ésta aplicación empieza se ve como la figura 6.15.

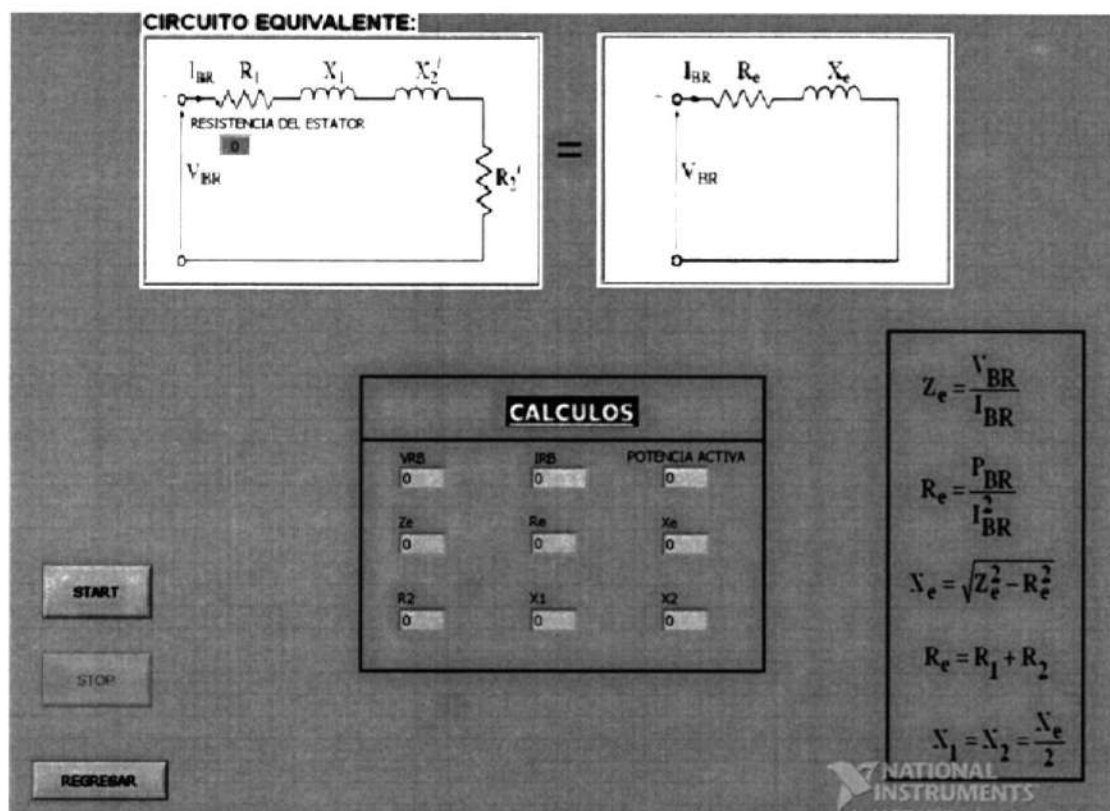


Figura 6.15, Prueba de rotor bloqueado.

En esta aplicación se muestra el circuito equivalente por fase para la prueba, considerando que el rotor no se mueve y por lo tanto el

desplazamiento es máximo. En el recuadro que dice "Resistencia del estator" se muestra un valor que corresponde al tipo de conexión de las bobinas, hay que recordar que son dos bobinas por fase que pueden estar conectadas en serie o en paralelo, por lo tanto el valor que se mostrará será el doble del que se ingresó en la aplicación de la figura 6.14 (conexión serie) o la mitad de dicho valor (conexión paralela).

Las siglas BR se refieren a "Blocked rotor" que significa rotor bloqueado, por lo tanto VBR e IBR significan voltaje con rotor bloqueado y corriente con rotor bloqueado. Del lado derecho de la pantalla aparecen el grupo de ecuaciones que se utilizan para realizar los cálculos de los parámetros, los cuales se muestran en el centro de la pantalla.

Para iniciar la prueba se presiona "START", si las condiciones para realizar la prueba son las correctas se conectará el motor e iniciará la captura de las señales y el cálculo de los parámetros, de lo contrario aparecerán mensajes de error para corregir las condiciones iniciales para la prueba (ej: el voltaje del VARIAC debe ser cero al inicio de la prueba). Cuando se esté conforme con los datos mostrados (sean estables) se presiona "STOP", se puede volver a hacer la prueba o se puede presionar "REGRESAR" para volver a la aplicación de la figura

6.13. Cabe mencionar que los parámetros calculados se guardan para luego ser utilizados en la base de datos y en la gráfica de las curvas representativas del motor. Al regresar a la aplicación de la figura 6.13 adquiere brillo el recuadro que dice "PRUEBA DE VACIO" y se inicia la misma como se muestra en la figura 6.16.

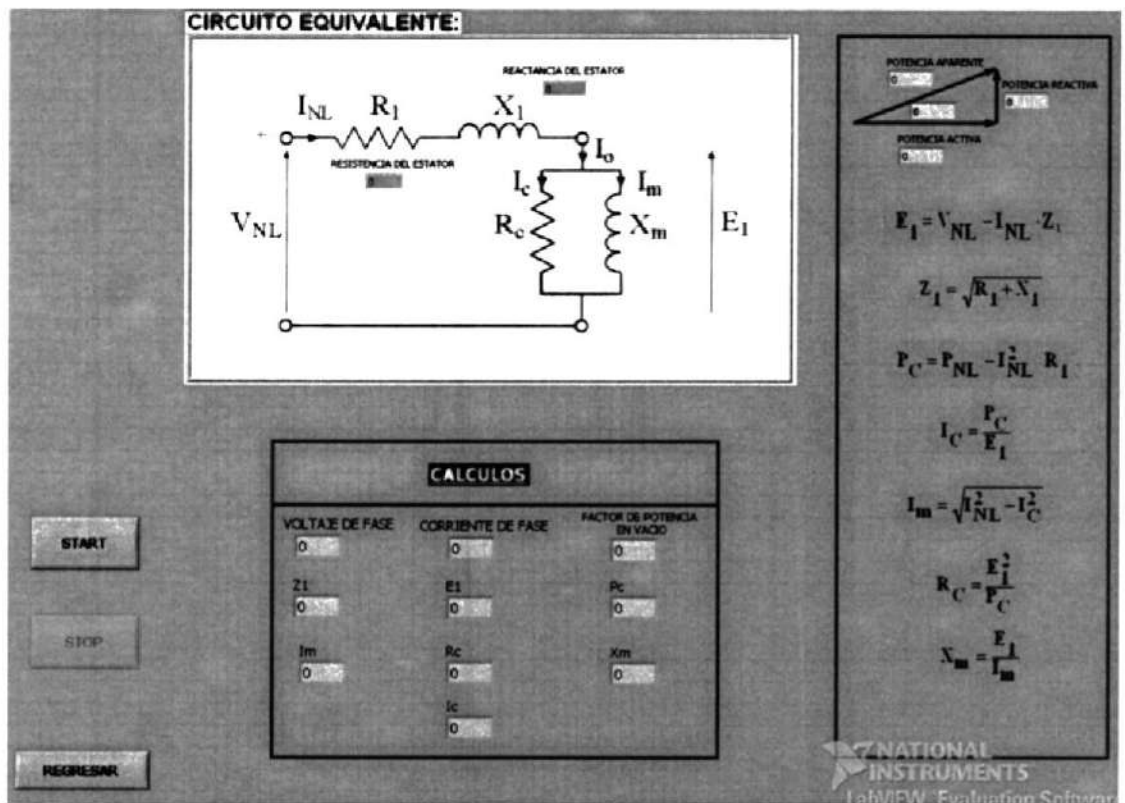


Figura 6.16, Prueba de vacío

Esta aplicación funciona exactamente igual que la prueba de rotor bloqueado.

Al regresar a la aplicación de la figura 6.13, adquiere brillo el último recuadro: "Circuito equivalente final por fase" y se abre otra pantalla en la que se muestra el circuito equivalente definitivo, como se muestra en la figura 6.17.

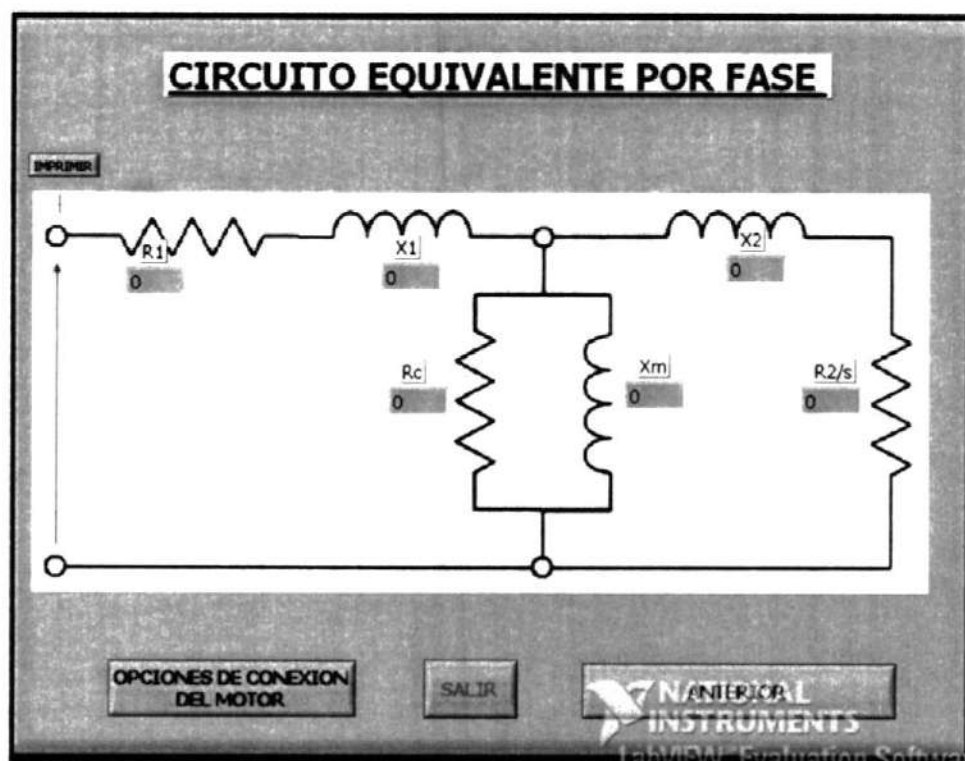


Figura 6.17, Circuito equivalente por fase.

En ésta aplicación encontramos 4 opciones, "SALIR" para cerrar todas las aplicaciones, "ANTERIOR" para volver a la aplicación "PRUEBAS DEL MOTOR AC.vi" para realizar las pruebas otra vez si se lo desea, la

opción "IMPRIMIR" para imprimir al circuito equivalente y la opción "OPCIONES DE CONEXIÓN DEL MOTOR" la cual llama a la aplicación que sirve para configurar cómo se hará funcionar la máquina AC: ya sea como motor AC o como generador AC, ésta aplicación se muestra en la Figura 6.18.



Figura 6.18, Configuración de la máquina AC.

Esta aplicación es el punto de partida para hacer funcionar a la máquina AC y a la máquina DC como se desea. Para empezar se debe especificar si la máquina funcionará como motor AC o como generador AC, esto se lo define presionando el botón gris con letras rojas que aparece como encabezado de la pantalla, si se lo presiona cambia de "MOTOR AC TRIFASICO" a "GENERADOR AC TRIFASICO". Si se elige trabajar como motor AC se debe especificar el tipo de arranque con los 4 botones que se encuentran debajo del botón de encabezado, "ARR-DIR" se refiere a arranque directo, "ARR-YD" se refiere a arranque Ye-Delta, "ARR-DR" se refiere a arranque en delta con resistencias y "ARR-YR" se refiere a arranque en ye con resistencias.

Si se selecciona un arranque con resistencias o el arranque Ye-Delta, se habilita el botón "DEF. TIEMPOS" que ejecuta una aplicación que sirve para definir los tiempos que estarán conectadas las resistencias en el arranque o el cambio de la conexión Ye a la conexión Delta como se muestra en la Figura 6.19.

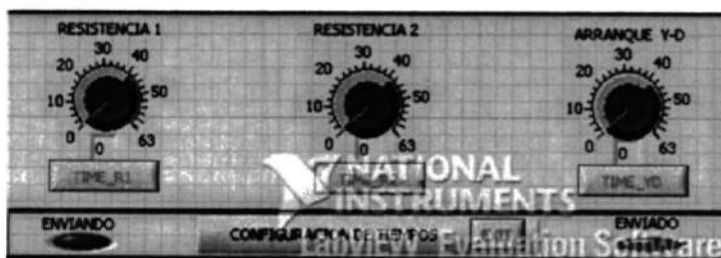


Figura 6.19, Definición de tiempos.

El uso de ésta aplicación es muy sencillo simplemente se giran los controles en forma de perilla hasta obtener el tiempo (en segundos) deseado para cada resistencia o para el cambio de conexión de Ye a Delta y luego se presionan los botones “TIME_R1” para enviar el valor del tiempo para la resistencia 1, “TIME_R2” para el tiempo de la resistencia 2 y “TIME_YD” para el cambio de Ye a Delta en el arranque. Al finalizar el envío de los tiempos se presiona “EXIT” para regresar a la pantalla de la Figura 6.18.

Volviendo a los controles de la aplicación de la Figura 6.18, cuando se selecciona “ARR_DIR” para realizar un arranque directo se debe especificar qué conexión tendrá el motor (o generador) con los controles en forma de paleta que se encuentran a continuación de los botones de los arranques, los cuales tienen las etiquetas “YE/DELTA”, “SERIE/PARALELO” y “DER/IZQ”. Cuando se selecciona que la

máquina AC funcione como motor todas estas opciones pueden modificarse, pero cuando funciona como generador el control para el sentido de giro "IZQ/DER" se deshabilita porque el sentido de giro lo define el motor DC.

Luego se especifica el tipo de rotor que tiene el motor en su interior: Jaula de ardilla o rotor devanado, esto se lo realiza marcando una de las cajas de opciones que tienen las etiquetas "MIJA" y "ROTOR DEVANADO". Cuando se selecciona "ROTOR DEVANADO" se habilita el botón "CONF. ROTOR", este sirve para conectar una o dos resistencias al rotor devanado antes de arrancar el motor. Para cambiar la conexión del rotor devanado en las siguientes aplicaciones se encontrará un botón que hace referencia al rotor devanado y desde allí se podrá cambiar la conexión del rotor devanado, es decir desconectar o conectar las resistencias en cualquier momento para cambiar la velocidad del motor.

A continuación de los controles mencionados aparece un botón con la etiqueta "GENERADOR DC-OFF", si se presiona el botón la etiqueta cambia a "GENERADOR DC-ON", esto define si se conecta o no el

generador DC en el arranque con lo cual se arranca el motor AC con carga o sin carga. Por defecto el generador estará desactivado en el arranque, pero si se selecciona conectarlo se habilitará un pequeño botón con la etiqueta "OK" para enviar la orden a la tarjeta de control y energizar el contactor C_{12} para conectar el generador DC. Las aplicaciones de LabView verifican si el generador DC puede conectarse o no, esto lo realiza muestreando el voltaje DC que existe entre los terminales A_1 y A_2 , si el voltaje DC es igual a cero la aplicación de LabView permitirá la conexión de la máquina DC como generador y enviará el código correspondiente a la tarjeta de control para su conexión; esta verificación se realiza para evitar que se conecte a la máquina DC como motor cuando la máquina AC también está funcionando como motor.

Por último se encuentra un interruptor que tiene dos opciones: "SEÑALES" y "CONTACTORES", este interruptor define qué se mostrará en la pantalla cuando el motor empiece a funcionar después de presionar el botón "CONECTAR". Si se selecciona "CONTACTORES" aparece la aplicación que se muestra en la figura 6.20 y si se selecciona "SEÑALES" la de la figura 6.21.

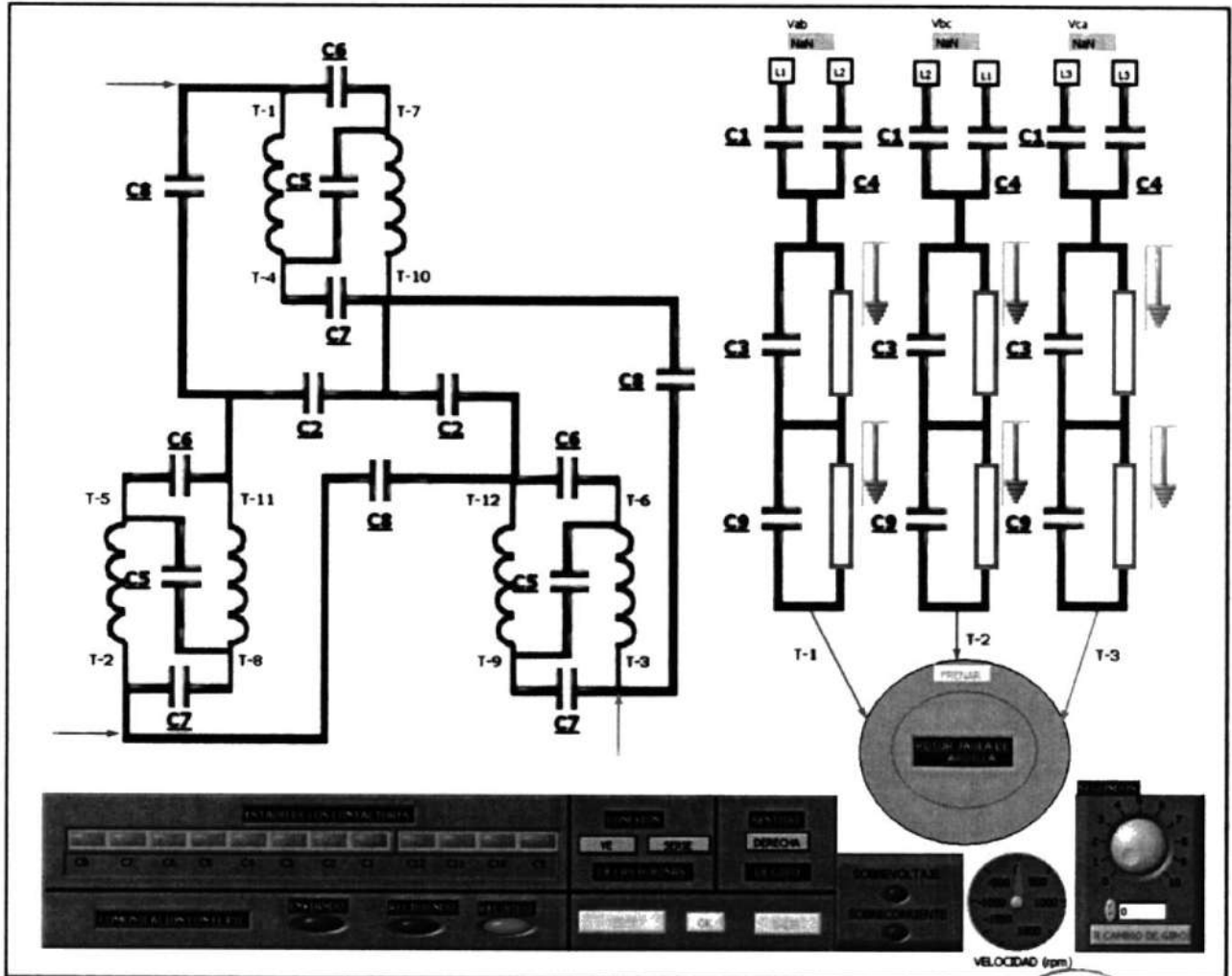


Figura 6.20, Pantalla de contactores.



CIB - ESPOL

Esta aplicación muestra la conexión que tiene el motor cerrando los contactos que aparecen en el gráfico. Desde esta aplicación es posible realizar algunas operaciones sobre el motor como: cambiar la conexión

del motor sobre la marcha, frenar el motor, cambiar de giro, conectar resistencias externas al rotor devanado y desconectar al motor.

Para cambiar la conexión del motor se utilizan los botones con las etiquetas "YE / DELTA", "SERIE / PARALELO" y "DERECHA /IZQUIERDA"; con esos botones se coloca la conexión deseada y luego se presiona el botón "OK" para ejecutar los cambios. Se recomienda hacer no más de un cambio por vez, es decir si se quiere cambiar de Ye-Serie a Delta-Paralelo primero se hace el cambio de serie a paralelo y luego de Ye a delta o viceversa pero no ambos al mismo tiempo.

Para realizar cambios de giro primero se debe especificar el tiempo que estará conectada la resistencia de cambio de giro, esto se lo realiza con la perilla que se encuentra en la parte inferior derecha de la figura 6.20, luego de especificar el tiempo (en segundos) con la perilla se presiona el botón "R_CAMBIO DE GIRO", luego de lo cual se utiliza el botón "DERECHA/IZQUIERDA" y luego "OK" para efectuar el cambio de giro. Durante el cambio de giro los botones "OK", "RESET", "FRENAR" y "SALIR" se deshabilitan hasta que la operación haya terminado.

En esta pantalla también hay indicadores de sobrecorriente y sobrevoltaje, los cuales utilizan los valores nominales de placa del motor para encenderse.

Para desconectar al motor y regresar a la pantalla de arranques (Figura 6.18) se presiona "RESET", para frenarlo y regresar a la pantalla de arranques "FRENAR" y para salir "SALIR".

A continuación se muestra y explica la aplicación "SEÑALES" que se mencionó en la explicación de la aplicación de la Figura 6.18.

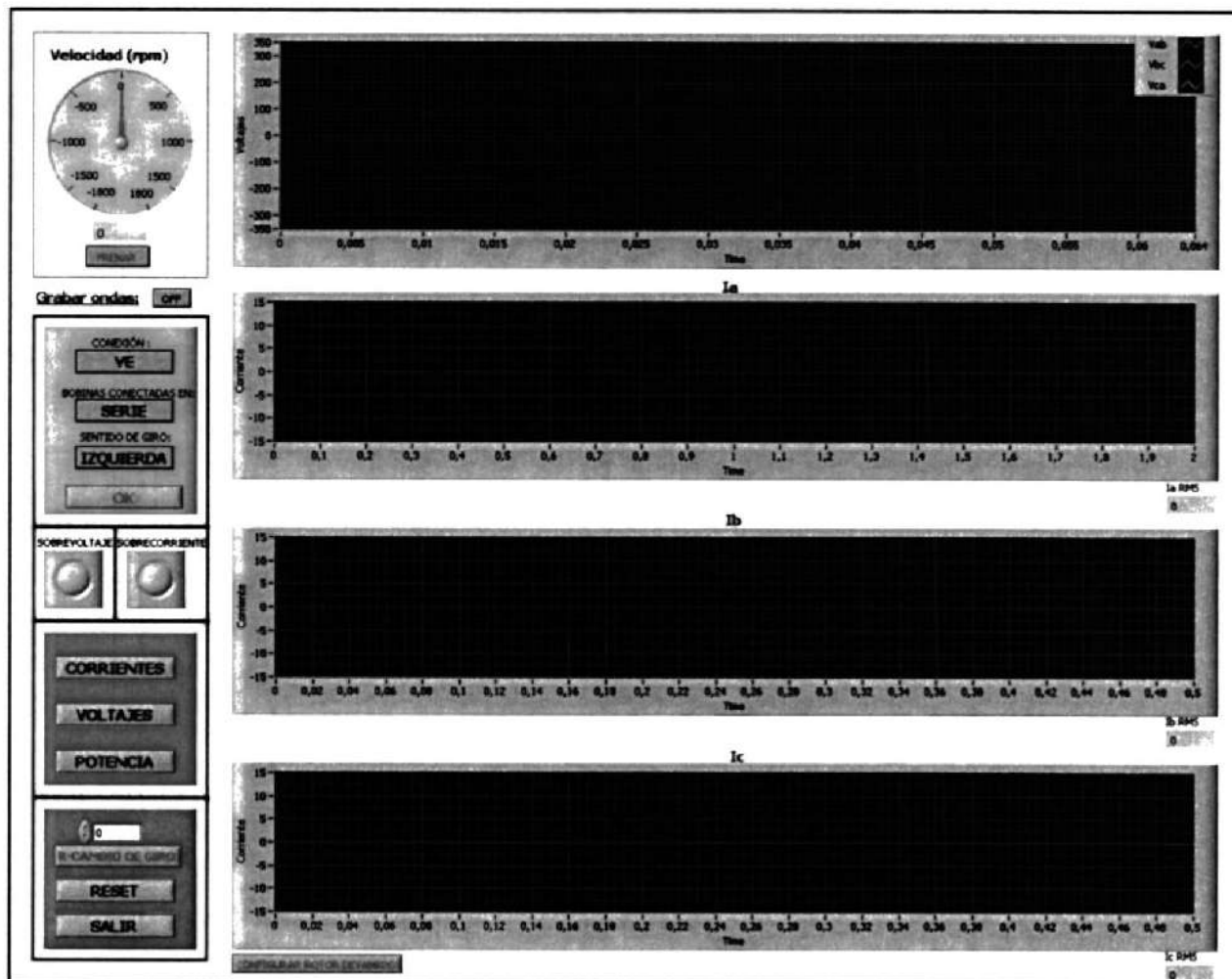


Figura 6.21, Visualización de señales

Con esta aplicación se pueden realizar los mismos cambios que con la aplicación anterior, pero se diferencian en qué se visualiza. En esta aplicación no se visualiza el estado de los contactores que forman la conexión del motor, se visualizan las señales eléctricas internas del motor. En el primer cuadro negro se visualizan los tres voltajes de línea

que alimentan al motor o los voltajes generados por el generador AC. A continuación en cada cuadro posterior se muestran las corrientes de fase. También se visualiza la velocidad de la máquina (en rpm) en la parte superior izquierda.

Otra característica importante de esta pantalla es que permite grabar las ondas que se están visualizando, cuando la aplicación se ejecuta se advierte de que se grabarán las ondas por defecto. Para desactivar esta opción se presiona el botón que está con la etiqueta "GRABAR ONDAS" y se lo coloca en "OFF". El objetivo de grabar las ondas es el de poder analizar los transientes que ocurren, para lo cual existe otra aplicación que se explicará más adelante.

Para visualizar las características de los voltajes y las corrientes se presionan los botones "VOLTAJES" y "CORRIENTES" los cuales muestran pantalla similares a la figura 6.22. La potencia aparente, real y reactiva se muestran con el botón "POTENCIA", su pantalla aparece en la figura 6.23.

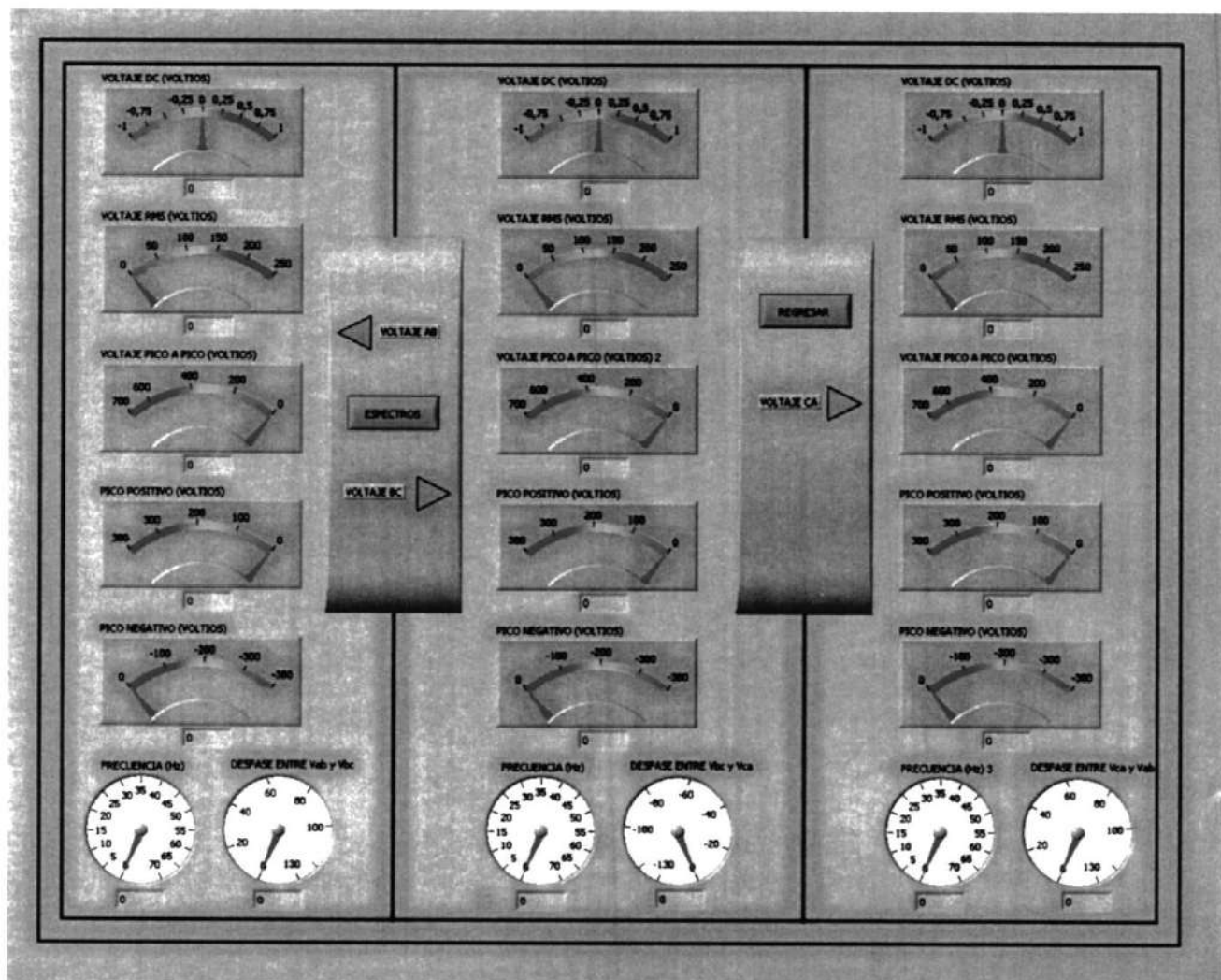


Figura 6.22, Análisis de los voltajes de línea

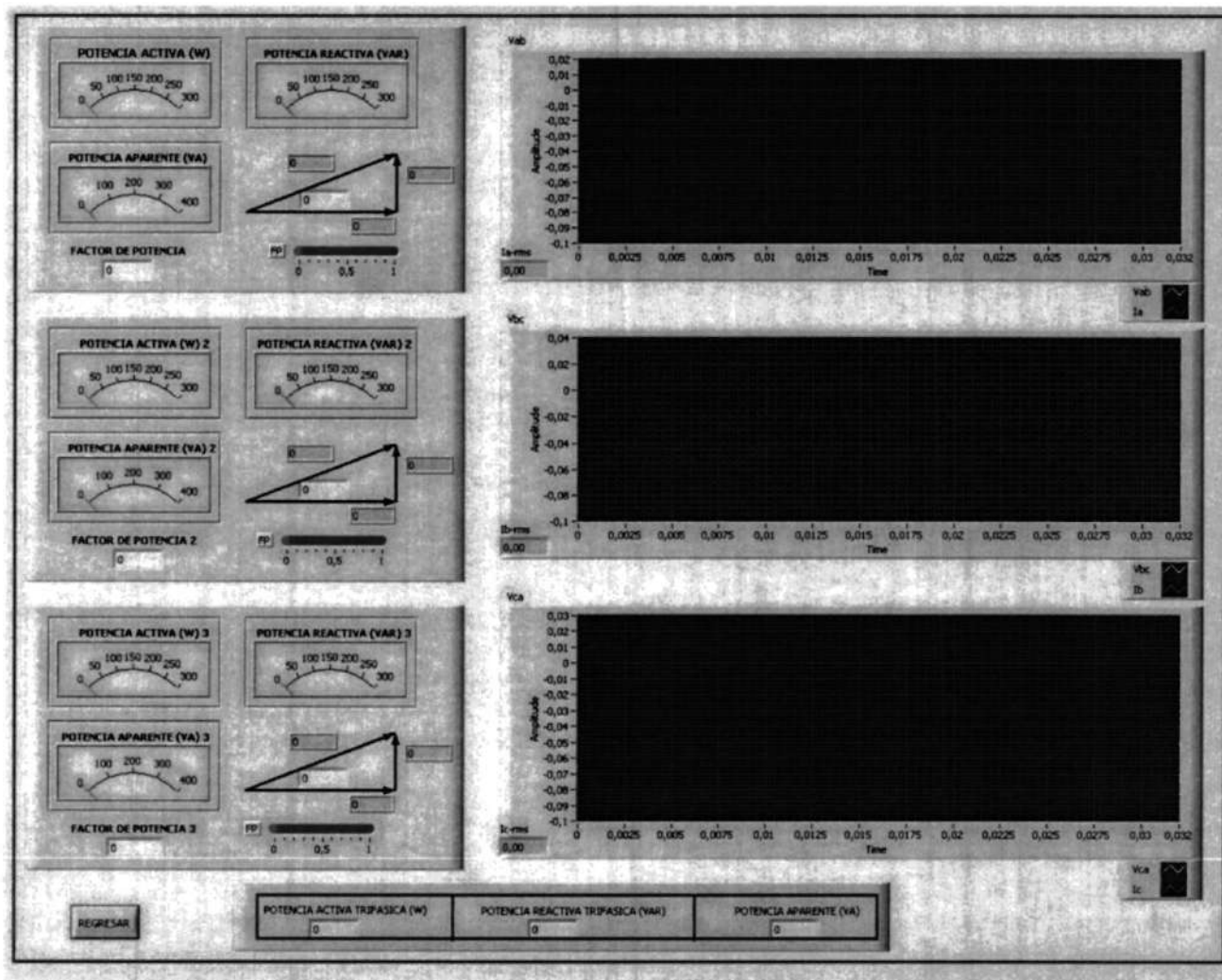


Figura 6.23, Análisis de potencia.

Cuando se desea analizar las ondas que se grabaron en la aplicación de la figura 6.21 se debe ejecutar la aplicación "RECORD DE ONDAS", la cual se muestra en la figura 6.24.

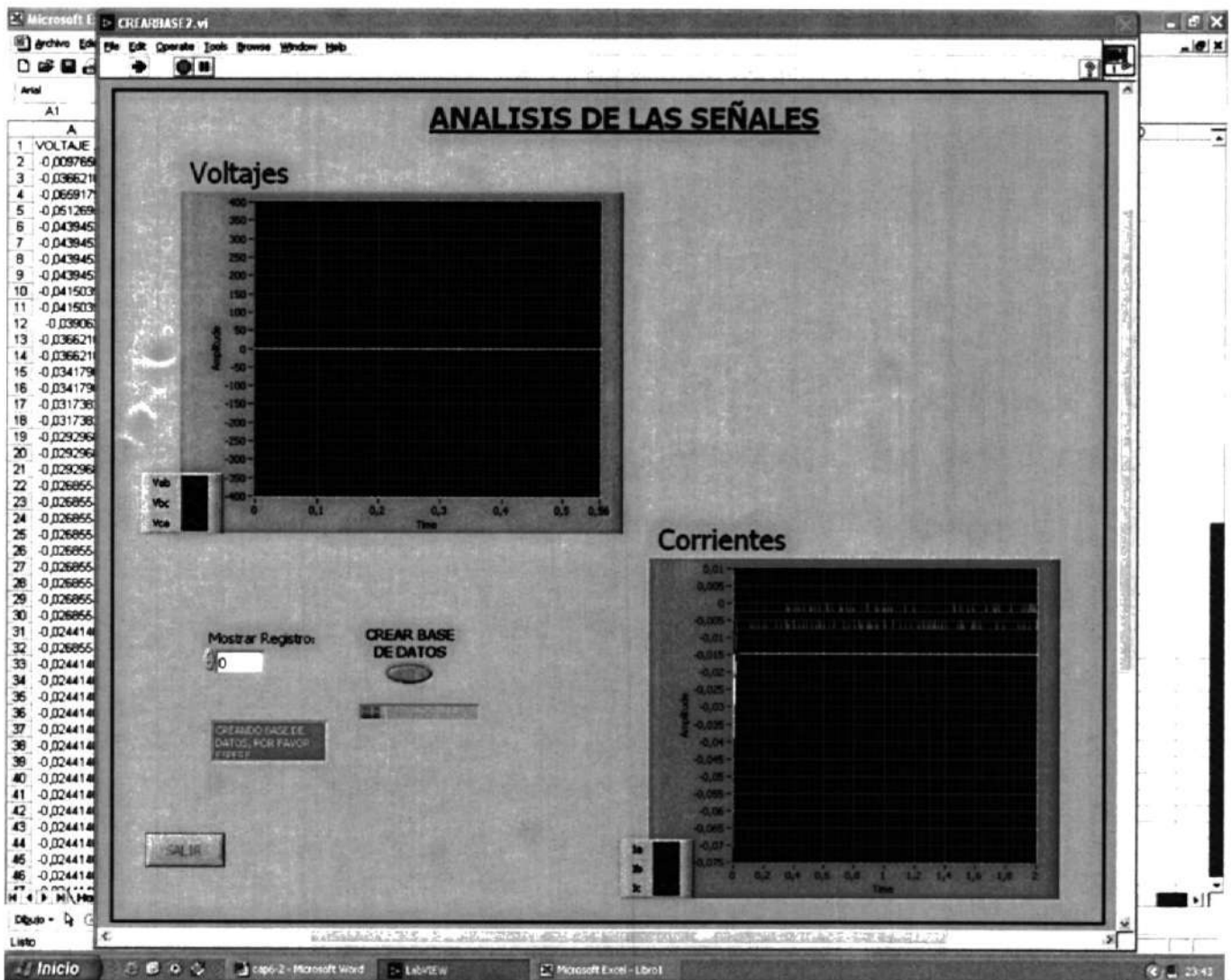


Figura 6.24, Record de ondas.

Cuando se ejecuta esta aplicación se abre un libro de "EXCEL" como se muestra en el fondo de la figura 6.22. El record de las ondas esta formado por "fotos" o "registros", en cada registro hay una "foto instantánea" de los voltajes de línea y las corrientes de fase. Para ver los diferentes "registros" o "fotos" se utiliza el control que tiene la etiqueta "MOSTRAR REGISTRO", en este control se ingresa un número el cual corresponde a un "registro" con el grupo de ondas antes mencionados. Los voltajes de línea que corresponden al "registro" seleccionado se muestran en la parte superior izquierda y las corrientes en la parte inferior derecha, de esta forma se puede buscar al grupo de ondas que contiene información interesante. Cuando se localice al grupo de ondas de interés se presiona el botón que tiene la etiqueta "CREAR BASE DE DATOS" con lo cual empieza a llenarse la hoja de EXCEL que se abrió, más información sobre el contenido de la hoja de EXCEL se encuentra en el capítulo 9.

CAPITULO 7

7. CIRCUITOS EQUIVALENTES DE LAS MÁQUINAS ELÉCTRICAS

7.1 Ecuaciones para calcular los circuitos equivalentes.

7.1.1 Generalidades de los circuitos equivalentes.

Los circuitos equivalentes son modelos matemáticos que representan a máquinas reales y nos sirven para predecir el comportamiento de dichas máquinas en diversas situaciones.

Para obtener los circuitos equivalentes se deben hallar experimentalmente los parámetros que contiene el circuito equivalente, esto se logra con diversas pruebas como son los

ensayos de corto circuito y circuito abierto para un transformador y los ensayos de rotor bloqueado y en vacío de un motor AC.

7.1.2 Circuito equivalente del motor AC trifásico KATO.

El circuito equivalente del motor AC KATO se representa en la figura 7.1.

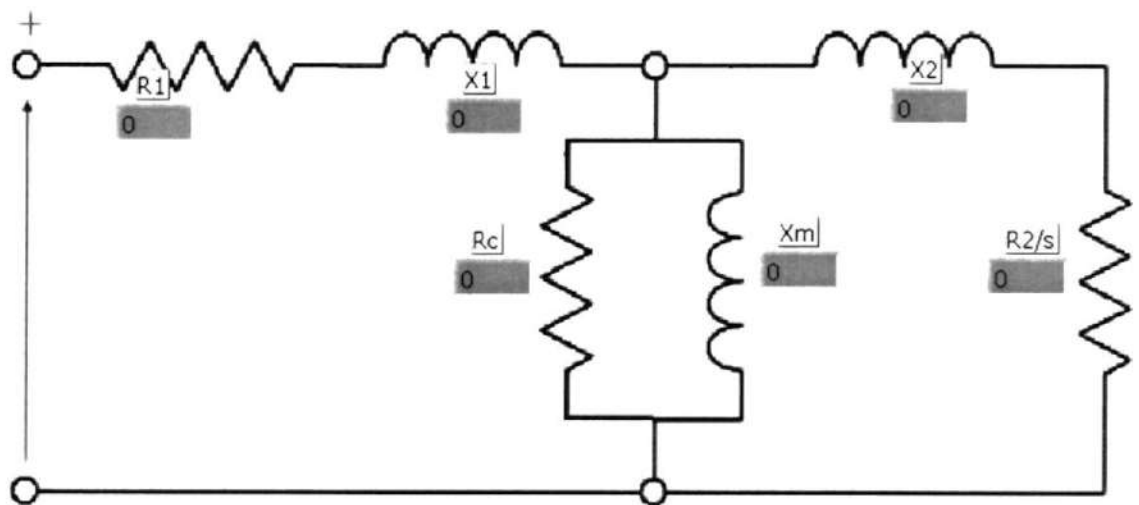


Figura 7.1, Circuito equivalente del motor KATO.

Los parámetros mostrados en el circuito equivalente representan:

R1: Resistencia del Estator (por fase).

R2/s: Resistencia del rotor, la cual varía con el deslizamiento.

X1: Reactancia del Estator (por fase).

X2: Reactancia del rotor.

Xm: Reactancia de magnetización.

Rc: Resistencia de magnetización.

Para encontrar los parámetros antes mencionados se deben realizar 2 pruebas: el ensayo en vacío y el ensayo de rotor bloqueado.

El ensayo de rotor bloqueado consiste en energizar el motor con el rotor bloqueado y hacer fluir por los embobinados la corriente nominal del motor, al hacer esto el deslizamiento es máximo (es decir 1) y la frecuencia eléctrica del estator y el rotor es la misma. Hay que recordar que la corriente que se muestra en las diferentes pantallas de LabView es la corriente de línea, por lo tanto depende de la conexión del motor (ye o delta) el

valor de la corriente nominal que se hará fluir, LabView indica si hay sobrecorriente ya que tiene un algoritmo (Icheck) que determina la corriente de fase máxima tomando en cuenta la conexión del motor.

Tomando en cuenta las consideraciones antes expuestas el circuito equivalente del motor con el rotor bloqueado se muestra en la figura 7.2.

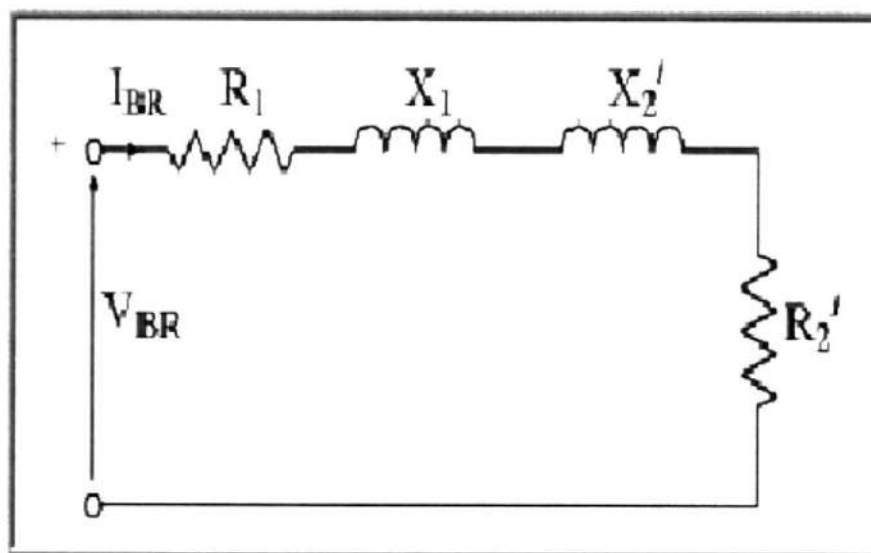


Figura 7.2, Circuito equivalente con el rotor bloqueado.

Donde V_{BR} es el voltaje en el cual se consigue la corriente nominal con el rotor bloqueado, X_1 es la reactancia del estator, X_2 es la reactancia reflejada del rotor y R_2 la resistencia reflejada del rotor. Después de realizar este ensayo se obtiene X_1+X_2 lo cual es llamada X_e (reactancia equivalente), no es posible saber de forma sencilla y con certeza los valores de X_1 y X_2 ya que se necesita la relación de vueltas entre el rotor y el estator, esto no afecta a los posteriores cálculos porque en todas las ecuaciones aparece X_1+X_2 , es decir X_e , la cual sí se encuentra con éste ensayo. También se halla el valor de R_1+R_2 al que se llama R_e (resistencia equivalente), debido a que R_1 se conoce fácilmente (midiendo la resistencia en las bobinas del estator), se determina R_2 por simple sustracción. En resumen las ecuaciones utilizadas en este ensayo son:

$$Z_e = \frac{V_{BR}}{I_{BR}}$$

$$R_e = \frac{P_{BR}}{I_{BR}^2}$$

$$X_e = \sqrt{Z_e^2 - R_e^2}$$

$$R_e = R_1 + R_2$$

$$P_{BR} = V_{BR} \cdot I_{BR} \cdot \text{fp}$$

El ensayo en vacío consiste en energizar con el motor con voltaje nominal y dejar que funcione sin carga alguna, esto ocasiona un deslizamiento muy pequeño y $R2/s$ es muy grande de tal forma que la corriente en el rotor es despreciable; en el lado del estator toda la corriente fluye por sus embobinados y por la rama de magnetización (o el núcleo) del motor. Este ensayo sirve para hallar los parámetros restantes, es decir: R_c y X_m , pero la pantalla de LabView que muestra este ensayo muestra otras magnitudes como son: la corriente que pasa por R_c y la potencia que ésta absorbe, la corriente que pasa por X_m y el voltaje en la rama de magnetización E_1 . Una vez más LabView controla que el voltaje con el que se alimenta al motor no esté sobre el valor máximo que acepta el motor con la conexión que se esté realizando. El algoritmo que determina el voltaje máximo de fase que el motor puede recibir se llama V_{check} .

Este algoritmo realiza la conversión cuando es necesaria de los voltajes de línea en voltajes de fase y determina el máximo voltaje que el motor puede recibir, ya que depende de la conexión de las bobinas (serie o paralelo) en cada fase.

El circuito equivalente para éste ensayo se muestra en la figura 7.3.

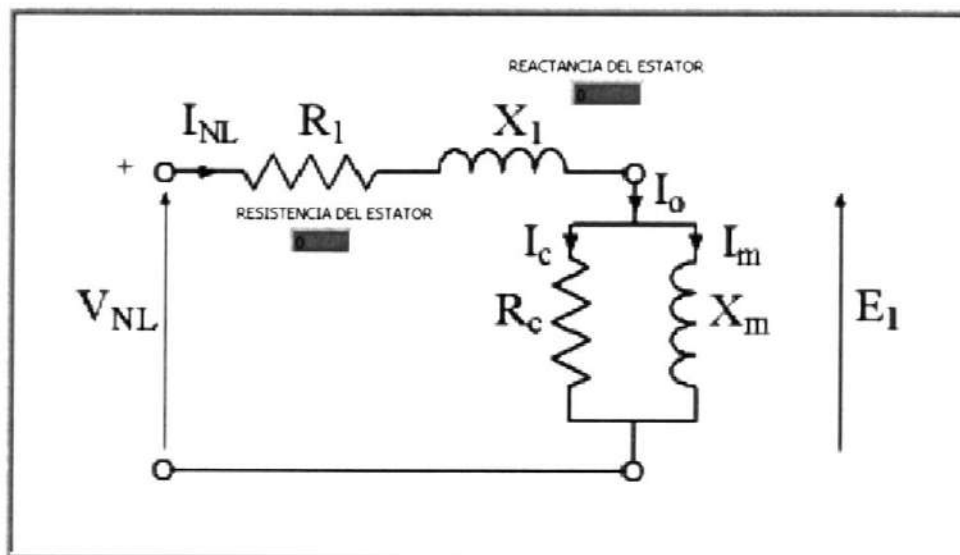


Figura 7.3, Circuito equivalente del motor en vacío.

Donde V_{NL} es el voltaje de fase aplicado al motor sin carga, I_{NL} es la corriente que fluye por la fase del motor sin carga, I_c e I_m son las componentes de la corriente I_0 que fluye por el núcleo. Las ecuaciones que se utilizaron para realizar los cálculos en este ensayo son:



CIB-ESPOL

$$E_1 = V_{NL} - I_{NL} \cdot Z_1$$

$$Z_1 = \sqrt{R_1 + X_1}$$

$$P_C = P_{NL} - I_{NL}^2 \cdot R_1$$

$$I_C = \frac{P_C}{E_1}$$

$$I_m = \sqrt{I_{NL}^2 - I_C^2}$$

$$R_C = \frac{E_1^2}{P_C}$$

7.2. Pantallas de LabView para las diferentes pruebas.

A continuación se muestran las pantallas de LabView encargadas de controlar las pruebas o ensayos antes descritos y obtener los valores o parámetros utilizando las ecuaciones anteriormente expuestas.

La primera pantalla que se mostrará es la que indica qué prueba o ensayo se realizará, en esta pantalla se debe indicar qué tipo de conexión se le hará al motor, luego de que se ingrese la conexión deseada presionando los botones con las etiquetas de YE/DELTA y

SERIE/PARALELO se presiona "EMPEZAR" para iniciar las pruebas. La figura 7.4 muestra la aplicación "PRUEBAS PARA EL MOTOR AC":

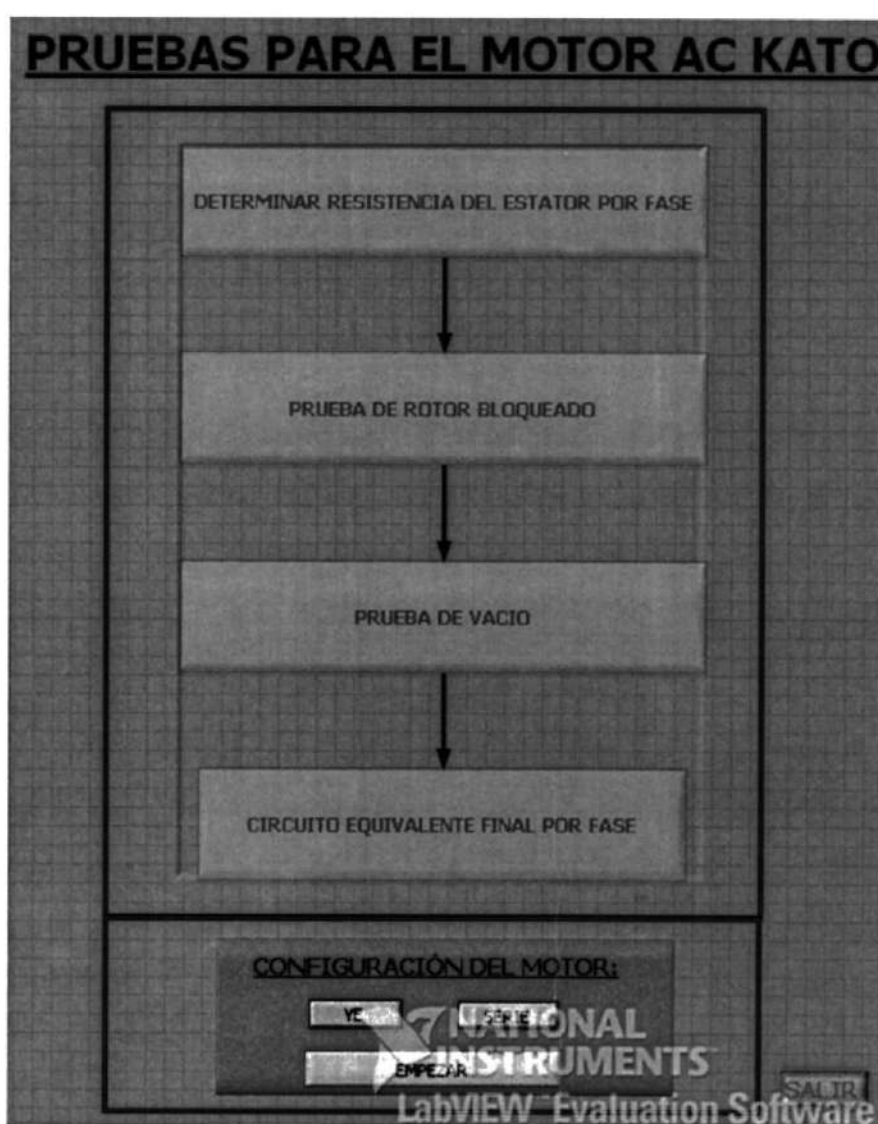


Figura 7.4, Pruebas para el motor AC

Luego de que se presiona “EMPEZAR” se abre una pequeña ventana en la que se pide ingresar la resistencia del estator. (Figura 7.5)

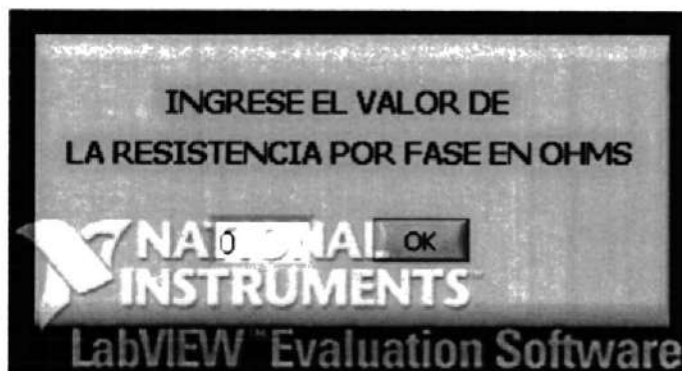


Figura 7.5, Resistencia del estator.

Se ingresa el valor del resistencia en ohmios y se presiona “OK”, si se ingresa el valor “0” como resistencia, LabView no lo acepta y le pide que ingrese un valor diferente de cero. (Figura 7.6)

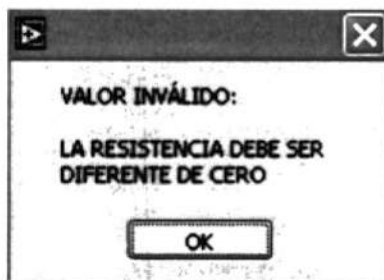
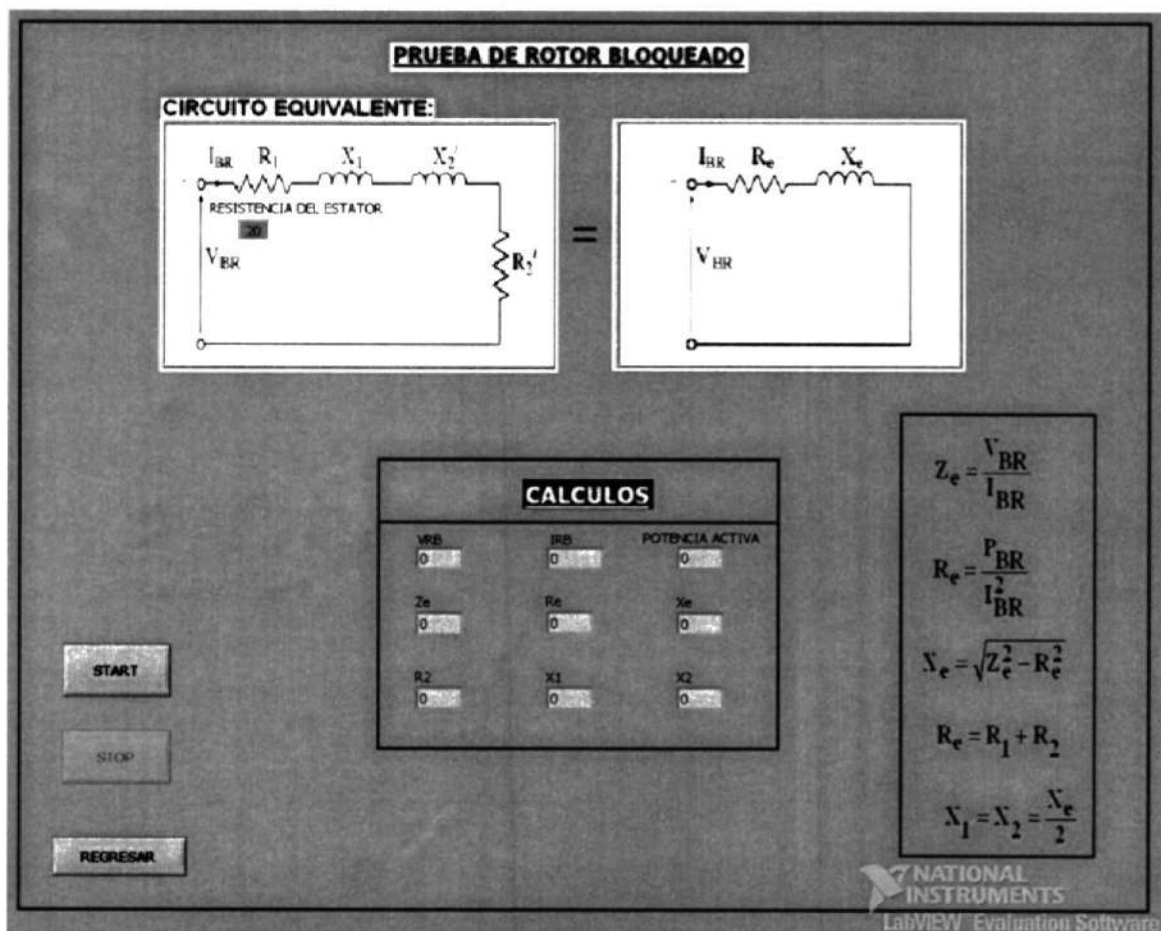
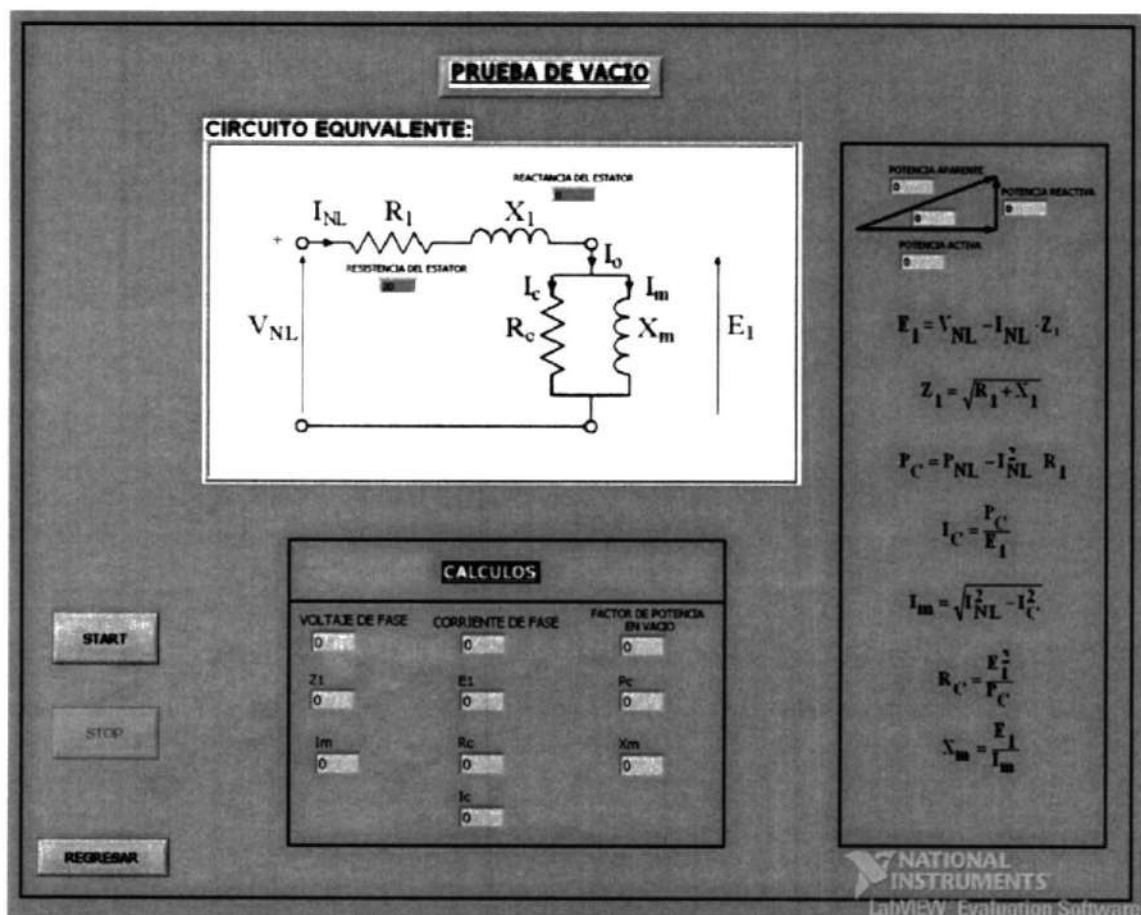


Figura 7.6, Valor de resistencia inválido.

La siguiente pantalla que se muestra es la del ensayo o prueba de rotor bloqueado, en ésta pantalla se presiona "START" después de que se ha puesto el VARIAC en cero para iniciar la prueba, en caso de que el VARIAC no esté en cero LabView no autorizará el inicio de la prueba, luego de que se ha puesto en cero el VARIAC y se ha trabado el rotor, se inicia la prueba, cuando se tienen los datos y los parámetros calculados en la pantalla se debe presionar "STOP" para terminar el ensayo y luego "REGRESAR" para seguir con la siguiente prueba. A continuación se muestra la pantalla del ensayo de rotor bloqueado:



Después de presionar "REGRESAR" se abre la pantalla del ensayo de vacío o la prueba de vacío, para ésta prueba se debe energizar el motor con los voltajes correctos y después se presiona "START", cuando se tengan los datos se presiona "STOP" para terminar la prueba, luego se debe presionar "REGRESAR" para mostrar el circuito equivalente final, la pantalla de la prueba de vacío se muestra a continuación:

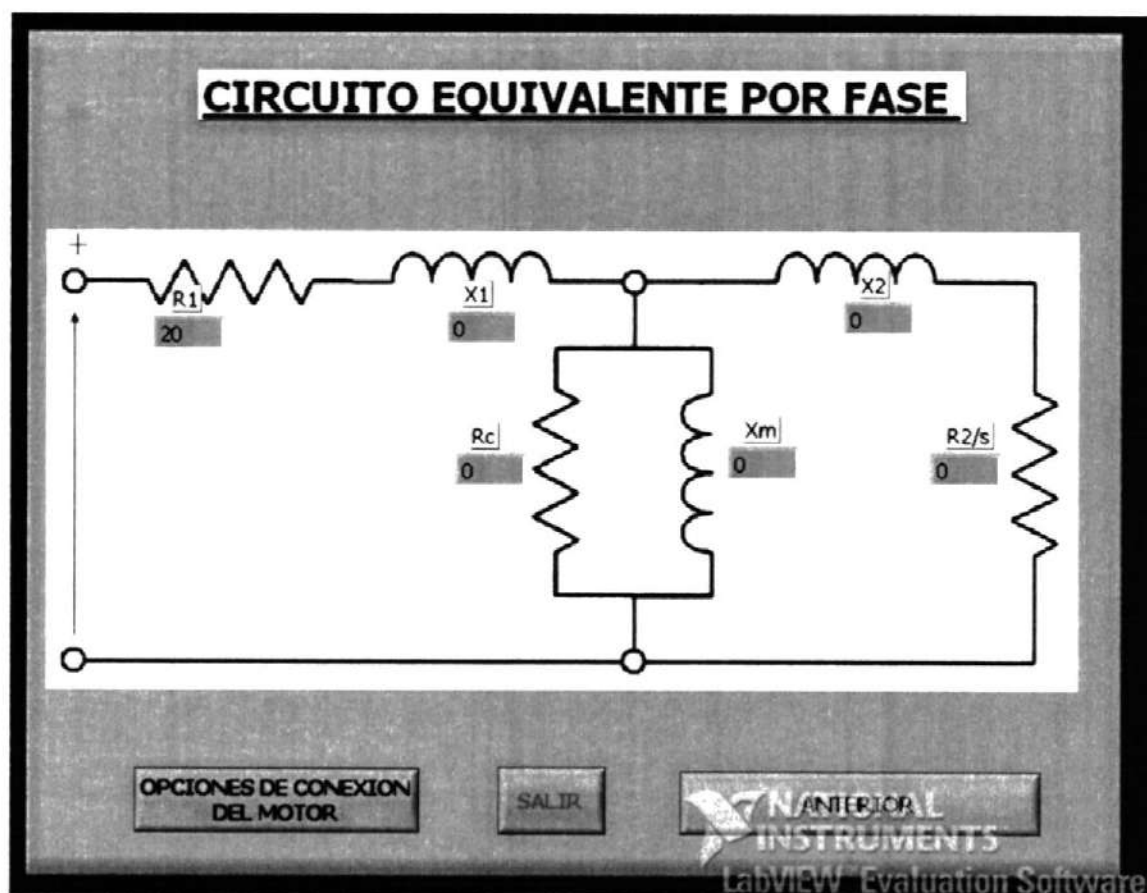


La última pantalla corresponde al circuito equivalente final por fase del motor KATO, en ésta pantalla se muestran los valores de todos los parámetros del circuito equivalente. Esta pantalla ofrece 3 opciones: "OPCIONES DE CONEXIÓN DEL MOTOR", "SALIR" y "ANTERIOR".

Si se presiona el botón de "OPCIONES DE CONEXIÓN DEL MOTOR", se abre una pantalla para configurar el arranque del motor entre otras cosas. Esta pantalla se explicó en detalle en el manual de usuario (capítulo 6).

Si se presiona "SALIR" se cierra la aplicación (se cierra todo LabView) y si se presiona "ANTERIOR" se regresa a la pantalla "PRUEBAS PARA EL MOTOR AC KATO", es decir se vuelven a hacer las pruebas.

La pantalla en mención es la siguiente:



CAPITULO 8

8. MONITOREO DE SEÑALES

8.1. Pantallas de adquisición de datos.

En este capítulo se mostrarán las pantallas que se encargan de la adquisición de datos y se explicará su funcionamiento.

Hay varias pantallas o aplicaciones que se encargan de la adquisición de datos pero todas se basan en el mismo principio, por lo tanto se mostrará y se explicará la más importante, de allí se originan el resto.



CIB -ESPOL

La aplicación de adquisición de datos más importante se llama "SEÑALES" y se abre después del menú de arranques del motor, el diagrama de bloques (o la programación) de la parte encargada de la adquisición de datos se muestra en la figura 8.1.

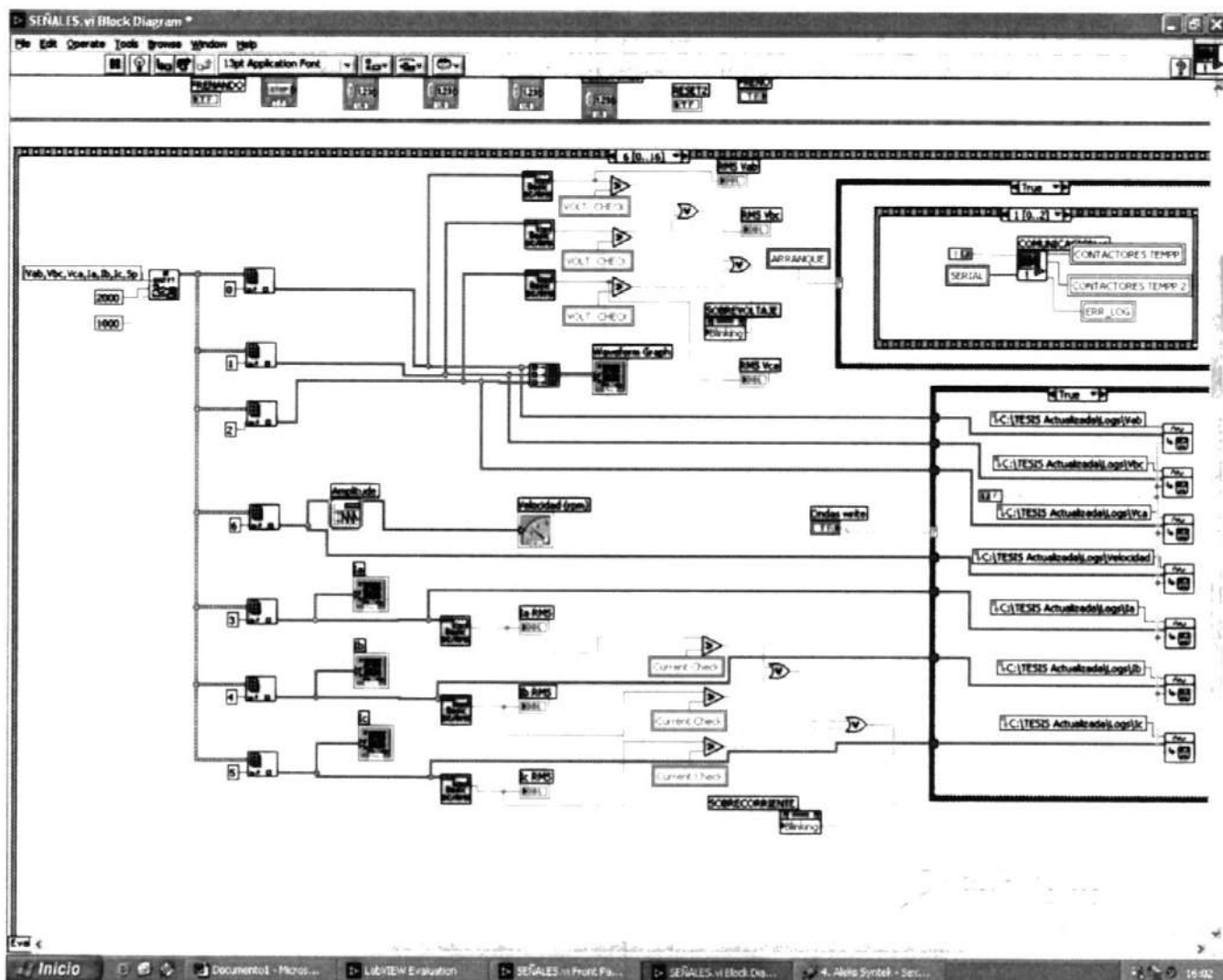


Figura 8.1, Diagrama de bloques para la adquisición de datos.

En esta parte de la aplicación se adquieren varias señales a la vez por medio de la función "AI Acquire Waveforms.vi" que se encuentra en la paleta de funciones "Data acquisition" dentro el menú "Analog Input". Esta función permite adquirir datos de varios canales a la vez (figura 8.2).

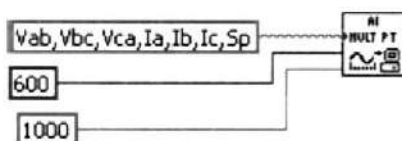


Figura 8.2, AI Acquire Waveforms.vi

Para la adquisición de datos se tomó en cuenta el criterio de Nyquist que manifiesta que la frecuencia de muestreo debe ser al menos 4 veces la frecuencia de la señal muestreada, por esta razón y por motivos experimentales se escogieron 600 muestras y un muestreo de 1000 muestras por segundo. La mayoría de las señales que se muestrean son de 60 Hz.

Para la adquisición de datos se usa el concepto de los canales virtuales que se explicaron en el capítulo 4. En la programación que se mostró se adquieren los 3 voltajes de línea, las 3 corrientes de línea y la velocidad.

La función "AI Sample Waveforms.vi" adquiere las señales y las agrupa en un arreglo (array) de tipo tabla (un array de tablas), en las cuales se guardan los valores de la variable muestreada vs el tiempo relativo al inicio de la adquisición.

Cuando ya se tiene el array de tablas se muestra la gráfica de cada señal, para lo cual hay que ingresar al array de tablas, seleccionar una tabla y luego mostrarla en un gráfico utilizando el objeto "Waveform Graph". Para la velocidad se utilizó un indicador numérico. Luego de extraer cada señal se procede a su análisis o al cálculo de valores DC y RMS y se los compara con los valores máximos permitidos para las diferentes conexiones del motor.

También se procede a grabar las señales si así lo permite el botón “GRABAR ONDAS”.

El procedimiento indicado para adquirir y manipular las ondas se lo utilizó en otras pantallas de manera idéntica, lo único que varió es la cantidad de señales que se adquirieron, por ejemplo para las pantallas de adquisición de datos de las pruebas del motor, sólo se adquieren el voltaje y la corriente de una fase.

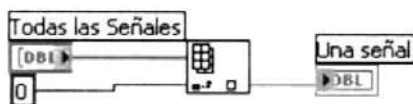
8.2. Análisis de las señales adquiridas.

El análisis de las señales de voltaje y corriente se lo realiza utilizando funciones incorporadas en LabView que permiten analizar ondas de una forma rápida y eficiente.

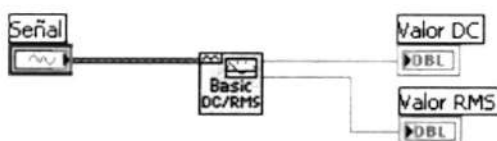
En LabView existen funciones que calculan valores DC y RMS de señales periódicas, otra calcula o indica la frecuencia de la señal, otra calcula el desfase entre dos señales y otras hacen análisis de las señales en el dominio de la frecuencia. También se detectan y

muestran los picos de voltajes y corrientes tanto negativos como positivos.

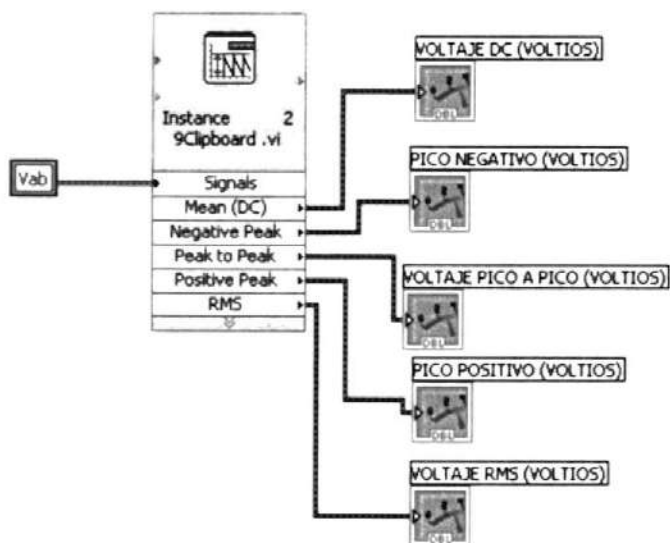
A continuación se explican las funciones utilizadas para realizar los diferentes cálculos o mediciones de las señales adquiridas:



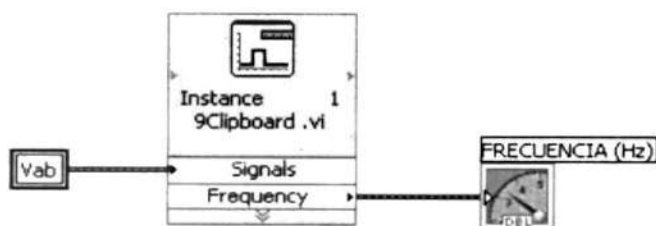
Index Array: Esta función permite extraer un elemento de un arreglo, en el caso de adquisición de datos extrae una señal (la posición 0 del ejemplo) del arreglo de señales que contiene algunas señales.



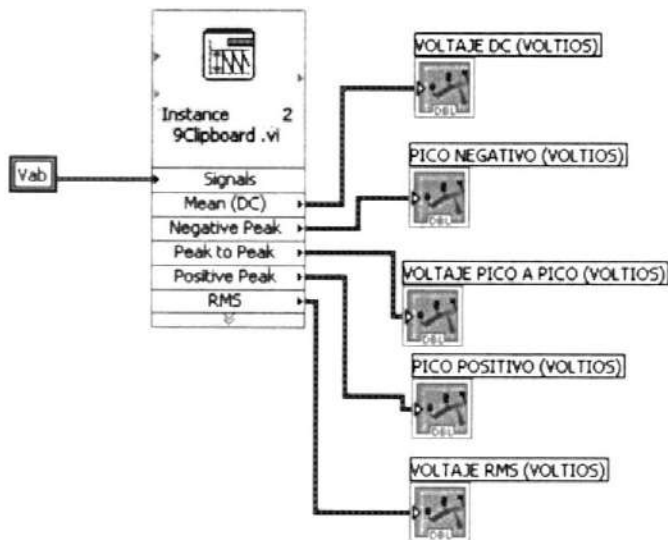
Basic DC/RMS: Esta función calcula el valor DC y el valor RMS de la señal que está ingresando en ella.



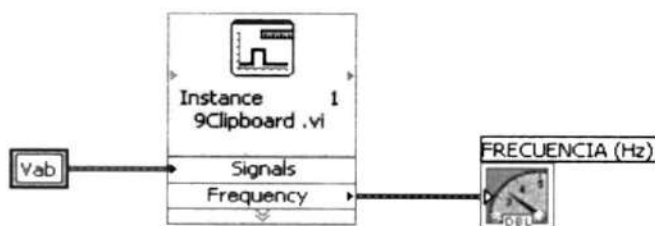
Amplitude and Level Measurements: Esta función contiene muchas funciones incorporadas, ingresa una señal (V_{ab}) y se obtienen como salidas el valor DC, el pico negativo, el pico positivo, el valor pico a pico y el valor RMS de la señal que ingresó.



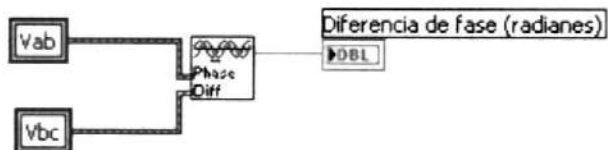
Timing and Transition Measurements: Esta función halla el valor de la frecuencia de la señal que ingresa (V_{ab}).



Amplitude and Level Measurements: Esta función contiene muchas funciones incorporadas, ingresa una señal (V_{ab}) y se obtienen como salidas el valor DC, el pico negativo, el pico positivo, el valor pico a pico y el valor RMS de la señal que ingresó.



Timing and Transition Measurements: Esta función halla el valor de la frecuencia de la señal que ingresa (V_{ab}).



Calculate Phase Difference:

Esta función calcula la diferencia de fase entre dos señales que ingresan en ella (Vab y Vbc), el resultado lo calcula en radianes.

Los valores RMS obtenidos se utilizan para comparar si los voltajes y corrientes son menores que los máximos permitidos en las diferentes conexiones del motor.

CAPITULO 9

9. BASE DE DATOS

9.1 Objetivos.

LabView ofrece la facilidad de comunicarse con dispositivos de adquisición de datos y también ofrece la posibilidad de grabar o registrar las señales que están siendo monitoreadas.

El objetivo principal de crear una base de datos se debe a la necesidad de poder analizar las señales que se obtienen y para lograr aquello se debe tener registradas las señales ya que en las pantallas de LabView se observa lo que está pasando (con un pequeño retraso debido al muestreo) y lo que pasó o los datos anteriores se borran de la pantalla.

Otros objetivos de la base de datos es mostrar las corrientes de arranque, cómo cambia la velocidad, cómo cambian las señales con cambios de carga, etc. Todo esto se puede analizar con los datos que se guardan en la base de datos.

Con la disponibilidad de los datos pertenecientes a las señales del motor en diversas situaciones, los alumnos estarán en capacidad de analizar y graficar otras curvas que sean de su interés, lo único que deben hacer es buscar la ecuación adecuada que relacione los datos que se necesitan para hacer la gráfica.

En la base de datos que se crea se guardan también los parámetros hallados en las pruebas de rotor bloqueado y de vacío del motor; también se registra si estuvieron conectadas resistencias externas en el caso de un rotor devanado y todos éstos parámetros son tomados en cuenta para el cálculo del torque inducido.

9.2 Estructura de la base de datos.

La base de datos se crea en EXCEL automáticamente desde LabView por medio de archivos de texto, archivos especiales de LabView para registrar ondas y ActiveX (el cual se explicó en el capítulo 1) .

La base de datos se forma de la siguiente manera:

- 1.- LabView adquiere y muestra las señales en tiempo casi real.

- 2.- Las señales adquiridas son grabadas o registradas en archivos especiales de LabView.

- 3.- Los parámetros calculados en las pruebas de rotor bloqueado y de vacío son guardados en archivos de texto.

4.- Mediante AciveX se abre un libro de EXCEL y se trasladan los valores de las señales que estaban guardadas en los archivos especiales de LabView y de los archivos de texto a celdas en EXCEL.

5.- Se crea una hoja de EXCEL para cada parte de señal que se quiere analizar.

De ésta manera se crea la base de datos, la cual queda configurada de la siguiente manera: en las tres primeras columnas (A,B y C) se ubican los valores de los voltajes de línea a línea, en las siguientes tres columnas (D,E y F) se colocan los valores de las corrientes de línea, en la siguiente columna (G) se colocan los valores correspondientes a la velocidad, en las siguientes tres columnas (H, I y J) se ubica la conexión que tenía el motor en el momento que se registraron las señales, en la siguiente columna (K) se muestra el valor de la resistencia externa conectada al rotor (0 si no hay ninguna resistencia conectada).

En la siguiente columna (L) se muestran los valores de las pruebas de rotor bloqueado de vacío y la resistencia del estator: R_1 , R_2 , X_1 y X_m , también se muestra el Voltaje RMS de cada fase.

En la siguiente columna (M) se muestran los valores de cálculos preliminares al cálculo del torque inducido, estos son: resistencia de Thevenin y el voltaje de Thevenin de cada fase. En la última columna (N) se muestra el Torque inducido en el motor.

Cuando se presiona "CREAR BASE DE DATOS" se debe prestar atención al número de registro que se muestra debajo de la gráfica de voltajes, este valor indica qué parte de la señal se está viendo, es decir, si el número es el cero se están mostrando las primeras 600 muestras de todas las señales grabadas y se creará la base de datos en una hoja de EXCEL con sólo las 600 primeras muestras de las señales, se puede seleccionar cualquier parte de la señal por medio de el control que dice "MOSTRAR REGISTRO", para cada parte de la señal (o 600 muestras) se crea una hoja en EXCEL, es decir una base de datos para cada pedazo de la señal. Para cerrar la aplicación se presiona "SALIR".

CAPITULO 10

10. SOFTWARE DE LA APLICACIÓN.

10.1 Programa en Assembler para el PIC.

LIST P=PIC16F877A

RADIX	HEX
INCLUDE	"P16F877A.INC"
INCLUDE	"PIC_CONF.ASM"
INCLUDE	"TECLCD.ASM"
INCLUDE	"TXDATOS.ASM"
INCLUDE	"VCONEXION.ASM"



CIB -ESPOL

TEMPORALREG_1	EQU	0x30
TEMPORALREG_2	EQU	0x31
TEMPORALREG_3	EQU	0x32
TEMPORALREG_4	EQU	0x33
R_ERROR	EQU	0x34
TEMPO	EQU	0x40
TEMPOX	EQU	0x41
OPCODE	EQU	0x50

OPCODEANT	EQU	0x51
CONFIG	EQU	0x52
TARRANQUE	EQU	0x53
C_ORDENADA	EQU	0x54
C_ORDENADA_2	EQU	0x55
C_EJECUTADA	EQU	0x56
C_EJECUTADA_2	EQU	0x57
STOPLOG	EQU	0x58
YDLOG	EQU	0x59
ROTDEVAN	EQU	0x60
RROTDEV1	EQU	0x61
RROTDEV2	EQU	0x62
TIMER1	EQU	0x48
TIMER2	EQU	0x49
TIMEYD	EQU	0x4A
ARRANQUEYD	EQU	0x66
ARRANCO	EQU	0x67
INICIO	EQU	0x68
GIRO_1	EQU	0x69
GIRO_2	EQU	0x6A
GIRO_X	EQU	0x6B
CGIRO	EQU	0x6C
PARAMETRO	EQU	0x35
TECLADOIN	EQU	0x36
SALIDA	EQU	0x37
TIEMPOCG	EQU	0x38
TIEMPOCG_2	EQU	0x39
TIEMPOCG_3	EQU	0x3A
DDRAM	EQU	0x3B
TIEMPOR1L	EQU	0x3C
TIEMPOR1H	EQU	0x3D
TIEMPOR2L	EQU	0x3E
TIEMPOR2H	EQU	0x3F
TIEMPOYDL	EQU	0x42
TIEMPOYDH	EQU	0x43
DATO_D	EQU	0x44
INTERPC	EQU	0x45
RESISTENCIAS	EQU	0x46
YEDELTA	EQU	0x47
INTER1	EQU	0x6D
INTER2	EQU	0x6E
INTER3	EQU	0x6F
INFLAG	EQU	0x70
COMPARAR	EQU	0x71


```

CONTADOR                EQU        0x72

        ORG        0x00
        GOTO       PROGRAM

        ORG        0x04
        GOTO       INTERRUPCION

PROGRAM
        PIC_CONF
        NOP
        TECLCD

PROGRAMA
        CLRF       CONTADOR
        CLRF       INICIO
        CLRF       INFLAG
        CLRF       R_ERROR
        CLRF       STOPLOG
        CLRF       C_ORDENADA_2
        BSF        INTCON,7
        CLRF       RROTDEV1
        CLRF       RROTDEV2
        BSF        PORTC,5
        BCF        PORTC,4
        CALL       RECEPCION_SET_UP

DEF_OPERACION
        MOVF       INTERPC,F
        BZ        PC_INTER

NO_PC_INTER
        BSF        PORTC,5
        BCF        PORTC,4
        GOTO       COMPARACION

SINPC
        MOVF       OPCODE,W
        MOVWF      OPCODEANT
        MOVF       TARRANQUE,F
        BZ        DIR_COMUN
        MOVLW     .1
        SUBWF     TARRANQUE,W

```

	BZ	R_COMUN
	MOVLW	.2
	SUBWF	TARRANQUE,W
	BZ	YD_COMUN
ESPERAR		
	MOVF	OPCODE,W
	MOVWF	OPCODEANT
	CALL	CHECKX
	MOVF	R_ERROR,F
	BNZ	INERROR_2
	BSF	C_ORDENADA_2,4
	BSF	PORTA,3
	MOVF	PORTE,W
	MOVWF	TECLADOIN
	BTFS	TECLADOIN,1
	GOTO	CGIROX
	BTFS	TECLADOIN,0
	GOTO	AMBASRESISTENCIAS
	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFS	TECLADOIN,4
	GOTO	RESISTENCIAUNO
	BTFS	TECLADOIN,5
	GOTO	RESISTENCIADOS
	GOTO	ESPERAR
RESISTENCIAUNO		
	CALL	SOLTAR
	MOVLW	.64
	MOVWF	RROTDEV1
	BSF	PORTA,2
	BSF	C_ORDENADA_2,6
	BCF	PORTA,1
	BCF	C_ORDENADA_2,5
	CLRF	RROTDEV2
	CALL	TEMPORIZADOR_3
	CALL	TEMPORIZADOR_3
	GOTO	ESPERAR
RESISTENCIADOS		
	CALL	SOLTAR
	MOVLW	.32
	MOVWF	RROTDEV2

BSF	PORTA,1
BSF	C_ORDENADA_2,5
BCF	PORTA,2
BCF	C_ORDENADA_2,6
CLRF	RROTDEV1
CALL	TEMPORIZADOR_3
CALL	TEMPORIZADOR_3
GOTO	ESPERAR
AMBASRESISTENCIAS	
CALL	SOLTAR
MOVLW	.64
MOVWF	RROTDEV1
MOVLW	.32
MOVWF	RROTDEV2
BSF	PORTA,1
BSF	C_ORDENADA_2,5
BSF	PORTA,2
BSF	C_ORDENADA_2,6
CALL	TEMPORIZADOR_3
CALL	TEMPORIZADOR_3
GOTO	ESPERAR
CGIROX	
CALL	SOLTAR
MOVF	TIEMPOCG_3,W
MOVWF	TIMER2
BTFSS	OPCODE,4
GOTO	DCERAUNO
GOTO	DUNOACER
DCERAUNO	
BSF	OPCODE,4
GOTO	CCG2
DUNOACER	
BCF	OPCODE,4
GOTO	CCG2
PC_INTER	
BCF	INTCON,7
CALL	CHECKX
MOVF	R_ERROR,F

	BNZ	INERROR_2
	BSF	INTCON,7
	CALL	TEMPORIZADOR_3
	CALL	TEMPORIZADOR_3
	MOVF	INFLAG,F
	SKPNZ	
	GOTO	PC_INTER
TYU	CLRF	INFLAG
	BCF	INTCON,7
	MOVF	OPCODE,W
	MOVWF	OPCODEANT
	MOVF	TARRANQUE,F
	BZ	DIR_COMUN
	MOVLW	.1
	SUBWF	TARRANQUE,W
	BZ	R_COMUN
	MOVLW	.2
	SUBWF	TARRANQUE,W
	BZ	YD_COMUN
	MOVLW	.255
	SUBWF	TARRANQUE,W
	BZ	PROGRAMA
	GOTO	PC_INTER

INTERRUPCION

;ROUTINA DE COMUNICACION

	INCF	CONTADOR
	MOVLW	.1
	SUBWF	CONTADOR,W
	BZ	DATO1
	MOVLW	.2
	SUBWF	CONTADOR,W
	BZ	DATO2
	MOVLW	.3
	SUBWF	CONTADOR,W
	BZ	DATO3
DATO1	MOVF	RCREG,W
	MOVWF	INTER1
	RETFIE	
DATO2	MOVF	RCREG,W
	MOVWF	INTER2

```

    RETFIE
DATO3
    MOVF      RCREG,W
    MOVWF    INTER3
    CLRF     CONTADOR
    MOVLW   .204
    SUBWF   INTER1,W
    BNZ     NOCONF
    GOTO    CONFIGURACION

CONFBACK
    CLRF     INFLAG
    RETFIE

NOCONF
    MOVF     INTER1,W
    MOVWF   OPCODE
CCG2
    MOVF     INICIO,F
    BNZ     CCG
    MOVF     OPCODE,W
    MOVWF   OPCODEANT

CCG
    MOVLW   B'00010000'
    ANDWF   OPCODE,W
    MOVWF   GIRO_1
    MOVLW   B'00010000'
    ANDWF   OPCODEANT,W
    MOVWF   GIRO_2
    MOVF    GIRO_1,W
    SUBWF   GIRO_2,W
    BZ      COMPARACION
    BCF     PORTB,7
    BCF     PORTB,4
    BCF     C_ORDENADA,7
    BCF     C_ORDENADA,4
    CALL    TEMPORIZADOR_3
    CALL    TEMPORIZADOR_3
    CALL    CHECKX
    MOVF    R_ERROR,F
    BNZ    INERROR_2

;REVISAR LA CONEXION ANTERIOR(SER/PAR)
    MOVLW   B'00100000'
    ANDWF   OPCODEANT,W

```

```

                SKPNZ
                GOTO          L32
                MOVLW        .32
                MOVWF        TEMPORALREG_4
                GOTO          L33
L32
                CLRF         TEMPORALREG_4
L33
;REVISAR EL SENTIDO DE GIRO ANTERIOR
                MOVLW        B'00010000'
                ANDWF        OPCODEANT,W
                SKPZ
                GOTO          L30
                MOVLW        .16
                MOVWF        GIRO_X
                GOTO          L31
L30
                CLRF         GIRO_X
L31
                MOVLW        B'00000111'
                ANDWF        OPCODEANT,W

;REVISAR OPCODE ANTERIOR
                MOVWF        TEMPORALREG_3
                MOVLW        .1
                MOVWF        TARRANQUE
                MOVLW        .1
                SUBWF        TEMPORALREG_3,W
                BZ            YER
                MOVLW        .2
                SUBWF        TEMPORALREG_3,W
                BZ            DELTAR
                MOVLW        .3
                SUBWF        TEMPORALREG_3,W
                BZ            DELTAR
                MOVLW        .4
                SUBWF        TEMPORALREG_3,W
                BZ            YER
                MOVLW        .5
                SUBWF        TEMPORALREG_3,W
                BZ            DELTAR

```

YER

MOVLW	.4
MOVWF	OPCODE
MOVF	GIRO_X,W
ADDWF	OPCODE,F
MOVF	TEMPORALREG_4,W
ADDWF	OPCODE,F
CLRF	TIMER1
GOTO	COMPARACION

DELTAR

MOVLW	.5
MOVWF	OPCODE
MOVF	GIRO_X,W
ADDWF	OPCODE,F
MOVF	TEMPORALREG_4,W
ADDWF	OPCODE,F
CLRF	TIMER1
GOTO	COMPARACION

COMPARACION

;ASIGNACIÓN DEL CÓDIGO PARA CADA CONEXIÓN

MOVLW	.17
SUBWF	OPCODE,W
BZ	YSD
MOVLW	.1
SUBWF	OPCODE,W
BZ	YSI
MOVLW	.50
SUBWF	OPCODE,W
BZ	DPD
MOVLW	.34
SUBWF	OPCODE,W
BZ	DPI
MOVLW	.18
SUBWF	OPCODE,W
BZ	DSD
MOVLW	.2
SUBWF	OPCODE,W
BZ	DSI
MOVLW	.49
SUBWF	OPCODE,W
BZ	YPD
MOVLW	.33
SUBWF	OPCODE,W

BZ	YPI
MOVLW	.20
SUBWF	OPCODE,W
BZ	YSDR
MOVLW	.4
SUBWF	OPCODE,W
BZ	YSIR
MOVLW	.52
SUBWF	OPCODE,W
BZ	YPDR
MOVLW	.36
SUBWF	OPCODE,W
BZ	YPIR
MOVLW	.21
SUBWF	OPCODE,W
BZ	DSDR
MOVLW	.5
SUBWF	OPCODE,W
BZ	DSIR
MOVLW	.53
SUBWF	OPCODE,W
BZ	DPDR
MOVLW	.37
SUBWF	OPCODE,W
BZ	DPIR
MOVLW	.19
SUBWF	OPCODE,W
BZ	YSD
MOVLW	.3
SUBWF	OPCODE,W
BZ	YDSI
MOVLW	.51
SUBWF	OPCODE,W
BZ	YDPD
MOVLW	.35
SUBWF	OPCODE,W
BZ	YDPI
GOTO	PROGRAMA

;ASIGNACIÓN DE LAS SALIDAS PARA ENERGIZAR
;LOS CONTACTORES NECESARIOS
YSD

MOVLW	B'11101000'
MOVWF	C_ORDENADA

YSI	CLRF GOTO	TARRANQUE FIN2
YPD	MOVLW MOVWF CLRF GOTO	B'01111000' C_ORDENADA TARRANQUE FIN2
YPI	MOVLW MOVWF CLRF GOTO	B'11100110' C_ORDENADA TARRANQUE FIN2
DSD	MOVLW MOVWF CLRF GOTO	B'01110110' C_ORDENADA TARRANQUE FIN2
DSI	MOVLW MOVWF CLRF GOTO	B'10101001' C_ORDENADA TARRANQUE FIN2
DPD	MOVLW MOVWF CLRF GOTO	B'00111001' C_ORDENADA TARRANQUE FIN2
DPI	MOVLW MOVWF CLRF GOTO	B'10100111' C_ORDENADA TARRANQUE FIN2
YSDR	MOVLW MOVWF MOVLW MOVWF GOTO	B'00110111' C_ORDENADA TARRANQUE FIN2
	MOVLW MOVWF MOVLW MOVWF GOTO	B'11001000' C_ORDENADA .1 TARRANQUE FIN2

YSIR	MOVLW MOVWF MOVLW MOVWF GOTO	B'01011000' C_ORDENADA .1 TARRANQUE FIN2
YPDR	MOVLW MOVWF MOVLW MOVWF GOTO	B'11000110' C_ORDENADA .1 TARRANQUE FIN2
YPIR	MOVLW MOVWF MOVLW MOVWF GOTO	B'01010110' C_ORDENADA .1 TARRANQUE FIN2
DSDR	MOVLW MOVWF MOVLW MOVWF GOTO	B'10001001' C_ORDENADA .1 TARRANQUE FIN2
DSIR	MOVLW MOVWF MOVLW MOVWF GOTO	B'00011001' C_ORDENADA .1 TARRANQUE FIN2
DPDR	MOVLW MOVWF MOVLW MOVWF GOTO	B'10000111' C_ORDENADA .1 TARRANQUE FIN2
DPIR	MOVLW MOVWF MOVLW MOVWF GOTO	B'00010111' C_ORDENADA .1 TARRANQUE FIN2

YDSD	MOVLW MOVWF MOVLW MOVWF GOTO	.2 TARRANQUE .1 ARRANQUEYD FIN2
YDSI	MOVLW MOVWF MOVLW MOVWF GOTO	.2 TARRANQUE .2 ARRANQUEYD FIN2
YDPD	MOVLW MOVWF MOVLW MOVWF GOTO	.2 TARRANQUE .3 ARRANQUEYD FIN2
YDPI	MOVLW MOVWF MOVLW MOVWF GOTO	.2 TARRANQUE .4 ARRANQUEYD FIN2
FIN2	MOVF SUBWF BZ CLRF CALL MOVF MOVWF MOVF BNZ	OPCODE,W OPCODEANT,W J7 PORTB TEMPORIZADOR_4 PORTD,W TEMPORALREG_3 TEMPORALREG_3,F INERROR_2
J7	MOVF BNZ MOVLW MOVWF	INTERPC,F J5 .1 INFLAG

```

                RETFIE
J5              MOVF      ARRANQUEYD,F
                BZ        SINPC
                MOVF      INICIO,F
                BZ        SINPC
                RETURN

```

```

;RUTINA PARA ARRANQUES DIRECTOS

```

```

DIR_COMUN

```

```

                MOVLW     .1
                MOVWF     INICIO
                BSF       C_ORDENADA_2,7
                MOVF      C_ORDENADA,W
                MOVWF     PORTB
                BSF       PORTA,0
                MOVF      RROTDEV1,F
                SKPZ
                BSF       PORTA,2
                MOVF      RROTDEV2,F
                SKPZ
                BSF       PORTA,1
                CALL      TEMPORIZADOR_3
                CALL      TEMPORIZADOR_3
                CALL      CHECKX
                MOVF      R_ERROR,F
                BZ        NOERROR
                GOTO      INERROR_2

```

```

;RUTINA DE PROCESAMIENTO DE ERRORES

```

```

INERROR_2

```

```

                CLRF      PORTA
                CLRF      PORTB
                MOVF      INTERPC,F
                BNZ       MOSTRAR_ERROR
                CALL      TEMPORIZADOR_3
                CALL      TEMPORIZADOR_3
                MOVF      INTERPC,F
                BNZ       J1
                CALL      TRANSMISION_SET_UP
                CALL      TXDATOS
                MOVF      C_EJECUTADA,W
                MOVWF     C_ORDENADA
                MOVF      C_EJECUTADA_2,W

```

```

MOVWF      C_ORDENADA_2
CALL       TEMPORIZADOR_3
CALL       TEMPORIZADOR_3
CALL       TRANSMISION_SET_UP
CALL       TXDATOS
MOVLW     .255
MOVWF     TARRANQUE
J1        CLRF      R_ERROR
          GOTO     PROGRAMA

NOERROR
          MOVF     INTERPC,F
          BNZ     J6
          CALL    TRANSMISION_SET_UP
          CALL    TXDATOS
          CALL    RECEPCION_SET_UP
          MOVF     ARRANQUEYD,F
          SKPNZ
          GOTO    PC_INTER
          RETURN

J6        MOVF     ARRANQUEYD,F
          SKPNZ
          GOTO    ESPERAR
          RETURN

;RUTINA PARA ARRANQUES CON RESISTENCIAS
R_COMUN
          MOVLW   .1
          MOVWF   INICIO
          BCF     C_ORDENADA_2,7 ;C9
          CLRF    TARRANQUE
          BCF     PORTA,0
          MOVF    TIMER1,F
          BZ     NOR1
L20       MOVF    C_ORDENADA,W
          MOVWF   PORTB
          MOVF    RROTDEV1,F
          SKPZ
          BSF    PORTA,2
          MOVF    RROTDEV2,F
          SKPZ
          BSF    PORTA,1
          CALL    TEMPORIZADOR_3

```

	CALL	TEMPORIZADOR_3
	CALL	CHECKX
	MOVF	R_ERROR,F
	BZ	NOERROR_2
	GOTO	INERROR_2
NOERROR_2		
	MOVF	INTERPC,F
	BNZ	J2
	CALL	TRANSMISION_SET_UP
	CALL	TXDATOS
J2	MOVF	TIMER1,W
	MOVWF	TEMPOX
	MOVF	TIMER1,F
	SKPZ	
	CALL	TEMPORIZADOR_1
	BSF	C_ORDENADA,5
	MOVF	C_ORDENADA,W
	MOVWF	PORTB
	CALL	TEMPORIZADOR_3
	CALL	TEMPORIZADOR_3
	CALL	CHECKX
	MOVF	R_ERROR,F
	BZ	NOERROR_3
	GOTO	INERROR_2
NOERROR_3		
	MOVF	INTERPC,F
	BNZ	J3
	CALL	TXDATOS
	NOP	
J3	MOVF	STOPLOG,F
	BNZ	KLI
	MOVF	TIMER2,F
	BZ	NOR2
	MOVF	TIMER2,W
	MOVWF	TEMPOX
	MOVF	TIMER2,F
	SKPZ	
	CALL	TEMPORIZADOR_1
L21	BSF	C_ORDENADA_2,7
	BSF	PORTA,0
	CALL	TEMPORIZADOR_3
	CALL	TEMPORIZADOR_3



	CALL	CHECKX
	MOVF	R_ERROR,F
	BZ	NOERROR_4
	GOTO	INERROR_2
KLI	BCF	C_ORDENADA_2,7
	BCF	PORTA,0
	GOTO	NOERROR_4
NOERROR_4	MOVF	INTERPC,F
	BNZ	ESPERAR
	CALL	TXDATOS
	CALL	RECEPCION_SET_UP
	GOTO	PC_INTER
NOR1	BSF	C_ORDENADA,5
	GOTO	L20
NOR2	BSF	C_ORDENADA_2,7
	GOTO	L21
;RUTINAS PARA ARRANQUES YE-DELTA		
YD_COMUN		
	MOVLW	.1
	MOVWF	INICIO
	MOVLW	.1
	SUBWF	ARRANQUEYD,W
	BZ	YDSD2
	MOVLW	.2
	SUBWF	ARRANQUEYD,W
	BZ	YDSI2
	MOVLW	.3
	SUBWF	ARRANQUEYD,W
	BZ	YDPD2
	MOVLW	.4
	SUBWF	ARRANQUEYD,W
	BZ	YDPI2

YDSD2

	CALL	YSD
	CALL	DIR_COMUN
	MOVF	TIMEYD,W
	MOVWF	TEMPOX
	CALL	TEMPORIZADOR_1
	CLRF	PORTB
	CALL	TEMPORIZADOR_4
	MOVF	PORTD,W
	MOVWF	TEMPORALREG_3
	MOVF	TEMPORALREG_3,F
	BNZ	INERROR_2
	CALL	DSD
	CALL	DIR_COMUN
	CLRF	ARRANQUEYD
	CLRF	TARRANQUE
	CLRF	INFLAG
	MOVF	INTERPC,F
	BNZ	ESPERAR
	GOTO	PC_INTER
YDSI2		
	CALL	YSI
	CALL	DIR_COMUN
	MOVF	TIMEYD,W
	MOVWF	TEMPOX
	CALL	TEMPORIZADOR_1
	CLRF	PORTB
	CALL	TEMPORIZADOR_4
	MOVF	PORTD,W
	MOVWF	TEMPORALREG_3
	MOVF	TEMPORALREG_3,F
	BNZ	INERROR_2
	CALL	DSI
	CALL	DIR_COMUN
	CLRF	ARRANQUEYD
	CLRF	TARRANQUE
	CLRF	INFLAG
	MOVF	INTERPC,F
	BNZ	ESPERAR
	GOTO	PC_INTER
YDPD2		
	CALL	YPD
	CALL	DIR_COMUN
	MOVF	TIMEYD,W


```

MOVWF    TEMPOX
CALL     TEMPORIZADOR_1
CLRF    PORTB
CALL     TEMPORIZADOR_4
MOVF    PORTD,W
MOVWF    TEMPORALREG_3
MOVF    TEMPORALREG_3,F
BNZ     INERROR_2
CALL     DPD
CALL     DIR_COMUN
CLRF    ARRANQUEYD
CLRF    TARRANQUE
CLRF    INFLAG
MOVF    INTERPC,F
BNZ     ESPERAR
GOTO    PC_INTER

YDPI2
CALL     YPI
CALL     DIR_COMUN
MOVF    TIMEYD,W
MOVWF    TEMPOX
CALL     TEMPORIZADOR_1
CLRF    PORTB
CALL     TEMPORIZADOR_4
MOVF    PORTD,W
MOVWF    TEMPORALREG_3
MOVF    TEMPORALREG_3,F
BNZ     INERROR_2
CALL     DPI
CALL     DIR_COMUN
CLRF    ARRANQUEYD
CLRF    TARRANQUE
CLRF    INFLAG
MOVF    INTERPC,F
BNZ     ESPERAR
GOTO    PC_INTER

;TEMPORIZADOR PROGRAMABLE DE SEGUNDOS
;VARIABLE TEMPO DEFINE LOS SEGUNDOS A TEMPORIZAR
TEMPORIZADOR_1
CLRF    TEMPORALREG_1
CLRF    TEMPO
LA2     CLRF    TMR0
LA1     CALL    CHECKX

```

	MOVF	R_ERROR,F
	BNZ	INERROR_2
	BTFSS	INTCON,2
	GOTO	LA1
	BCF	INTCON,2
	INCF	TEMPO
	MOVLW	.38
	SUBWF	TEMPO,W
	BZ	LA3
	GOTO	LA2
LA3		
	INCF	TEMPORALREG_1
	MOVF	TEMPOX,W
	SUBWF	TEMPORALREG_1,W
	SKPNZ	
	RETURN	
	CLRF	TEMPO
	GOTO	LA2
	;TEMPORIZACIÓN FIJA	
	TEMPORIZADOR_2	
	CLRF	TEMPO
	CLRF	TMR0
L4		
	BTFSS	INTCON,2
	GOTO	L4
	BCF	INTCON,2
	INCF	TEMPO
	MOVLW	.3
	SUBWF	TEMPO,W
	BZ	L5
	CLRF	TMR0
	GOTO	L4
L5		
	RETURN	
	;TEMPORIZACIÓN FIJA	
	TEMPORIZADOR_3	
	CLRF	TEMPO
	CLRF	TMR0
LE4	BTFSS	INTCON,2
	GOTO	LE4
	BCF	INTCON,2
	INCF	TEMPO

```

                MOVLW        .5
                SUBWF        TEMPO,W
                BZ            LE5
                CLRF         TMR0
                GOTO         LE4
LE5
                RETURN

;TEMPORIZACIÓN FIJA
TEMPORIZADOR_4
                CLRF         TEMPO
                CLRF         TMR0
LT4            BTFSS        INTCON,2
                GOTO         LT4
                BCF         INTCON,2
                INCF         TEMPO
                MOVLW        .2
                SUBWF        TEMPO,W
                BZ            LT5
                CLRF         TMR0
                GOTO         LT4
LT5
                RETURN

;PREPARA AL USART PARA LA TRANSMISIÓN
;A 9600 BAUDIOS, 8 BITS DE DATOS
;1 BIT DE INICIO, 1 BIT DE PARADA
;SIN PARIDAD NI CONTROL DE FLUJO.

TRANSMISION_SET_UP
                BCF         STATUS,RP0
                BCF         RCSTA,CREN
                BSF         STATUS,RP0
                MOVLW        .64
                MOVWF       SPBRG
                BCF         TXSTA,SYNC
                BSF         TXSTA,BRGH
                BSF         TXSTA,TXEN
                BCF         STATUS,RP0
                BSF         RCSTA,SPEN
                RETURN

;PREPARA AL USART PARA LA RECEPCIÓN

```

```

;A 9600 BAUDIOS, 8 BITS DE DATOS
;1 BIT DE INICIO, 1 BIT DE PARADA
;SIN PARIDAD NI CONTROL DE FLUJO.

```

```
RECEPCION_SET_UP
```

```

    BSF        STATUS,RP0
    BCF        STATUS,RP1
    MOVLW     .64
    MOVWF     SPBRG
    BCF        TXSTA,SYNC
    BSF        TXSTA,BRGH
    BCF        STATUS,RP0
    BSF        RCSTA,SPEN
    BSF        RCSTA,CREN
    RETURN

```

```
;RUTINAS PARA CONFIGURAR VARIABLES
```

```
;DE OPERACIÓN
```

```
CONFIGURACION
```

```

    CLRF      INFLAG
    CLRF      TARRANQUE

```

```
;VERIFICA SI HAY COMUNICACIÓN FIABLE
```

```
;CON EL PIC
```

```
PICCOM
```

```

    MOVLW     .47
    SUBWF     INTER2,W
    BNZ      STOP
    CALL      TRANSMISION_SET_UP
    CALL      TEMPORIZADOR_2
    CALL      TEMPORIZADOR_2
    MOVLW     .200
    MOVWF     TXREG
OKTX      BTFSS     PIR1,TXIF
          GOTO     OKTX
          BANKSEL  TXSTA
OKTX2     BTFSS     TXSTA,TRMT
          GOTO     OKTX2
          BCF      STATUS,RP0
          CALL     RECEPCION_SET_UP
          CLRF     TARRANQUE
          GOTO     CONFBACK

STOP
    MOVLW     .55

```

```

SUBWF      INTER2,W
BNZ        DEF_TIME_R1
MOVLW     .100
MOVWF     STOPLOG
CLRF      TARRANQUE
GOTO      CONFBACK

```

```

;DEFINE EL TIEMPO PARA LA RESISTENCIA 1
DEF_TIME_R1

```

```

MOVLW     .63
SUBWF     INTER2,W
BZ        DEFTIMER1
GOTO      DEF_TIME_R2

```

```

;DEFINE EL TIEMPO PARA LA RESISTENCIA 2
DEF_TIME_R2

```

```

MOVLW     .62
SUBWF     INTER2,W
BZ        DEFTIMER2
GOTO      DEF_TIME_YD

```

```

;DEFINE EL TIEMPO PARA EL CAMBIO DE YE A DELTA
DEF_TIME_YD

```

```

MOVLW     .61
SUBWF     INTER2,W
BZ        DEFTIMEYD
GOTO      ROTORDEV

```

```

;DEFINE CUÁNTAS RESISTENCIAS SE LE CONECTARÁN
;AL ROTOR DEVANADO
ROTORDEV

```

```

MOVLW     .58
SUBWF     INTER2,W
BZ        ROTORDEVANADO
GOTO      RESET

```

```

ROTORDEVANADO

```

```

MOVF      INTER3,W
MOVWF     ROTDEVAN
MOVLW     .1
SUBWF     ROTDEVAN,W
BZ        R_ROTDEV_1
MOVLW     .2
SUBWF     ROTDEVAN,W

```

	BZ	R_ROTDEV_2	
	MOVLW	.3	
	SUBWF	ROTDEVAN,W	
	BZ	R_ROTDEV_12	
	CLRF	RROTDEV1	
	CLRF	RROTDEV2	
	BCF	PORTA,1	
	BCF	PORTA,2	
	GOTO	V1	
R_ROTDEV_1			
	MOVLW	.64	;RC6
	MOVWF	RROTDEV1	
	BSF	PORTA,2	
	BSF	C_ORDENADA_2,6	
	BCF	PORTA,1	
	BCF	C_ORDENADA_2,5	
	CLRF	RROTDEV2	
	GOTO	V1	
R_ROTDEV_2			
	MOVLW	.32	;RC5
	MOVWF	RROTDEV2	
	BSF	PORTA,1	
	BSF	C_ORDENADA_2,5	
	BCF	PORTA,2	
	BCF	C_ORDENADA_2,6	
R_ROTDEV_12			
	MOVLW	.64	
	MOVWF	RROTDEV1	
	MOVLW	.32	
	MOVWF	RROTDEV2	
	CLRF	RROTDEV1	
	BSF	PORTA,1	
	BSF	C_ORDENADA_2,5	
	BSF	PORTA,2	
	BSF	C_ORDENADA_2,6	
	GOTO	V1	
V1			
	CALL	TEMPORIZADOR_3	
	CALL	TEMPORIZADOR_3	
	CLRF	TARRANQUE	
	GOTO	CONFBACK	

```
;DEFINE LOS TIEMPOS PARA LAS RESISTENCIAS
```

```
DEFTIMER1
```

```
    MOVF          INTER3,W
    MOVWF        TIMER1
    CALL         TEMPORIZADOR_3
    CLRF         TARRANQUE
    GOTO         CONFBACK
```

```
DEFTIMER2
```

```
    MOVF          INTER3,W
    MOVWF        TIMER2
    CALL         TEMPORIZADOR_3
    CLRF         TARRANQUE
    GOTO         CONFBACK
```

```
DEFTIMEYD
```

```
    MOVF          INTER3,W
    MOVWF        TIMEYD
    CALL         TEMPORIZADOR_3
    CLRF         TARRANQUE
    GOTO         CONFBACK
```

```
;HACE UN RESET POR SOFTWARE
```

```
RESET
```

```
    MOVLW        .60
    SUBWF        INTER2,W
    SKPZ
    GOTO         MAQ_DC_ON
    CALL         TEMPORIZADOR_3
    CLRF         PORTB
    CLRF         PORTC
    CLRF         PORTA
    CLRF         OPCODE
    CLRF         OPCODEANT
    NOP
    GOTO         PROGRAMA
```

```
MAQ_DC_ON
```

```
    MOVLW        .242
    SUBWF        INTER2,W
    SKPZ
    GOTO         MAQ_DC_OFF
    BSF         C_ORDENADA_2,4
```

```

                BSF          PORTA,3
                GOTO        CONFBACK

MAQ_DC_OFF
                MOVLW       .243
                SUBWF       INTER2,W
                SKPZ
                GOTO        PROGRAMA
                BCF         C_ORDENADA_2,4
                BCF         PORTA,3
                GOTO        CONFBACK

;RUTINA QUE MUESTRA MENSAJE DE ERROR
;EN LA LCD.
TABLA3        MOVLW       0x06
                MOVWF       PCLATH
                MOVLW       0xAA
                SUBWF       PARAMETRO,W
                ADDWF       PCL

MENSAJE_16
                RETLW       ''
                RETLW       ''
                RETLW       ''
                RETLW       ''
                RETLW       'E'
                RETLW       'R'
                RETLW       'R'
                RETLW       'O'
                RETLW       'R'
                RETLW       0x00

MOSTRAR_ERROR
                BSF         PORTC,4
                BCF         PORTC,5
                MOVLW       MENSAJE_16
                BZ          MOSTRAR_MENSAJE_3

MOSTRAR_MENSAJE_3
KI2           MOVWF       PARAMETRO
                CALL       TABLA3
                MOVWF       DATO_D
                MOVF        DATO_D,F

```



```

        BZ          ESPERAR2
        CALL        LCD_DATOS
        INCF        PARAMETRO
        GOTO        KI2

ESPERAR2
        CALL        TEMPORIZADOR_3
        CALL        TEMPORIZADOR_3
        CALL        BORRAR
        CALL        TEMPORIZADOR_3
        CALL        TEMPORIZADOR_3
        CALL        TEMPORIZADOR_3
        MOVF        PORTA,W
        MOVWF       TECLADOIN
        BTFSS       TECLADOIN,4
        GOTO        MOSTRAR_ERROR
        GOTO        PROGRAM

;LLAMADAS A MACROS
CHECKX
        CHECK_CONEXION
        RETURN

TXDATOS
        TX_DATOS
        RETURN

END
```

PIC_CONF MACRO

;CONFIGURACION INICIAL DEL PIC PARA TECLADO Y PANTALLA LCD

```

BCF          STATUS,RP0
CLRF        PORTA
CLRF        PORTB
CLRF        PORTC
CLRF        PORTD
CLRF        PORTE
MOVLW      B'11000000'
MOVWF      INTCON
BSF        STATUS,RP0
MOVLW      B'00100000'
MOVWF      PIE1
MOVLW      0x30
MOVWF      TRISA
CLRF        TRISB
MOVLW      0xCF
MOVWF      TRISC
MOVLW      0xFF
MOVWF      TRISD
MOVLW      0x03
MOVWF      TRISE

```

; CONFIGURACION DEL TMR0

```

MOVLW      B'00000111'
MOVWF      OPTION_REG

```

;configuración del modulo analógico / digital

```

BCF          STATUS,RP0
MOVLW      .1
MOVWF      INTERPC
BSF        STATUS,RP0
MOVLW      B'00000110'
MOVWF      ADCON1
BCF        STATUS,RP0
CLRF        OPCODE
CLRF        OPCODEANT
CLRF        TEMPORALREG_1
CLRF        TEMPORALREG_2
CLRF        TEMPORALREG_3
CLRF        TEMPORALREG_4
CLRF        TEMPO

```

```
CLRF      TEMPOX
CLRF      ARRANCO
CLRF      TARRANQUE
CLRF      C_ORDENADA
CLRF      C_ORDENADA_2
CLRF      C_EJECUTADA
CLRF      C_EJECUTADA_2
CLRF      RROTDEV1
CLRF      RROTDEV2
CLRF      SALIDA
CLRF      ARRANQUEYD
CLRF      PORTB
CLRF      PORTA
CLRC

ENDM
```

TECLCD	MACRO	
TABLA	GOTO	MENU_INICIAL
	MOVLW	0x00
	MOVWF	PCLATH
	MOVLW	0X3E
	SUBWF	PARAMETRO,W
	ADDWF	PCL
MENSAJE_1		
	RETLW	'K'
	RETLW	'A'
	RETLW	'T'
	RETLW	'O'
	RETLW	''
	RETLW	'C'
	RETLW	'O'
	RETLW	'N'
	RETLW	'T'
	RETLW	'R'
	RETLW	'O'
	RETLW	'L'
	RETLW	0x00
MENSAJE_2_1		
	RETLW	'T'
	RETLW	'I'
	RETLW	'P'
	RETLW	'O'
	RETLW	''
	RETLW	'D'
	RETLW	'E'
	RETLW	''
	RETLW	'A'
	RETLW	'R'
	RETLW	'R'
	RETLW	'A'
	RETLW	'N'
	RETLW	'Q'
	RETLW	'U'
	RETLW	'E'
	RETLW	':'
	RETLW	0x00

```
MENSAJE_2_2
    RETLW    'I'
    RETLW    ')'
    RETLW    'D'
    RETLW    'I'
    RETLW    'R'
    RETLW    ''
    RETLW    ''
    RETLW    '2'
    RETLW    ')'
    RETLW    'C'
    RETLW    'O'
    RETLW    'N'
    RETLW    ''
    RETLW    'R'
    RETLW    ''
    RETLW    ''
    RETLW    '3'
    RETLW    ')'
    RETLW    'Y'
    RETLW    'D'
    RETLW    0x00
```

```
MENSAJE_3_1
    RETLW    'C'
    RETLW    'O'
    RETLW    'N'
    RETLW    'E'
    RETLW    'X'
    RETLW    'I'
    RETLW    'O'
    RETLW    'N'
    RETLW    ':'
    RETLW    0x00
```

```
MENSAJE_3_2
    RETLW    'I'
    RETLW    ')'
    RETLW    'Y'
    RETLW    'E'
    RETLW    ''
    RETLW    ''
    RETLW    ''
    RETLW    ''
```

```
RETLW    ''  
RETLW    '2'  
RETLW    ')''  
RETLW    'D'  
RETLW    'E'  
RETLW    'L'  
RETLW    'T'  
RETLW    'A'  
RETLW    0x00
```

MENSAJE_4_1

```
RETLW    'B'  
RETLW    'O'  
RETLW    'B'  
RETLW    'I'  
RETLW    'N'  
RETLW    'A'  
RETLW    'S'  
RETLW    ''  
RETLW    ''  
RETLW    'E'  
RETLW    'N'  
RETLW    ':'  
RETLW    0x00
```

MENSAJE_4_2

```
RETLW    'I'  
RETLW    ')''  
RETLW    'S'  
RETLW    'E'  
RETLW    'R'  
RETLW    'I'  
RETLW    'E'  
RETLW    ''  
RETLW    ''  
RETLW    '2'  
RETLW    ')''  
RETLW    'P'  
RETLW    'A'  
RETLW    'R'  
RETLW    'A'  
RETLW    'L'  
RETLW    'E'  
RETLW    'L'
```

```
        RETLW    'O'
        RETLW    0x00

MENSAJE_5
        RETLW    'R'
        RETLW    ''
        RETLW    'D'
        RETLW    'E'
        RETLW    ''
        RETLW    'C'
        RETLW    'A'
        RETLW    'M'
        RETLW    'B'
        RETLW    'T'
        RETLW    'O'
        RETLW    ''
        RETLW    'D'
        RETLW    'E'
        RETLW    ''
        RETLW    'G'
        RETLW    'T'
        RETLW    'R'
        RETLW    'O'
        RETLW    ':'
        RETLW    0x00

MENSAJE_6_1
        RETLW    'S'
        RETLW    'E'
        RETLW    'N'
        RETLW    'T'
        RETLW    'I'
        RETLW    'D'
        RETLW    'O'
        RETLW    ''
        RETLW    'D'
        RETLW    'E'
        RETLW    ''
        RETLW    'G'
        RETLW    'T'
        RETLW    'R'
        RETLW    'O'
        RETLW    ':'
        RETLW    0x00
```

```
MENSAJE_6_2
    RETLW    'I'
    RETLW    ')'
    RETLW    'D'
    RETLW    'E'
    RETLW    'R'
    RETLW    ':'
    RETLW    ''
    RETLW    ''
    RETLW    '2'
    RETLW    ')'
    RETLW    'I'
    RETLW    'Z'
    RETLW    'Q'
    RETLW    ':'
    RETLW    0x00
```

```
MENSAJE_7
    RETLW    'D'
    RETLW    'E'
    RETLW    'S'
    RETLW    'C'
    RETLW    'O'
    RETLW    'N'
    RETLW    'E'
    RETLW    'C'
    RETLW    'T'
    RETLW    'A'
    RETLW    'R'
    RETLW    ''
    RETLW    'R'
    RETLW    'I'
    RETLW    ''
    RETLW    'E'
    RETLW    'N'
    RETLW    ':'
    RETLW    0x00
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
```


NOP
 NOP
 NOP
 NOP
 NOP
 NOP
 NOP
 NOP

TABLA_2

MOVLW	0x01
MOVWF	PCLATH
MOVLW	0x09
SUBWF	PARAMETRO,W
ADDWF	PCL

MENSAJE_8

RETLW	'D'
RETLW	'E'
RETLW	'S'
RETLW	'C'
RETLW	'O'
RETLW	'N'
RETLW	'E'
RETLW	'C'
RETLW	'T'
RETLW	'A'
RETLW	'R'
RETLW	''
RETLW	'R'
RETLW	'2'
RETLW	''
RETLW	'E'
RETLW	'N'
RETLW	':'
RETLW	0x00

MENSAJE_9

RETLW	'C'
RETLW	'A'
RETLW	'M'
RETLW	'B'
RETLW	'T'
RETLW	'O'

```
RETLW    ''  
RETLW    'Y'  
RETLW    'E'  
RETLW    'L'  
RETLW    'D'  
RETLW    'E'  
RETLW    'L'  
RETLW    'T'  
RETLW    'A'  
RETLW    ''  
RETLW    'E'  
RETLW    'N'  
RETLW    ':'  
RETLW    0x00
```

MENSAJE_10

```
RETLW    'S'  
RETLW    'E'  
RETLW    'G'  
RETLW    'U'  
RETLW    'N'  
RETLW    'D'  
RETLW    'O'  
RETLW    'S'  
RETLW    0x00
```

MENSAJE_11_1

```
RETLW    'C'  
RETLW    'O'  
RETLW    'N'  
RETLW    'E'  
RETLW    'X'  
RETLW    'I'  
RETLW    'O'  
RETLW    'N'  
RETLW    ''  
RETLW    'C'  
RETLW    'O'  
RETLW    'N'  
RETLW    ''  
RETLW    'P'  
RETLW    'C'  
RETLW    '?'  
RETLW    0x00
```

```
MENSAJE_11_2
    RETLW    'I'
    RETLW    ')'
    RETLW    'S'
    RETLW    'T'
    RETLW    ''
    RETLW    ''
    RETLW    ''
    RETLW    ''
    RETLW    ''
    RETLW    ''
    RETLW    ''
    RETLW    '2'
    RETLW    ')'
    RETLW    'N'
    RETLW    'O'
    RETLW    0x00
```

```
MENSAJE_12
    RETLW    'F'
    RETLW    'U'
    RETLW    'N'
    RETLW    'C'
    RETLW    'T'
    RETLW    'O'
    RETLW    'N'
    RETLW    'A'
    RETLW    'N'
    RETLW    'D'
    RETLW    'O'
    RETLW    0x00
```

```
MENSAJE_13
    RETLW    'I'
    RETLW    'N'
    RETLW    'T'
    RETLW    'E'
    RETLW    'R'
    RETLW    'F'
    RETLW    'A'
    RETLW    'S'
    RETLW    'E'
    RETLW    ''
```

```

RETLW    'C'
RETLW    'O'
RETLW    'N'
RETLW    ''
RETLW    'P'
RETLW    'C'
RETLW    0x00

```

MENSAJE_14

```

RETLW    'E'
RETLW    'R'
RETLW    'R'
RETLW    'O'
RETLW    'R'
RETLW    ''
RETLW    'I'
RETLW    0x00

```

MENSAJE_15

```

RETLW    'C'
RETLW    'O'
RETLW    'N'
RETLW    'T'
RETLW    'R'
RETLW    'O'
RETLW    'L'
RETLW    ''
RETLW    'C'
RETLW    'O'
RETLW    'N'
RETLW    ''
RETLW    'P'
RETLW    'C'
RETLW    0x00

```



CIB -ESPOL

MENU_INICIAL

```

BCF      STATUS,RP0
CLRF     INTER1
CLRF     INTER2
CLRF     INTER3
CLRF     TIMER1
CLRF     TIMER2
CLRF     TIMEYD

```

	BSF	PORTC,4
	BCF	PORTC,5
	MOVF	INTERPC,F
	BZ	PRE_OPERA
	CLRF	INTERPC
	BCF	PORTA,2
	BCF	PORTA,0
	CALL	LCD_INI
	MOVLW	b'00001100'
	CALL	LCD_REG
	MOVLW	b'00000001'
	CALL	LCD_REG
	MOVLW	MENSAJE_1
	CALL	MOSTRAR_MENSAJE
TECLADO		
	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFSC	TECLADOIN,4
	GOTO	INTERFASE
	GOTO	TECLADO
INTERFASE		
	CLRF	INTERPC
	CALL	SOLTAR
	CALL	BORRAR
	MOVLW	MENSAJE_11_1
	CALL	MOSTRAR_MENSAJE_2
	MOVLW	B'11000000'
	CALL	LCD_REG
	MOVLW	MENSAJE_11_2
	CALL	MOSTRAR_MENSAJE_2
IUO	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFSC	TECLADOIN,4
	GOTO	PRE_OPERA
	BTFSC	TECLADOIN,5
	GOTO	TECLAUNO
	GOTO	IUO
TECLAUNO		
	CALL	SOLTAR
	CALL	BORRAR
	CLRF	RESISTENCIAS

	MOVLW	MENSAJE_2_1
	CALL	MOSTRAR_MENSAJE
	MOVLW	B'11000000'
	CALL	LCD_REG
	MOVLW	MENSAJE_2_2
T1	CALL	MOSTRAR_MENSAJE
	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFSC	TECLADOIN,4
	GOTO	MENU_3
	BTFSC	TECLADOIN,5
	GOTO	MENU_7
	MOVF	PORTE,W
	MOVWF	TECLADOIN
	BTFSC	TECLADOIN,0
	GOTO	MENU_9
	GOTO	T1
MENU_3		
	CALL	SOLTAR
	CALL	BORRAR
	MOVLW	MENSAJE_3_1
	CALL	MOSTRAR_MENSAJE
	MOVLW	B'11000000'
	CALL	LCD_REG
	MOVLW	MENSAJE_3_2
T2	CALL	MOSTRAR_MENSAJE
	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFSC	TECLADOIN,4
	GOTO	SYE
	BTFSC	TECLADOIN,5
	GOTO	SDE
	GOTO	T2
SYE		
	MOVF	RESISTENCIAS,F
	BZ	NORESISTENCIAS
	MOVLW	.4
	MOVWF	SALIDA
	GOTO	MENU_4
NORESISTENCIAS		
	CLRF	RESISTENCIAS

	MOVLW	.1
	MOVWF	SALIDA
SDE	GOTO	MENU_4
	MOVF	RESISTENCIAS,F
	BZ	NORESISTENCIAS_2
	MOVLW	.5
	MOVWF	SALIDA
	GOTO	MENU_4
NORESISTENCIAS_2	CLRF	RESISTENCIAS
	MOVLW	.2
	MOVWF	SALIDA
	GOTO	MENU_4
SOLTAR	MOVF	PORTA,W
	MOVWF	TECLADOIN
	MOVLW	B'11000000'
	ANDWF	TECLADOIN,F
	MOVF	TECLADOIN,F
	SKPZ	
	GOTO	SOLTAR
	MOVF	PORTE,W
	MOVWF	TECLADOIN
	MOVLW	B'011'
	ANDWF	TECLADOIN,F
	MOVF	TECLADOIN,F
	SKPZ	
	GOTO	SOLTAR
	RETURN	
MENU_4	CALL	SOLTAR
	CALL	BORRAR
	MOVLW	MENSAJE_4_1
	CALL	MOSTRAR_MENSAJE
	MOVLW	B'11000000'
	CALL	LCD_REG
	MOVLW	MENSAJE_4_2
	CALL	MOSTRAR_MENSAJE
T3	MOVF	PORTA,W
	MOVWF	TECLADOIN

	BTFSC	TECLADOIN,4
	GOTO	SSERIE
	BTFSC	TECLADOIN,5
	GOTO	SPARALELO
	GOTO	T3
SSERIE		
	MOVLW	.0
	ADDWF	SALIDA,F
	GOTO	MENU_5
SPARALELO		
	MOVLW	.32
	ADDWF	SALIDA,F
	GOTO	MENU_5
MENU_5		
	CALL	SOLTAR
	CALL	BORRAR
	MOVLW	MENSAJE_5
	CALL	MOSTRAR_MENSAJE
	CLRF	TIEMPOCG
	CLRF	TIEMPOCG_2
	CLRF	TIEMPOCG_3
LOP3		
	MOVLW	0xC5
	CALL	LCD_REG
	MOVLW	.10
	SUBWF	TIEMPOCG,W
	BZ	MAYOR
LOP2		
	MOVLW	0x30
	ADDWF	TIEMPOCG,W
	CALL	LCD_DATOS
	MOVLW	0xC7
	CALL	LCD_REG
	MOVLW	MENSAJE_10
	CALL	MOSTRAR_MENSAJE_2
LOP		
	MOVF	PORTE,W
	MOVWF	TECLADOIN
	BTFSC	TECLADOIN,0
	GOTO	PLUS
	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFSC	TECLADOIN,4
	GOTO	MENU_6
	GOTO	LOP

PLUS	CALL	SOLTAR
	INCF	TIEMPOCG
	INCF	TIEMPOCG_3,F
	GOTO	LOP3
MAYOR	MOVLW	0xC4
	CALL	LCD_REG
	INCF	TIEMPOCG_2,F
	MOVLW	0x30
	ADDWF	TIEMPOCG_2,W
	CALL	LCD_DATOS
	CLRF	TIEMPOCG
	GOTO	LOP3
MENU_6	CALL	SOLTAR
	CALL	BORRAR
	MOVLW	MENSAJE_6_1
	CALL	MOSTRAR_MENSAJE
	MOVLW	B'11000000'
	CALL	LCD_REG
	MOVLW	MENSAJE_6_2
	CALL	MOSTRAR_MENSAJE
T4	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFS	TECLADOIN,4
	GOTO	SDER
	BTFS	TECLADOIN,5
	GOTO	SIZQ
	GOTO	T4
SDER	MOVLW	.16
	ADDWF	SALIDA,F
	MOVF	SALIDA,W
	MOVWF	OPCODE
	GOTO	FUNCIONANDO
SIZQ	MOVLW	.0
	ADDWF	SALIDA,F
	MOVF	SALIDA,W
	MOVWF	OPCODE
	GOTO	FUNCIONANDO

```

MENU_7
    CALL    SOLTAR
    CALL    BORRAR
    MOVLW  .1
    MOVWF  RESISTENCIAS
    MOVLW  MENSAJE_7
    CALL   MOSTRAR_MENSAJE
    CLRF   TIEMPOR1L
    CLRF   TIEMPOR1H
LOPP3    MOVLW  0xC5
         CALL   LCD_REG
         MOVLW  .10
         SUBWF  TIEMPOR1L,W
         BZ     MAYOR2
LOPP2    MOVLW  0x30
         ADDWF  TIEMPOR1L,W
         CALL   LCD_DATOS
         MOVLW  0xC7
         CALL   LCD_REG
         MOVLW  MENSAJE_10
         CALL   MOSTRAR_MENSAJE_2
LOPP     MOVF   PORTE,W
         MOVWF  TECLADOIN
         BTFSC  TECLADOIN,0
         GOTO   PLUS2
         MOVF   PORTA,W
         MOVWF  TECLADOIN
         BTFSC  TECLADOIN,4
         GOTO   MENU_8
         GOTO   LOPP
PLUS2
         CALL   SOLTAR
         INCF   TIEMPOR1L,F
         INCF   TIMER1,F
         GOTO   LOPP3
MAYOR2
         MOVLW  0xC4
         CALL   LCD_REG
         INCF   TIEMPOR1H,F
         MOVLW  0x30
         ADDWF  TIEMPOR1H,W
         CALL   LCD_DATOS
         CLRF   TIEMPOR1L

```

	GOTO	LOPP3
MENU_8	CALL	SOLTAR
	CALL	BORRAR
	MOVLW	MENSAJE_8
	CALL	MOSTRAR_MENSAJE_2
	CLRF	TIEMPOR2L
	CLRF	TIEMPOR2H
OP3	MOVLW	0xC5
	CALL	LCD_REG
	MOVLW	.10
	SUBWF	TIEMPOR2L,W
	BZ	MAYOR3
OP2	MOVLW	0x30
	ADDWF	TIEMPOR2L,W
	CALL	LCD_DATOS
	MOVLW	0xC7
	CALL	LCD_REG
	MOVLW	MENSAJE_10
	CALL	MOSTRAR_MENSAJE_2
OP	MOVF	PORTE,W
	MOVWF	TECLADOIN
	BTFSF	TECLADOIN,0
	GOTO	PLUS3
	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFSF	TECLADOIN,4
	GOTO	MENU_3
	GOTO	OP
PLUS3	CALL	SOLTAR
	INCF	TIEMPOR2L,F
	INCF	TIMER2,F
	GOTO	OP3
MAYOR3	MOVLW	0xC4
	CALL	LCD_REG
	INCF	TIEMPOR2H,F
	MOVLW	0x30
	ADDWF	TIEMPOR2H,W
	CALL	LCD_DATOS
	CLRF	TIEMPOR2L

MENU_9	GOTO	OP3
	CALL	SOLTAR
	CALL	BORRAR
	MOVLW	.1
	MOVWF	YEDELTA
	MOVLW	.3
	MOVWF	SALIDA
	MOVLW	MENSAJE_9
	CALL	MOSTRAR_MENSAJE_2
	CLRF	TIEMPOYDL
	CLRF	TIEMPOYDH
LLOP3	MOVLW	0xC5
	CALL	LCD_REG
	MOVLW	.10
	SUBWF	TIEMPOYDL,W
	BZ	MAYOR4
LLOP2	MOVLW	0x30
	ADDWF	TIEMPOYDL,W
	CALL	LCD_DATOS
	MOVLW	0xC7
	CALL	LCD_REG
	MOVLW	MENSAJE_10
	CALL	MOSTRAR_MENSAJE_2
LLOP	MOVF	PORTE,W
	MOVWF	TECLADOIN
	BTFSF	TECLADOIN,0
	GOTO	PLUS4
	MOVF	PORTA,W
	MOVWF	TECLADOIN
	BTFSF	TECLADOIN,4
	GOTO	MENU_4
	GOTO	LLOP
PLUS4		
	CALL	SOLTAR
	INCF	TIEMPOYDL,F
	INCF	TIMEYD,F
	GOTO	LLOP3
MAYOR4		
	MOVLW	0xC4
	CALL	LCD_REG
	INCF	TIEMPOYDH,F
	MOVLW	0x30

	ADDWF	TIEMPOYDH,W
	CALL	LCD_DATOS
	CLRF	TIEMPOYDL
	GOTO	LLOP3
MOSTRAR_MENSAJE		
	MOVWF	PARAMETRO
KI	CALL	TABLA
	MOVWF	DATO_D
	MOVF	DATO_D,F
	SKPNZ	
	RETURN	
	CALL	LCD_DATOS
	INCF	PARAMETRO
	GOTO	KI
MOSTRAR_MENSAJE_2		
	MOVWF	PARAMETRO
KII	CALL	TABLA_2
	MOVWF	DATO_D
	MOVF	DATO_D,F
	SKPNZ	
	RETURN	
	CALL	LCD_DATOS
	INCF	PARAMETRO
	GOTO	KII
BORRAR		
	MOVLW	B'00000001'
	CALL	LCD_REG
	RETURN	
LCD_BUSY		
	NOP	
L_BUSY		
	CALL	TEMPORIZADOR_4
	BCF	PORTA,0
	BSF	STATUS,5
	CLRF	TRISB
	BCF	STATUS,5
	BCF	PORTA,1
	RETURN	

```

LCD_E
    BSF          PORTA,0
    NOP
    BCF          PORTA,0
    RETURN

LCD_DATOS
    BCF          PORTA,2
    MOVWF       PORTB
    CALL        LCD_BUSY
    BSF          PORTA,2
    GOTO        LCD_E

LCD_REG
    BCF          PORTA,2
    MOVWF       PORTB
    CALL        LCD_BUSY
    GOTO        LCD_E

LCD_INI
    MOVLW       B'00111000'
    CALL        LCD_REG
    CALL        DELAY_5MS
    MOVLW       B'00111000'
    CALL        LCD_REG
    CALL        DELAY_5MS
    MOVLW       B'00111000'
    CALL        LCD_REG
    CALL        DELAY_5MS
    RETURN

DELAY_5MS
    CALL        TEMPORIZADOR_2
    RETURN

FUNCIONANDO
    CALL        SOLTAR
    CALL        BORRAR
    MOVLW       B'10000100'
    CALL        LCD_REG
    MOVLW       MENSAJE_12
    CALL        MOSTRAR_MENSAJE_2
    MOVLW       .1

```

	MOVWF	INTERPC
	CLRF	PORTB
	CLRF	PORTA
	CLRF	PORTC
	GOTO	FIN1
PRE_OPERA		
	CALL	BORRAR
	MOVLW	MENSAJE_15
	CALL	MOSTRAR_MENSAJE_2
	CLRF	PORTB
	CLRF	PORTA
	CLRF	PORTC
	BSF	PORTC,5
	BCF	PORTC,4
	MOVLW	B'11000000'
	MOVWF	INTCON
	BSF	STATUS,RP0
	MOVLW	B'00100000'
	MOVWF	PIE1
	BCF	STATUS,RP0
	CALL	RECEPCION_SET_UP
FIN1		
	NOP	
	ENDM	



CIB - ESPOL

```

TX_DATOS      MACRO

                CALL      TEMPORIZADOR_2
                CALL      TEMPORIZADOR_2
                MOVF      C_ORDENADA,W
                MOVWF     TXREG
NOTX           BTFSS     PIR1,TXIF
                GOTO     NOTX
                CALL      TEMPORIZADOR_2
                MOVF      C_ORDENADA_2,W
                MOVWF     TXREG
NOTX2         BTFSS     PIR1,TXIF
                GOTO     NOTX2
                CALL      TEMPORIZADOR_2
                MOVF      R_ERROR,W
                MOVWF     TXREG
NOTX3         BTFSS     PIR1,TXIF
                GOTO     NOTX3
                BANKSEL  TXSTA
NOTX4         BTFSS     TXSTA,TRMT
                GOTO     NOTX4
                BCF      STATUS,RP0

                ENDM

```



```

CHECK_CONEXION      MACRO

    MOVF             INICIO,F
    BZ               FIN3
    MOVF             PORTD,W
    MOVWF           C_EJECUTADA
    MOVF             PORTC,W
    MOVWF           C_EJECUTADA_2
    MOVLW           B'00001111'
    ANDWF           C_EJECUTADA_2,F
    SWAPF           C_EJECUTADA_2,F
    NOP
    NOP
    NOP
    MOVF             C_ORDENADA,W
    SUBWF           C_EJECUTADA,W
    BNZ             INERROR
    MOVF             C_ORDENADA_2,W
    SUBWF           C_EJECUTADA_2,W
    BNZ             INERROR
    MOVF             C_ORDENADA,W
    MOVWF           C_EJECUTADA
    MOVF             C_ORDENADA_2,W
    MOVWF           C_EJECUTADA_2
    CLRF            R_ERROR
    GOTO            FIN3

INERROR
    MOVLW           .1
    MOVWF           R_ERROR

FIN3
    NOP
ENDM

```

10.2 Programa en LabView.

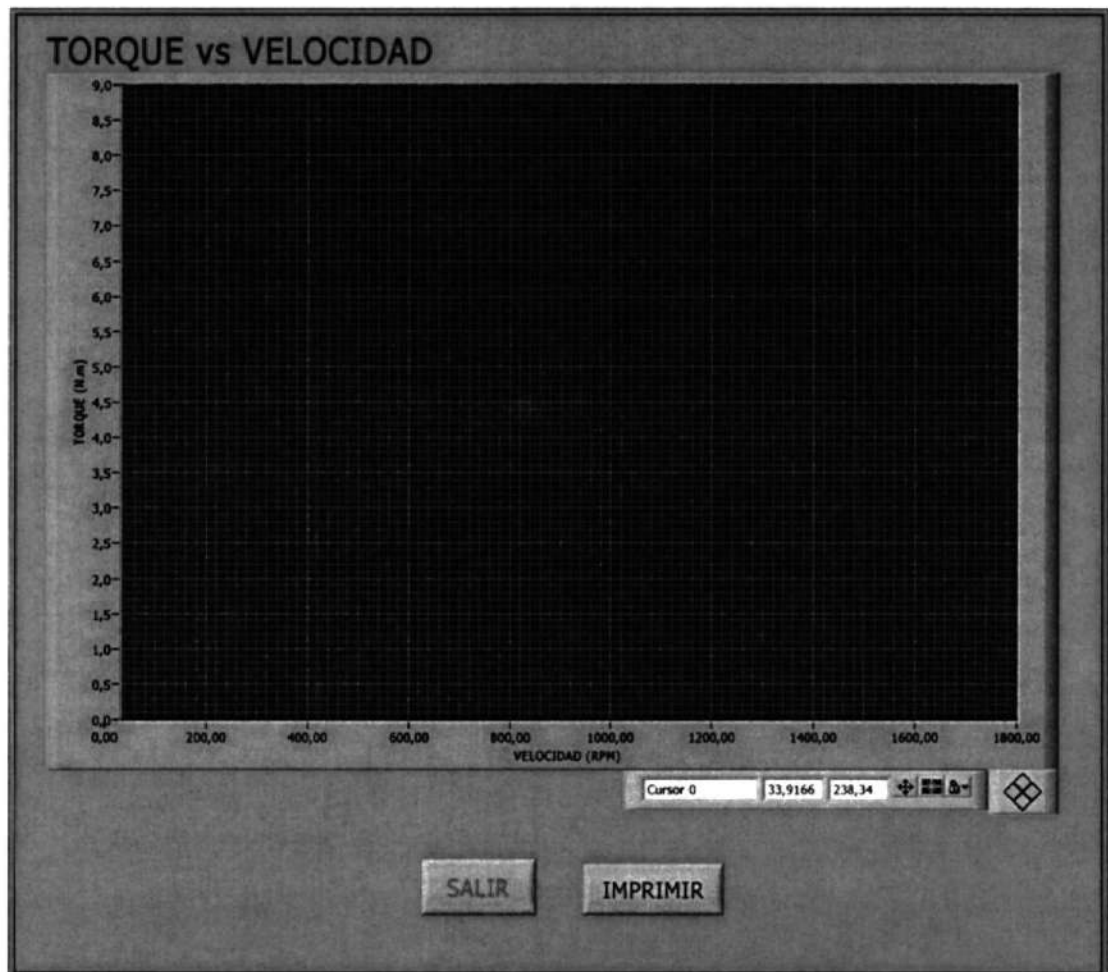
En las siguientes páginas se muestran todas las rutinas que conforman el programa en LabView las cuales hacen posible la adquisición de datos y el procesamiento de los mismos.

Connector Pane



TORQVEL.vi

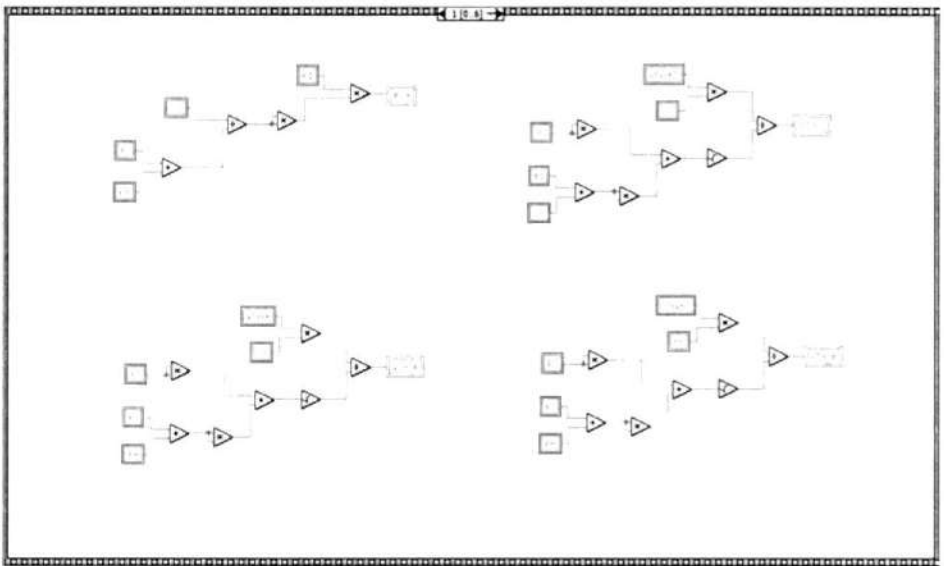
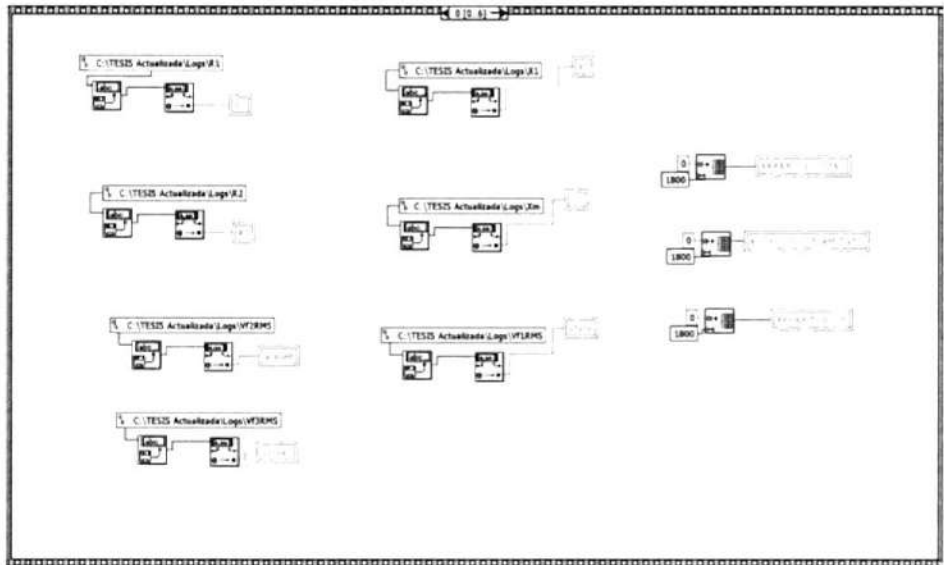
Front Panel

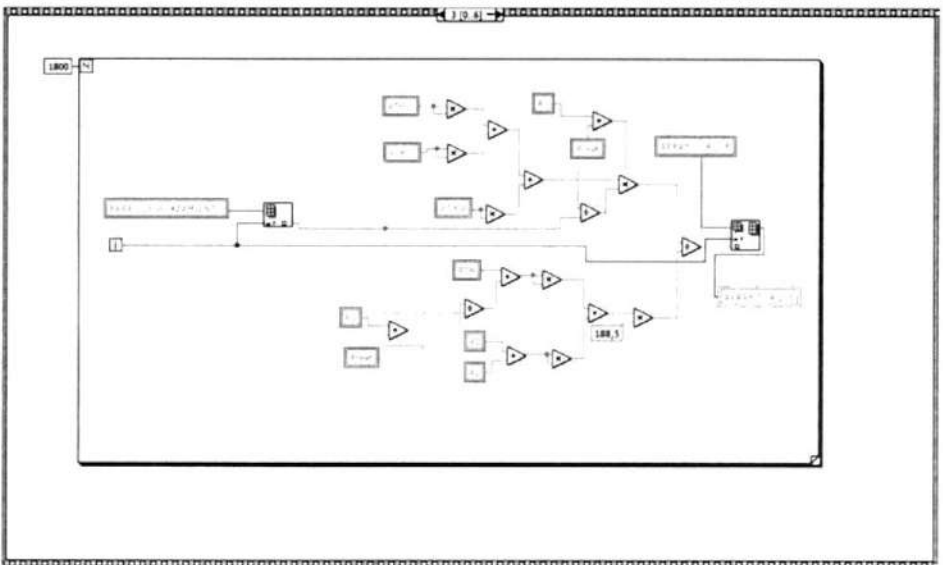
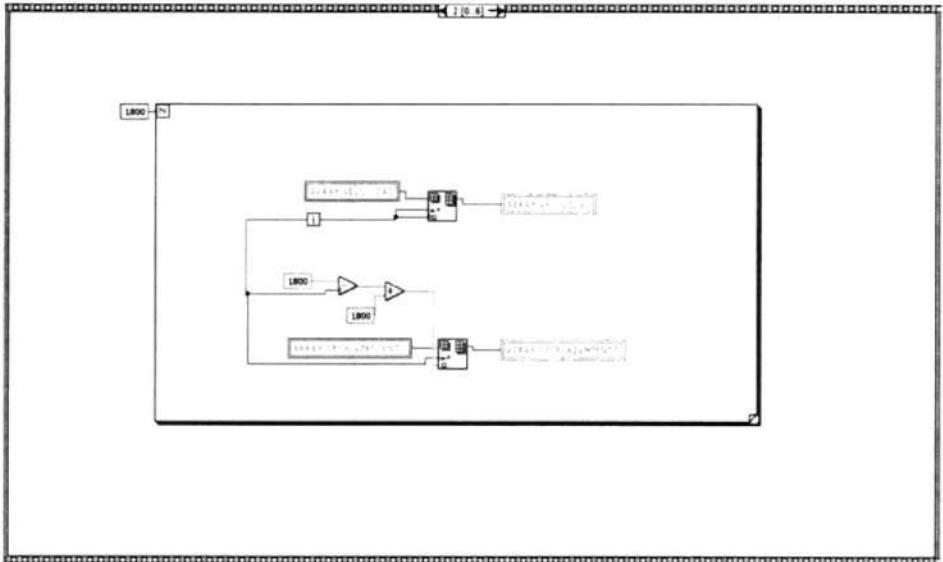


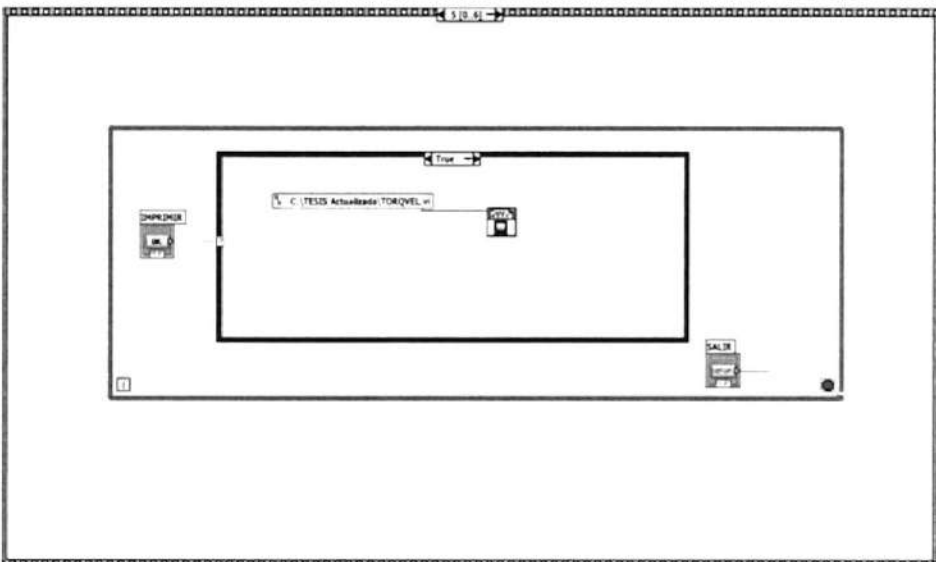
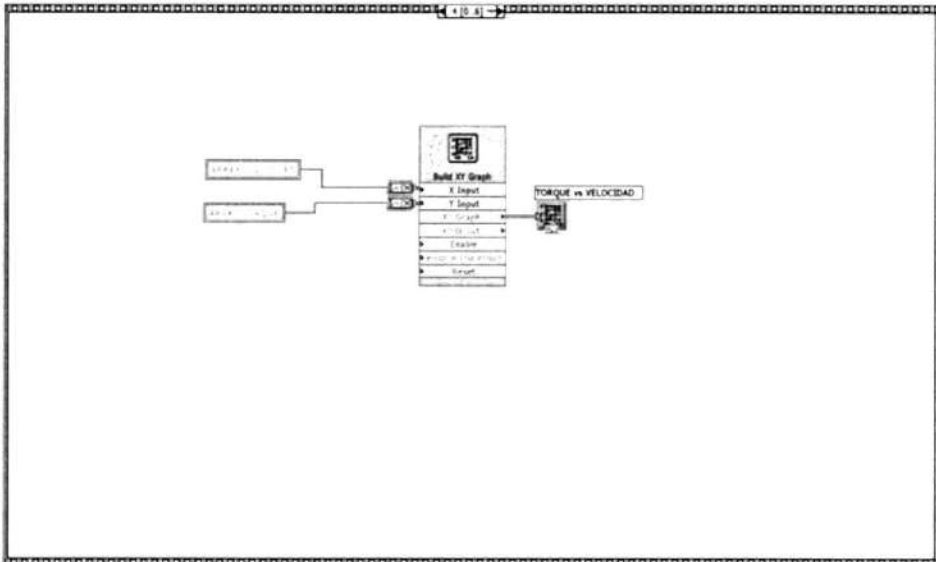
CIB -ESPOL

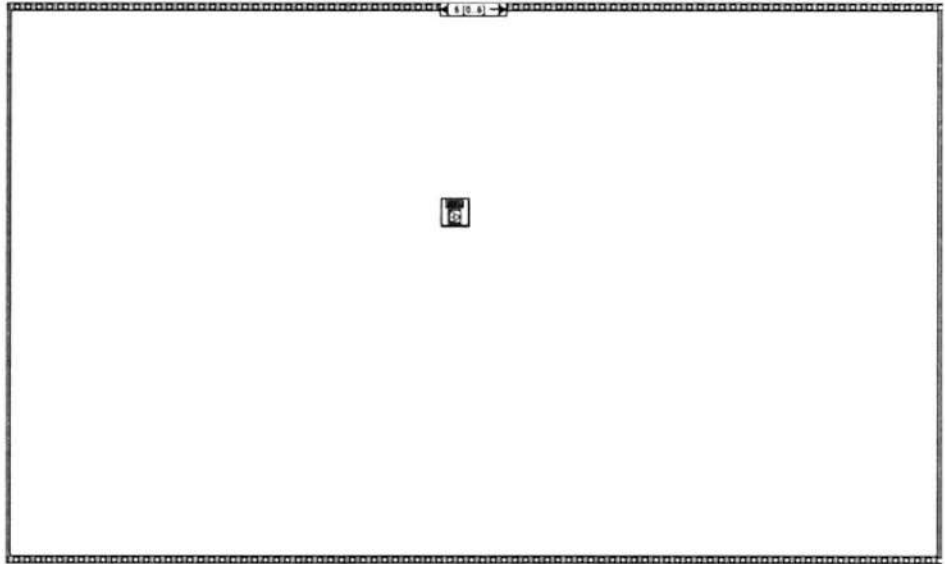
Block Diagram

- ARRAY DESPLAZAMIENTO
- ARRAY VELOCIDAD
- ARRAY TORQUE
- R1
- R2
- R3
- R4
- R5
- R6
- R7
- R8
- R9
- R10
- R11
- R12
- R13
- R14
- R15
- R16
- R17
- R18
- R19
- R20
- R21
- R22
- R23
- R24
- R25
- R26
- R27
- R28
- R29
- R30
- R31
- R32
- R33
- R34
- R35
- R36
- R37
- R38
- R39
- R40
- R41
- R42
- R43
- R44
- R45
- R46
- R47
- R48
- R49
- R50
- R51
- R52
- R53
- R54
- R55
- R56
- R57
- R58
- R59
- R60
- R61
- R62
- R63
- R64
- R65
- R66
- R67
- R68
- R69
- R70
- R71
- R72
- R73
- R74
- R75
- R76
- R77
- R78
- R79
- R80
- R81
- R82
- R83
- R84
- R85
- R86
- R87
- R88
- R89
- R90
- R91
- R92
- R93
- R94
- R95
- R96
- R97
- R98
- R99
- R100









Express VI Configuration Information



Build XY Graph

Build XY Graph
formats the data displayed on an X-Y Graph.

This Express VI is configured as follows:

Clear data on each call: On



Convert to Dynamic Data

Convert to Dynamic Data



Convert to Dynamic Data2

Convert to Dynamic Data



CIB -ESPOL

Connector Pane

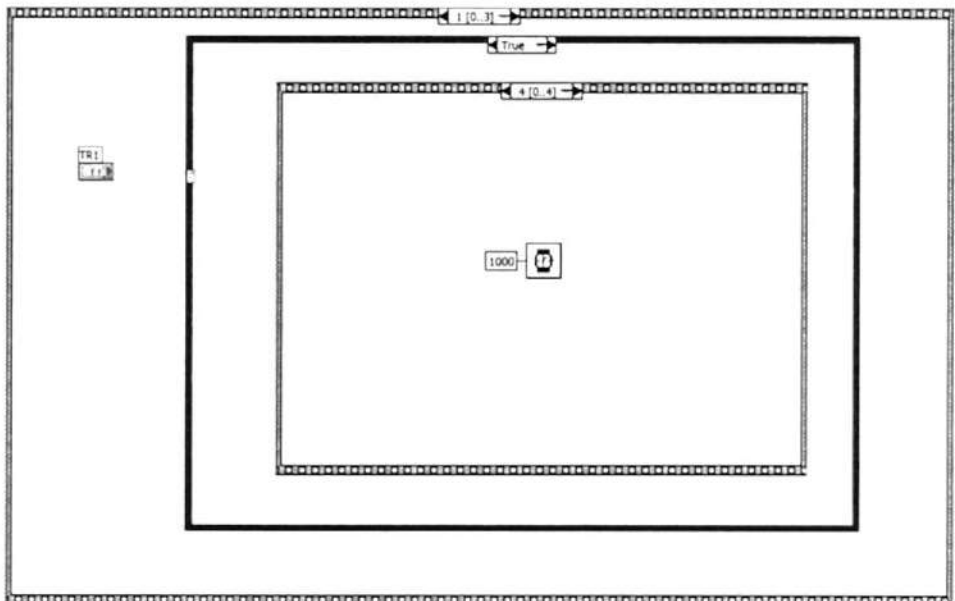
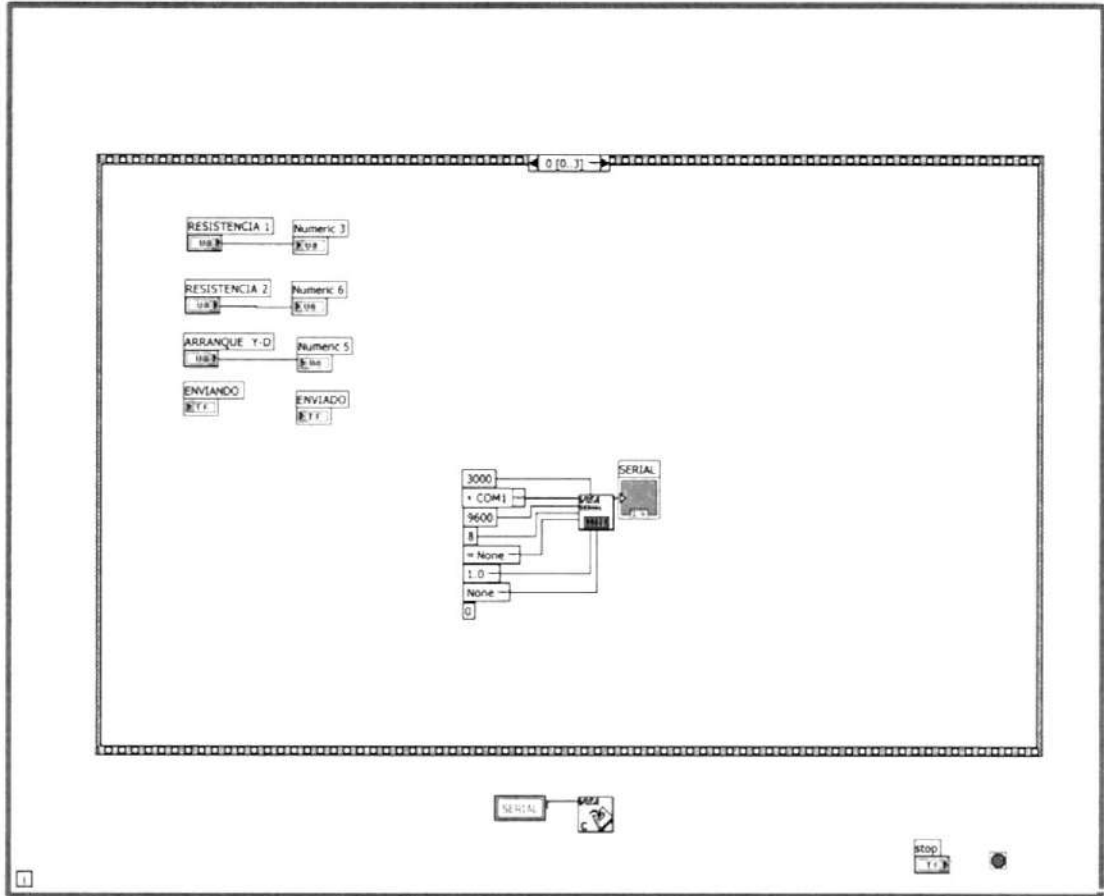


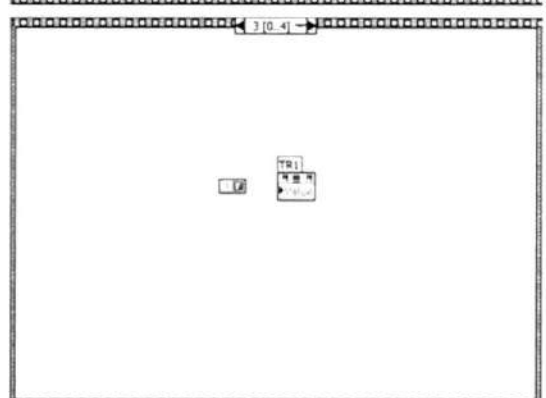
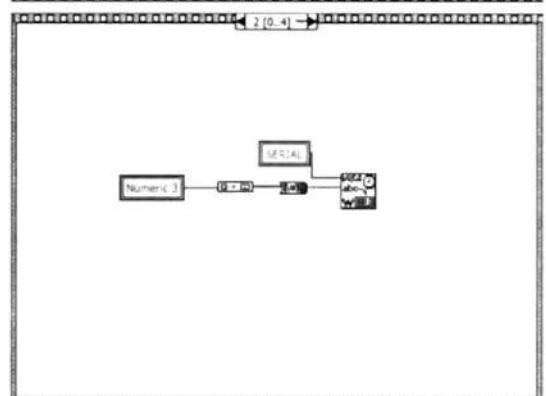
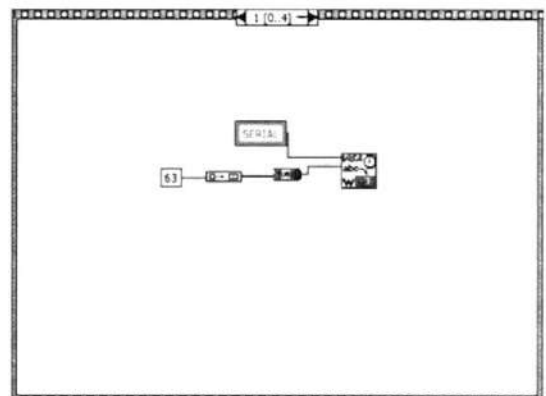
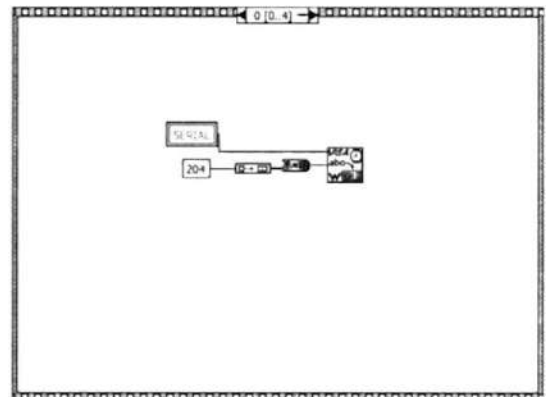
TIEMPOS.vi

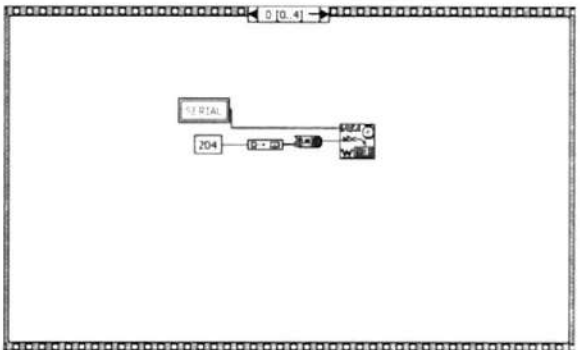
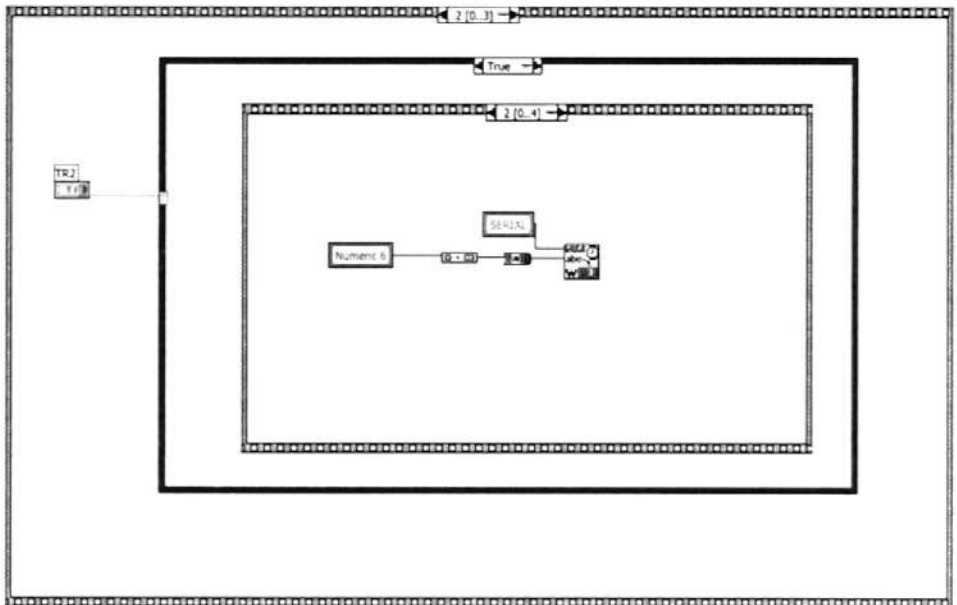
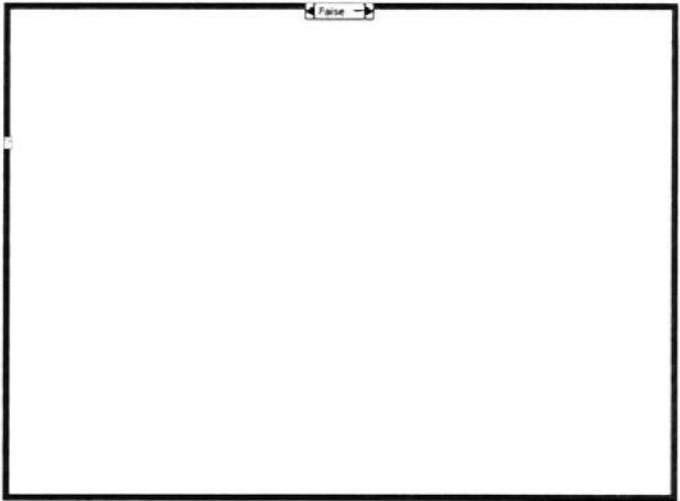
Front Panel

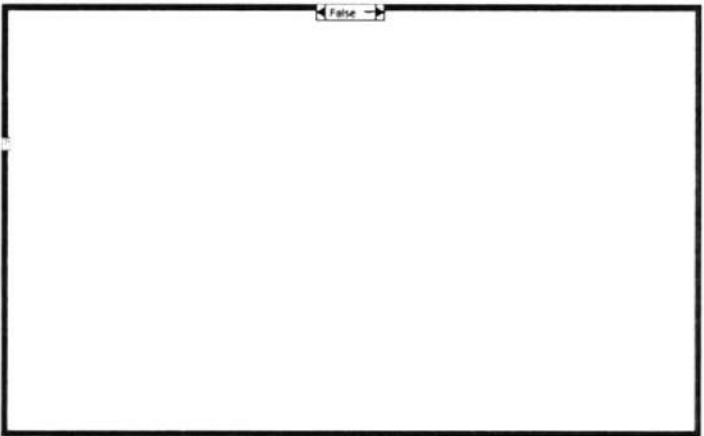
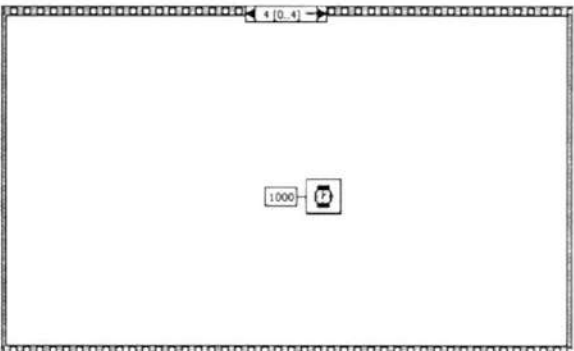
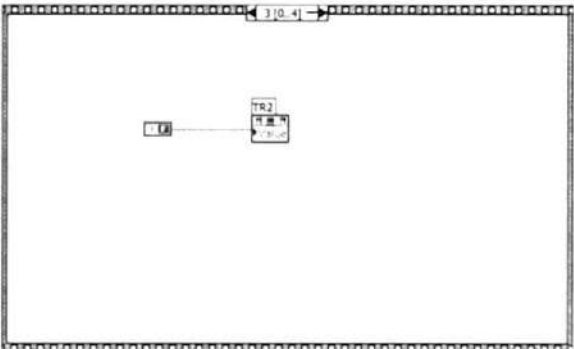
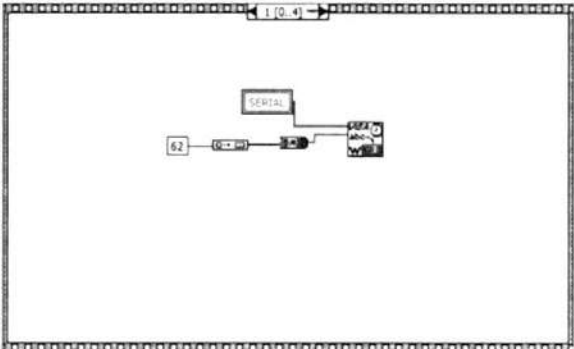


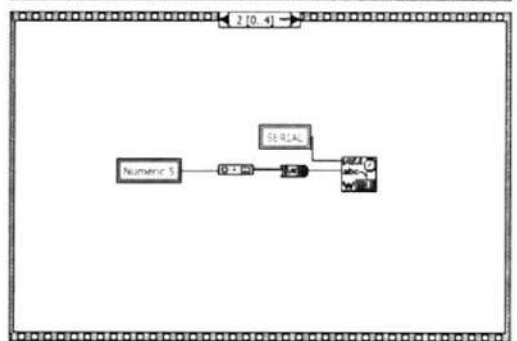
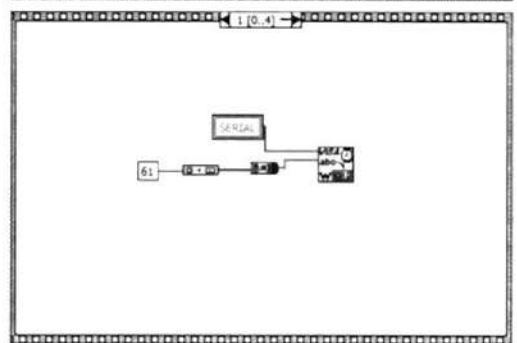
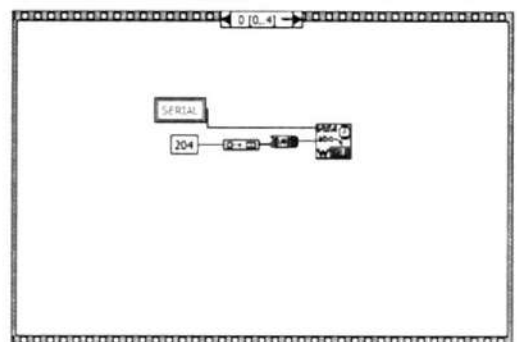
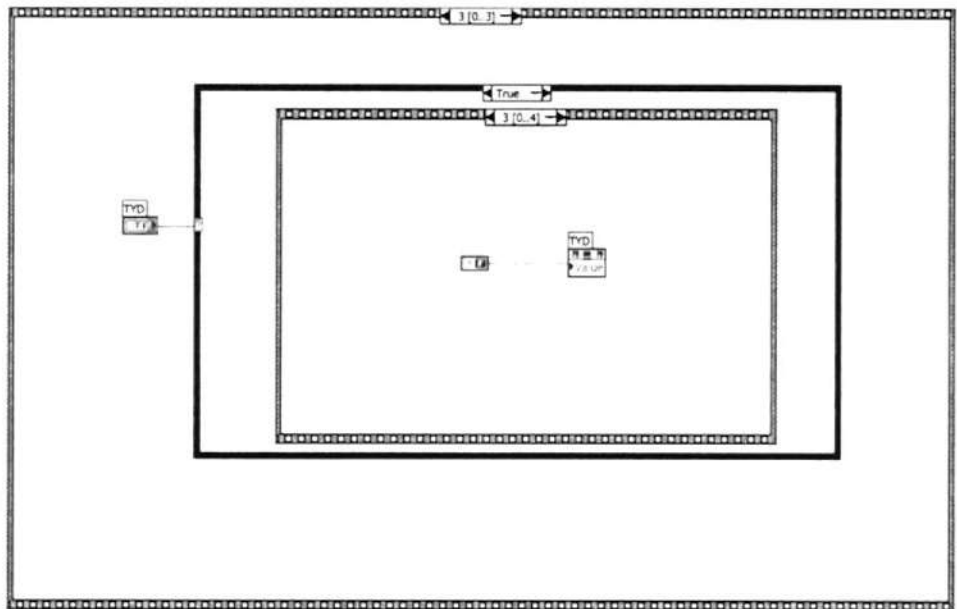
Block Diagram

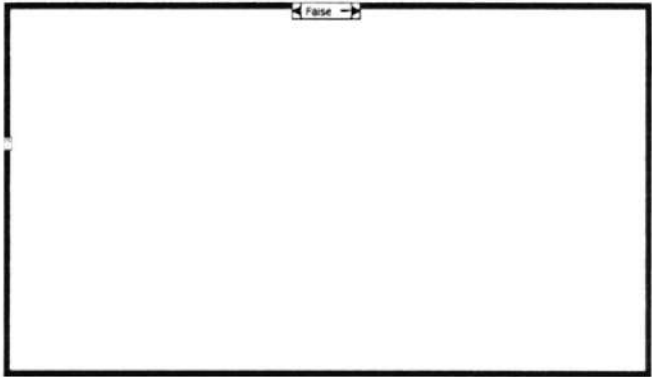
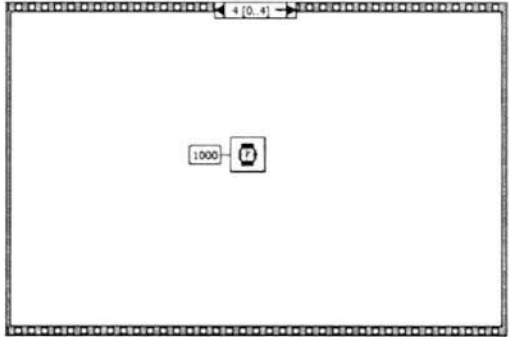










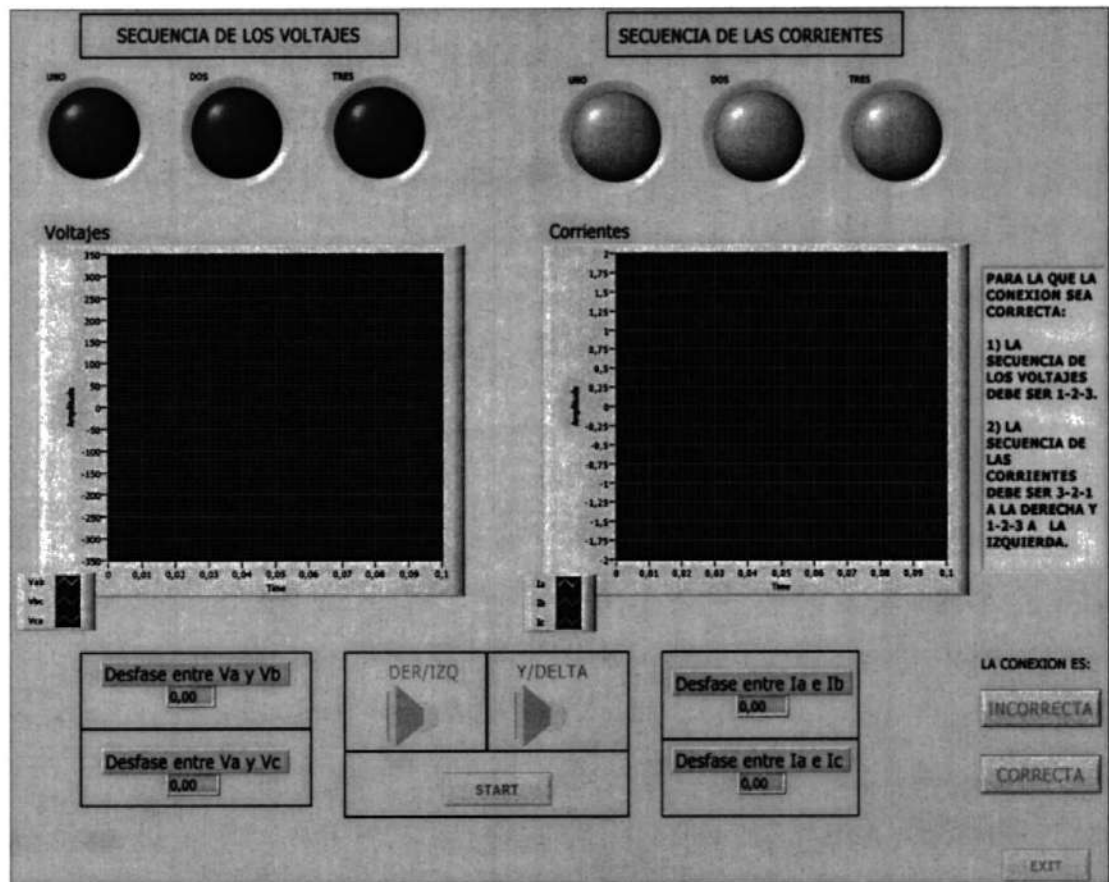


Connector Pane

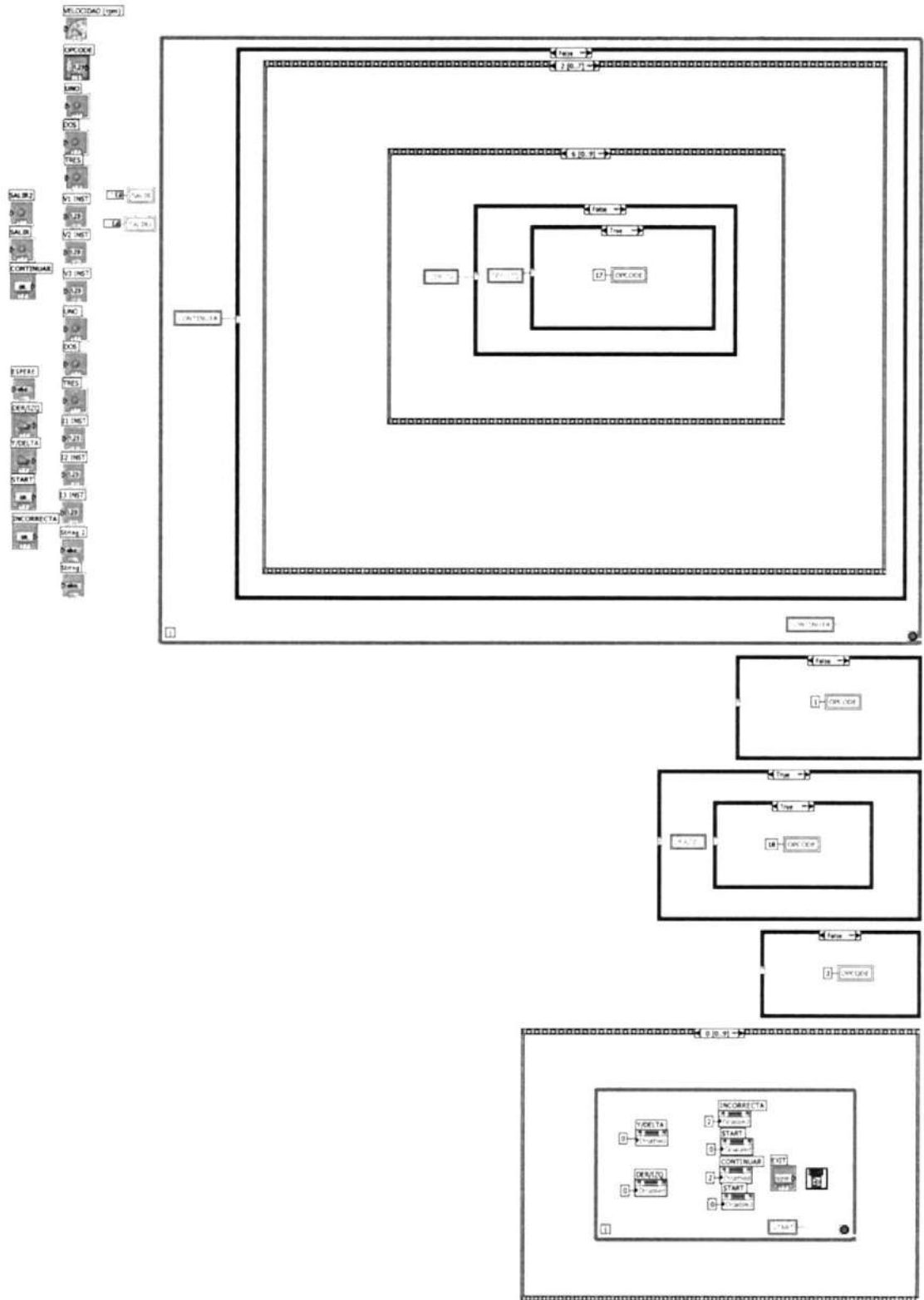


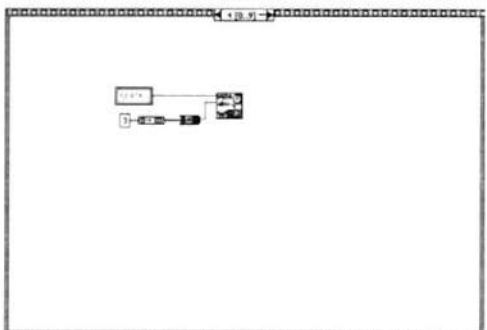
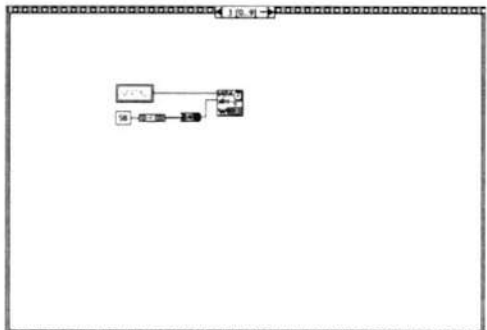
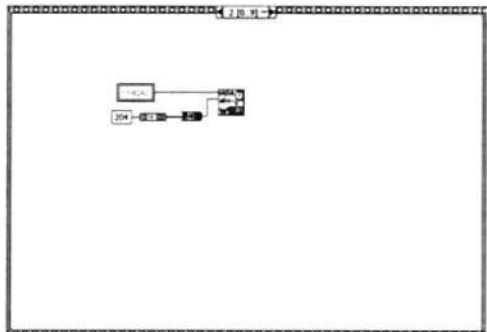
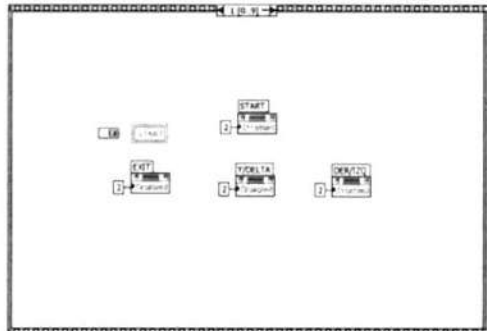
SINCRONIZACION.vi

Front Panel

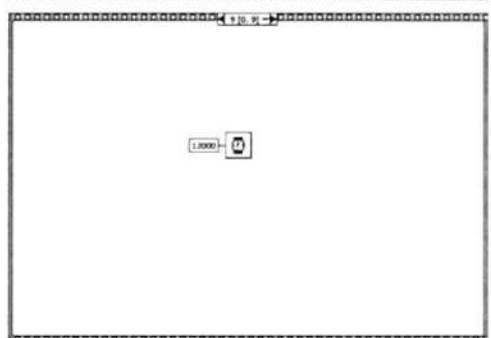
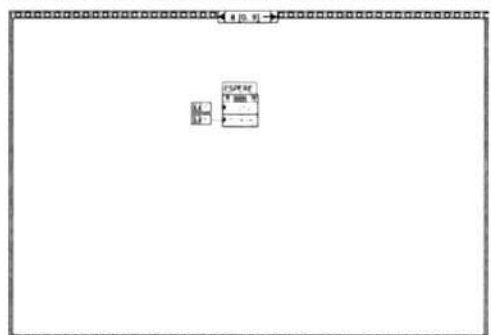
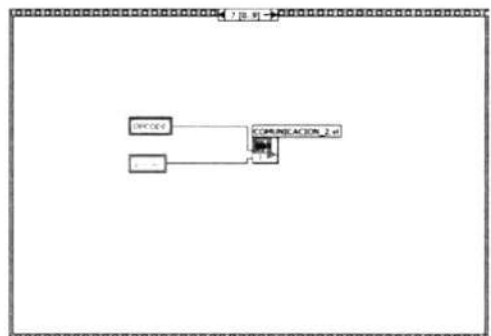
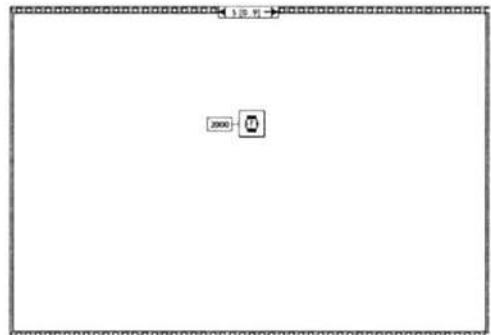


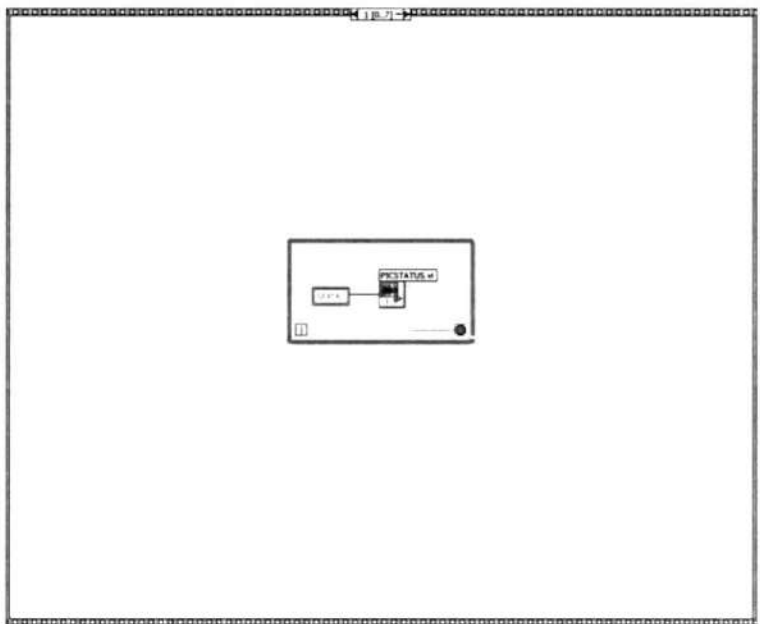
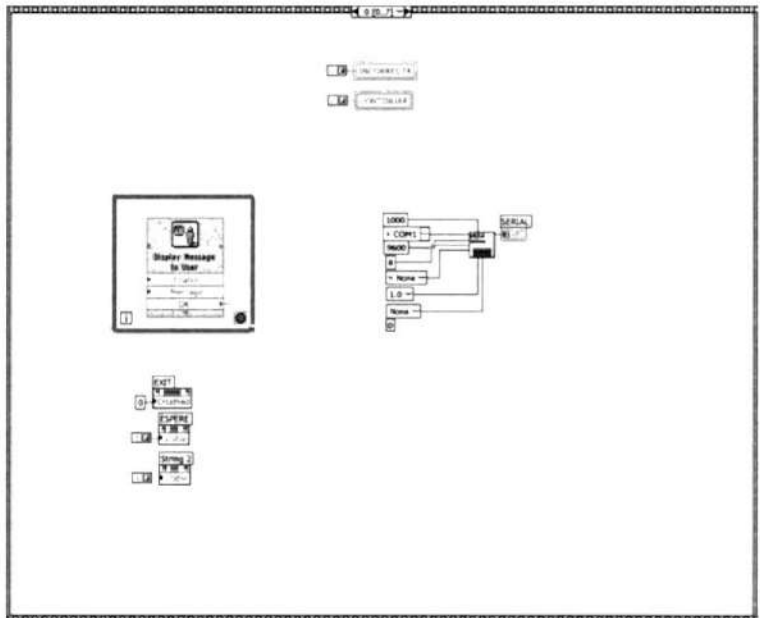
Block Diagram

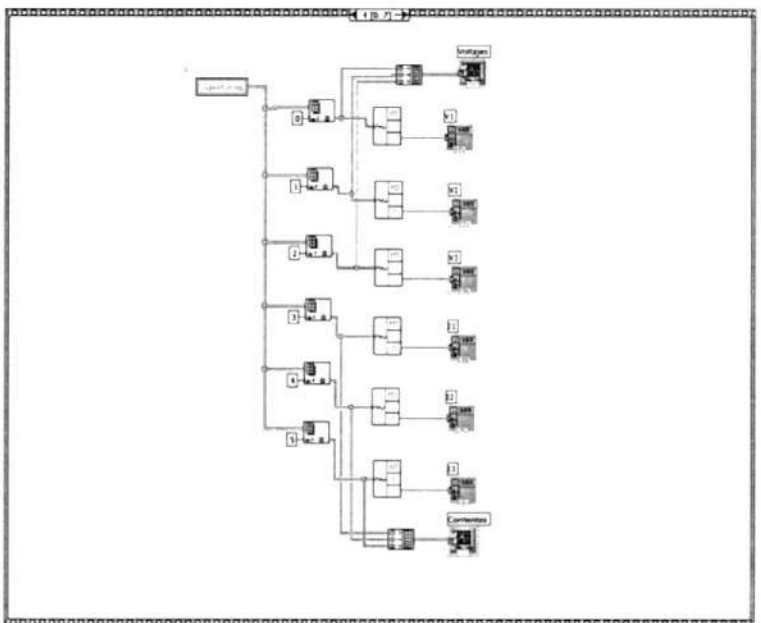
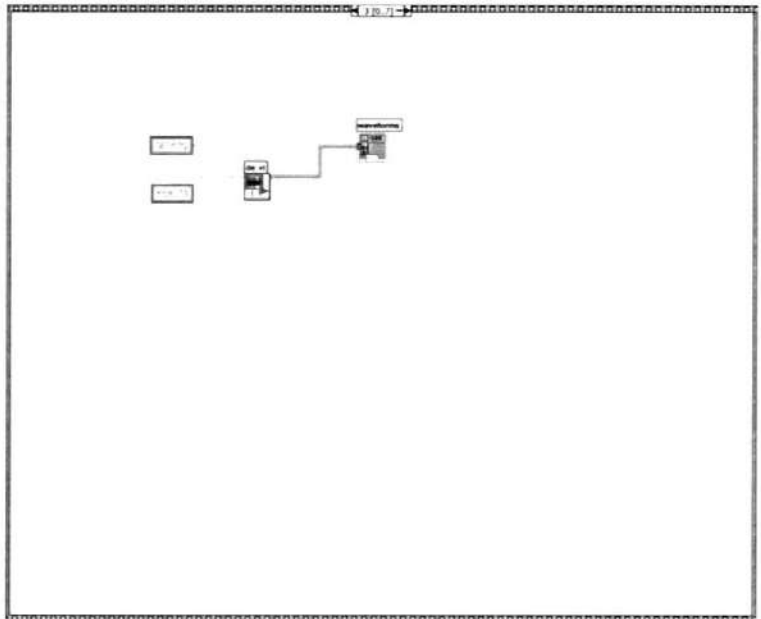


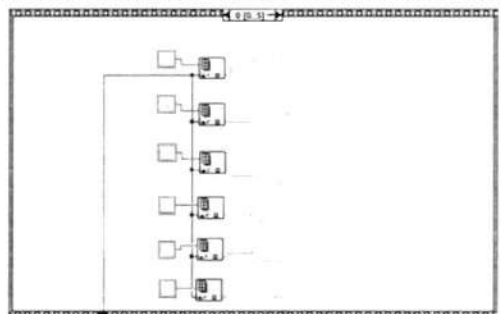
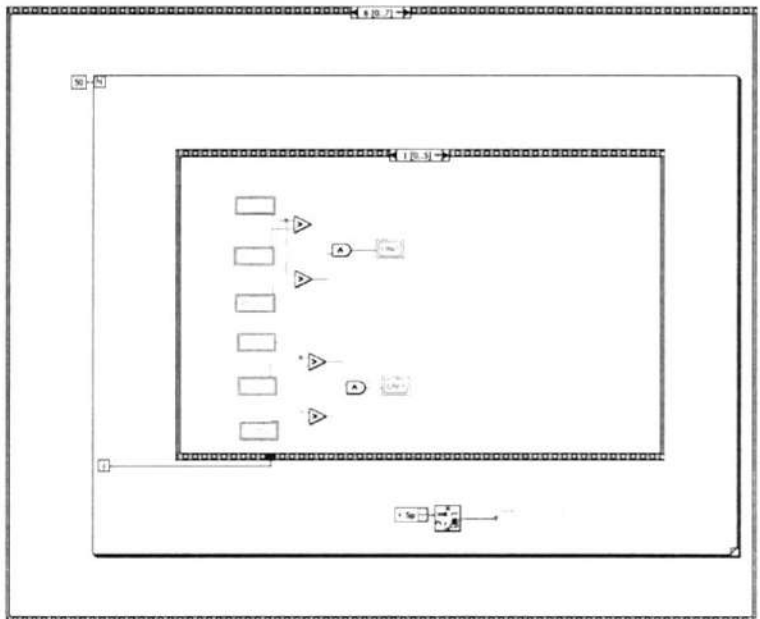
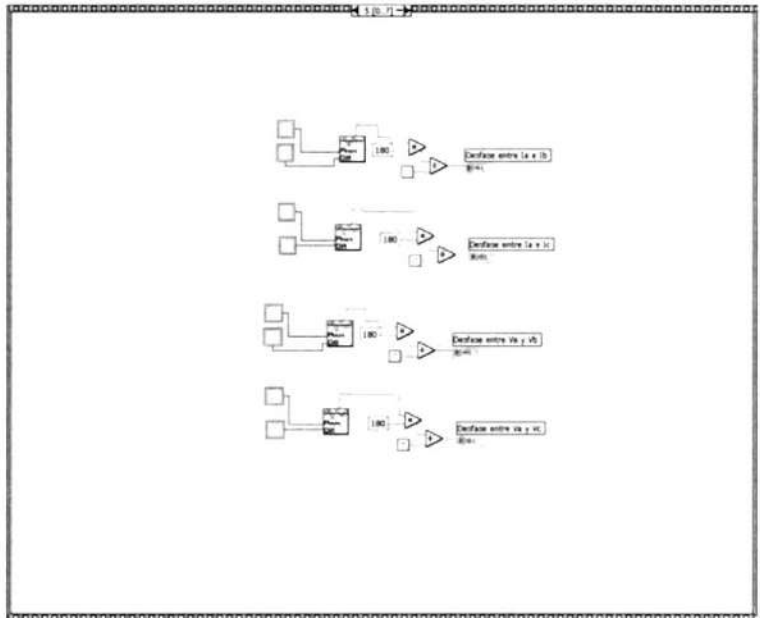


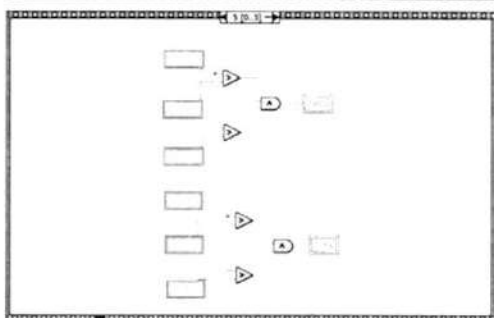
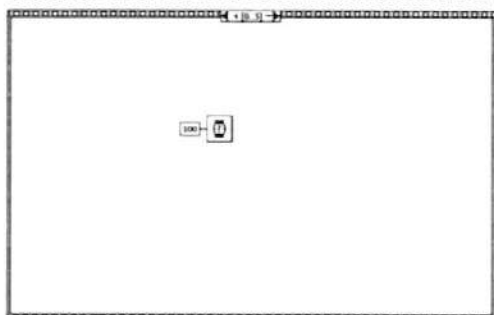
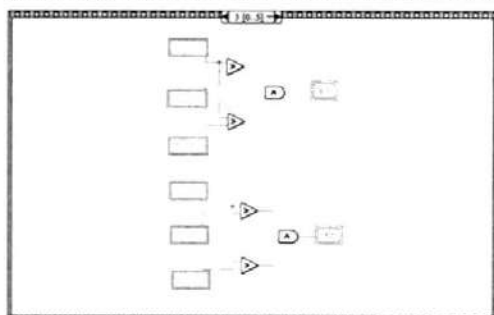
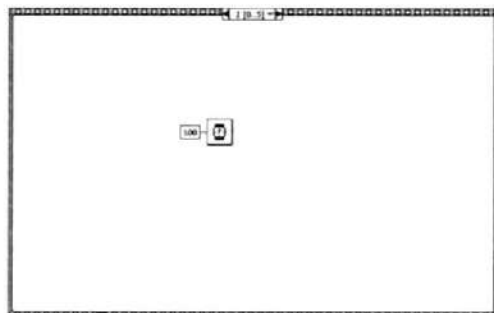
CIB -ESPOL

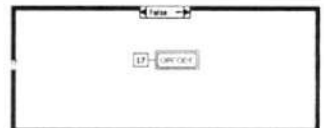
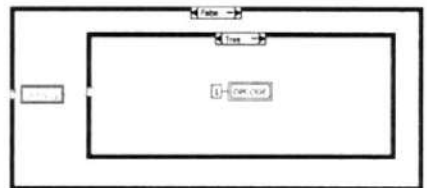
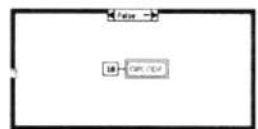
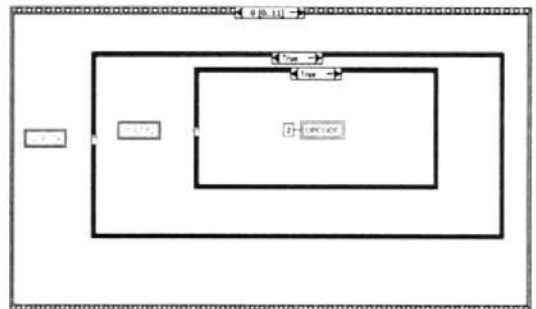
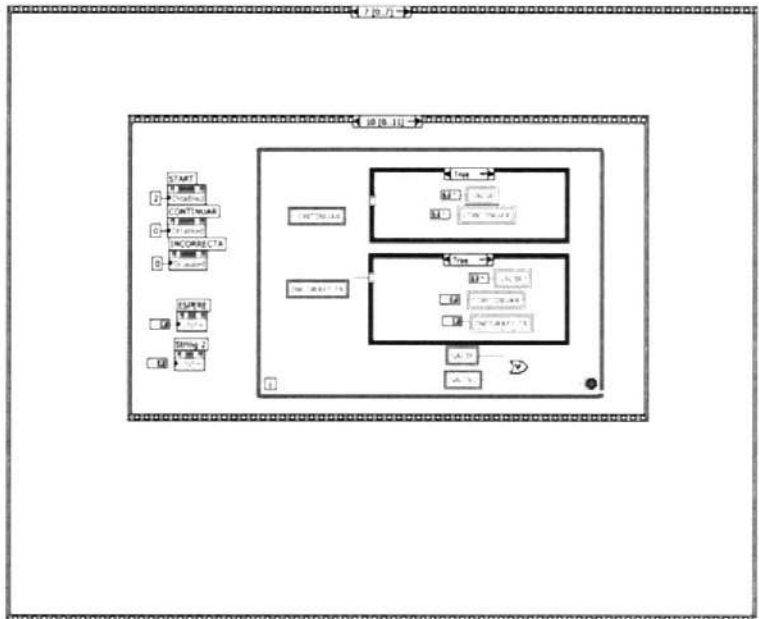


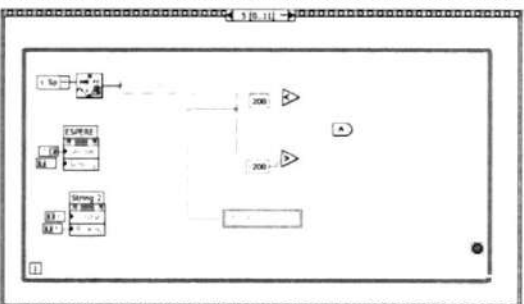
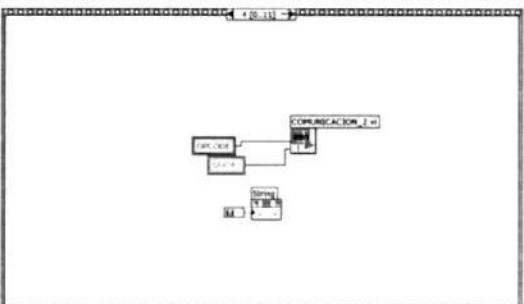
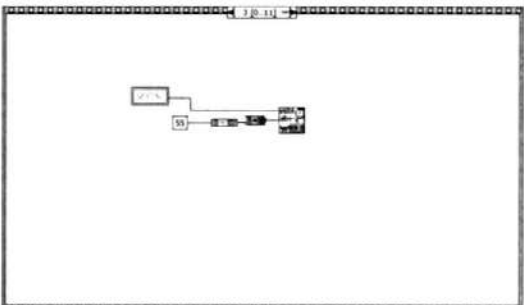
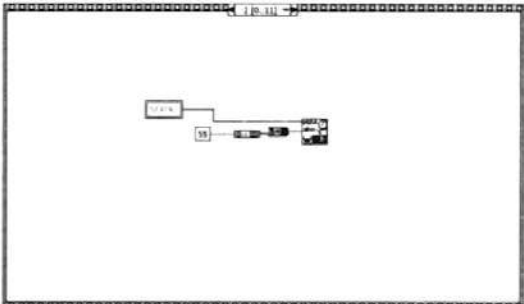
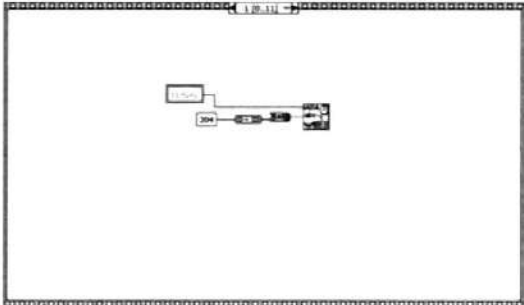


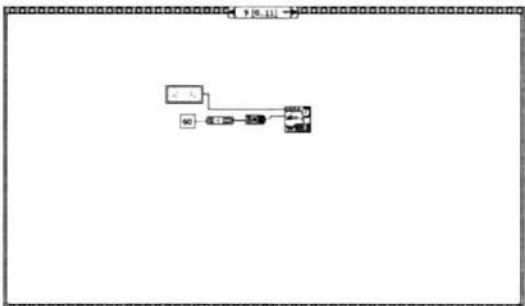
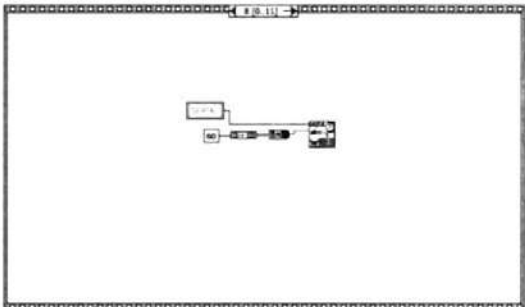
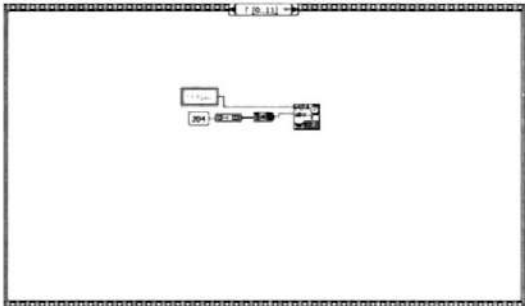


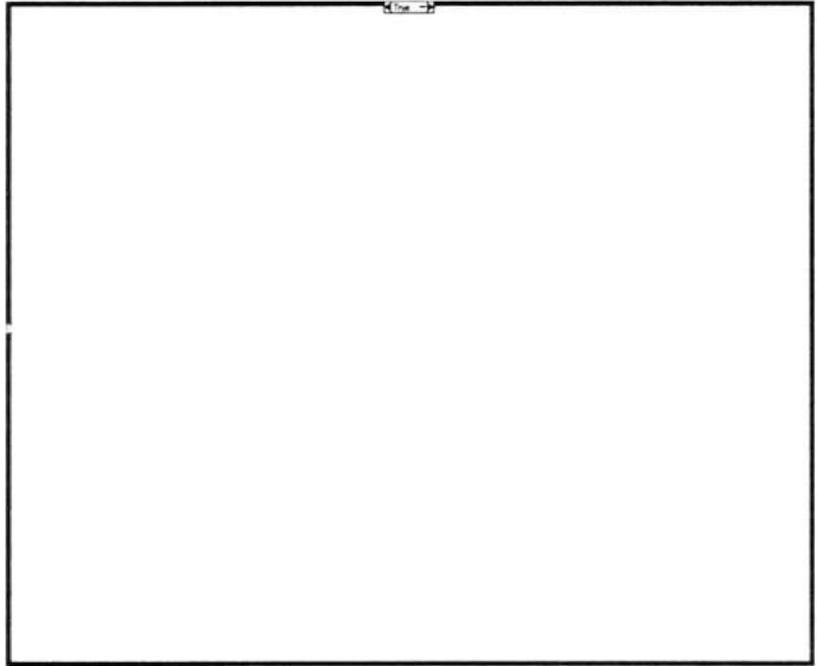












Express VI Configuration Information



Display Message to User

Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

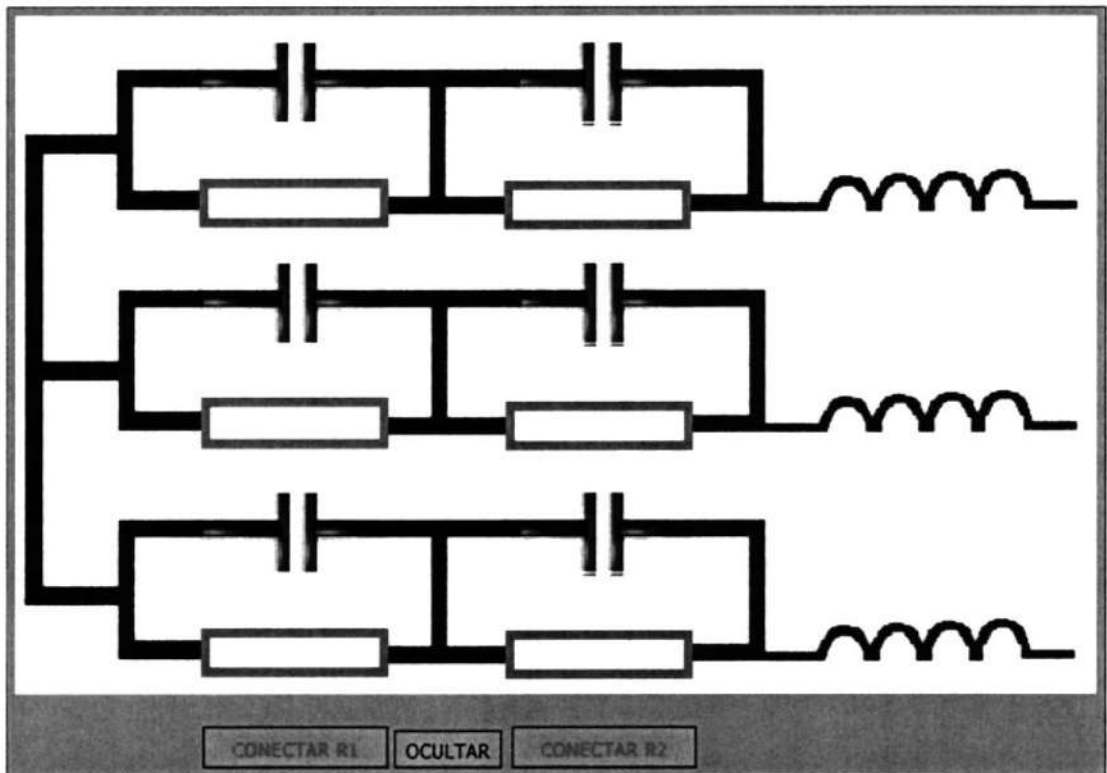
This Express VI is configured as follows:

Message:

Connector Pane

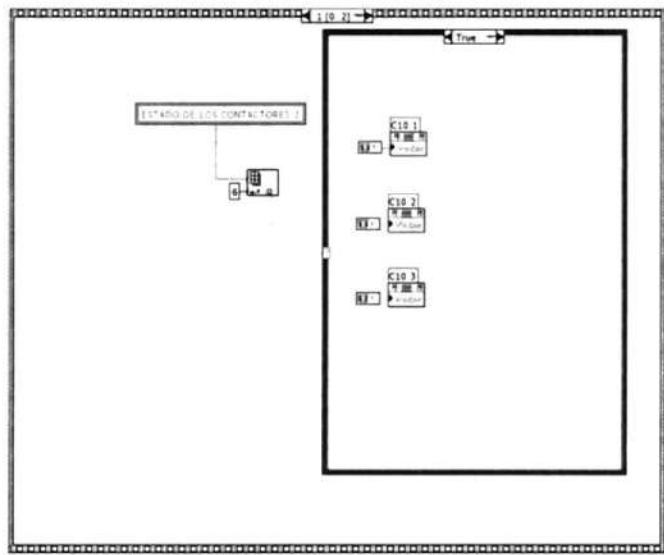
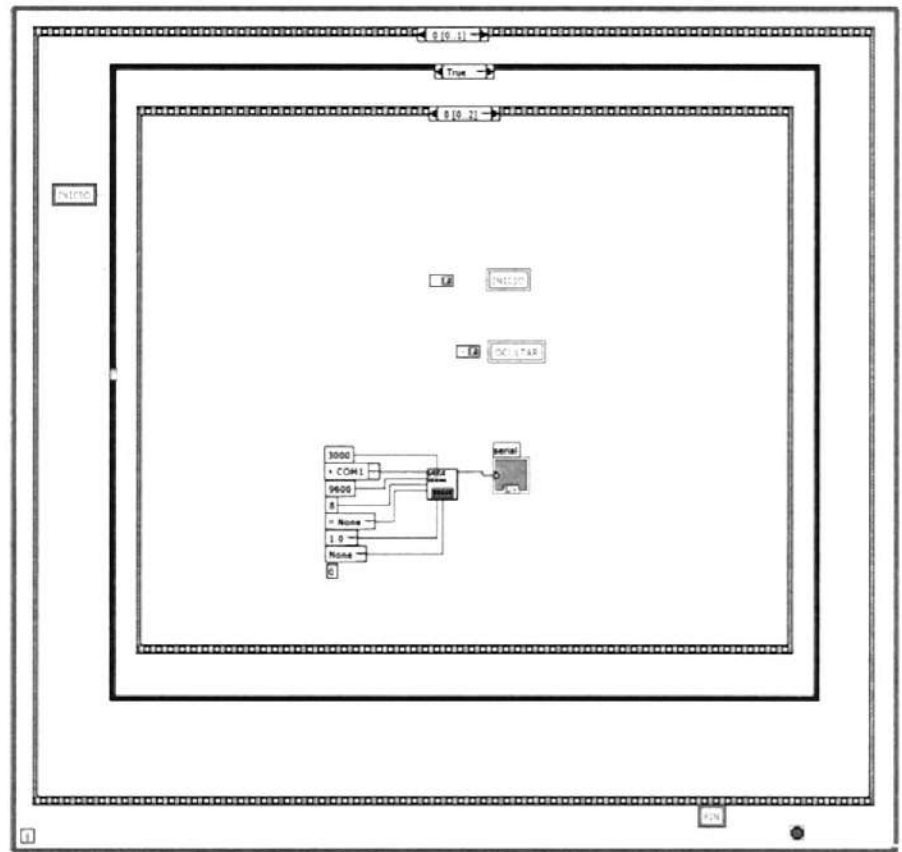


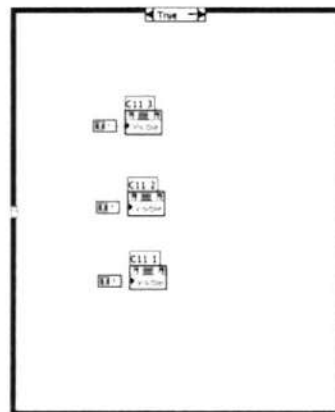
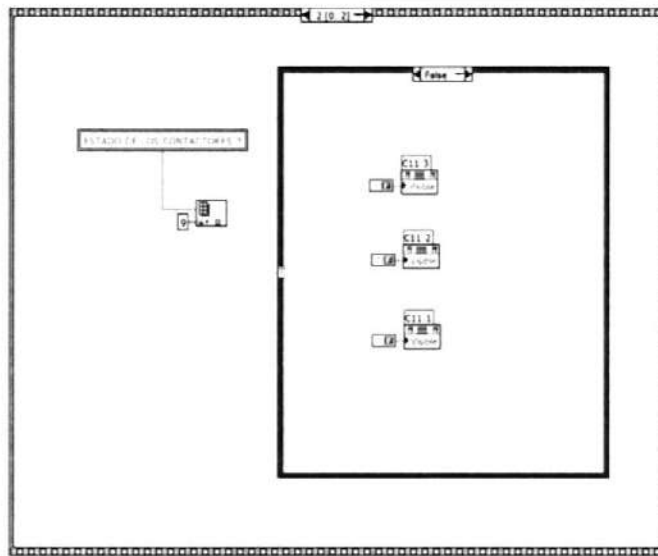
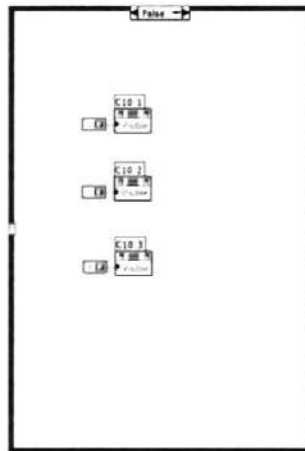
Front Panel

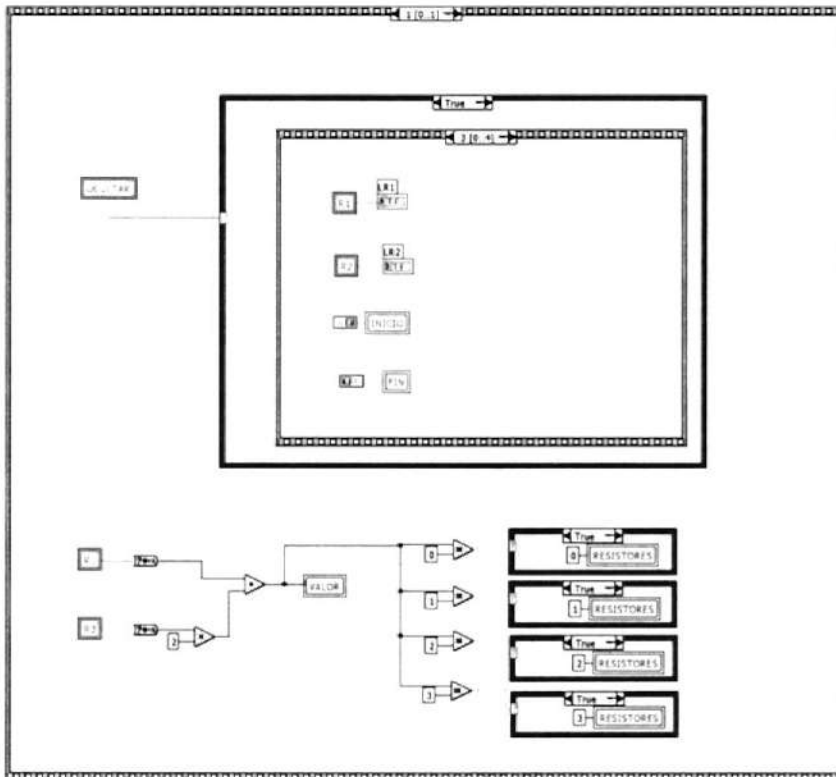
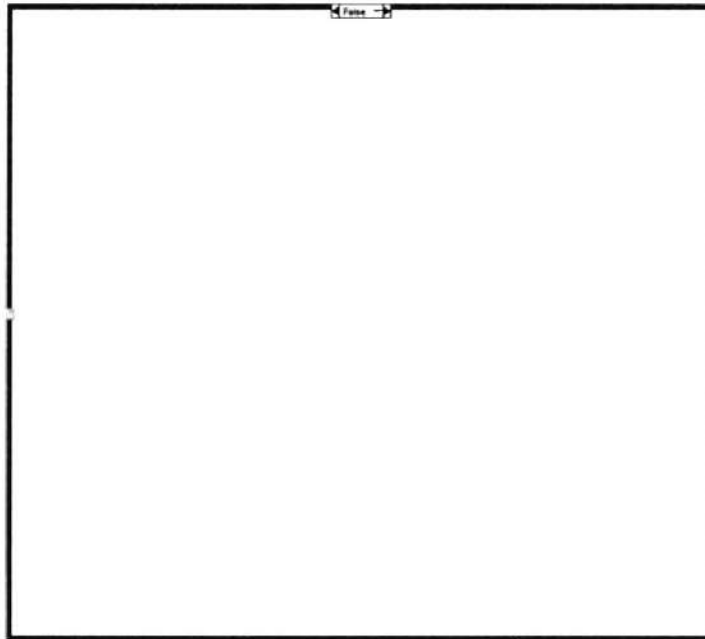


Block Diagram

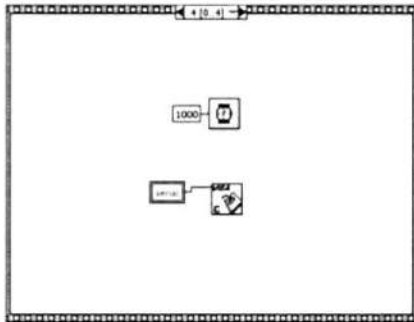
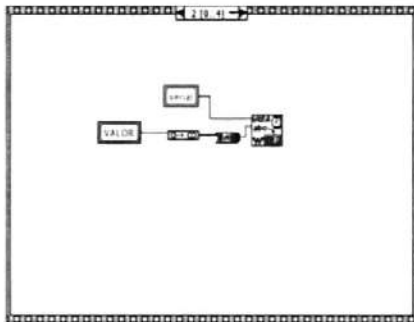
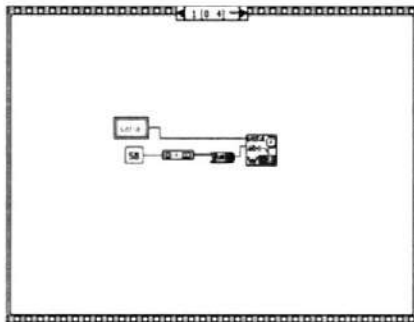
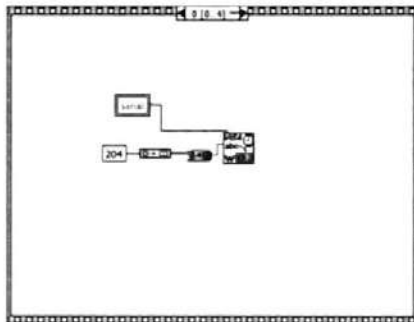
- VALOR
- ENVIANDO
- ESTADO DE LOS CONTACTORES 2
- C10 1
- C10 2
- C10 3
- INICIO
- R1
- R2
- OCULTAR
- RESISTORES
- R1N
- R2N

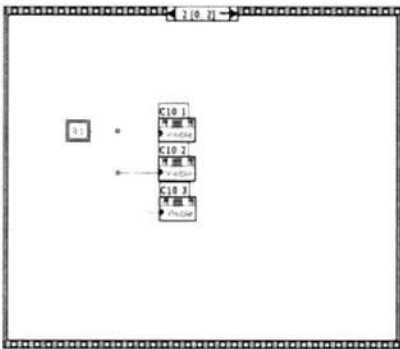
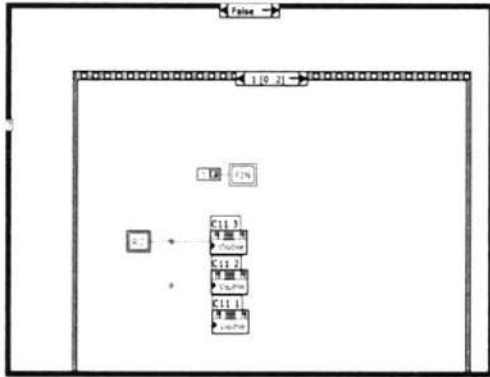




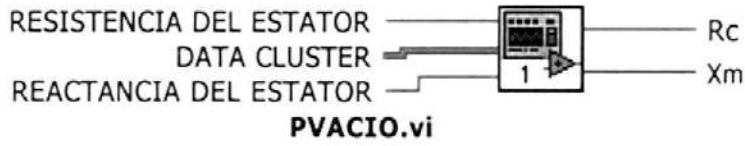








Connector Pane



Front Panel

PRUEBA DE VACIO

CIRCUITO EQUIVALENTE:

POTENCIAS TRIFASICAS DE ENTRADA:

P. APARENTE (VA)

P. REACTIVA (VAr)

P. ACTIVA (W)

CALCULOS

VOLTAJE DE FASE	CORRIENTE DE FASE	FACTOR DE POTENCIA EN VACIO
0,00 VOLTIOS	0,00 AMPERES	0
Z1 <input type="text"/>	E1 <input type="text"/>	Pc <input type="text"/>
Im <input type="text"/>	Rc <input type="text"/>	Xm <input type="text"/>
Ic <input type="text"/>	PERDIDAS ROTACIONALES <input type="text"/>	

NL=NO LOAD=EN CARGA (EN VACIO)

$$E_1 = V_{NL} - I_{NL} \cdot Z_1$$

$$Z_1 = \sqrt{R_1^2 + X_1^2}$$

$$P_C = P_{NL} - I_{NL}^2 \cdot R_1$$

$$I_C = \frac{P_C}{E_1}$$

$$I_m = \sqrt{I_{NL}^2 - I_C^2}$$

$$R_C = \frac{E_1^2}{P_C}$$

$$X_m = \frac{E_1}{I_m}$$

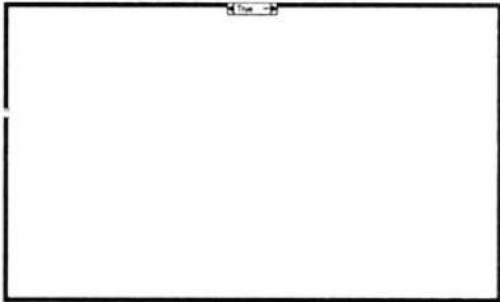
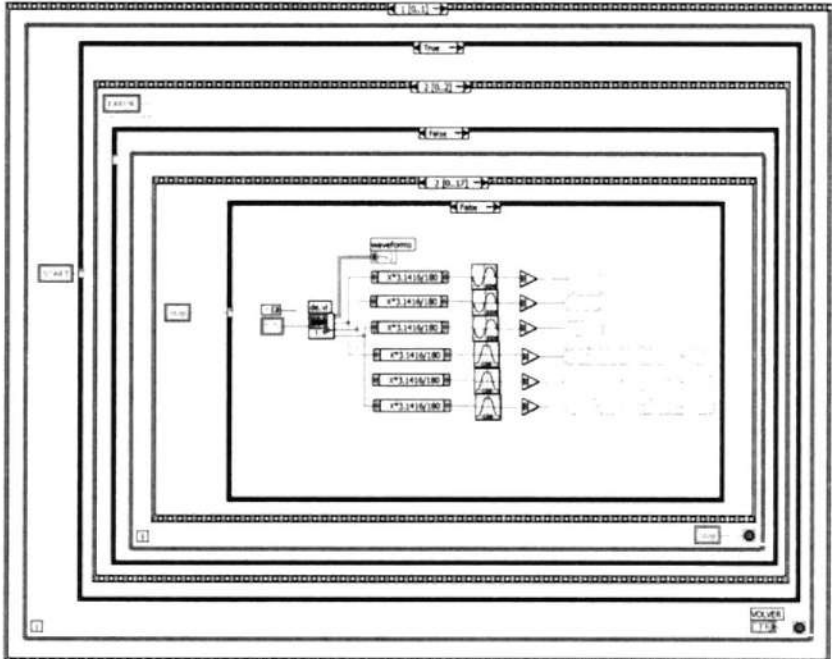
START

STOP

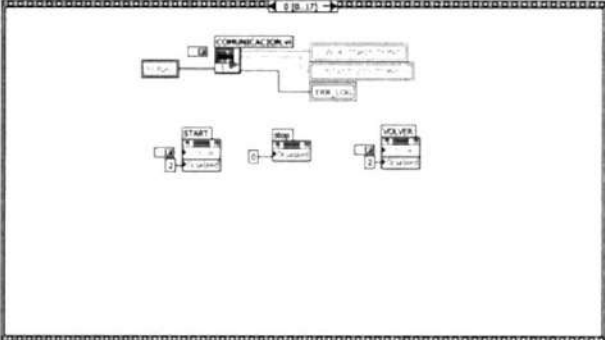
REGRESAR

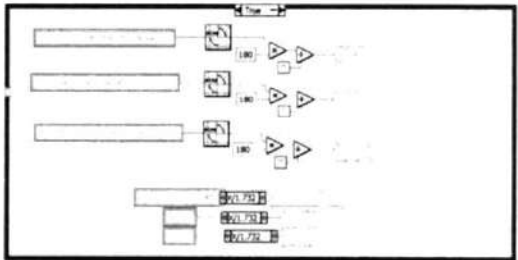
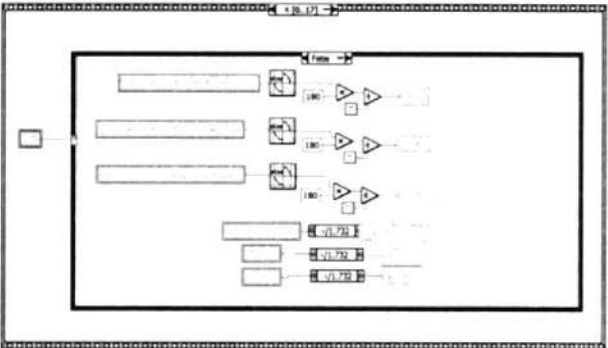
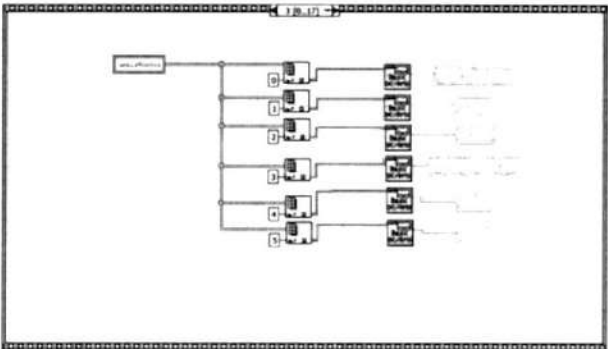
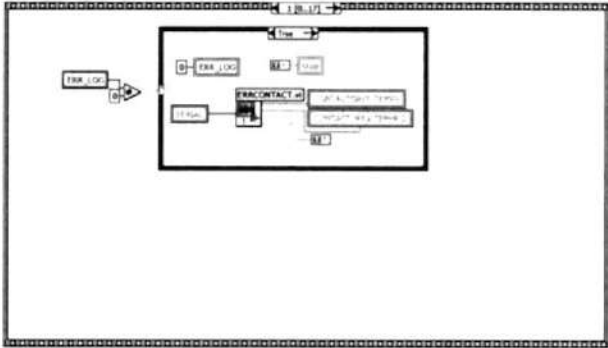
Block Diagram

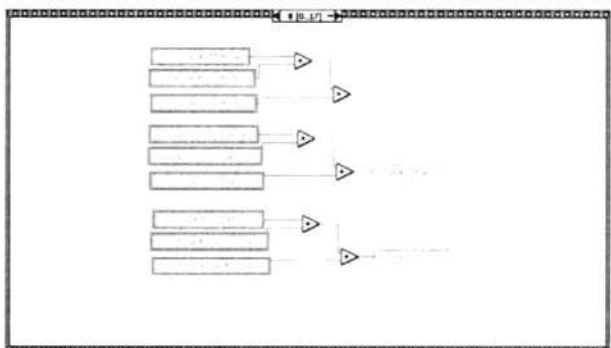
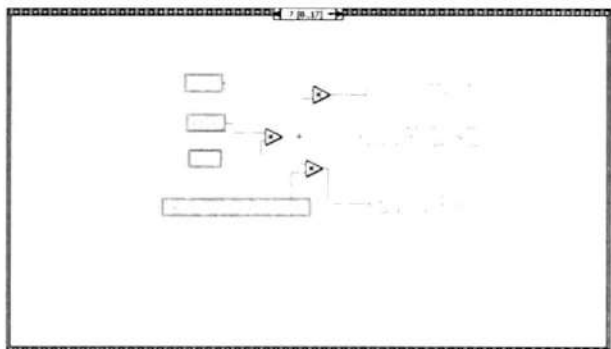
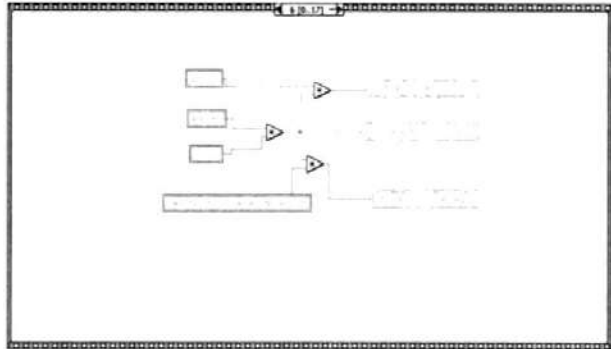
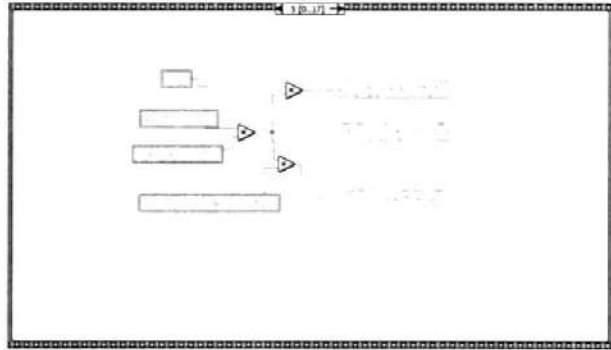
POTENCIA APARENTE (W)	POTENCIA ALTYVA (W)
POTENCIA REACTIVA (W)	CONTACTORES TEMP. 2
AMPLITUDE	BIT. 1
DATA CLUSTER	CONTACTORES TEMP. 1
SERIAL	BIT. 2
VOLT_CHECK	RESISTENCIA DEL ESTATOR
MAV.1.2	ERR. LOG
MAV.1.1	ERR. LOG
MAV.1	DATA CLUSTER
MAV.2	RESISTENCIA DEL ESTATOR
MAV.3	PERDIDAS ROTACIONALES
FACTO DE POTENCIA EN VACIO 2	START
FACTO DE POTENCIA EN VACIO 1	P. APARENTE (W)
VOLTAGE DE FASE	BITA
VAV.1MS	P. ACTIVA (W)
VAV.2MS	P. REACTIVA(W)
CORRIENTE DE FASE	BIT. 1
BIT.1MS	BIT. 2
BIT.2MS	BIT. 3
BIT.3MS	BIT. 4
BITA.1	POTENCIA REACTIVA(W) 2
BITA.2	POTENCIA REACTIVA(W) 1
BITA.3	POTENCIA ACTIVA (W) 2
BITA.4	POTENCIA ACTIVA (W) 1
POTENCIA APARENTE (W) 2	ERROR
POTENCIA APARENTE (W) 1	BIT. 1
ERROR	ESTATE DE LINEA PROMEDIO RMS
BIT. 1	POTENCIA APARENTE (W) 3
BIT. 2	CORRIENTE DE LINEA
BIT. 3	PROMEDIO RMS
BIT. 4	

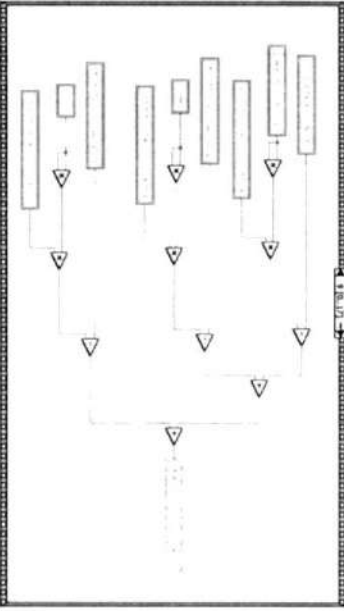
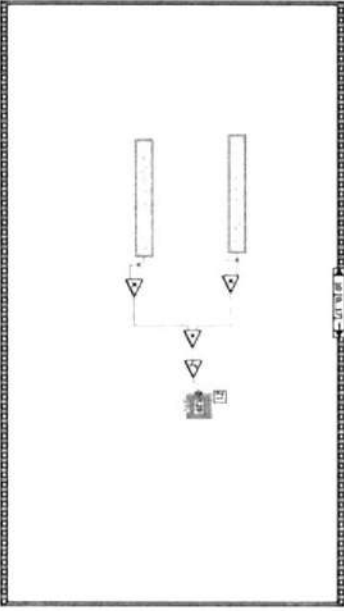
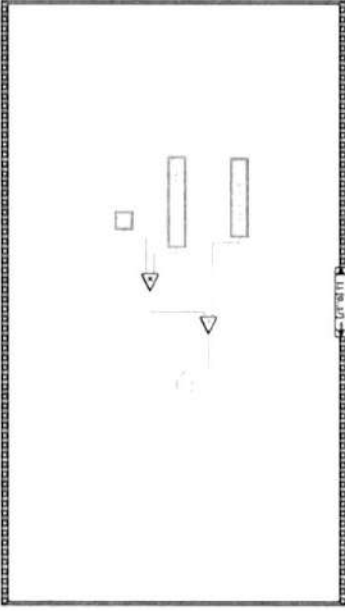
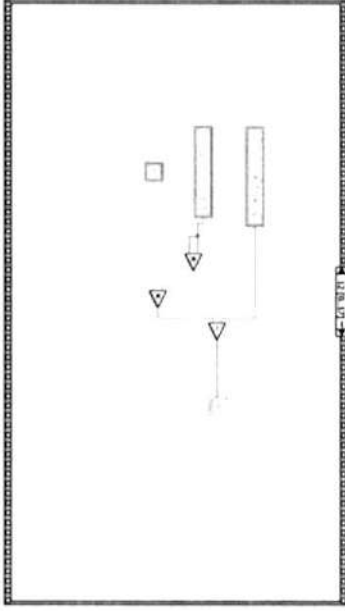


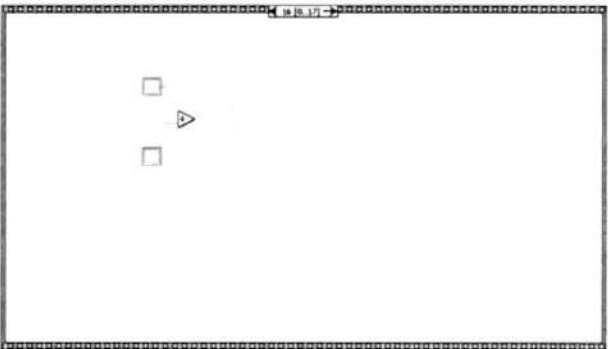
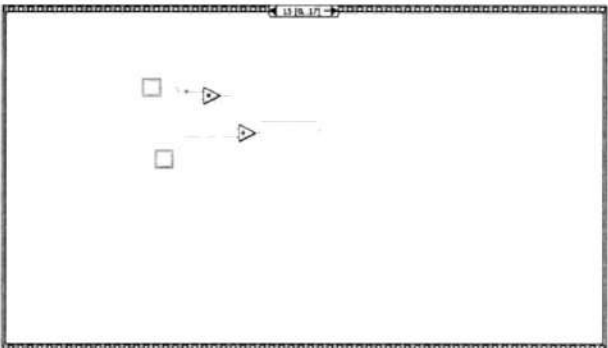
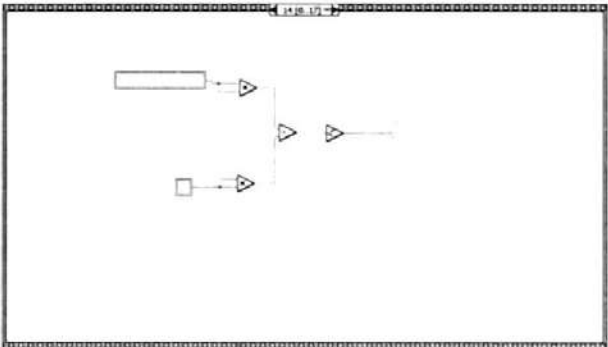
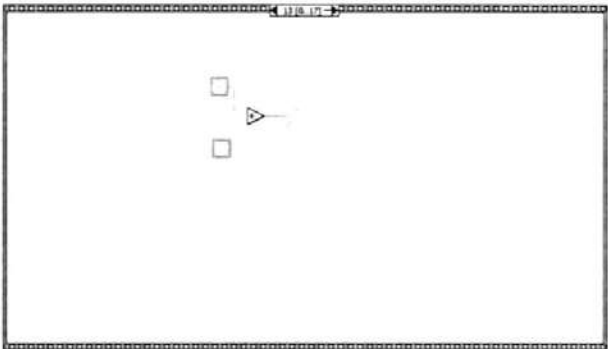
CIE - ESPOL

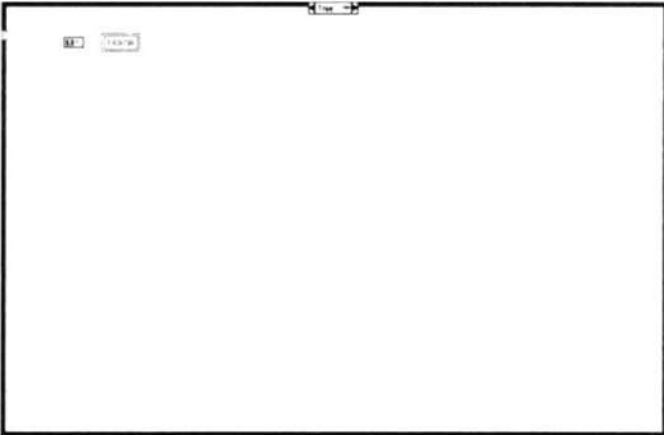
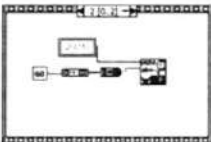
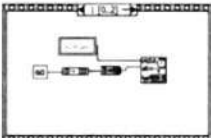
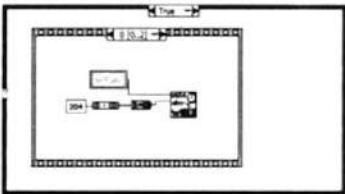
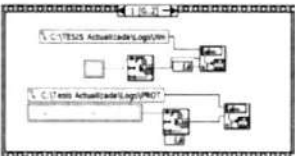
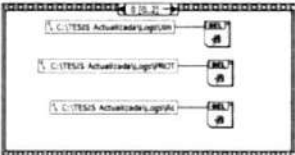
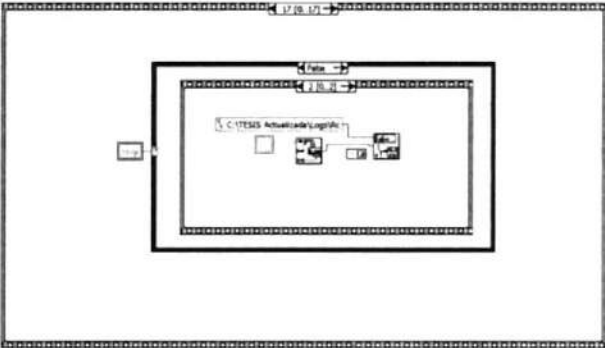


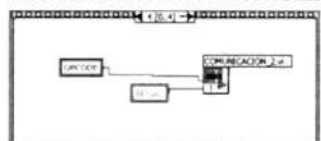
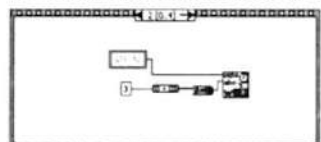
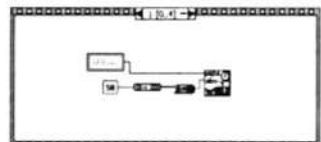
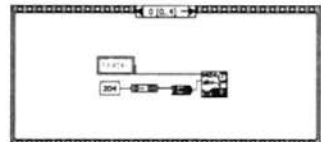
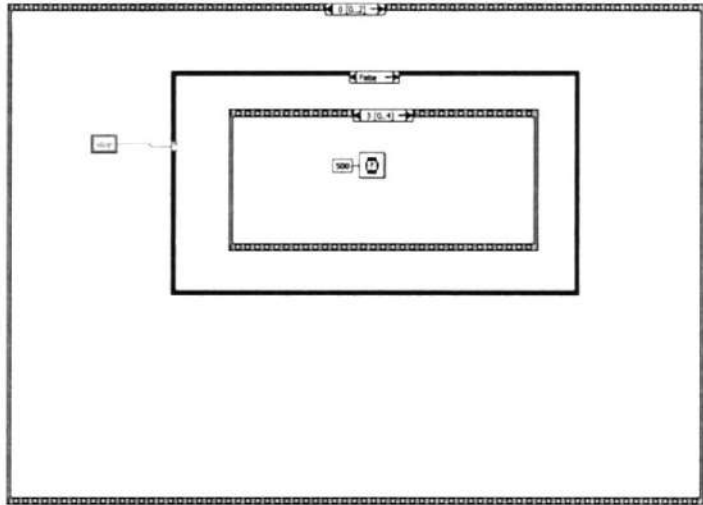


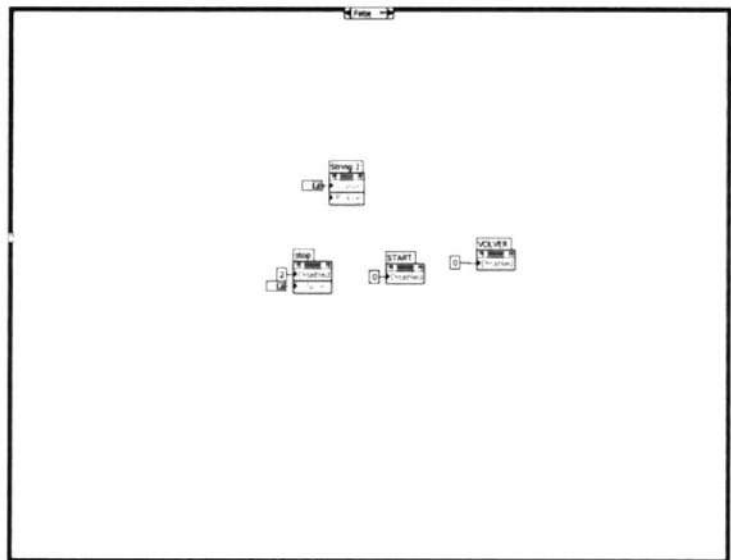
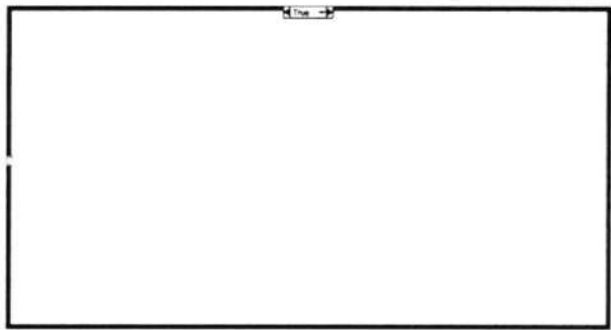
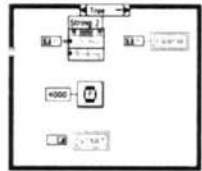
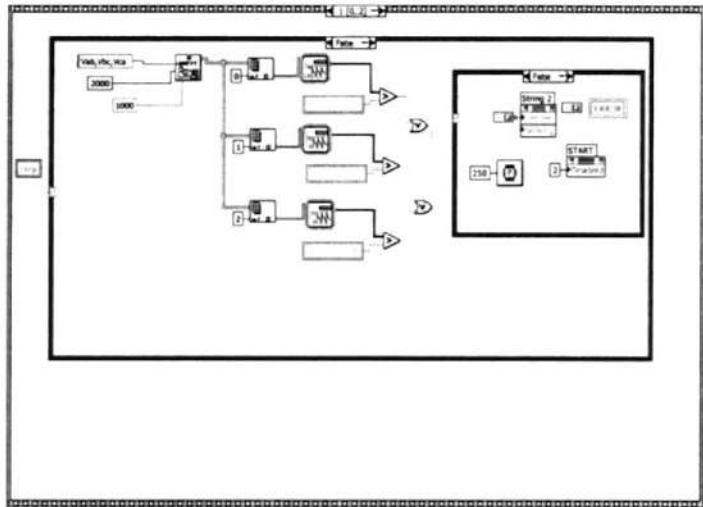


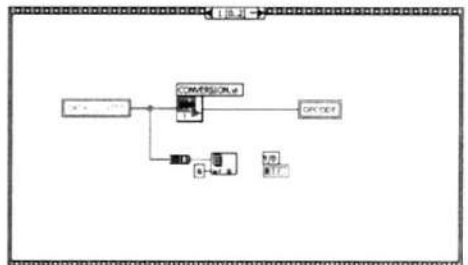
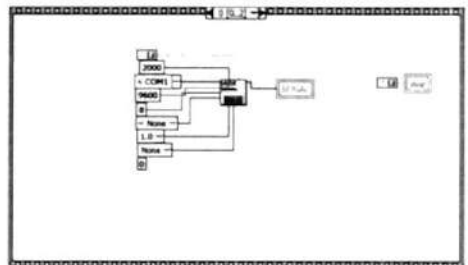
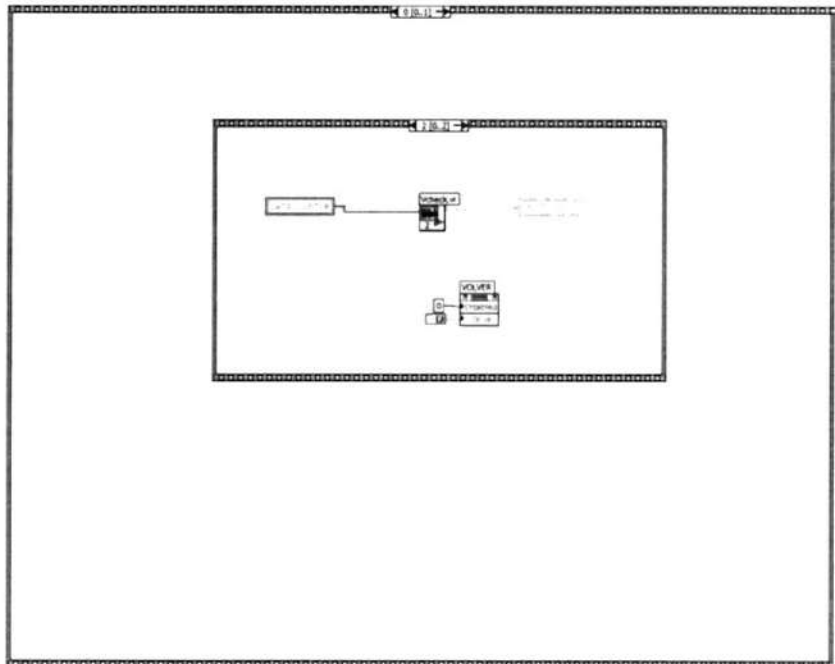












Express VI Configuration Information



Amplitude and Level Measurements

Amplitude and Level Measurements
Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: RMS
Voltage Window: Off



Amplitude and Level Measurements2

Amplitude and Level Measurements
Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: RMS
Voltage Window: Off



Amplitude and Level Measurements3

Amplitude and Level Measurements
Performs voltage measurements on a signal.

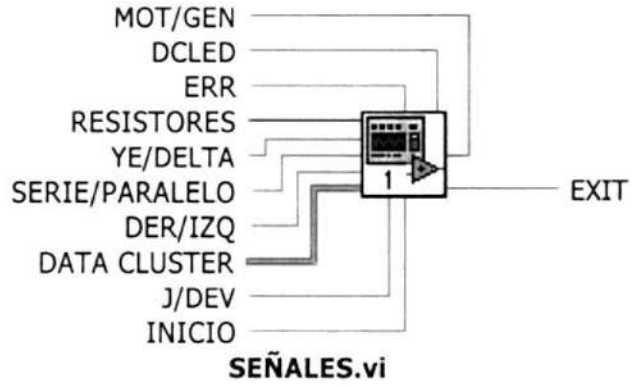
This Express VI is configured as follows:

Measurement: RMS
Voltage Window: Off



CIB -ESPOL

Connector Pane



Front Panel

Velocidad (rpm)

0
DETENER

Grabar ondas: OFF

CONEXIÓN:
YE

BOBINAS CONECTADAS EN:
SERIE

SENTIDO DE GIRO:
DERECHA

OK

SOBREVOLTAJE SOBRECORRIENTE

CORRIENTES
VOLTAJES
POTENCIA

5
CAMBIO DE GIRO
RESET
SALIR

350
300
200
100
0
-100
-200
-300
-350
0 0,005 0,01 0,015 0,02 0,025 0,03 0,035 0
Time

Ia

15
10
5
0
-5
-10
-15
0 0,1 0,2 0,3 0,4 0,5 0,6 0,7 0,8 0,9 1 1,1 1,2
Time

Ib

15
10
5
0
-5
-10
-15
0 0,1 0,2 0,3 0,4 0,5 0,6 0,7 0,8 0,9 1 1,1 1,2
Time

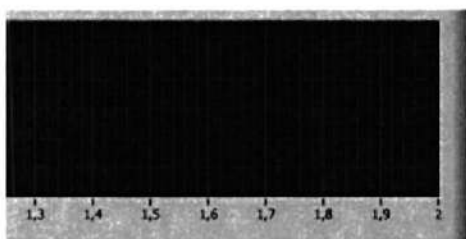
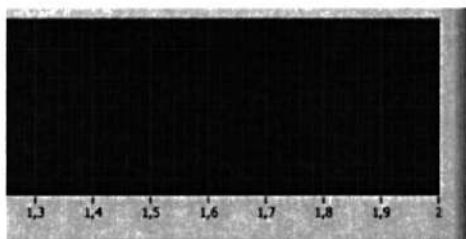
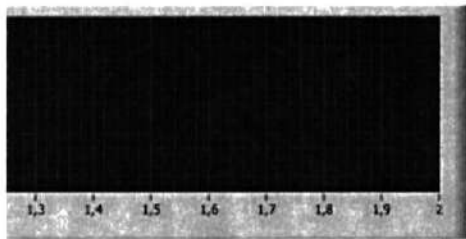
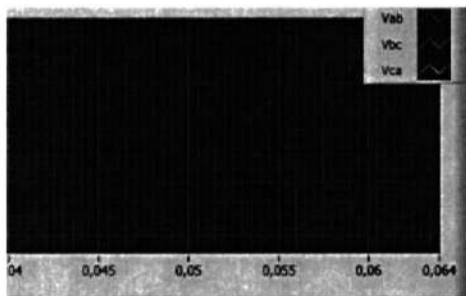
Ic

15
10
5
0
-5
-10
-15
0 0,1 0,2 0,3 0,4 0,5 0,6 0,7 0,8 0,9 1 1,1 1,2
Time

CONFIGURAR ROTOR DEVIADO DC-ON DC-CF

Fin

Current Check
0
RMS Vab
0,00



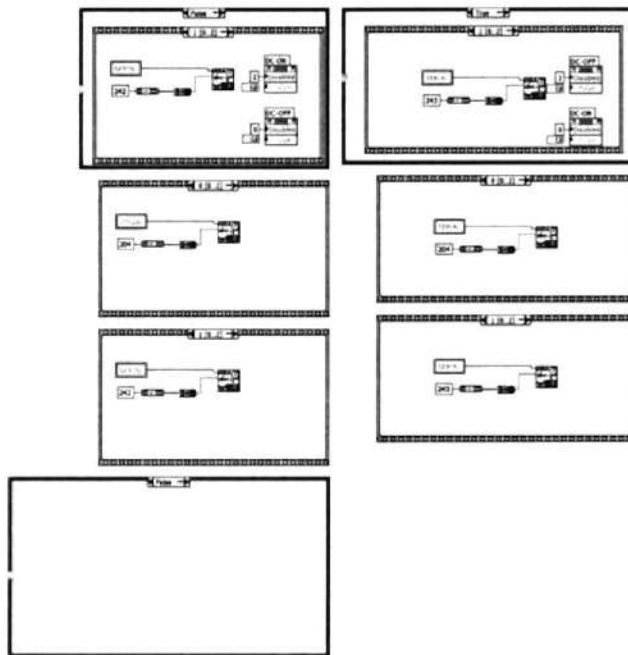
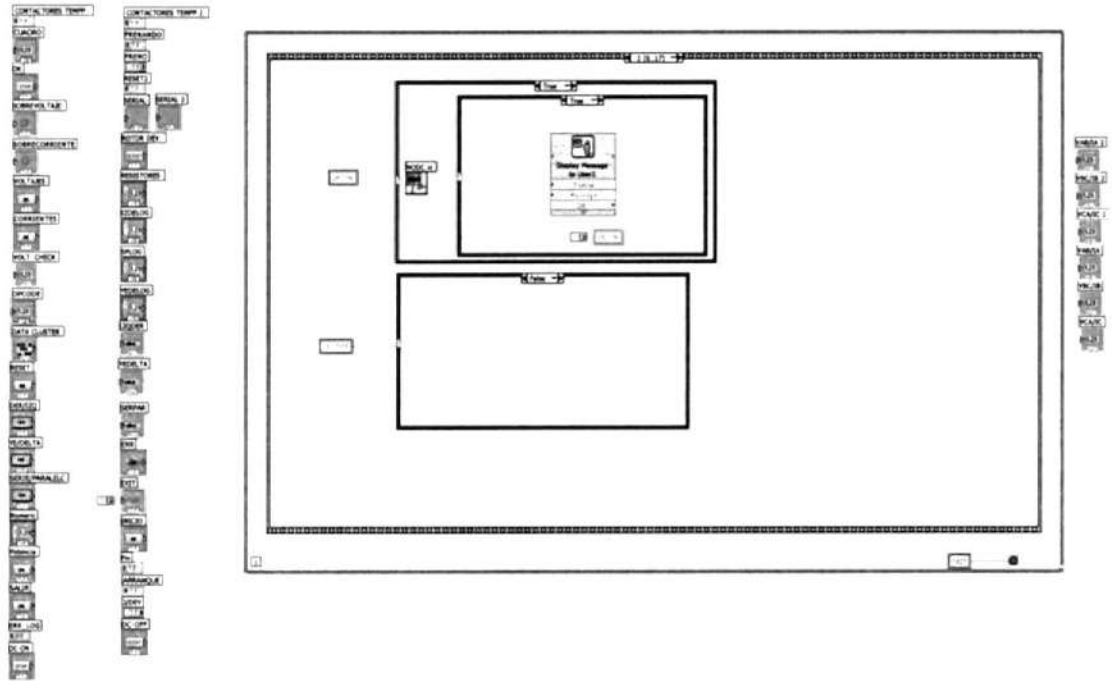
DATA CLUSTER

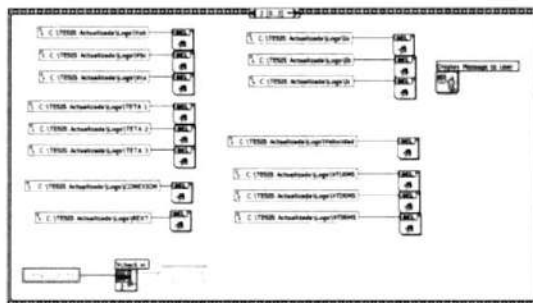
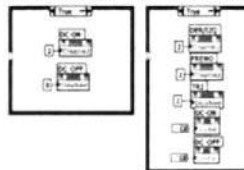
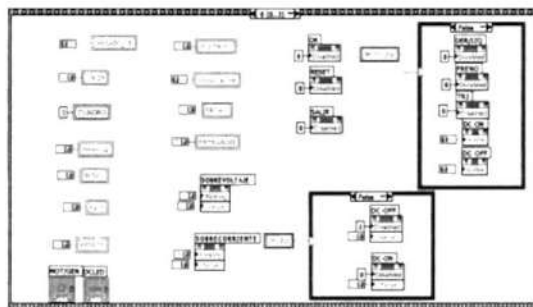
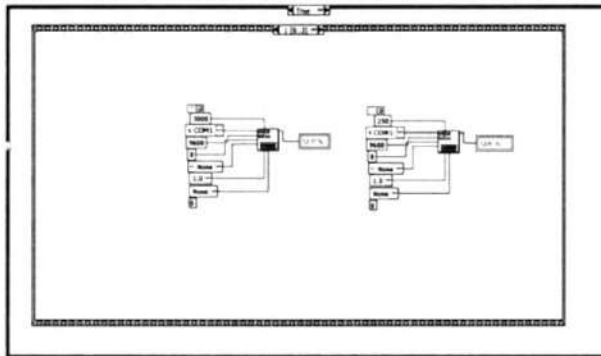
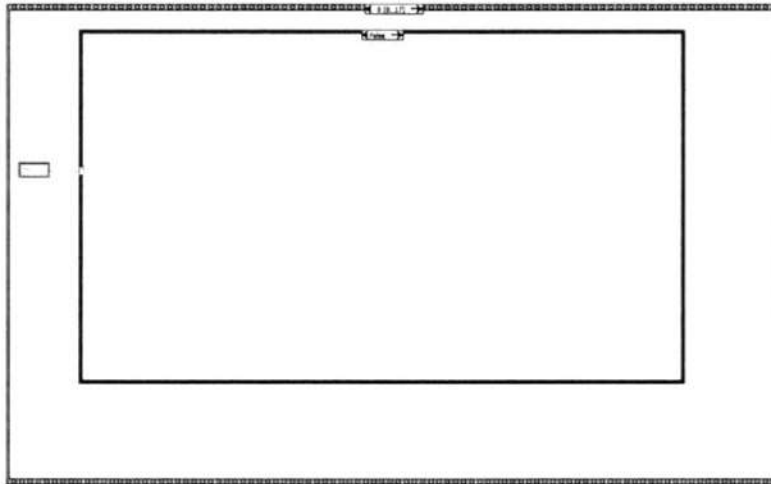
D0	D1	D2	D3	D4	D5	D6

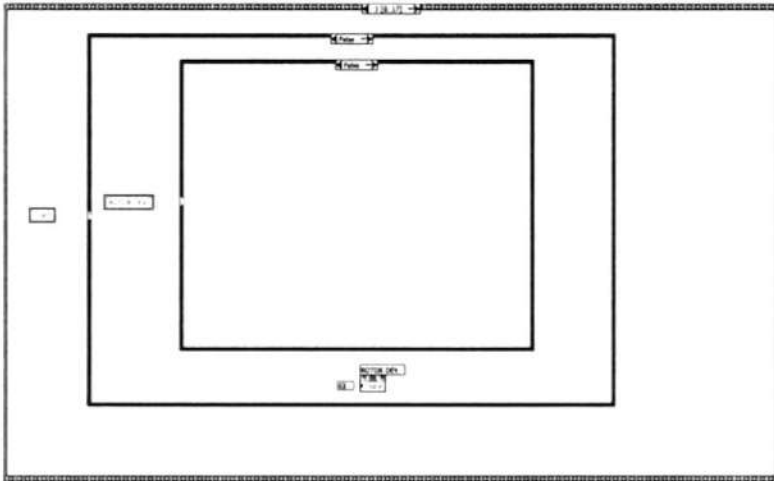
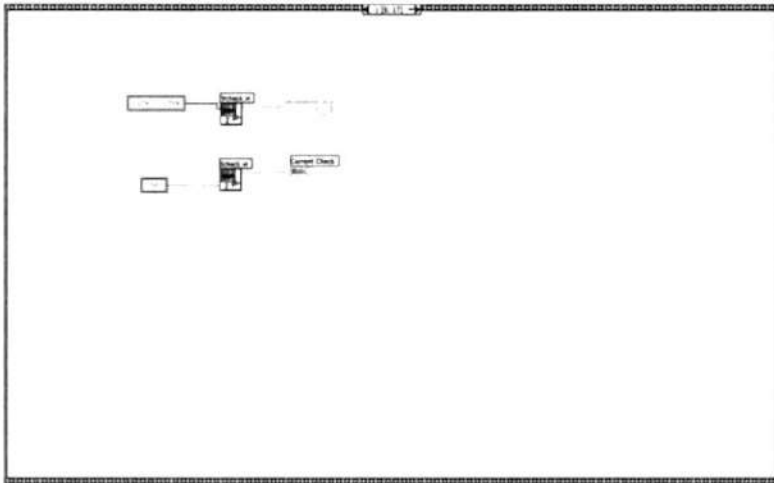
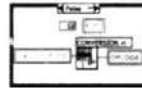
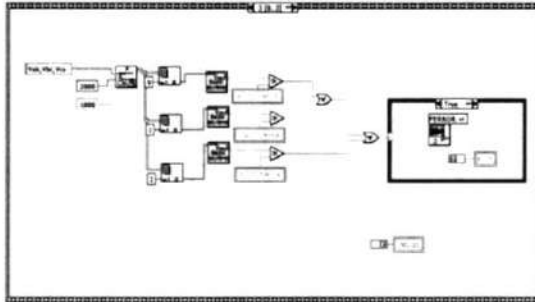
OPCODE

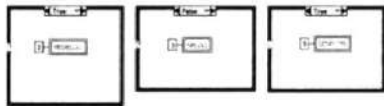
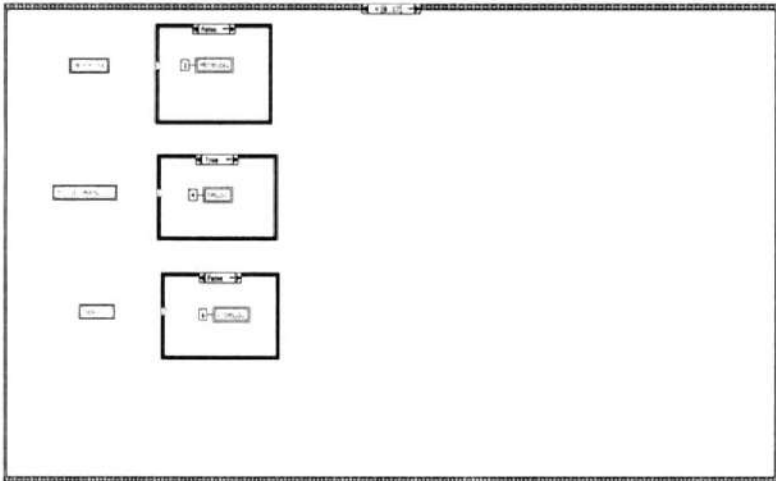
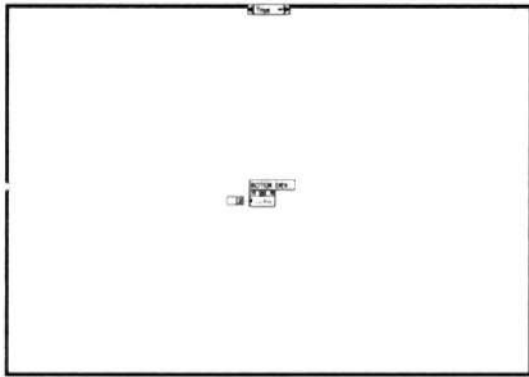
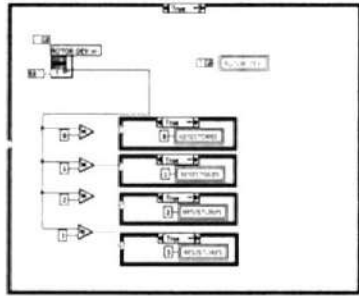
0

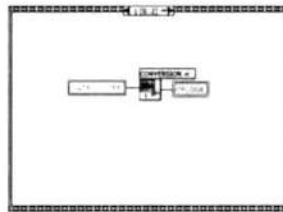
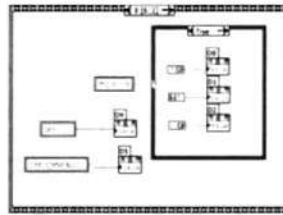
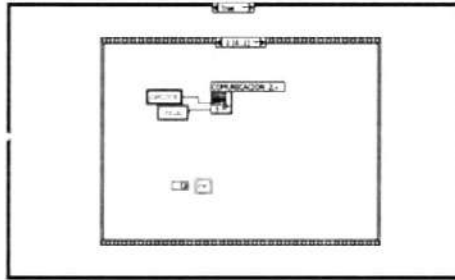
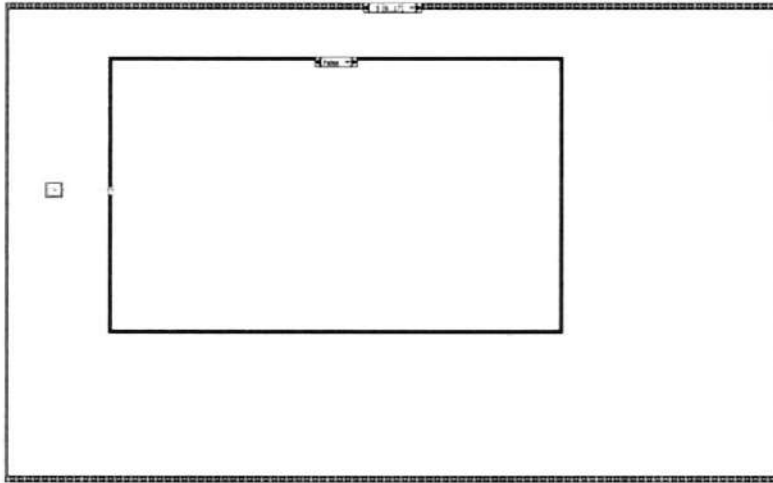
Block Diagram

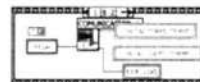
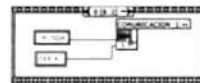
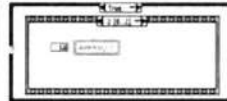
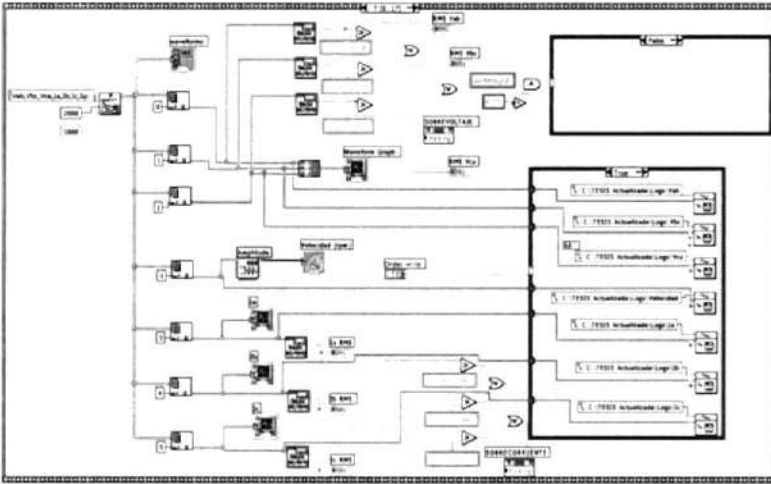
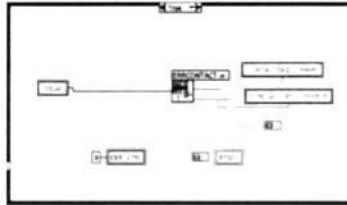
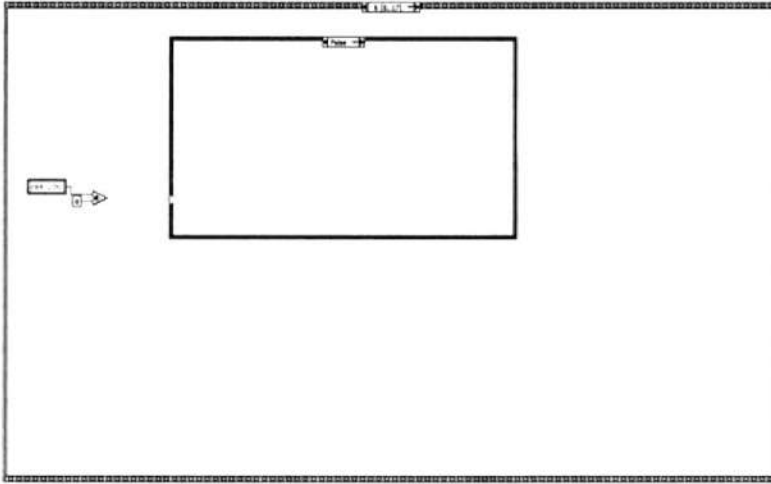


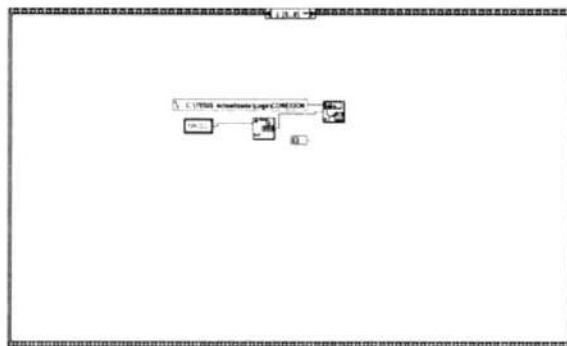
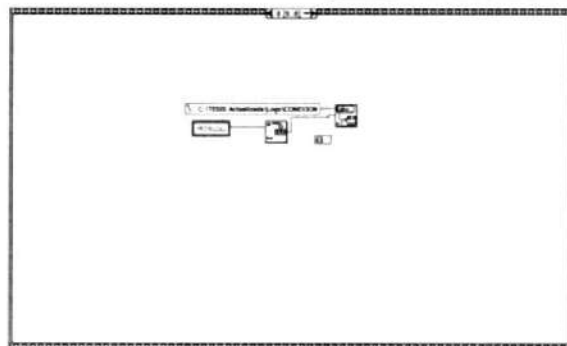
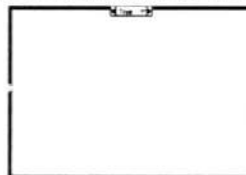
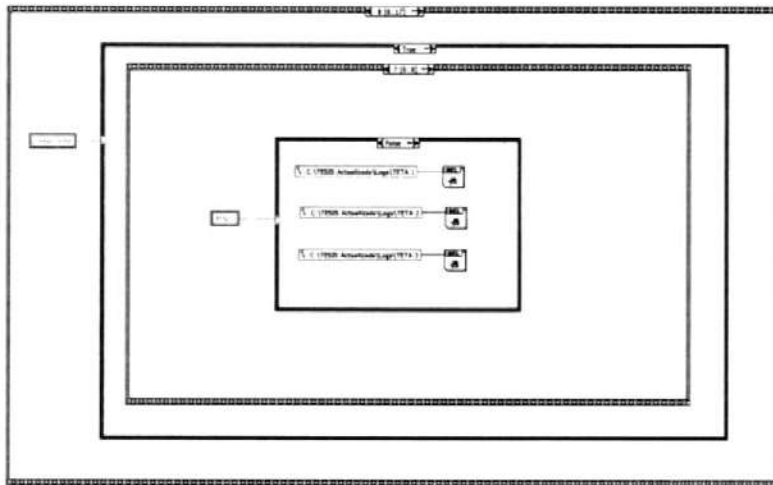


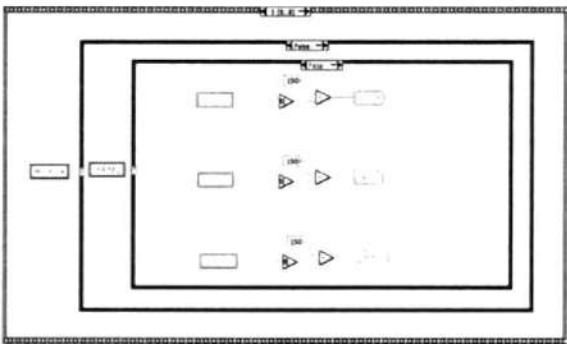
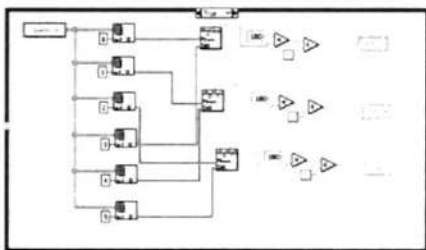
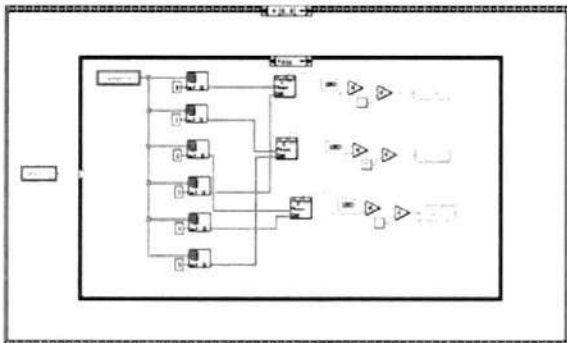
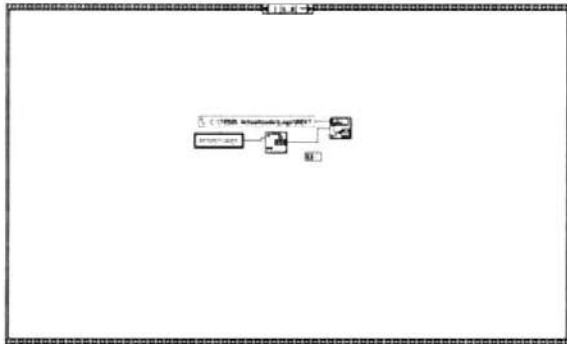
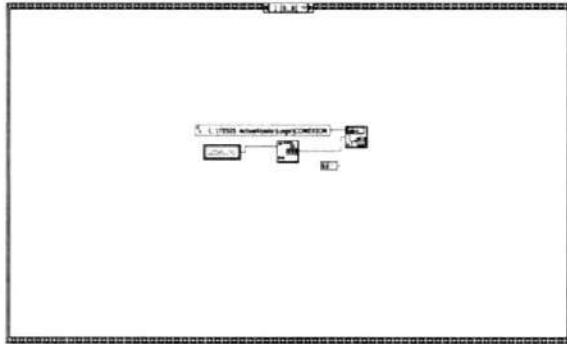




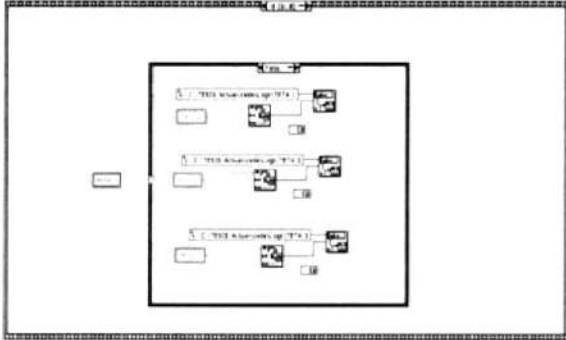
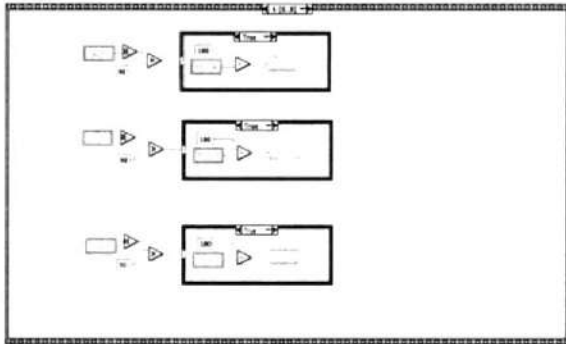
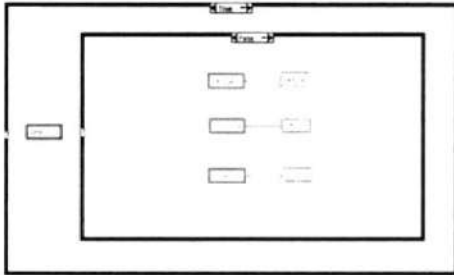
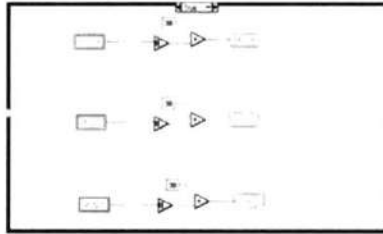


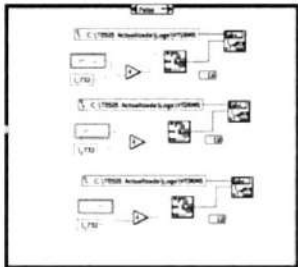
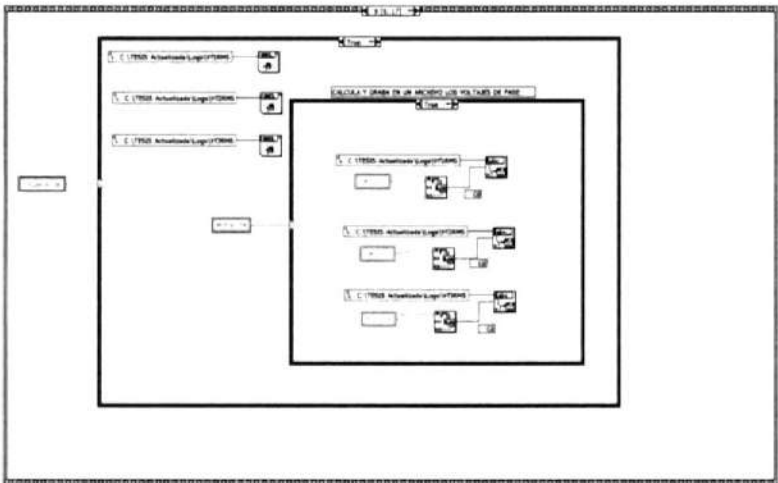
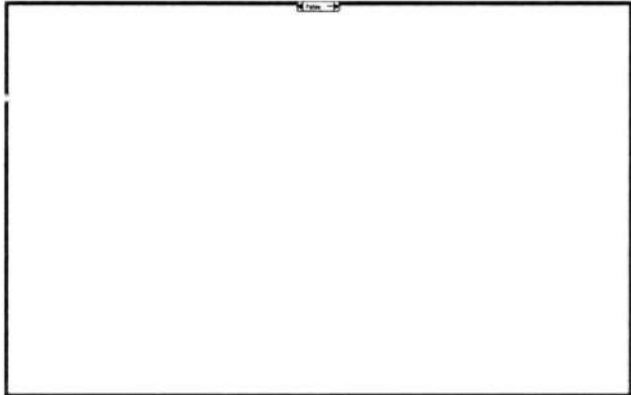
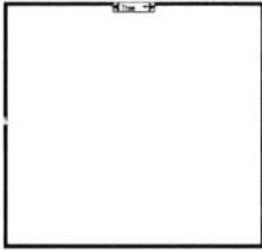


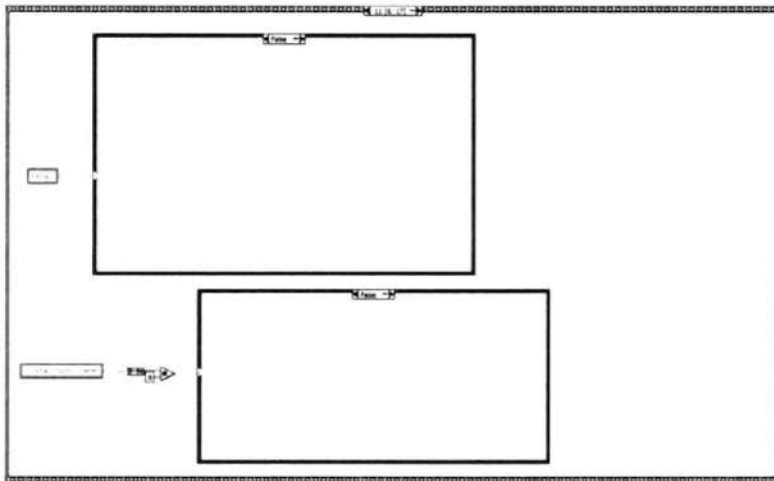
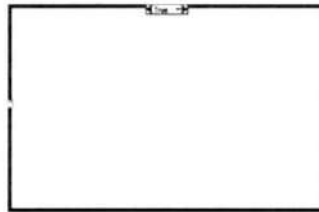
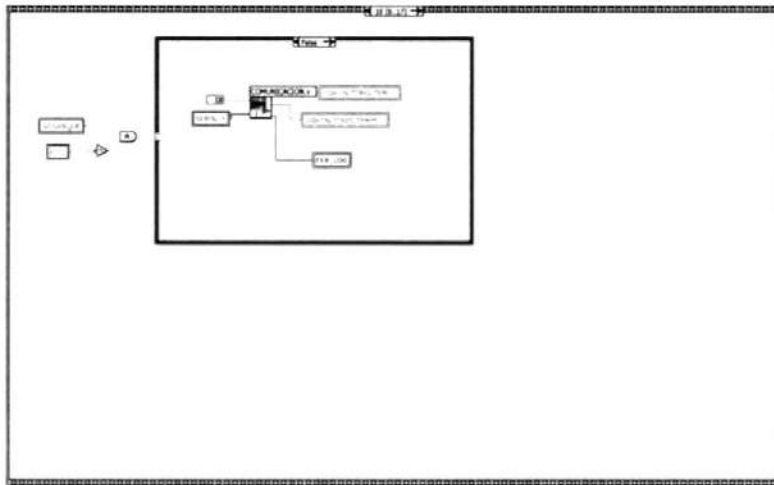
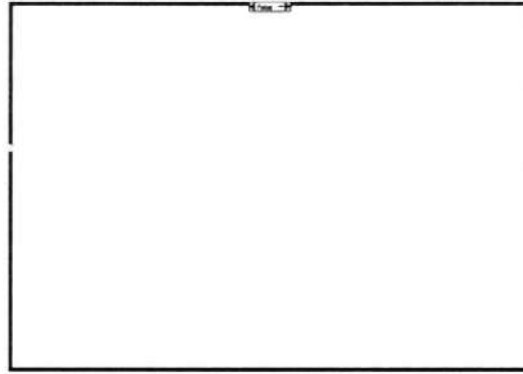


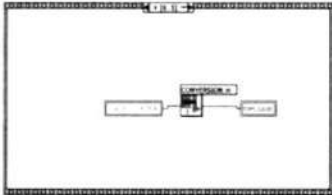
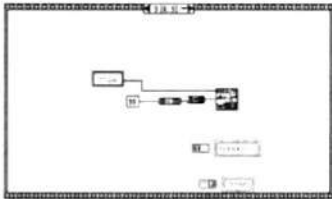
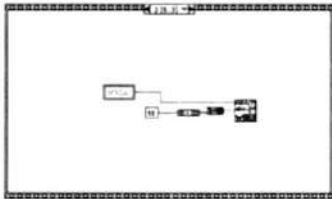
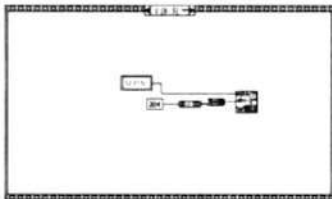
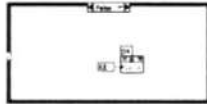
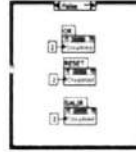
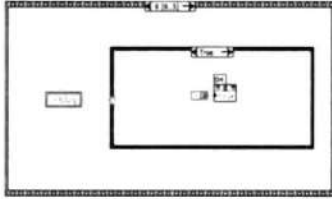
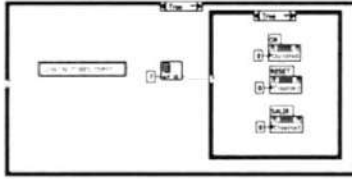
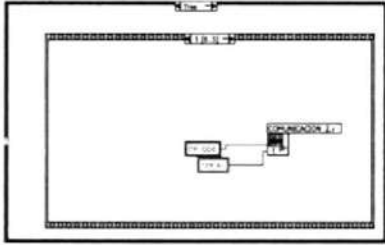


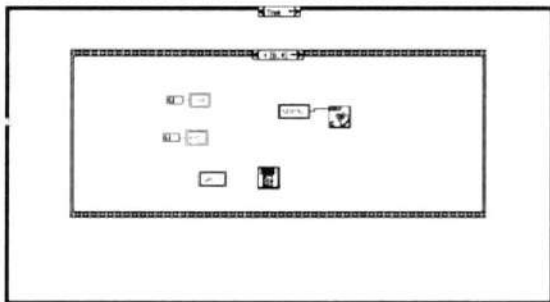
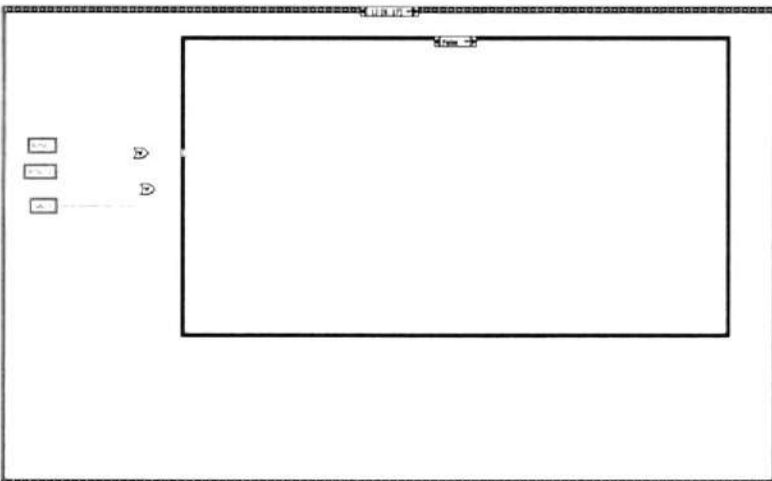
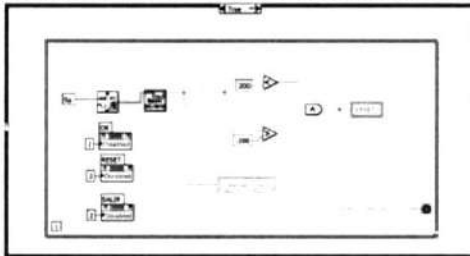
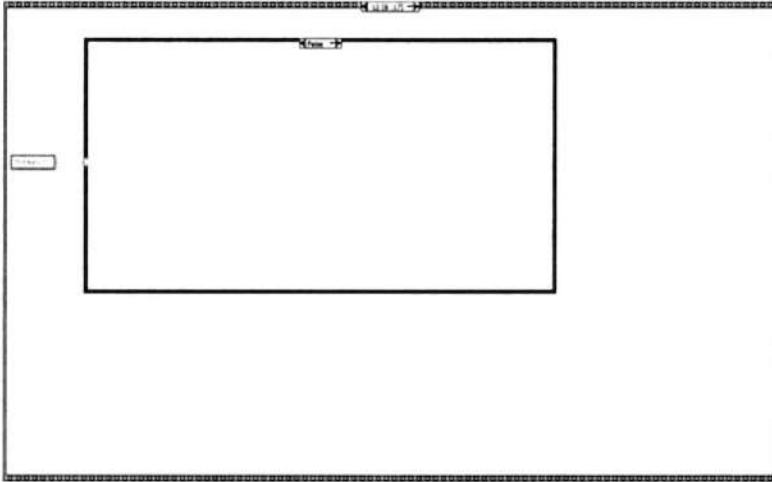
CIB - ESPOL

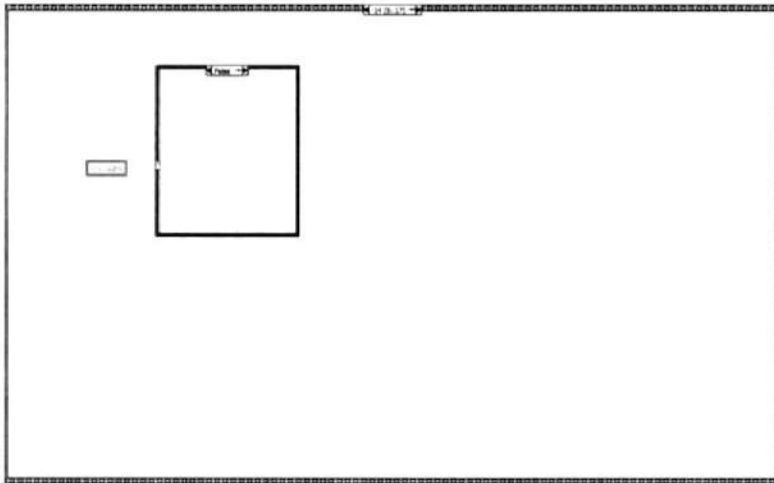
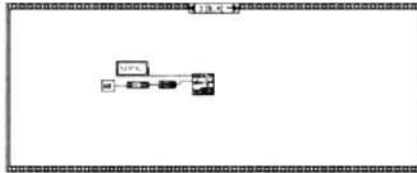
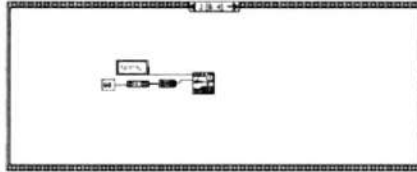


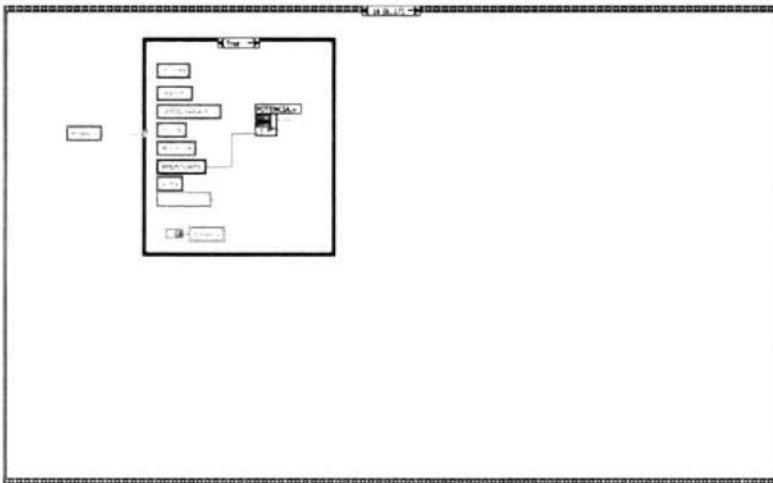
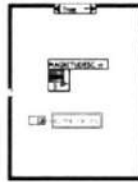
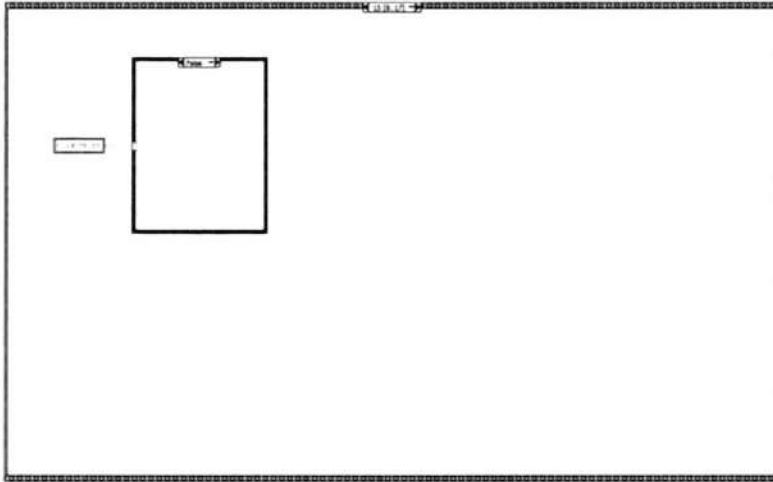


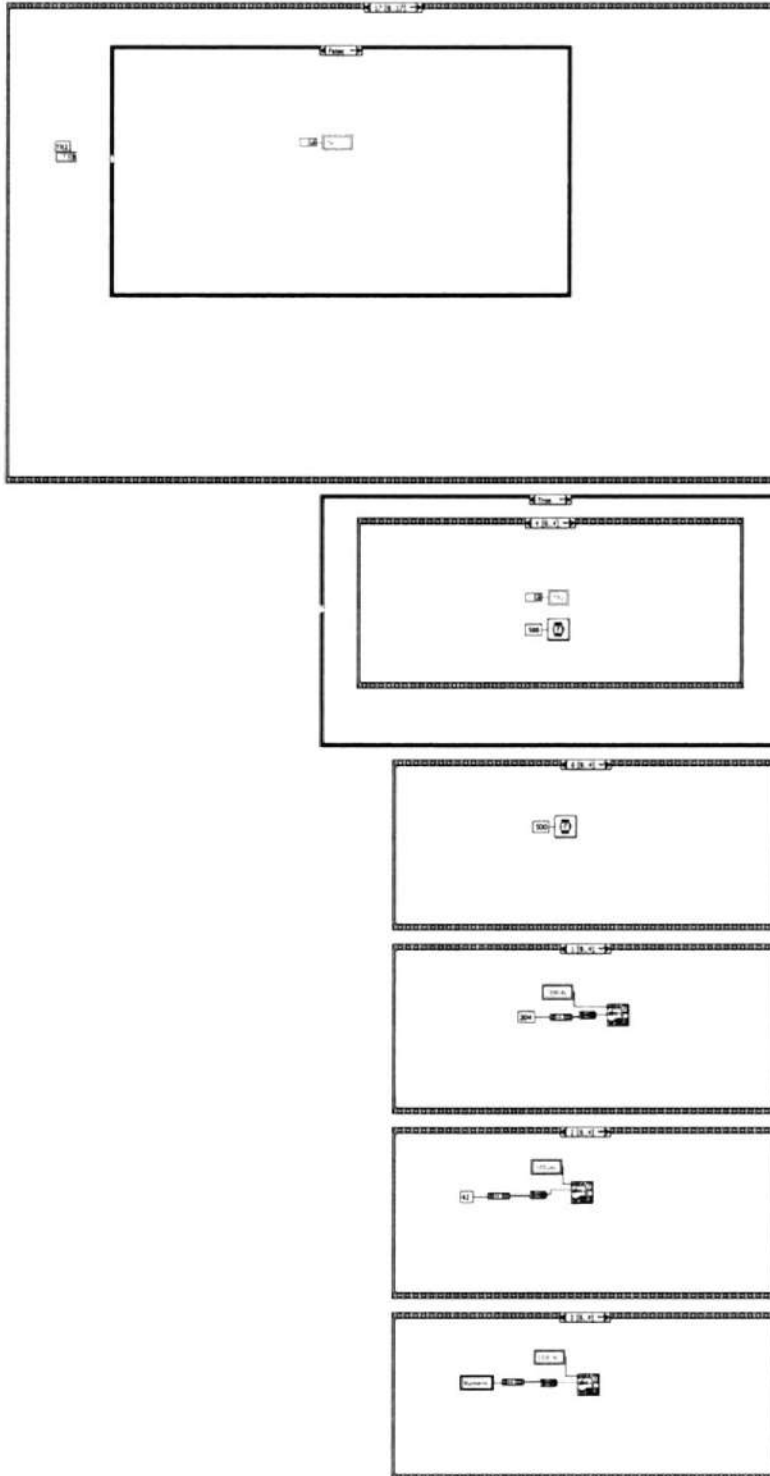












Express VI Configuration Information



Amplitude

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: DC

Voltage Window: Off



Display Message to User

Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

This Express VI is configured as follows:

Message:



Display Message to User2

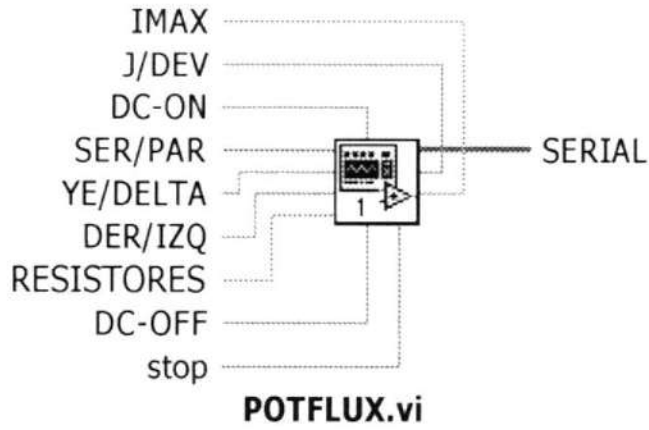
Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

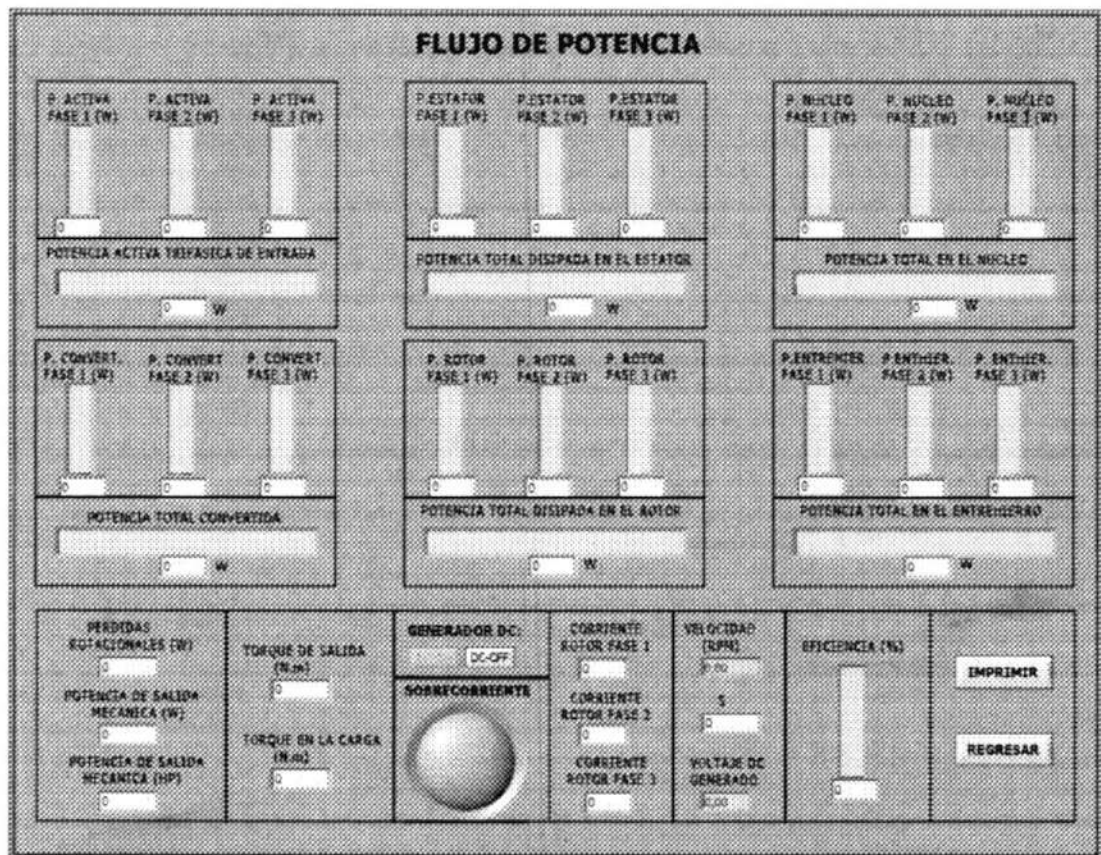
This Express VI is configured as follows:

Message:

Connector Pane

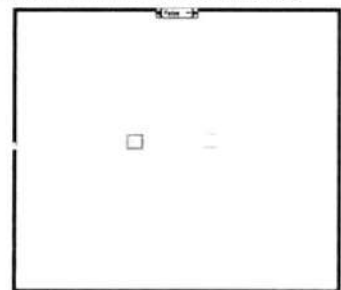
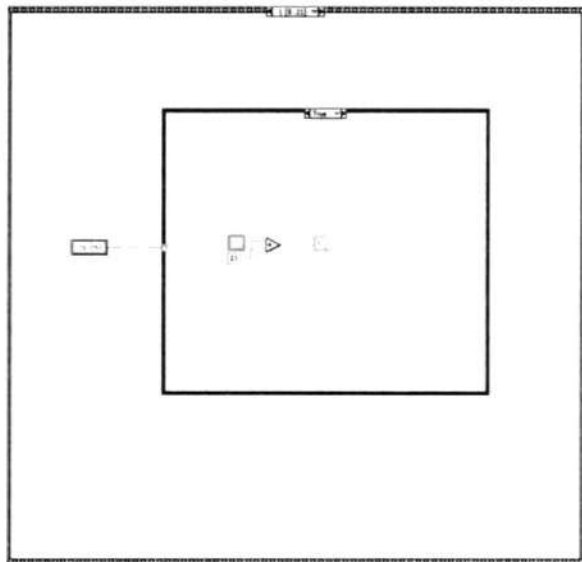
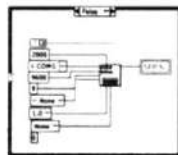
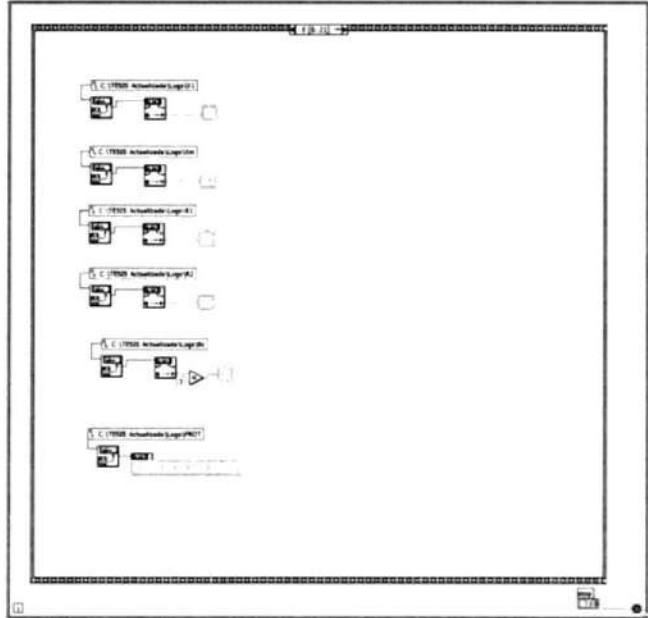
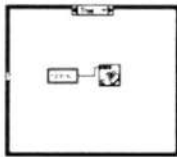


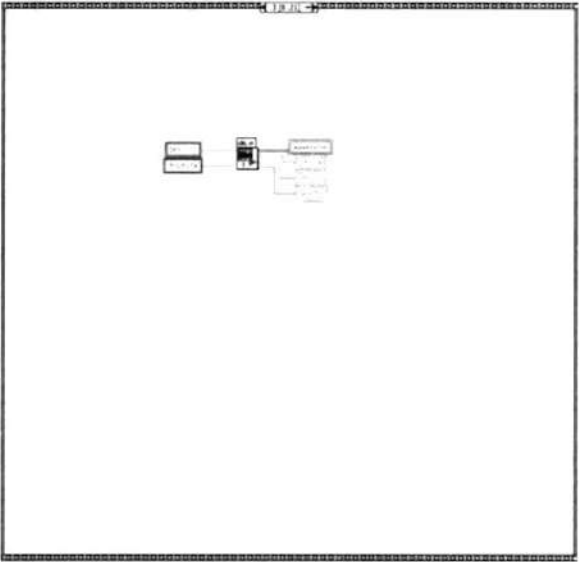
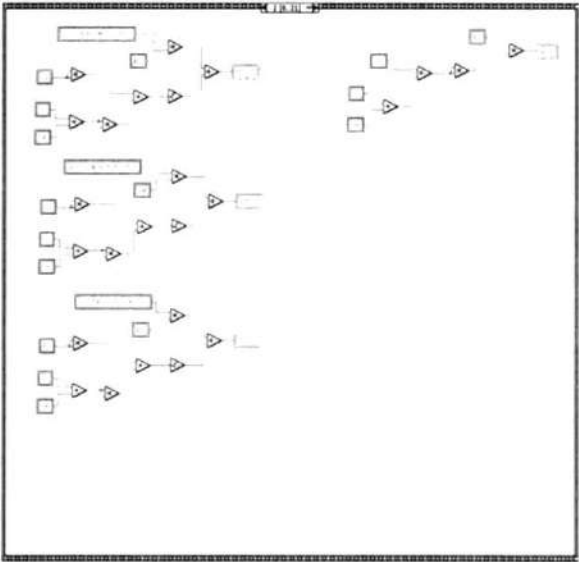
Front Panel

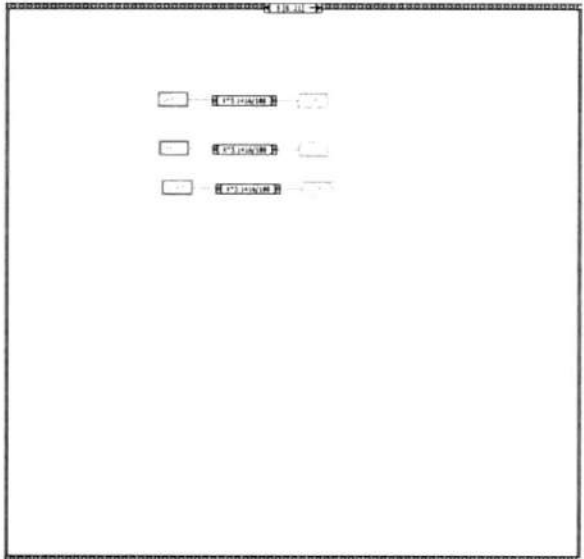
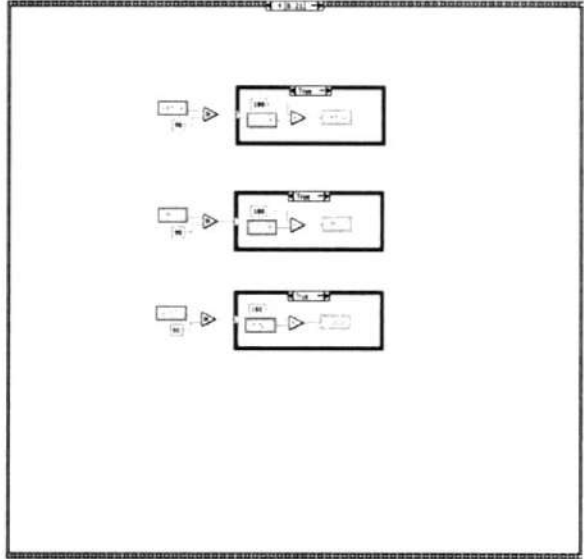


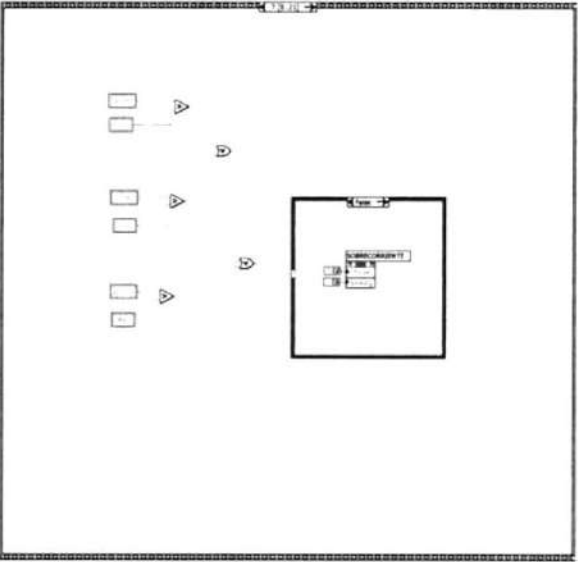
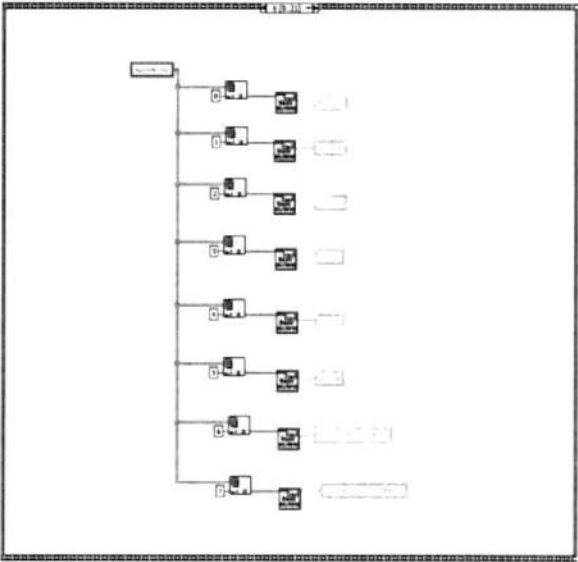
CIB-ESPOL

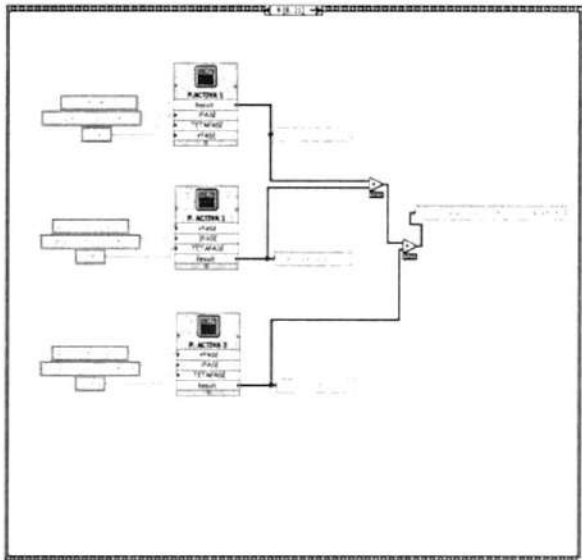
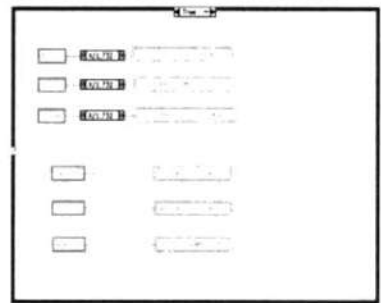
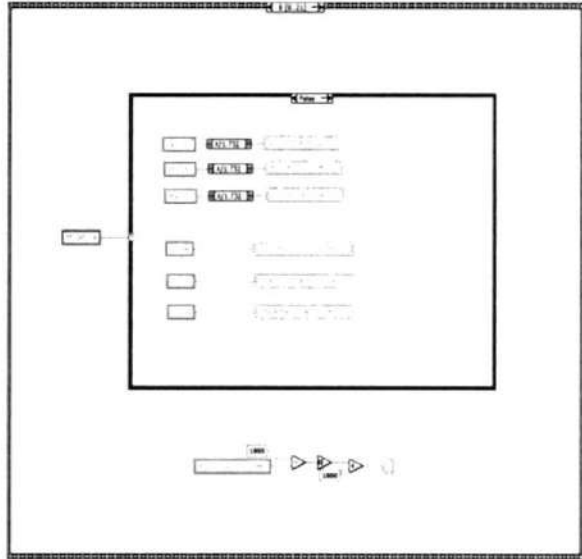
Block Diagram

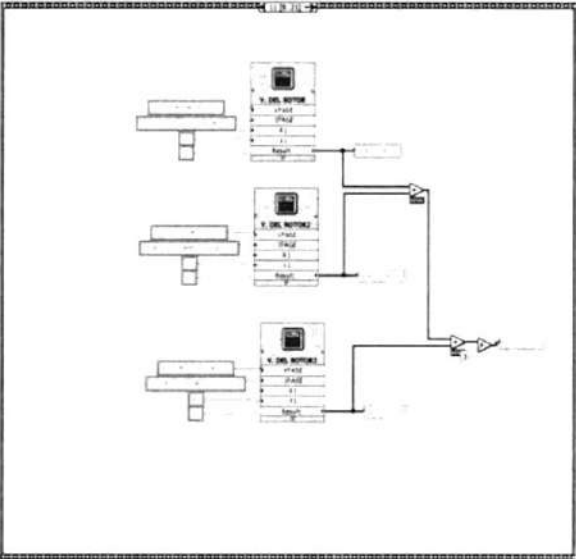
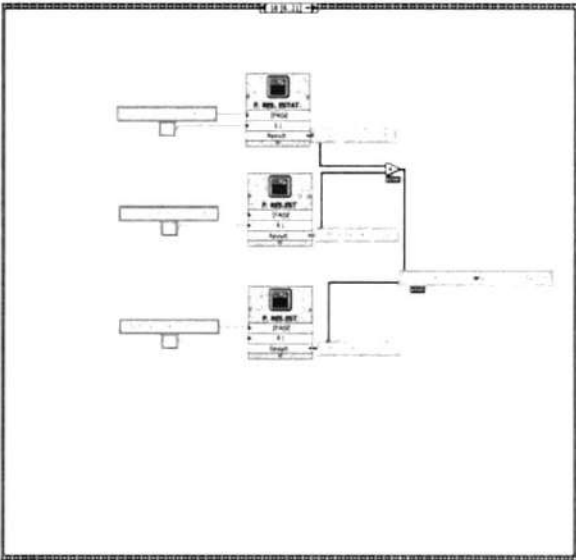


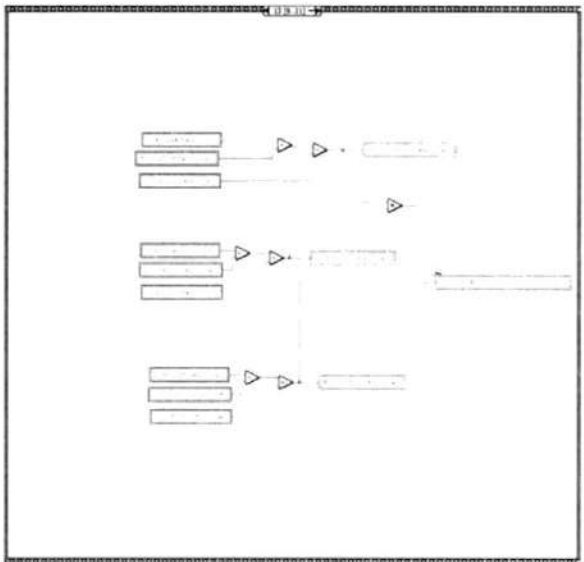
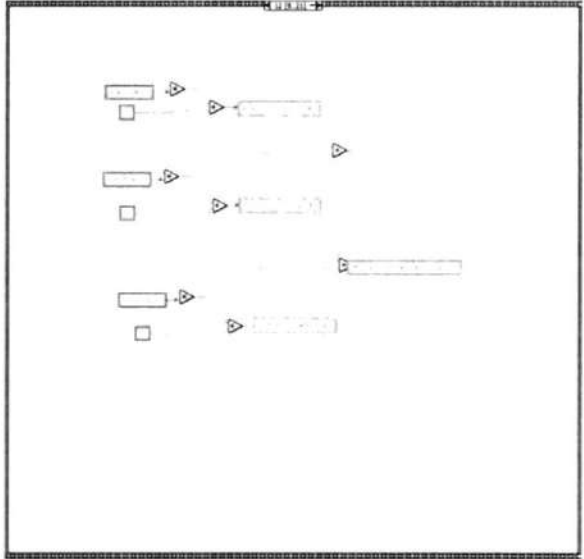


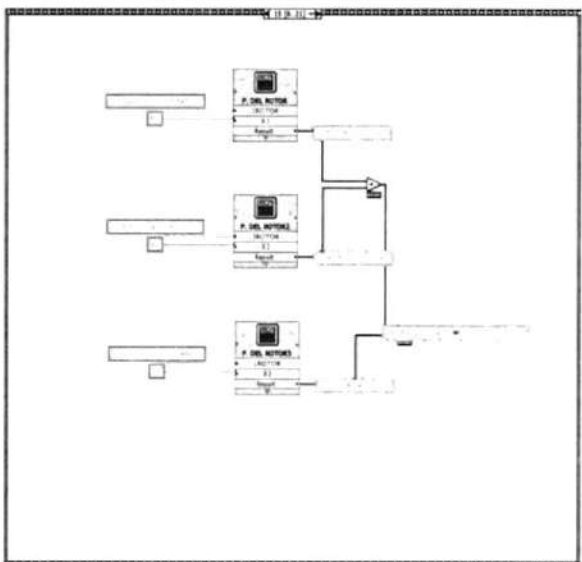
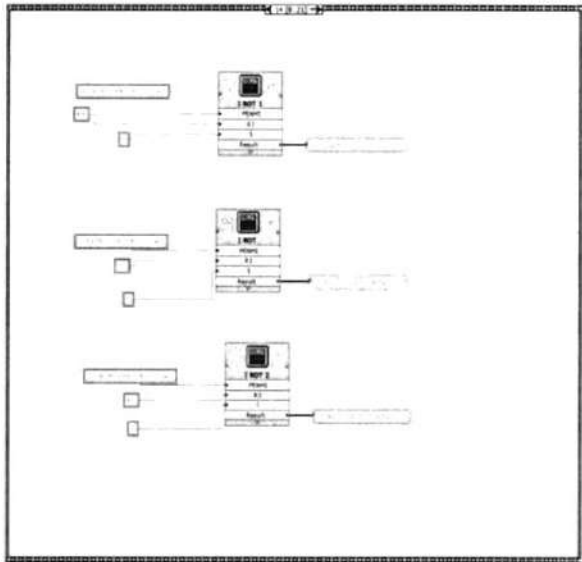


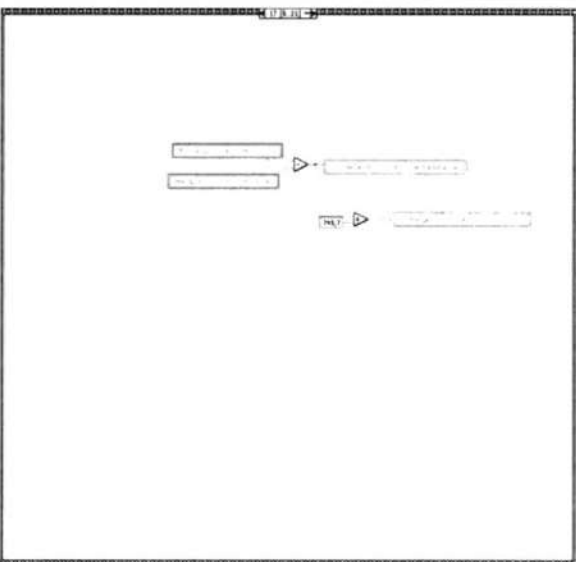
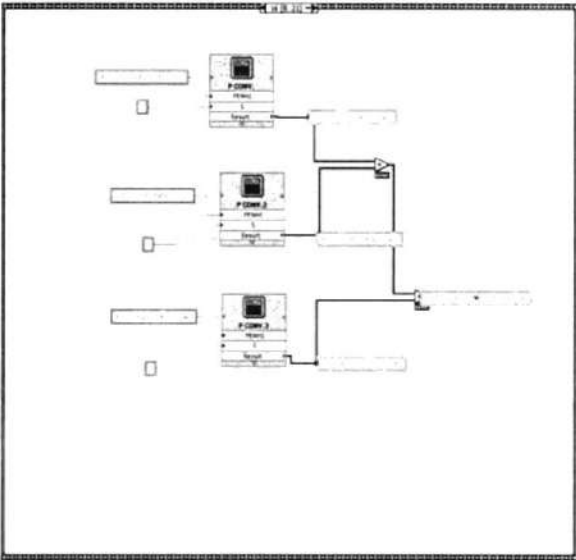


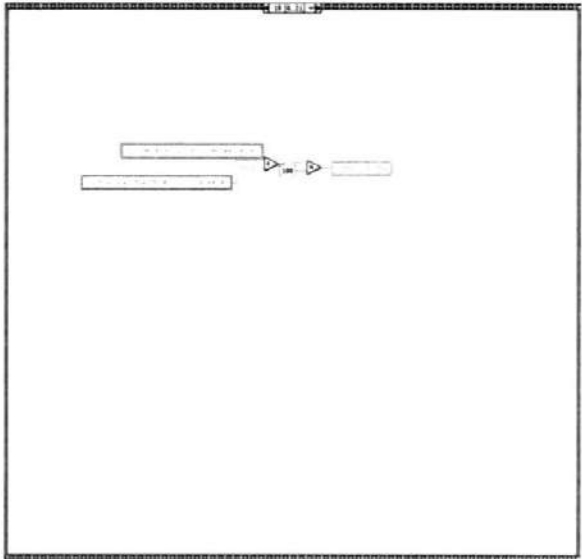
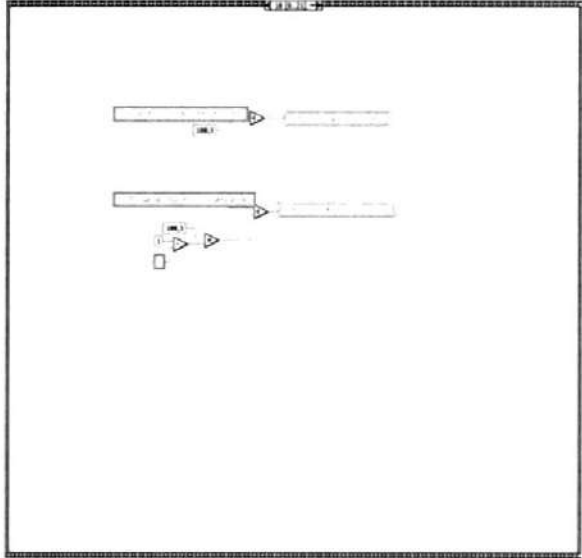


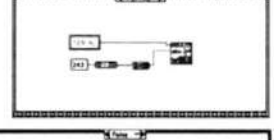
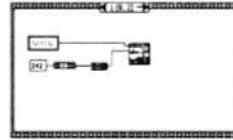
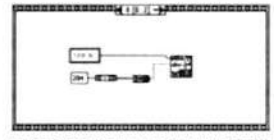
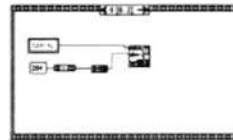
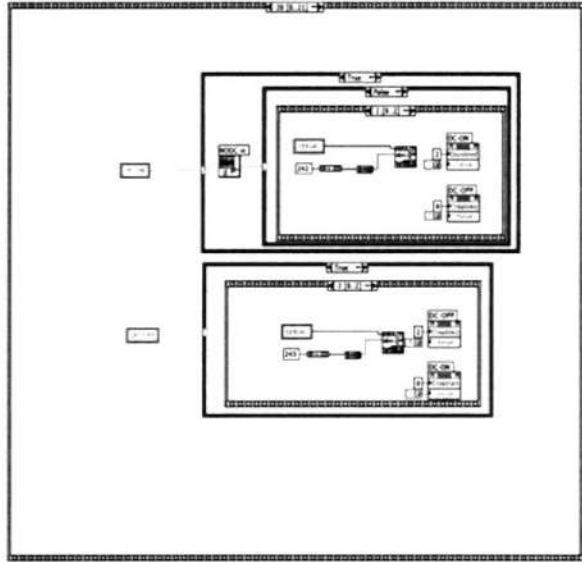


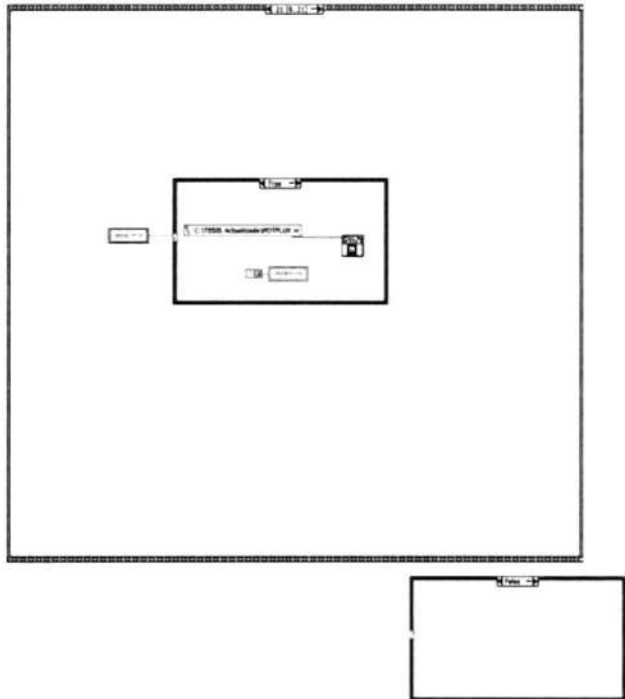












Express VI Configuration Information



P.ACTIVA 1

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $V_{FASE} * I_{FASE} * \cos(TETA_{FASE})$



P. ACTIVA 2

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $V_{FASE} * I_{FASE} * \cos(TETA_{FASE})$



P. ACTIVA 3

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $V_{FASE} * I_{FASE} * \cos(TETA_{FASE})$



P. RES. ESTAT.

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $I_{FASE} * I_{FASE} * R1$



P. RES. EST

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $I_{FASE} * I_{FASE} * R1$



P. RES. EST

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $I_{FASE} * I_{FASE} * R1$



CIB -ESPOL



V. DEL ROTOR

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $V_{\text{FASE}} - I_{\text{FASE}} \cdot \sqrt{(R_1 \cdot R_1) + (X_1 \cdot X_1)}$



V. DEL ROTOR2

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $V_{\text{FASE}} - I_{\text{FASE}} \cdot \sqrt{(R_1 \cdot R_1) + (X_1 \cdot X_1)}$



V. DEL ROTOR3

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $V_{\text{FASE}} - I_{\text{FASE}} \cdot \sqrt{(R_1 \cdot R_1) + (X_1 \cdot X_1)}$



P. DEL ROTOR

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $I_{\text{ROTOR}} \cdot I_{\text{ROTOR}} \cdot R_2$



P. DEL ROTOR2

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: IROTOR*IROTOR*R2



P. DEL ROTOR3

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: IROTOR*IROTOR*R2



I ROT 1

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: sqrt(PENHI*S/R2)



I ROT

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: sqrt(PENHI*S/R2)



I ROT 2

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $\text{sqrt}(\text{PENHI} * \text{S} / \text{R}2)$



P CONV.

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $(1 - \text{S}) * \text{PENHI}$



P CONV.2

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $(1 - \text{S}) * \text{PENHI}$



P CONV.3

Formula

Uses a calculator interface to create mathematical formulas. You can use this Express VI to perform most math functions that a basic scientific calculator can compute.

This Express VI is configured as follows:

Formula: $(1 - \text{S}) * \text{PENHI}$



Display Message to User

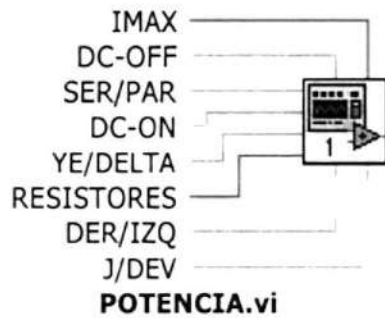
Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

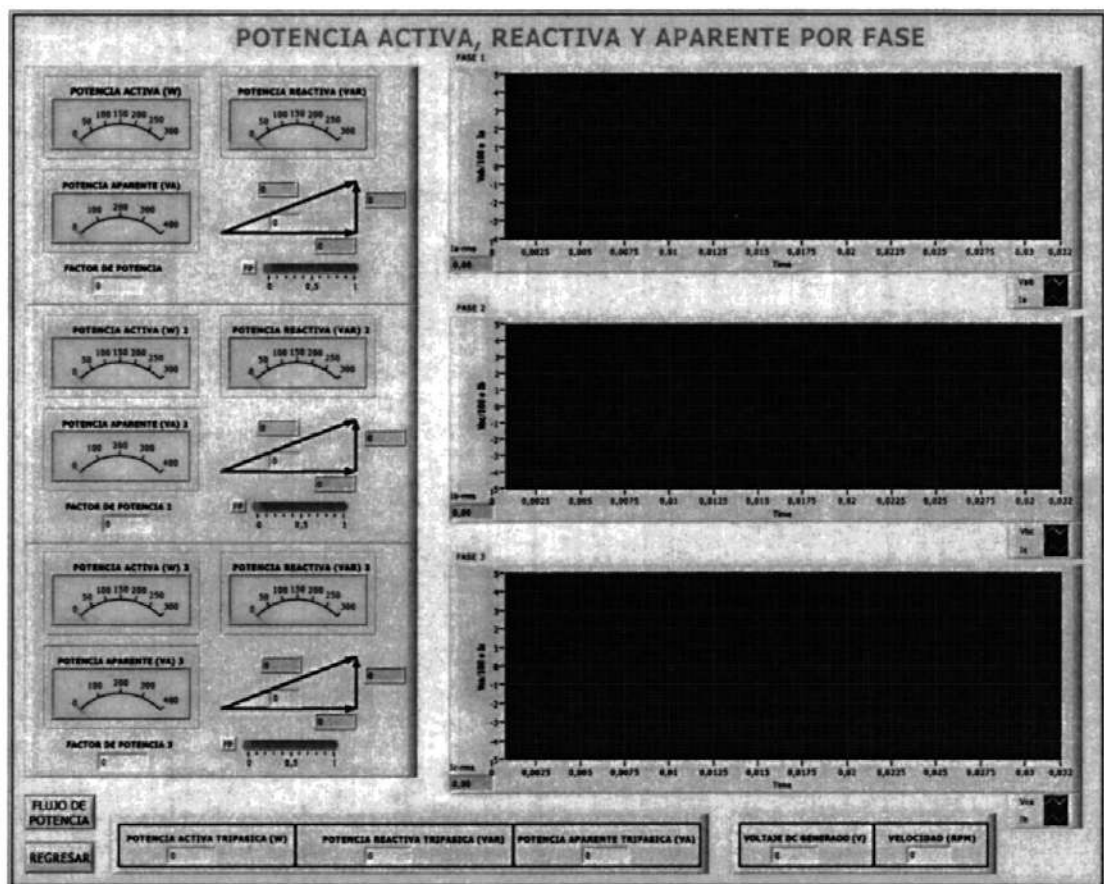
This Express VI is configured as follows:

Message:

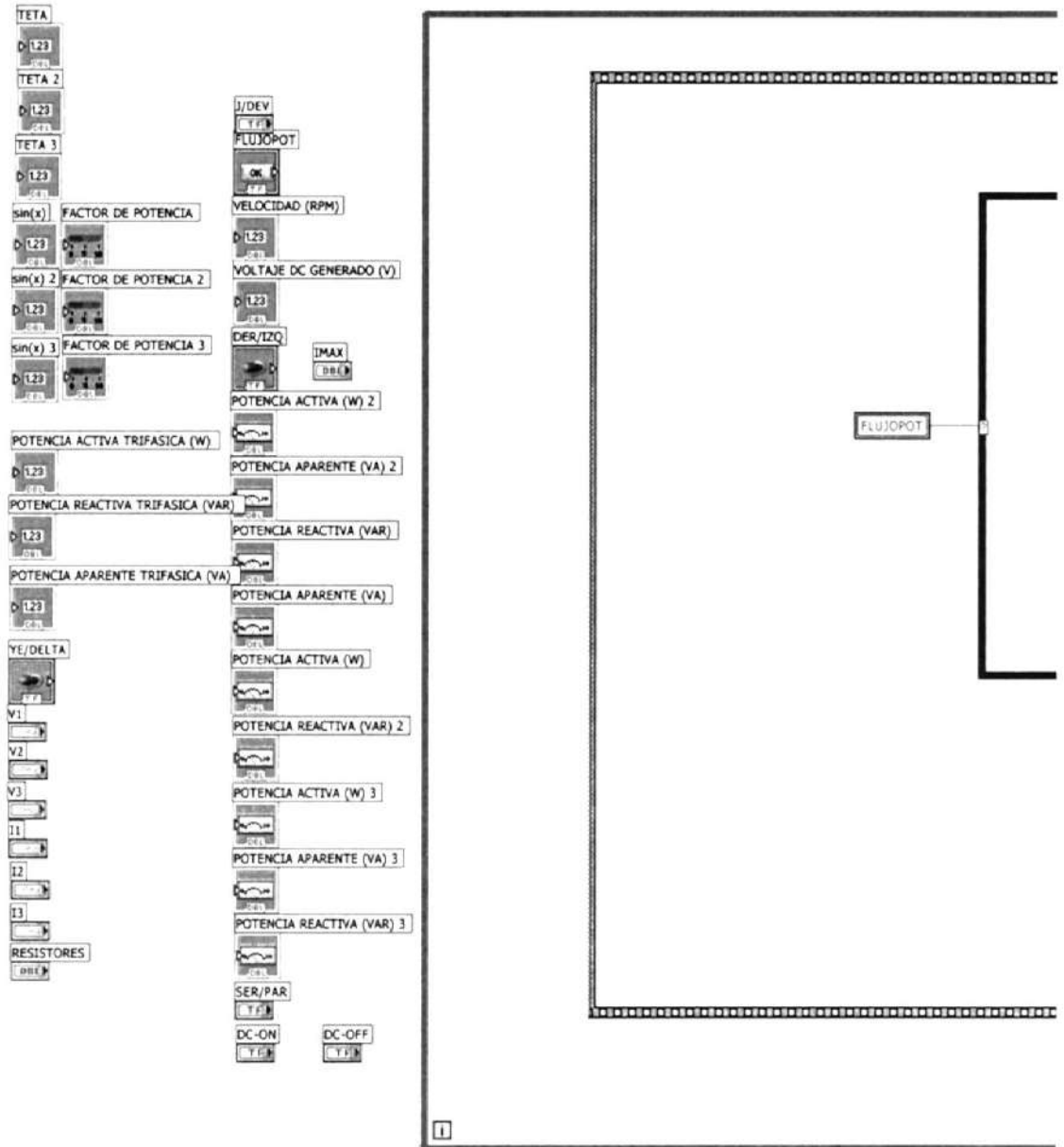
Connector Pane

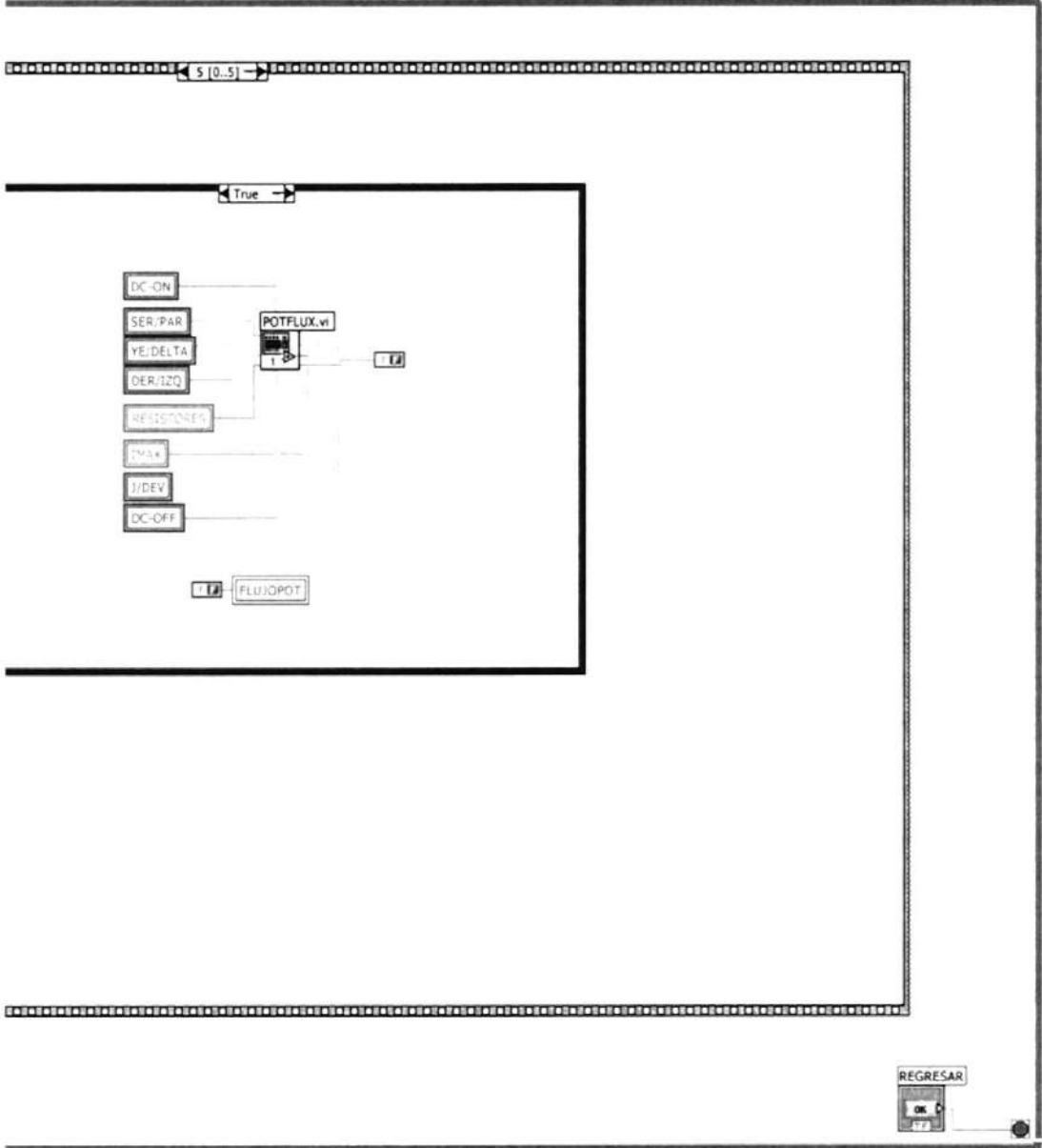


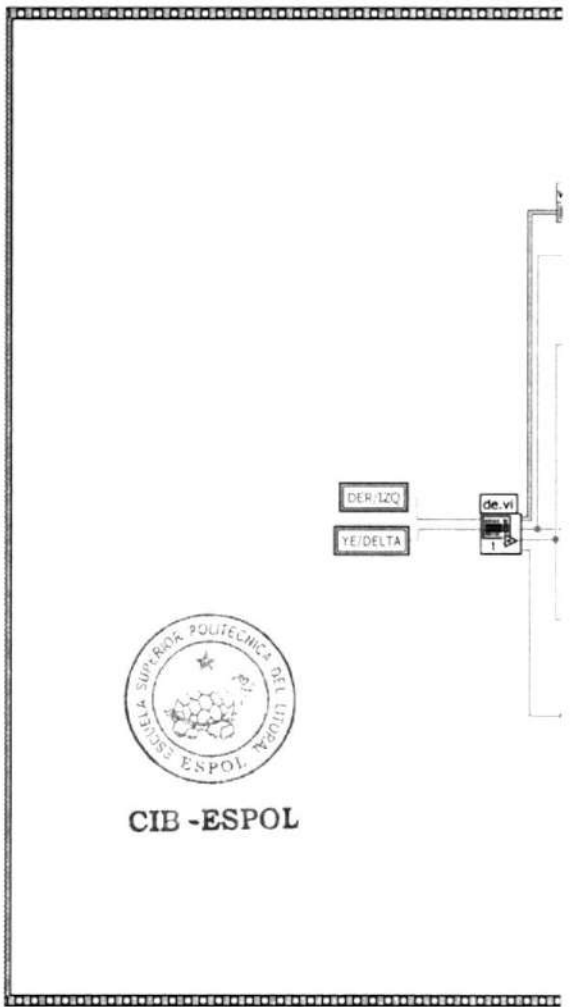
Front Panel

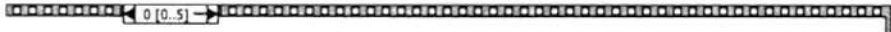
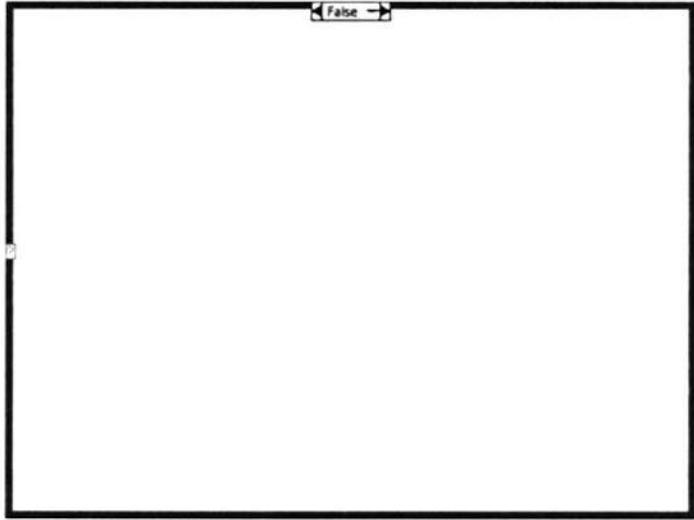


Block Diagram

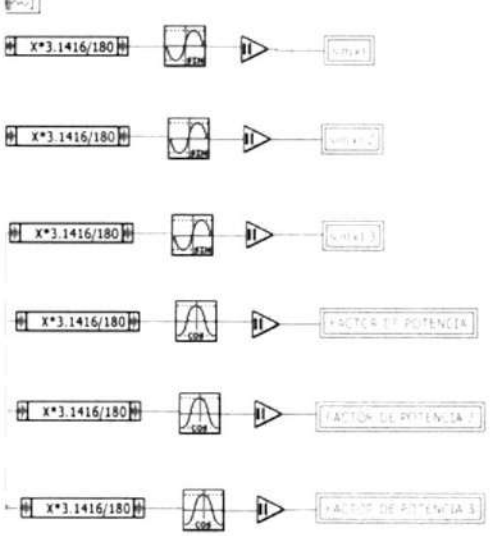


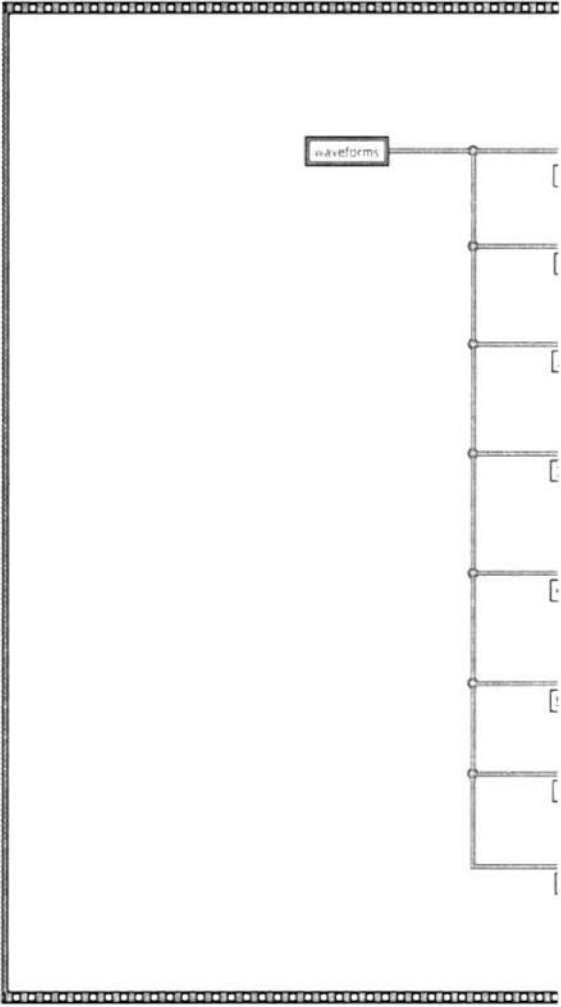


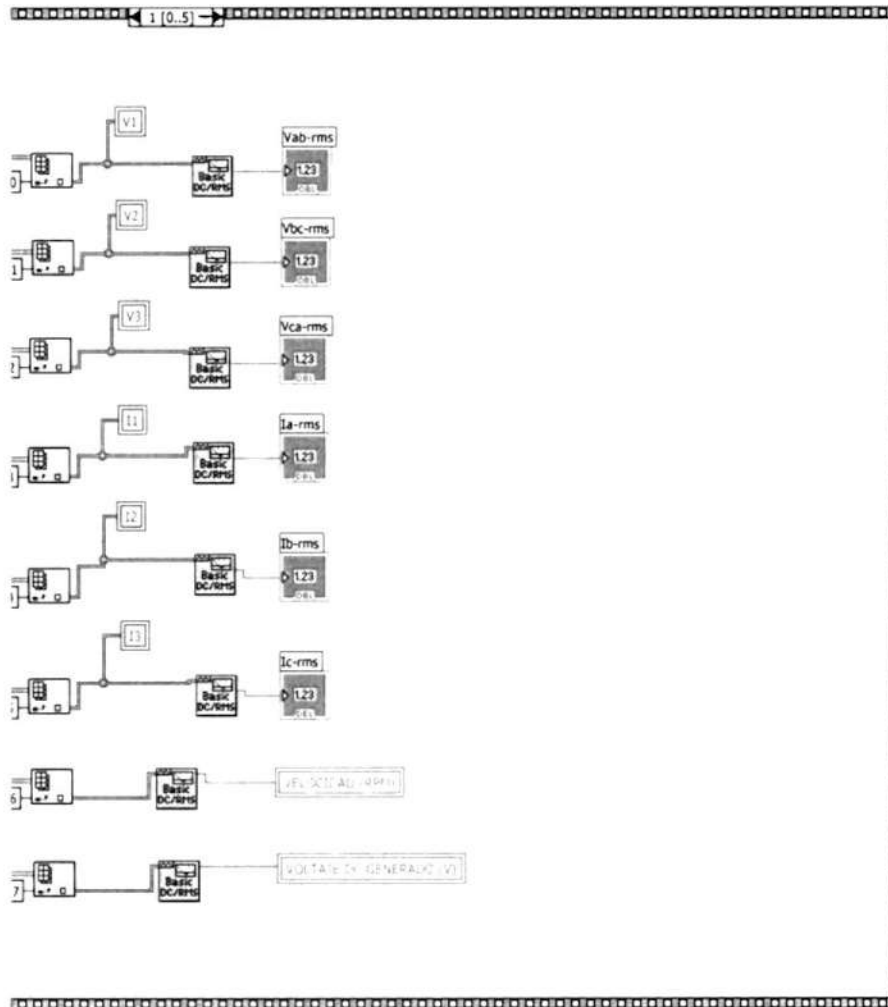


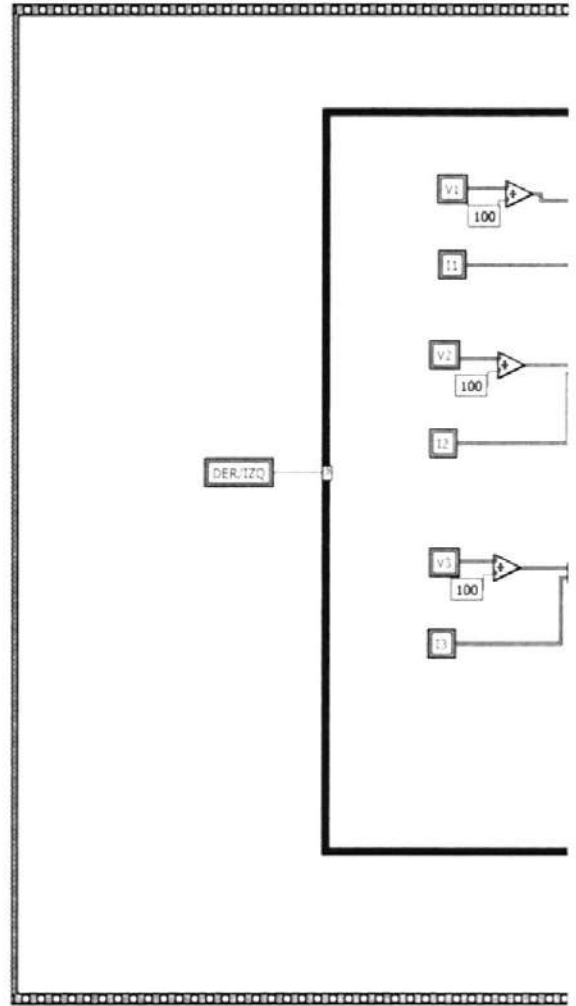


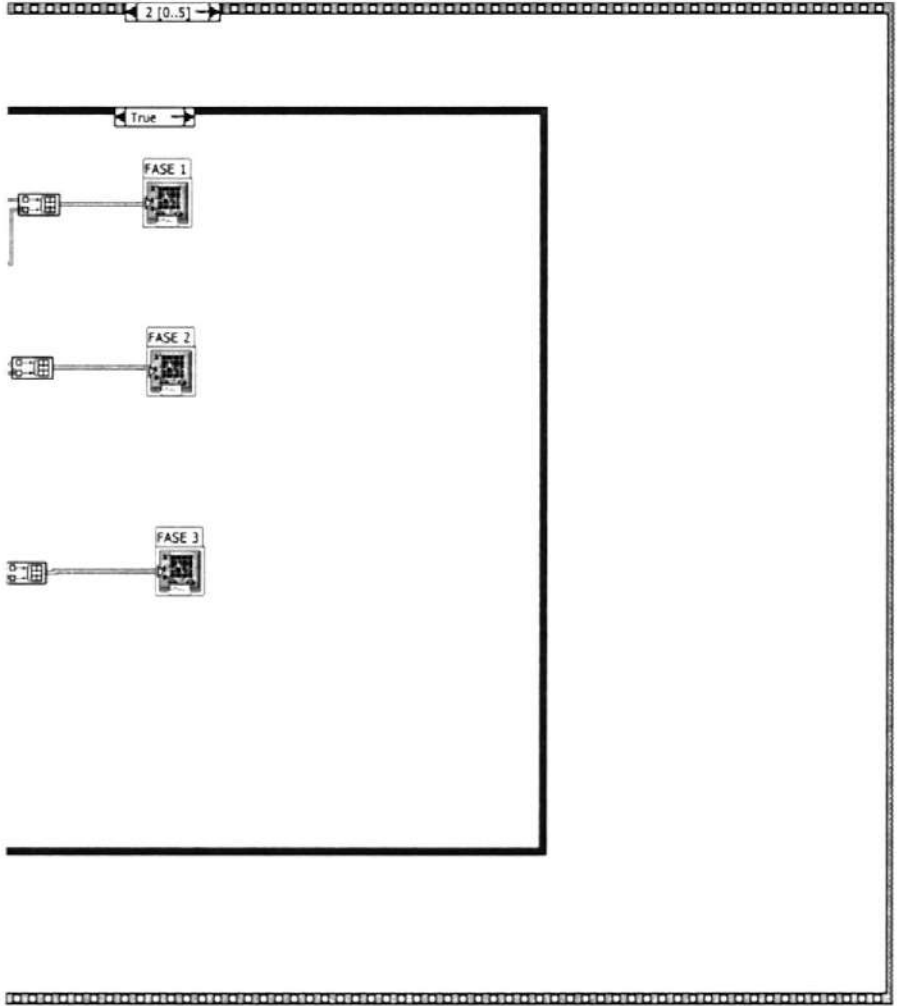
waveforms

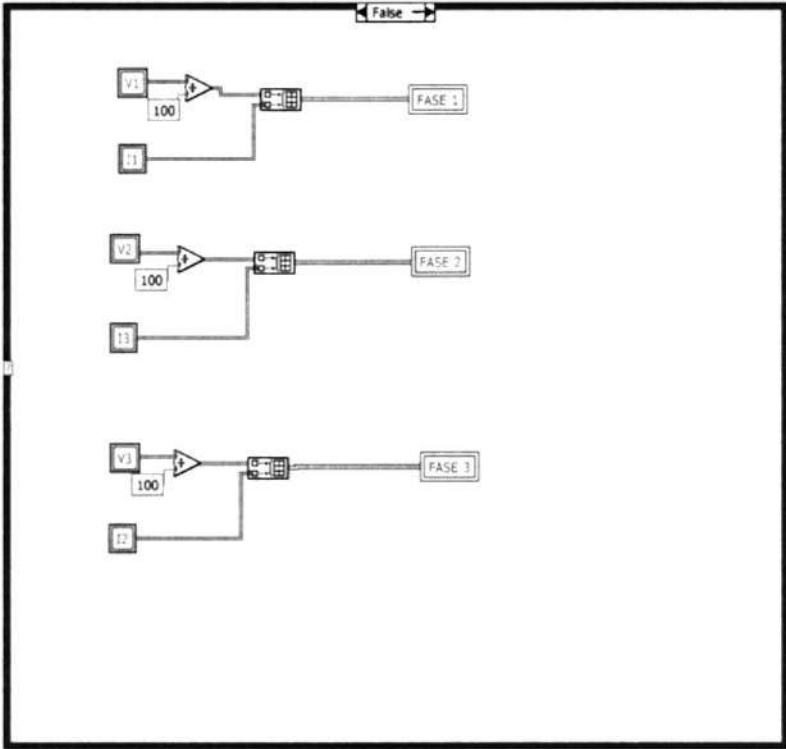


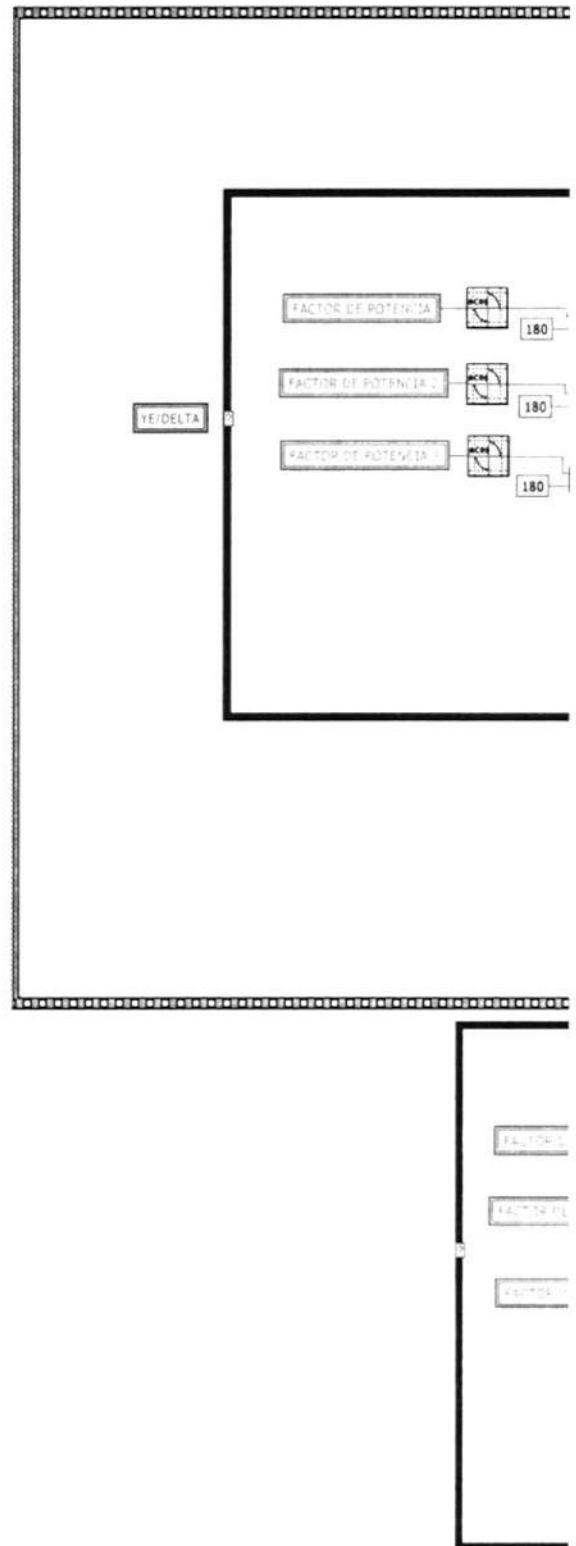


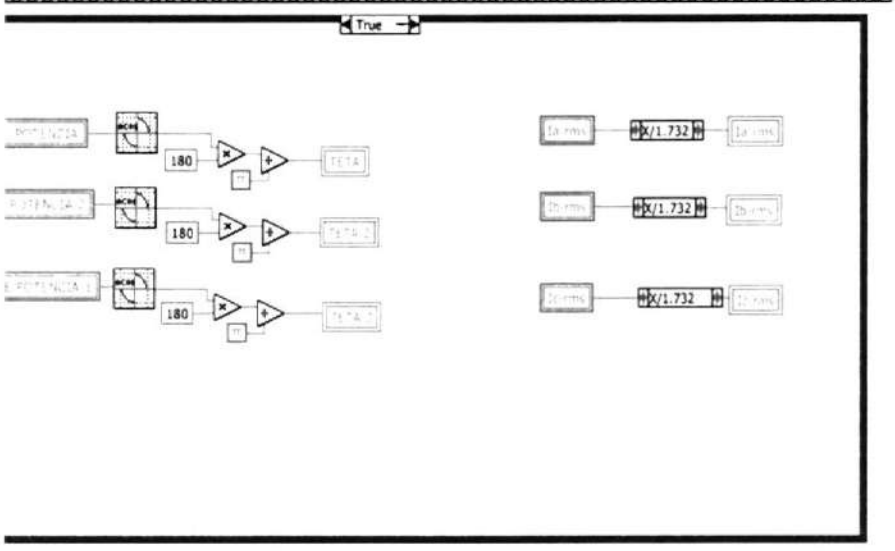
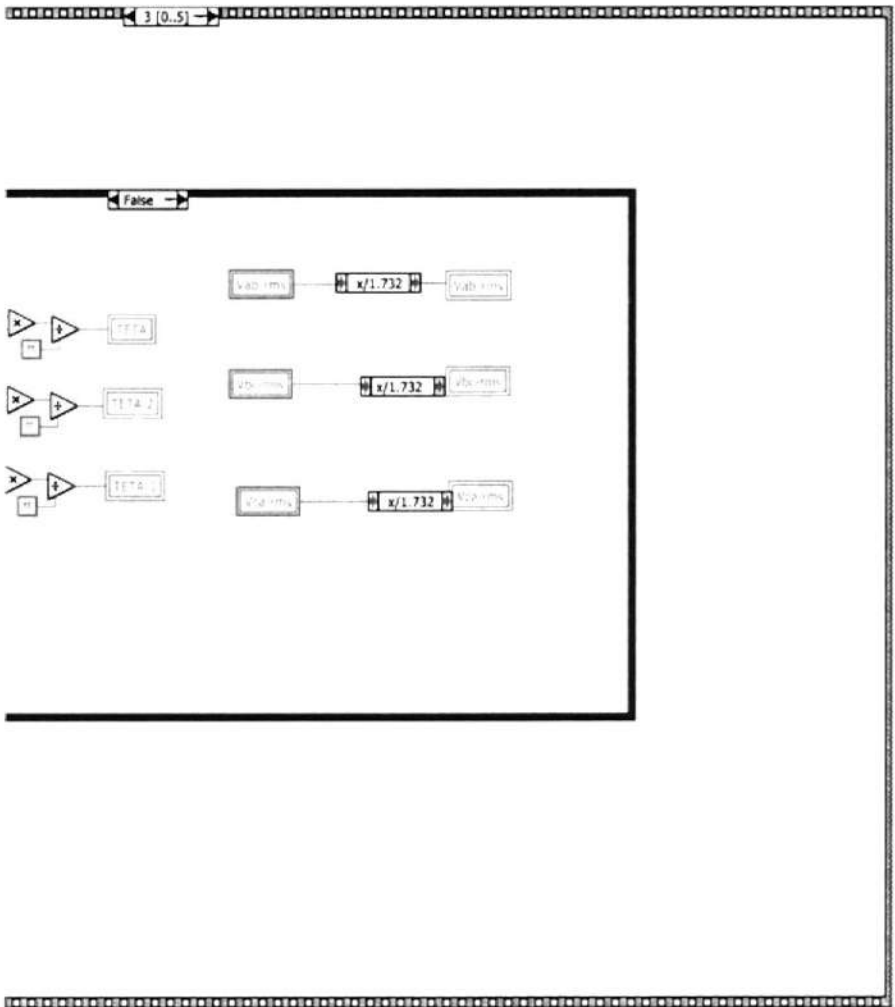


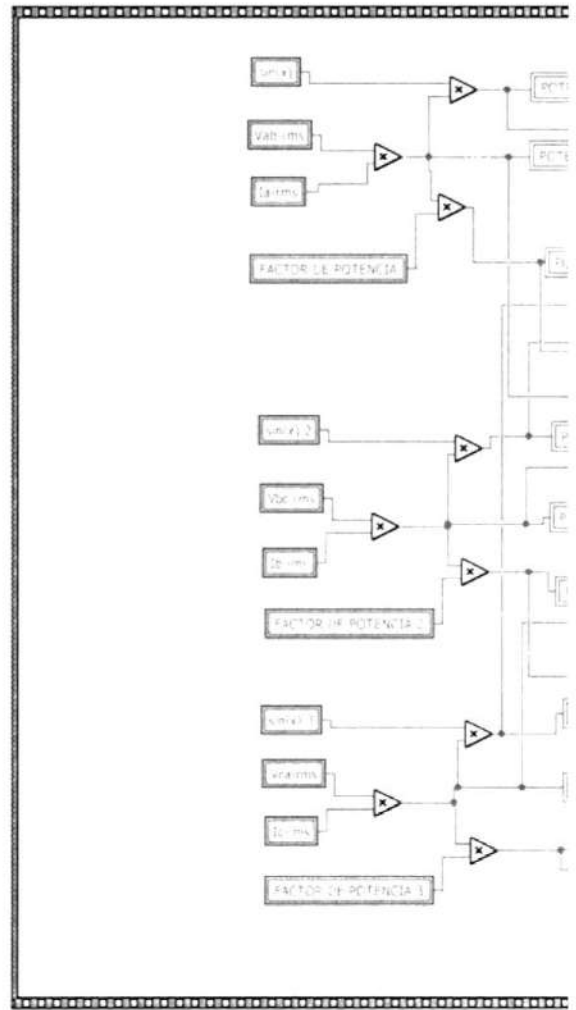


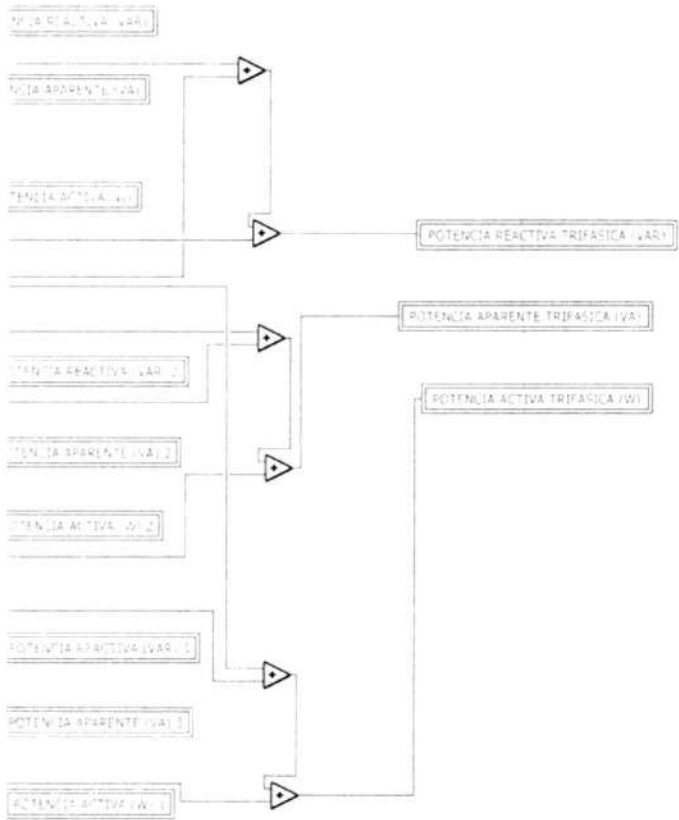




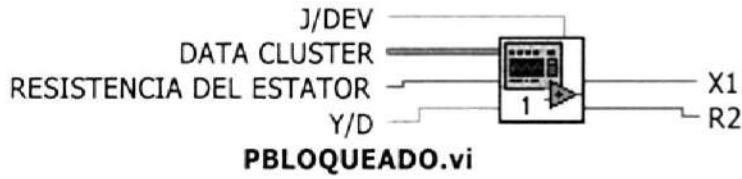








Connector Pane



Front Panel

PRUEBA DE ROTOR BLOQUEADO

CIRCUITO EQUIVALENTE:

=

CALCULOS

VBR (FASE)	IBR (FASE)	F. ACTIVA TRIFASICA DE ENTRADA
0 VOLTIOS	0 AMPERIOS	0 WATIOS
Ze	Re	Xe
0 OHMS	0 OHMS	0 OHMS
R2	X1	X2
0 OHMS	0 OHMS	0 OHMS

BR = BLOCK ROTOR - ROTOR BLOQUEADO

$$Z_e = \frac{V_{BR}}{I_{BR}}$$

$$R_e = \frac{P_{BR}}{I_{BR}^2}$$

$$X_e = \sqrt{Z_e^2 - R_e^2}$$

$$R_e = R_1 + R_2$$

$$X_1 = X_2 = \frac{X_e}{2}$$

START

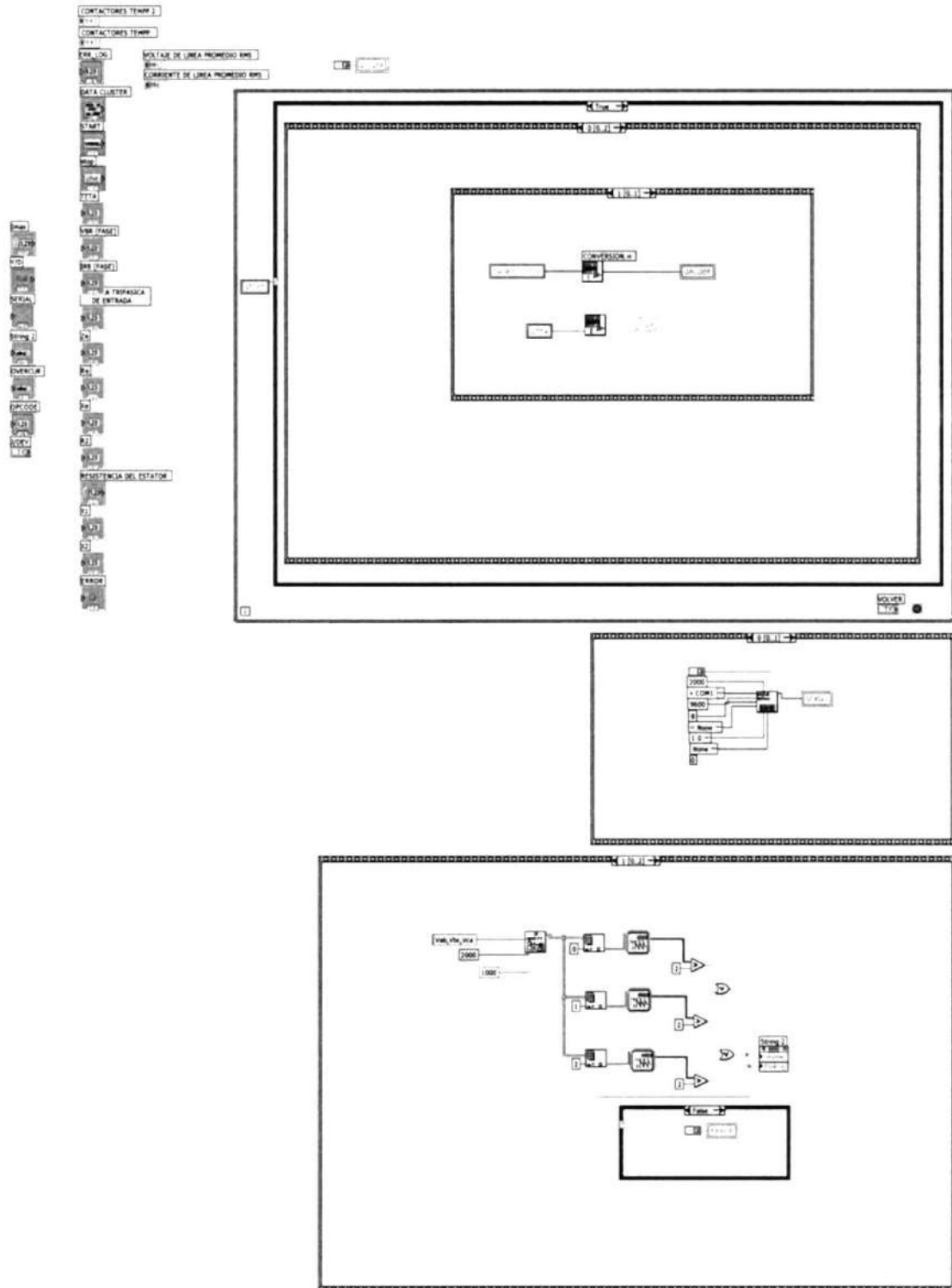
STOP

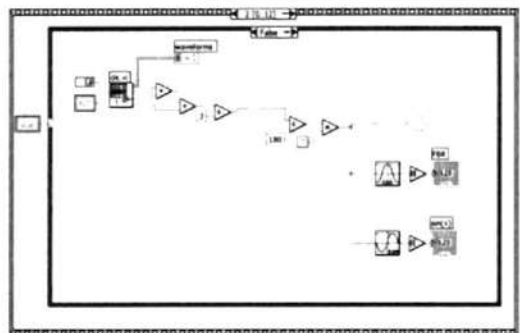
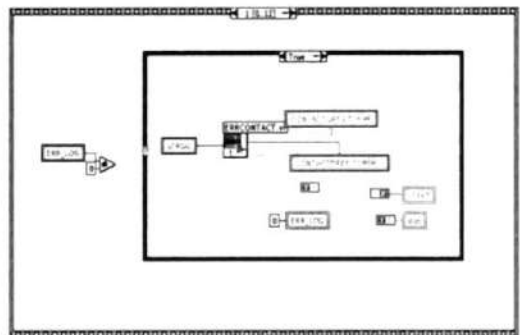
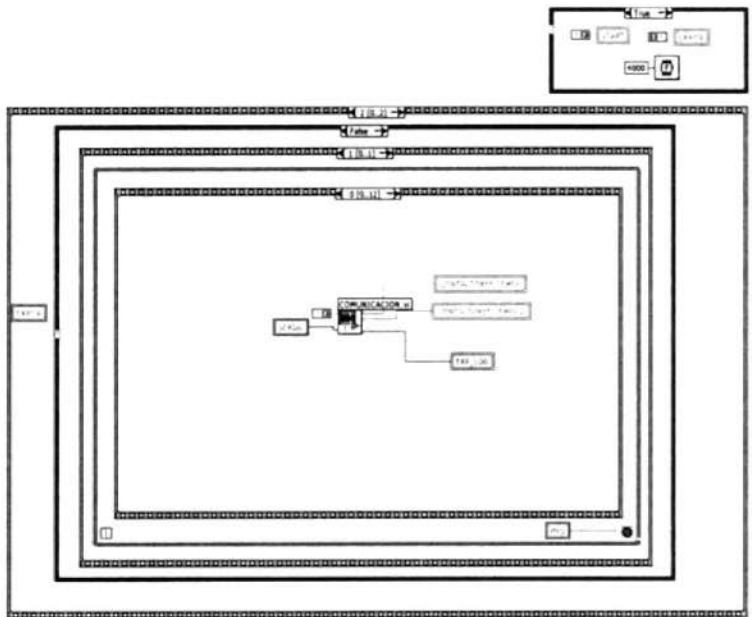
REGRESAR

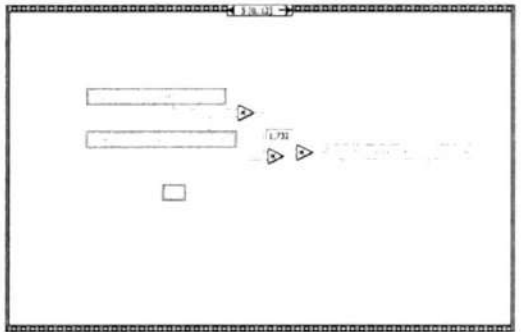
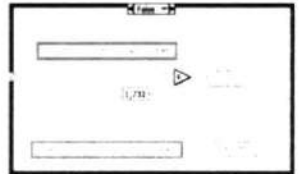
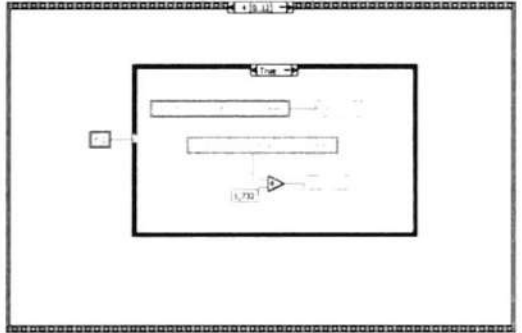
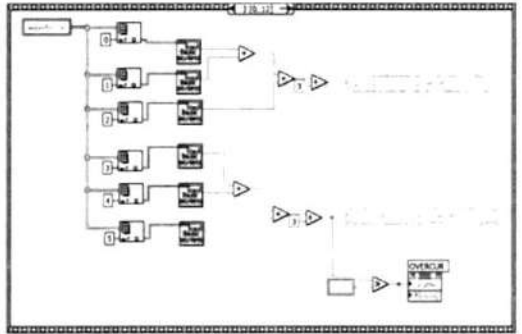


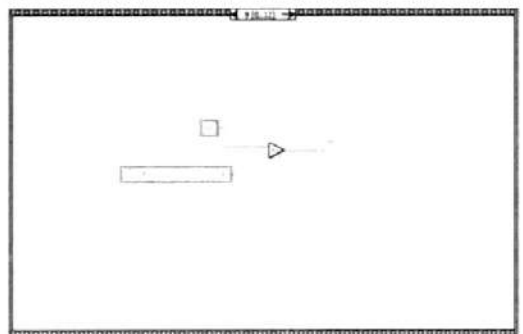
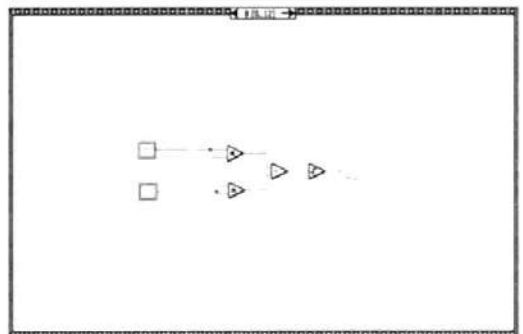
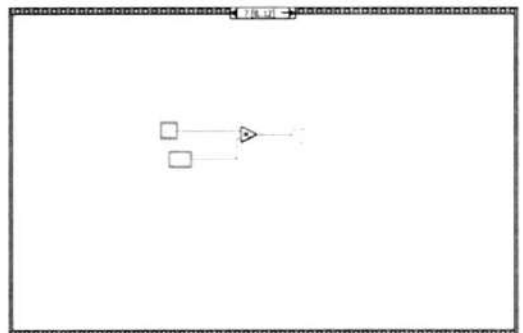
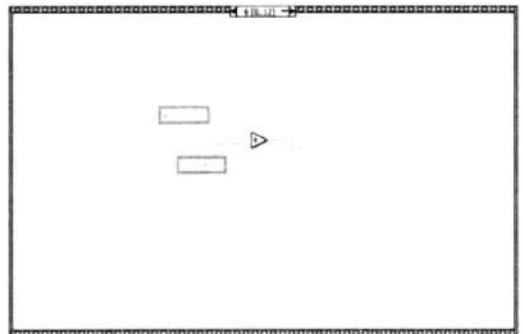
CIB - ESPOL

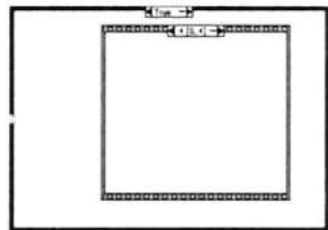
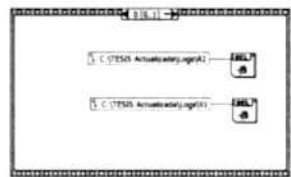
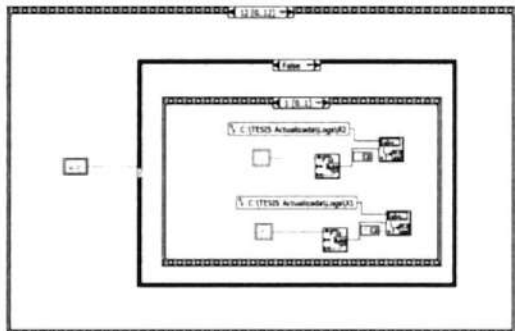
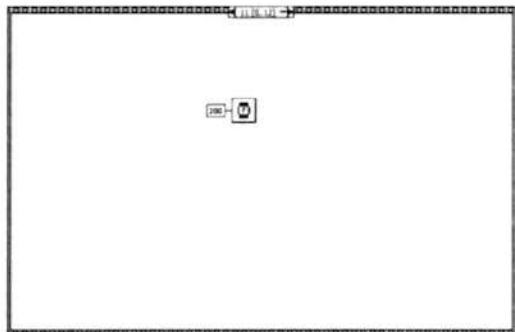
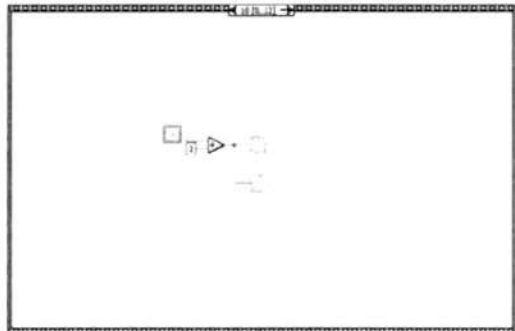
Block Diagram

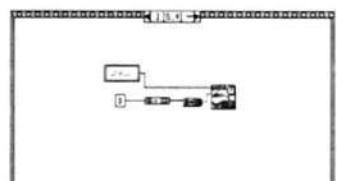
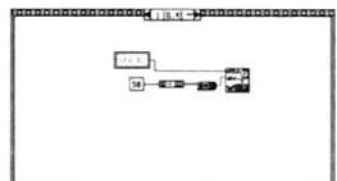
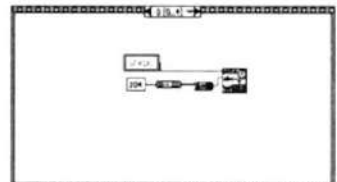
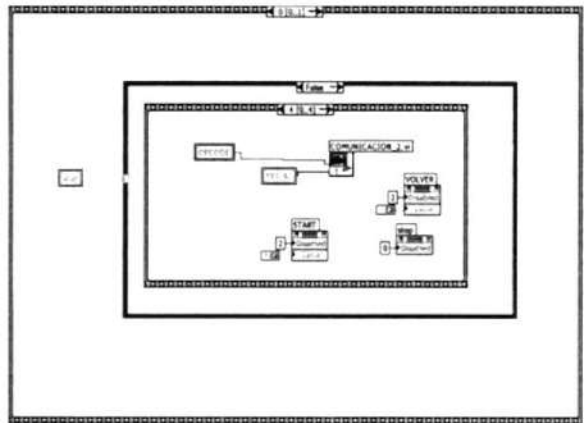
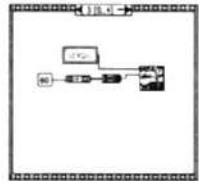
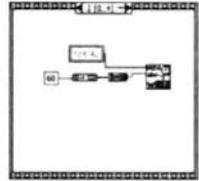
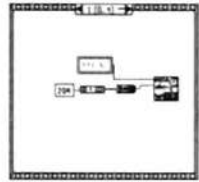


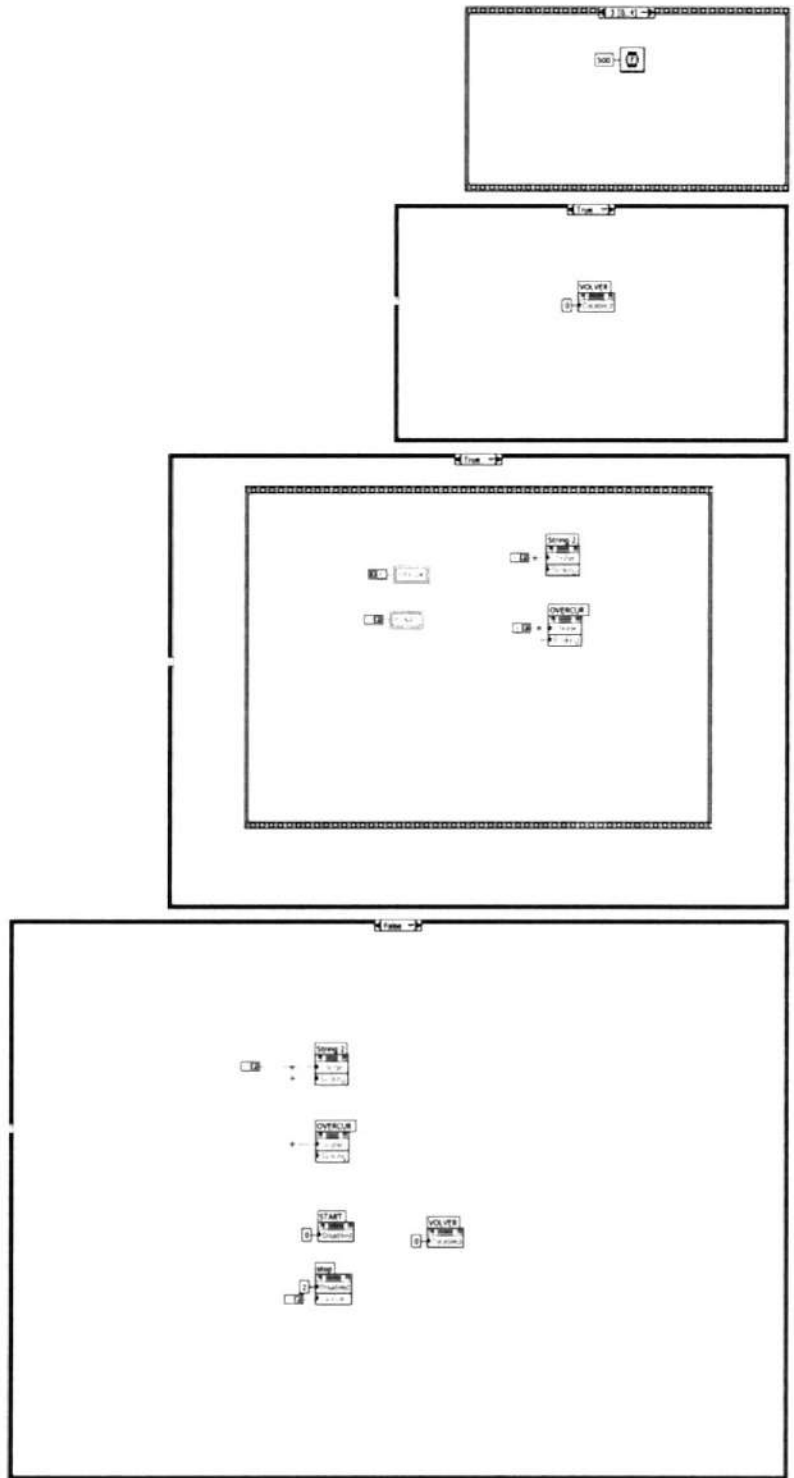












Express VI Configuration Information



Amplitude and Level Measurements2

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: RMS

Voltage Window: Off



Amplitude and Level Measurements3

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: RMS

Voltage Window: Off



Amplitude and Level Measurements4

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: RMS

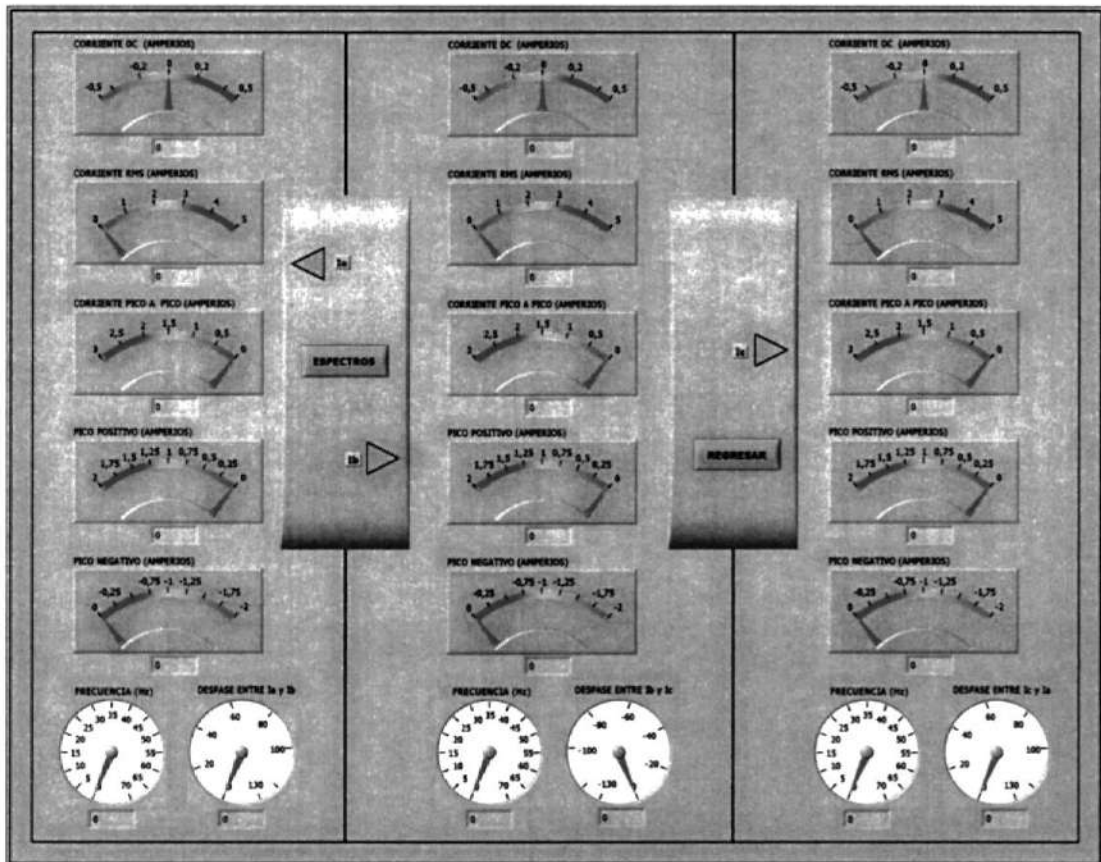
Voltage Window: Off

Connector Pane

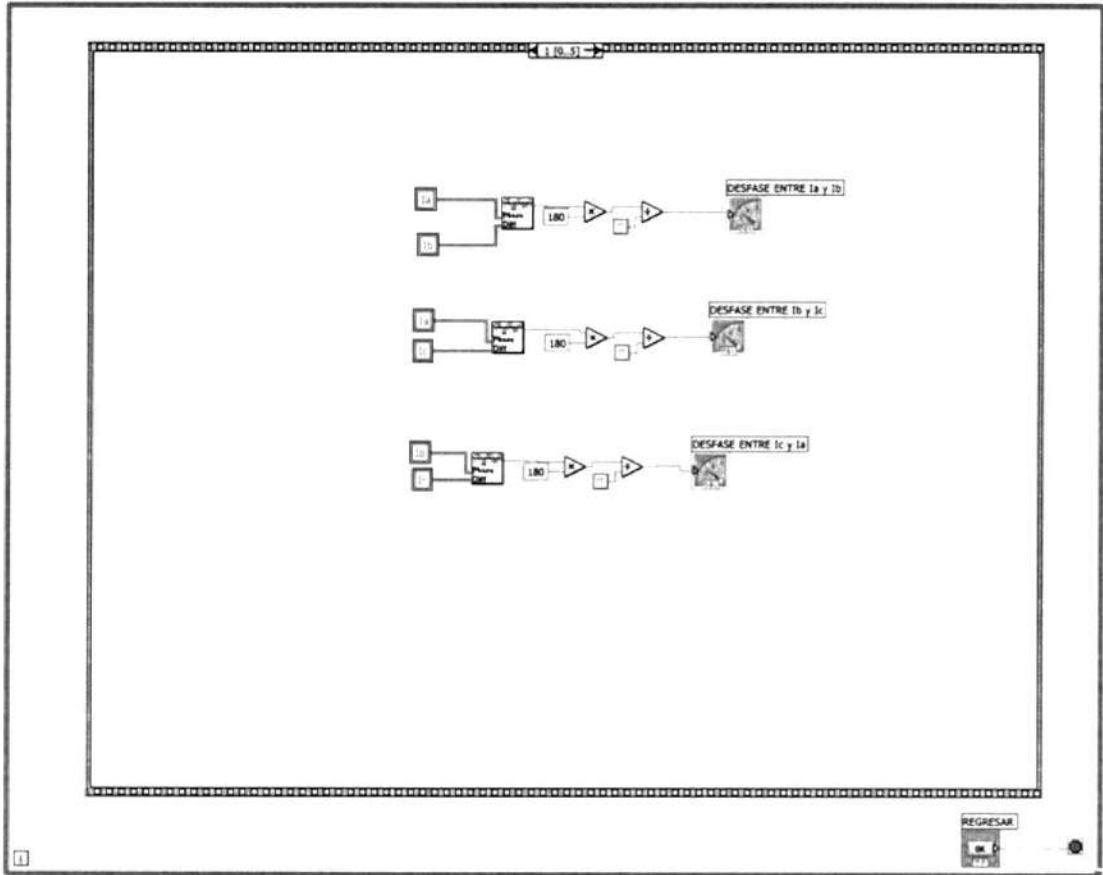


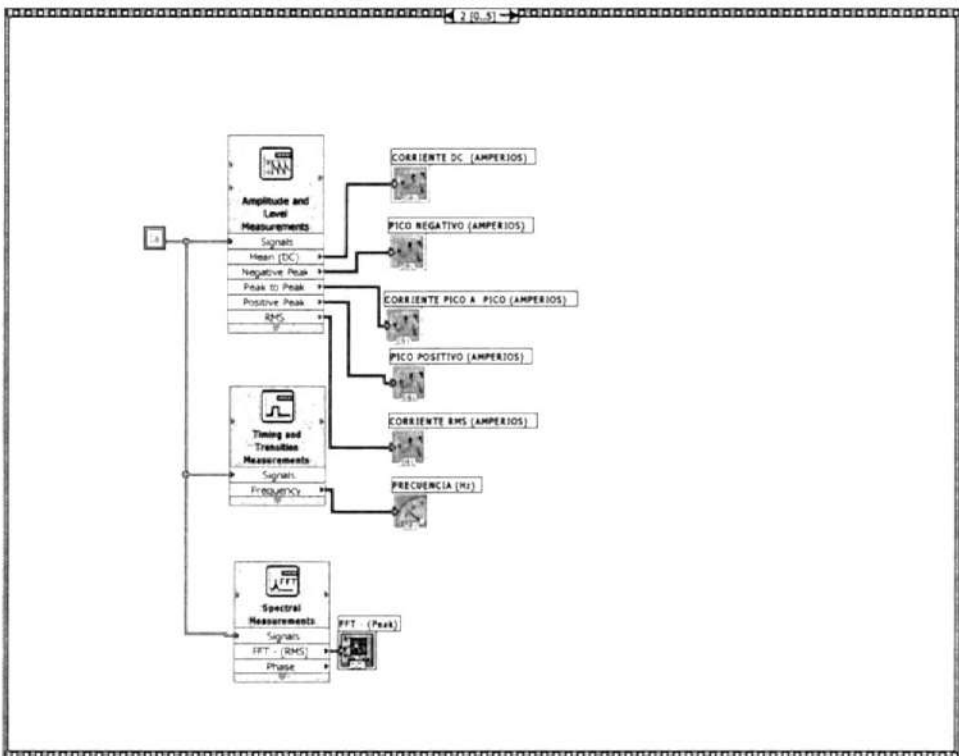
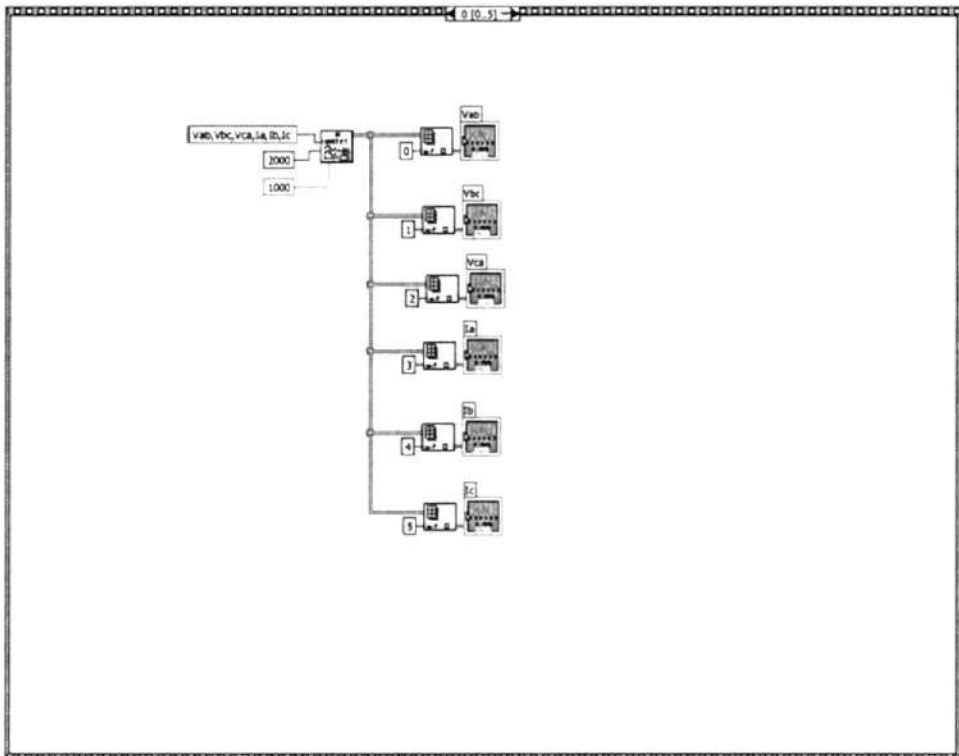
MAGNITUDESC.vi

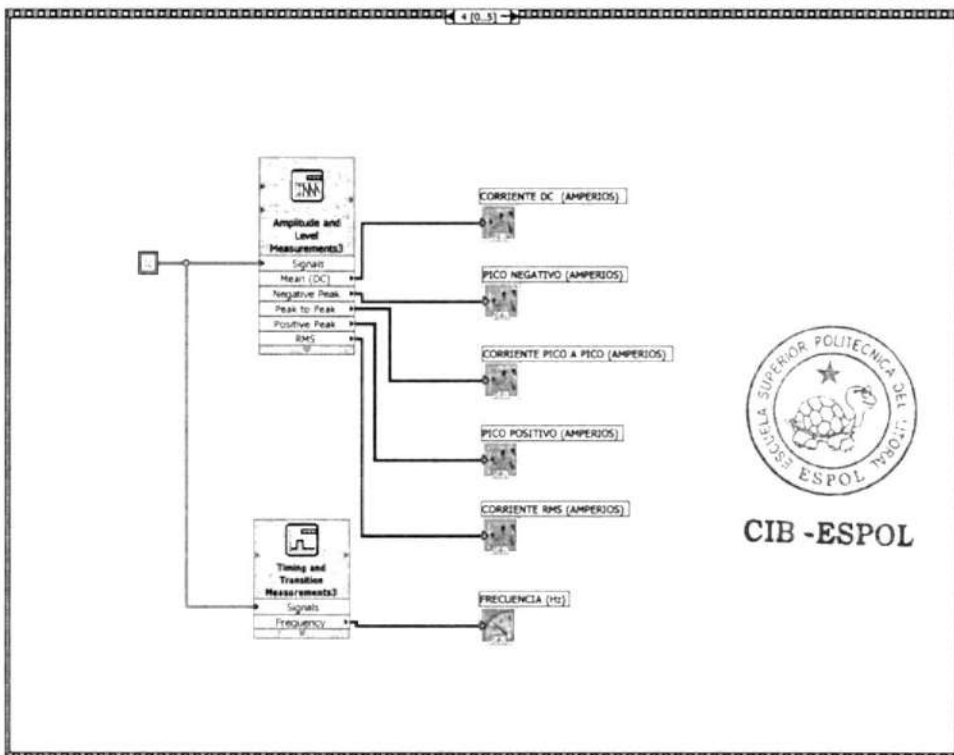
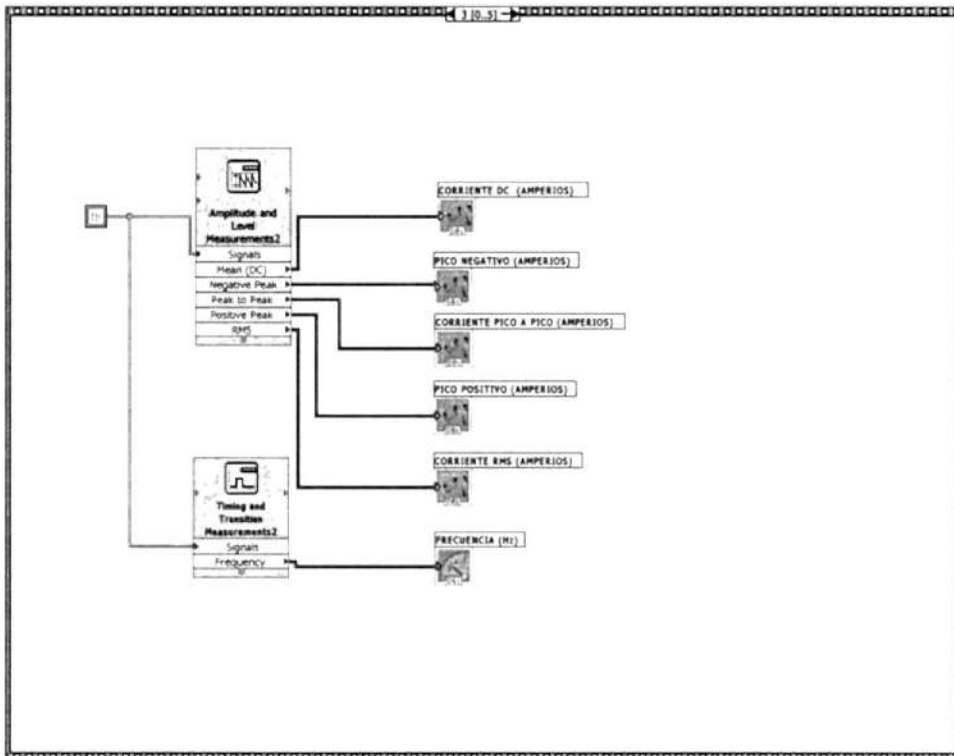
Front Panel

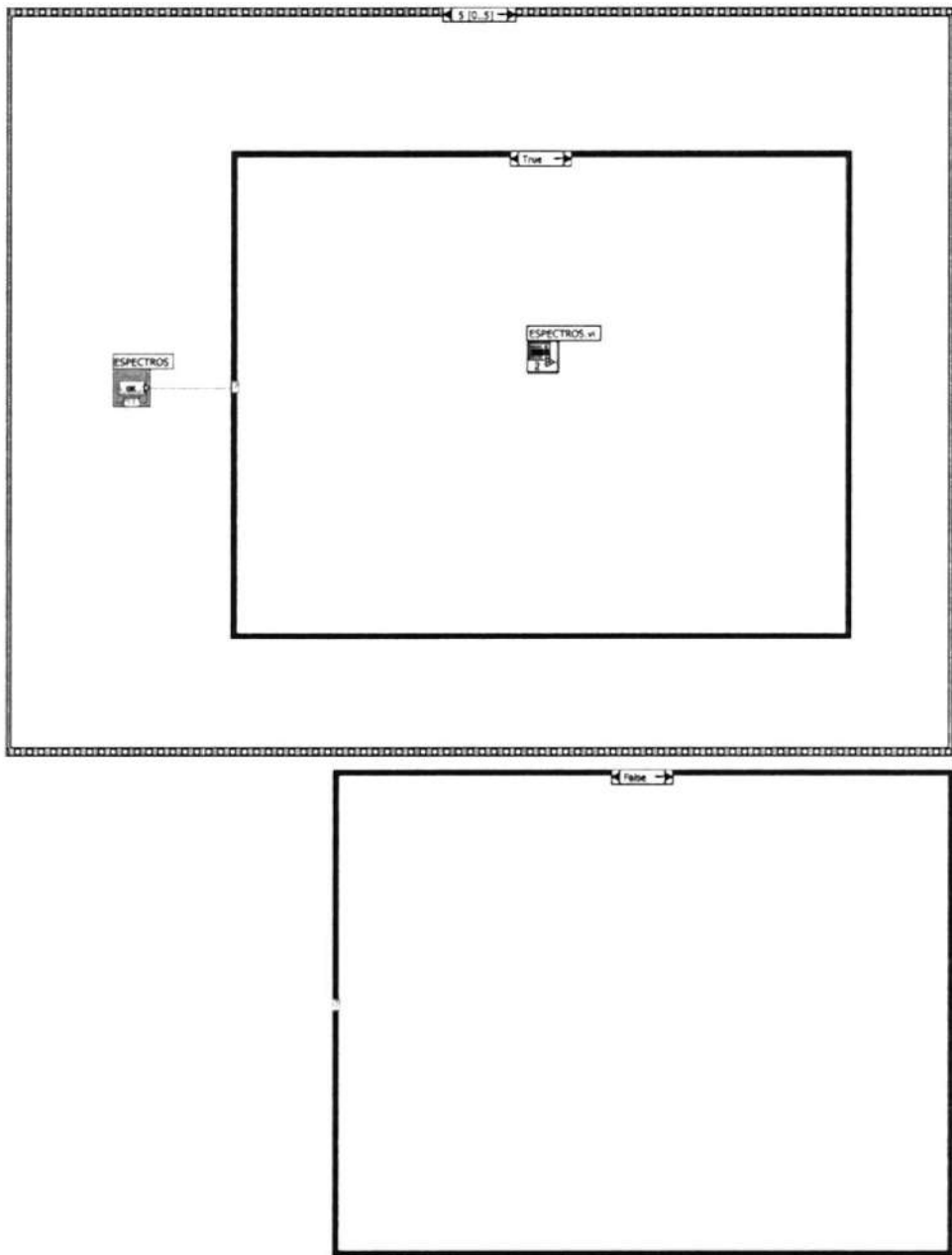


Block Diagram









Express VI Configuration Information



Amplitude and Level Measurements

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: DC

Measurement: RMS

Measurement: + Peak

Measurement: - Peak

Measurement: Peak to Peak

Voltage Window: Off



Timing and Transition Measurements

Timing and Transition Measurements

Performs timing and transition measurements on a signal, usually a pulse.

This Express VI is configured as follows:

Measurement: Frequency



Spectral Measurements

Spectral Measurements

Performs spectral measurements, such as peak spectrum and auto-power spectrum, on a signal.

This Express VI is configured as follows:

Selected Measurements: Magnitude (Peak)

View Phase: Wrapped and in Radians

Windowing: Hanning

Averaging: None



Amplitude and Level Measurements2

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: DC

Measurement: RMS

Measurement: + Peak

Measurement: - Peak

Measurement: Peak to Peak

Voltage Window: Off



Timing and Transition Measurements2

Timing and Transition Measurements

Performs timing and transition measurements on a signal, usually a pulse.

This Express VI is configured as follows:

Measurement: Frequency



Amplitude and Level Measurements3

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: DC

Measurement: RMS

Measurement: + Peak

Measurement: - Peak

Measurement: Peak to Peak

Voltage Window: Off



Timing and Transition Measurements3

Timing and Transition Measurements

Performs timing and transition measurements on a signal, usually a pulse.

This Express VI is configured as follows:

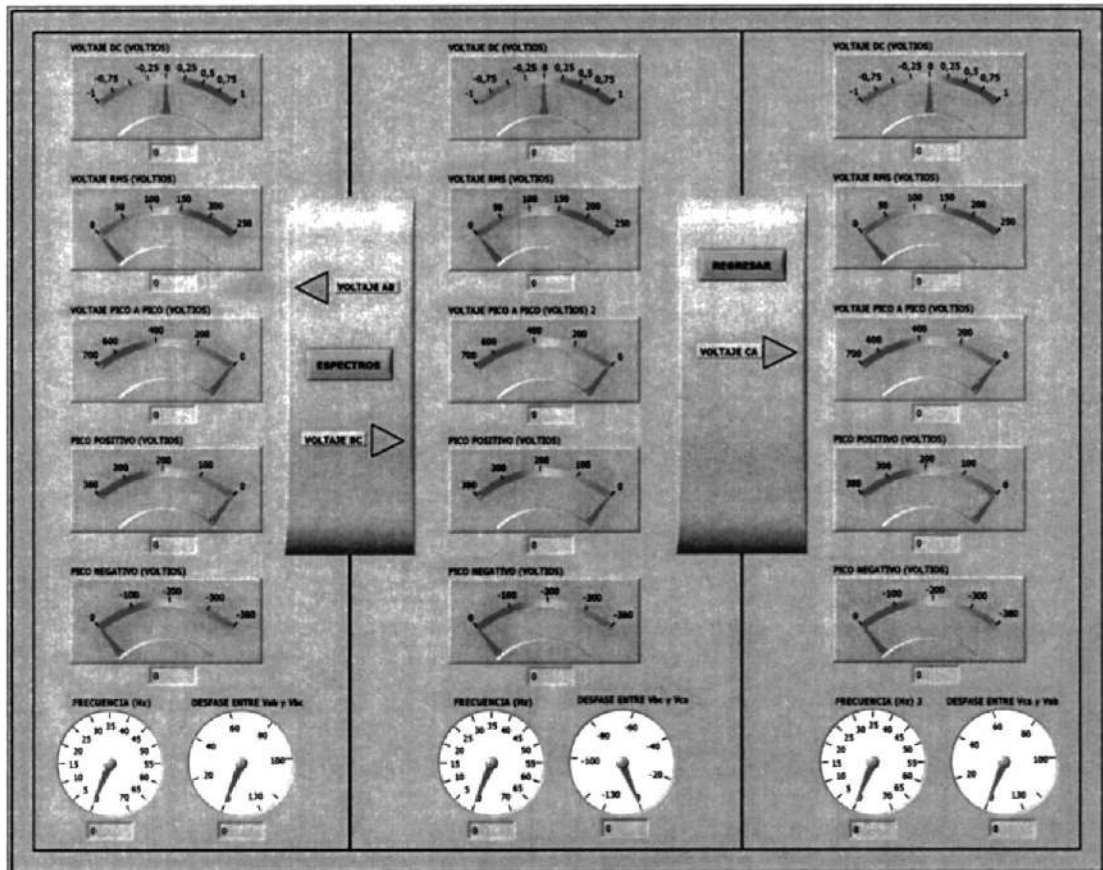
Measurement: Frequency

Connector Pane

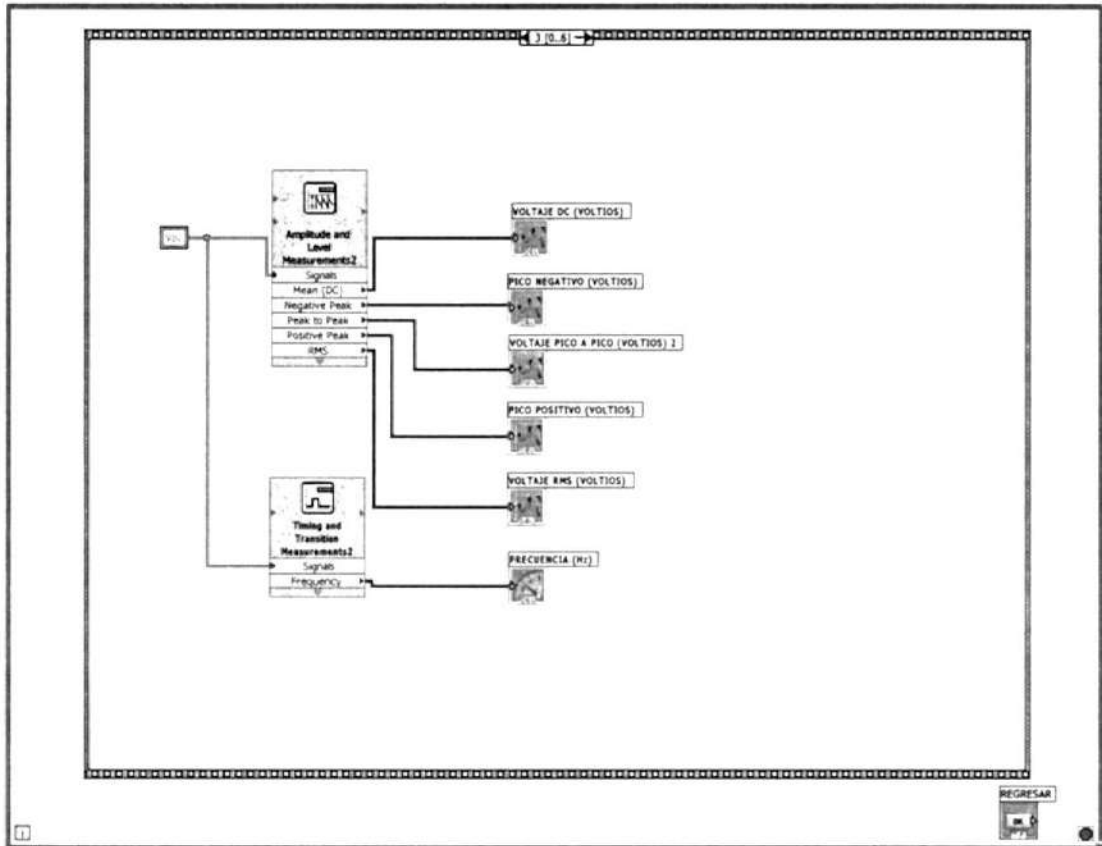


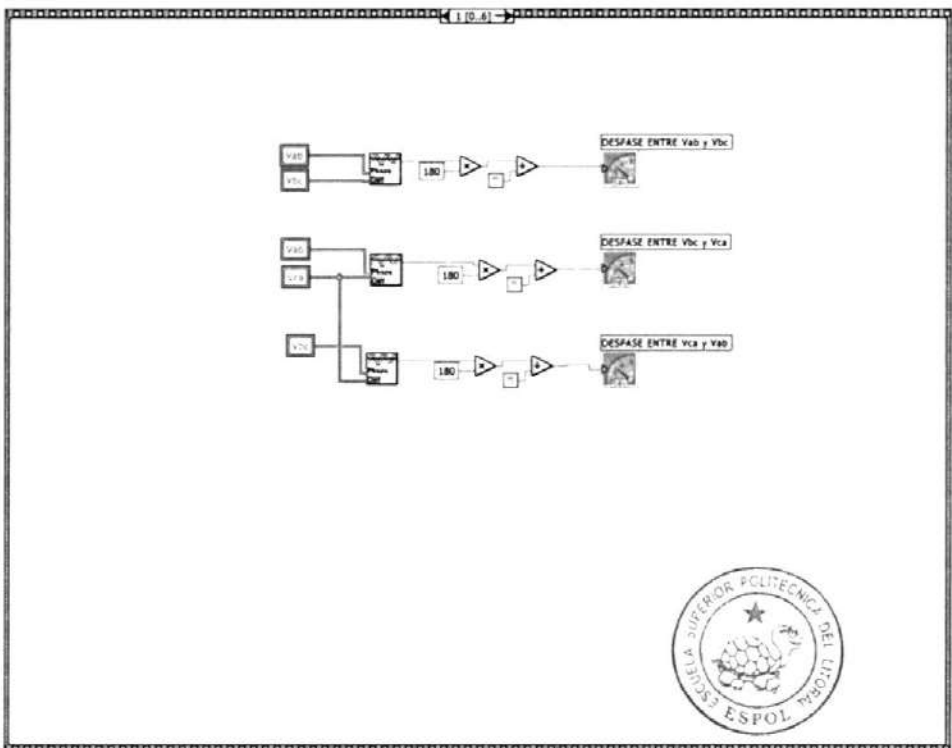
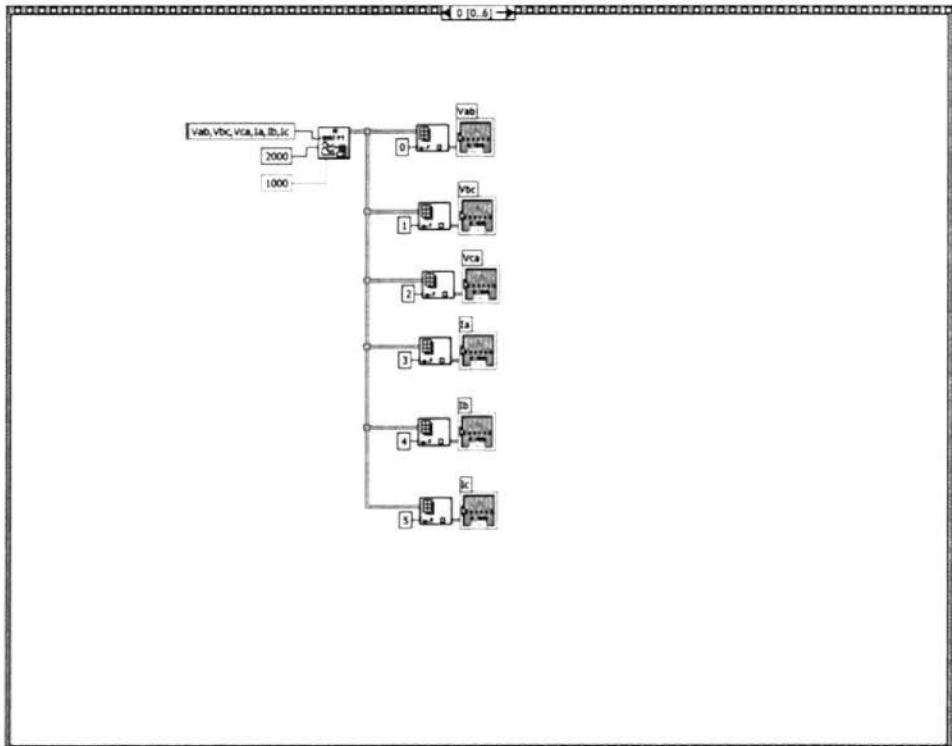
MAGNITUDES.vi

Front Panel



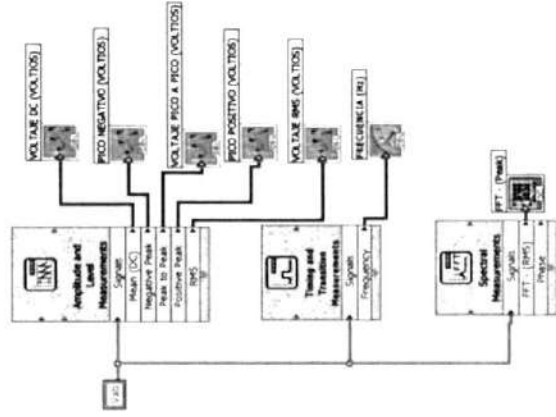
Block Diagram



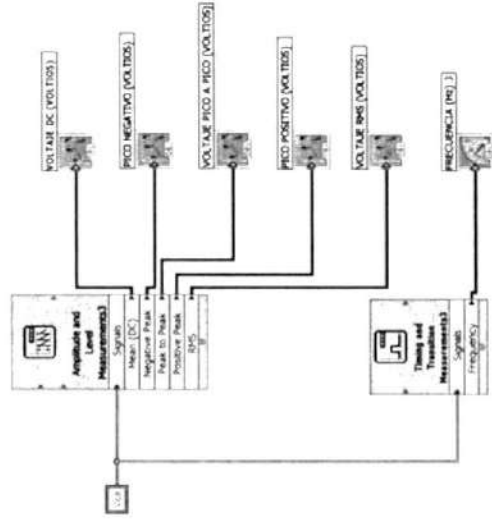


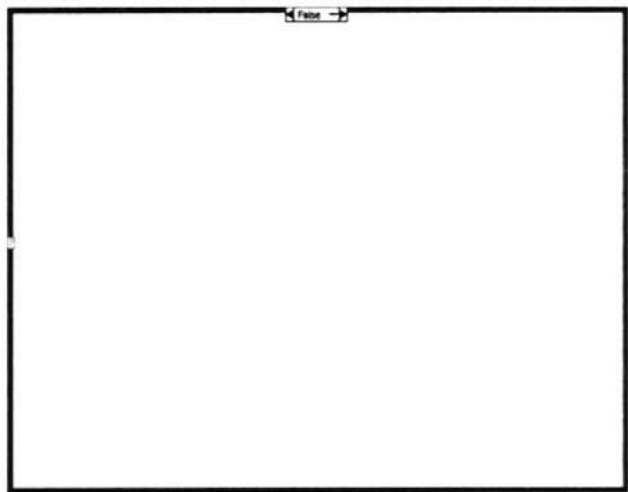
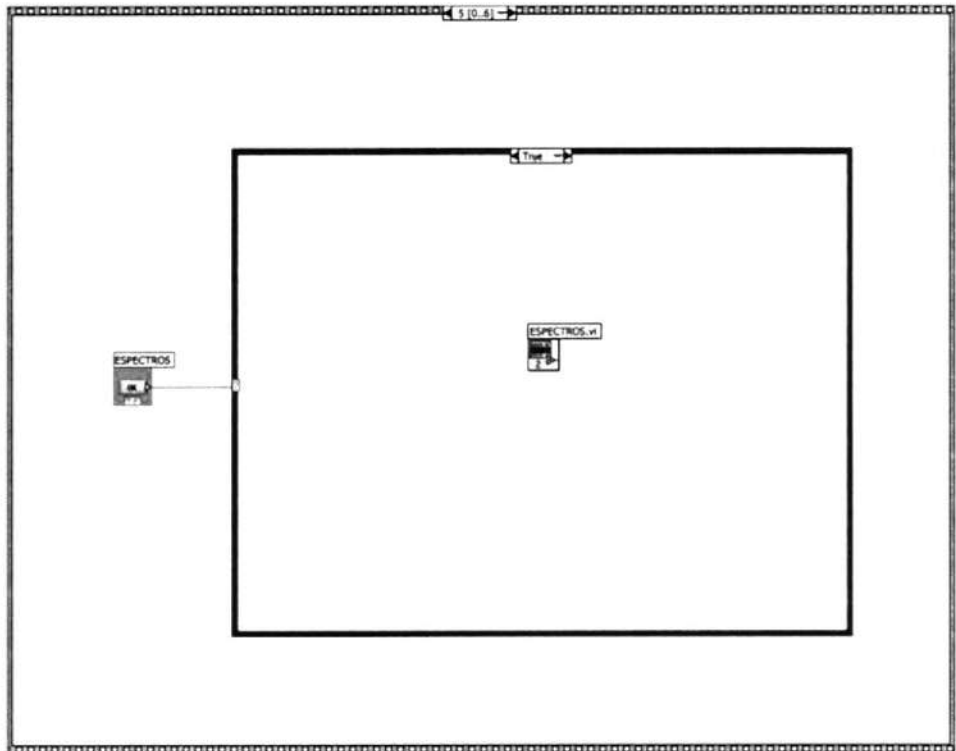
CIB - ESPOL

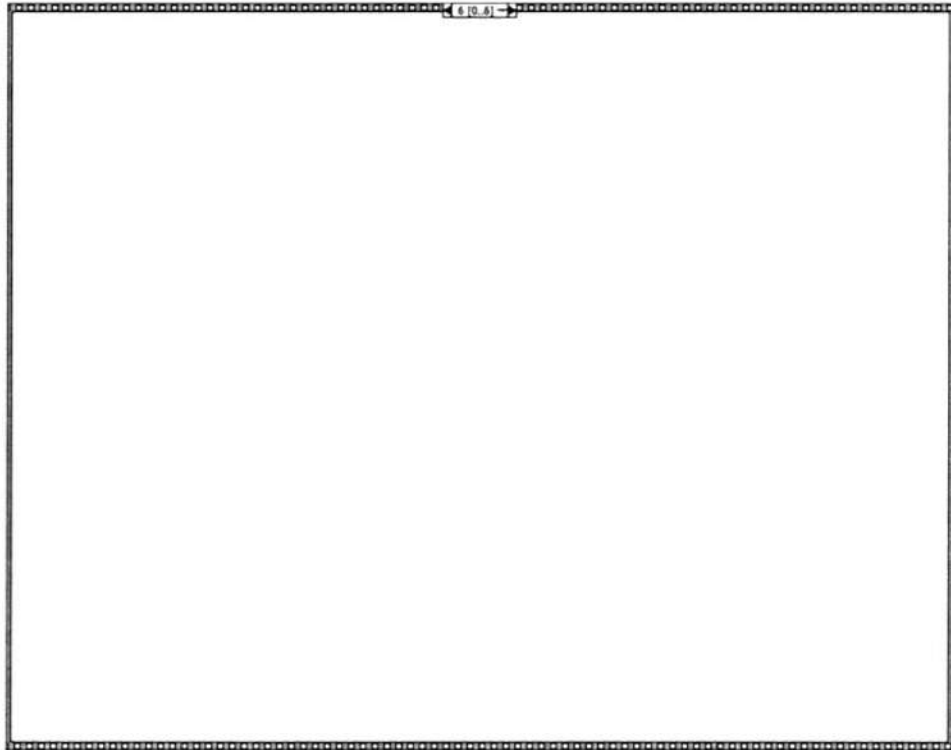
4 | 0.5 |



4 | 0.5 |







Express VI Configuration Information



Amplitude and Level Measurements

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: DC

Measurement: RMS

Measurement: + Peak

Measurement: - Peak

Measurement: Peak to Peak

Voltage Window: Off



Timing and Transition Measurements

Timing and Transition Measurements

Performs timing and transition measurements on a signal, usually a pulse.

This Express VI is configured as follows:

Measurement: Frequency



Spectral Measurements

Spectral Measurements

Performs spectral measurements, such as peak spectrum and auto-power spectrum, on a signal.

This Express VI is configured as follows:

Selected Measurements: Magnitude (Peak)

View Phase: Wrapped and in Radians

Windowing: Hanning

Averaging: None



Amplitude and Level Measurements2

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: DC

Measurement: RMS

Measurement: + Peak

Measurement: - Peak

Measurement: Peak to Peak

Voltage Window: Off



Timing and Transition Measurements2

Timing and Transition Measurements

Performs timing and transition measurements on a signal, usually a pulse.

This Express VI is configured as follows:

Measurement: Frequency



Timing and Transition Measurements3

Timing and Transition Measurements

Performs timing and transition measurements on a signal, usually a pulse.

This Express VI is configured as follows:

Measurement: Frequency



Amplitude and Level Measurements3

Amplitude and Level Measurements

Performs voltage measurements on a signal.

This Express VI is configured as follows:

Measurement: DC

Measurement: RMS

Measurement: + Peak

Measurement: - Peak

Measurement: Peak to Peak

Voltage Window: Off

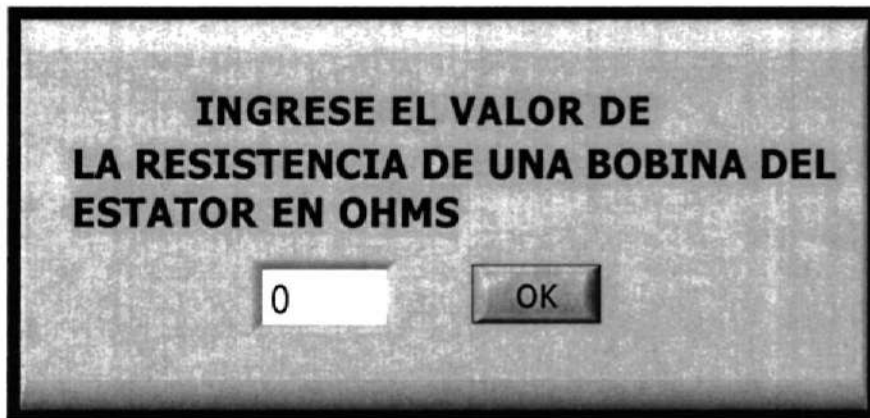
Connector Pane



RESISTENCIA 2

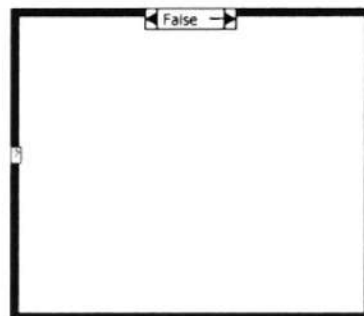
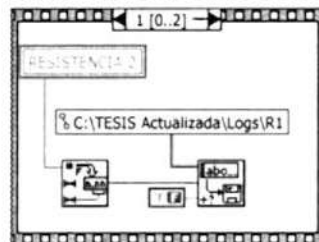
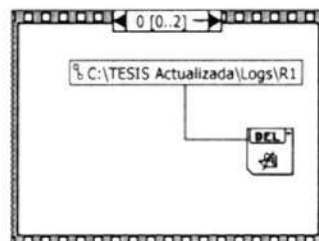
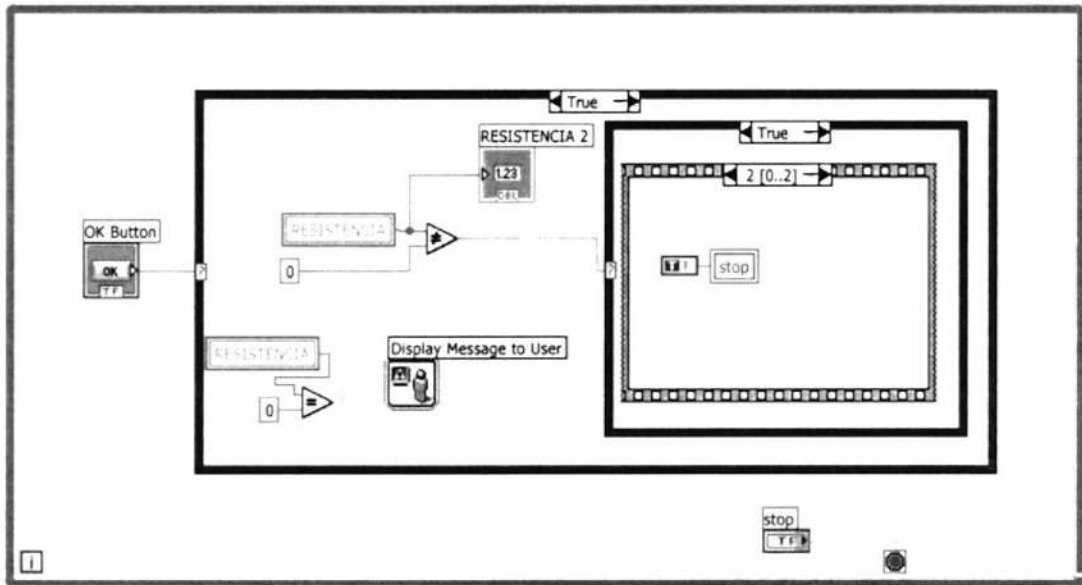
INRES.vi

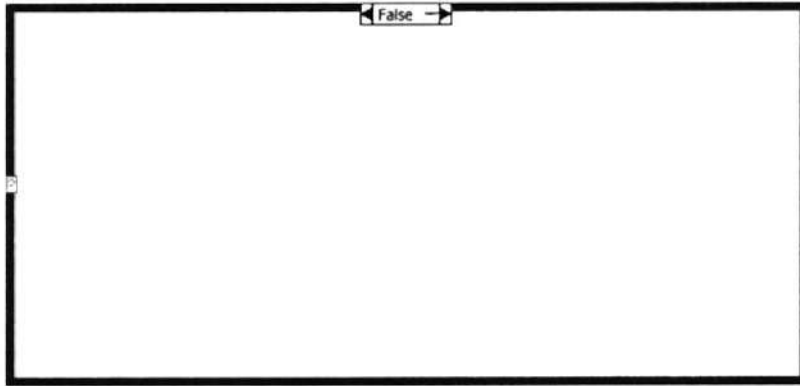
Front Panel



Block Diagram

RESISTENCIA





Express VI Configuration Information



Display Message to User

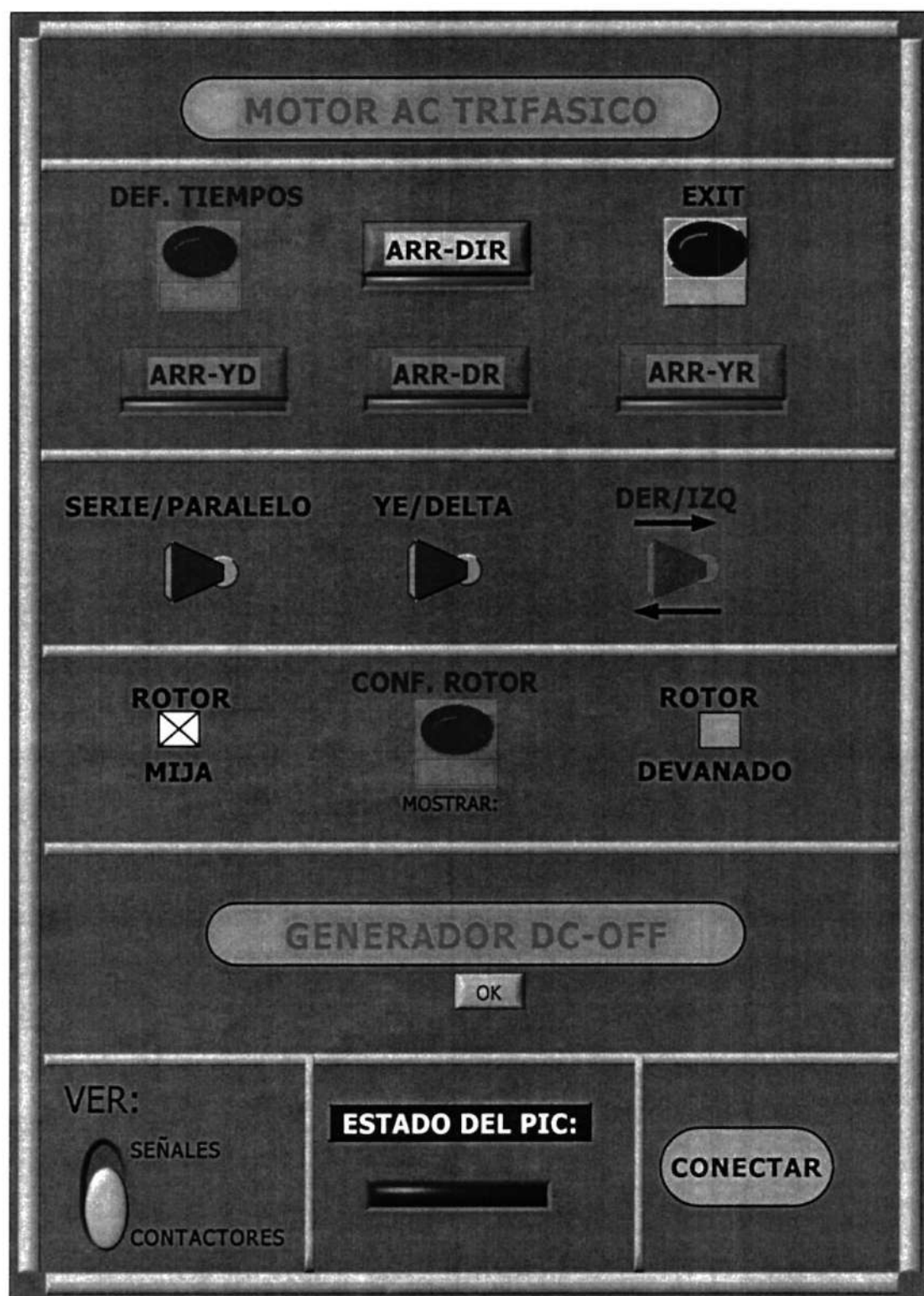
Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

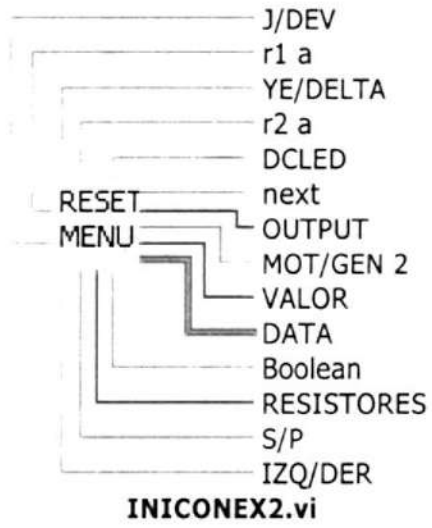
This Express VI is configured as follows:

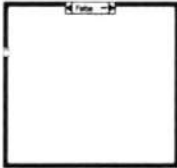
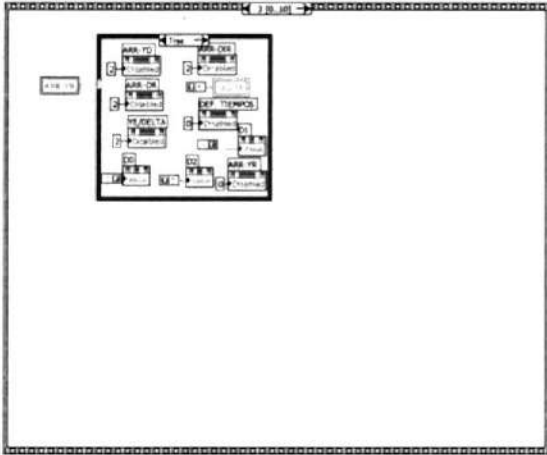
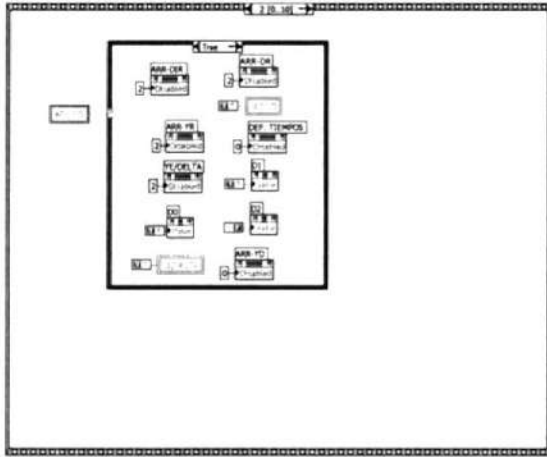
Message:

Front Panel

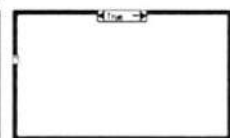
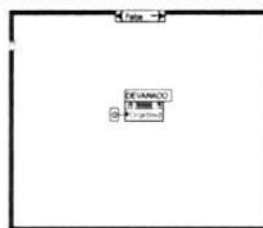
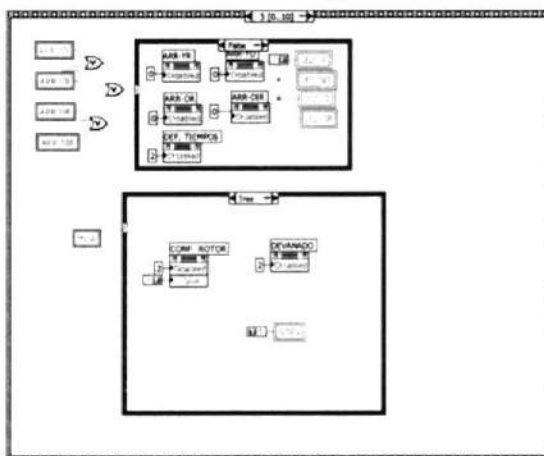
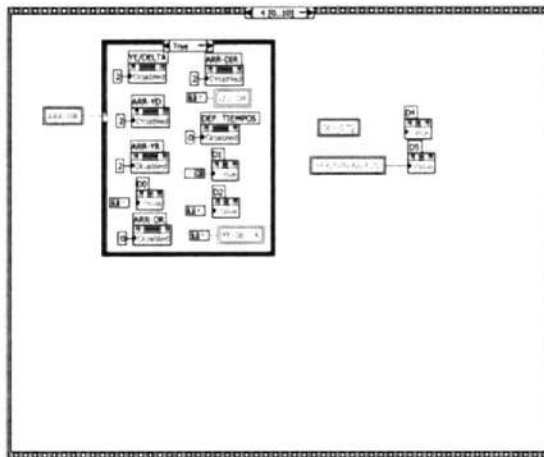


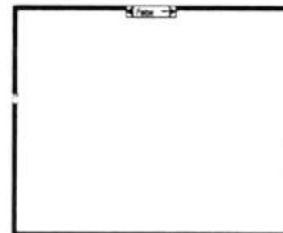
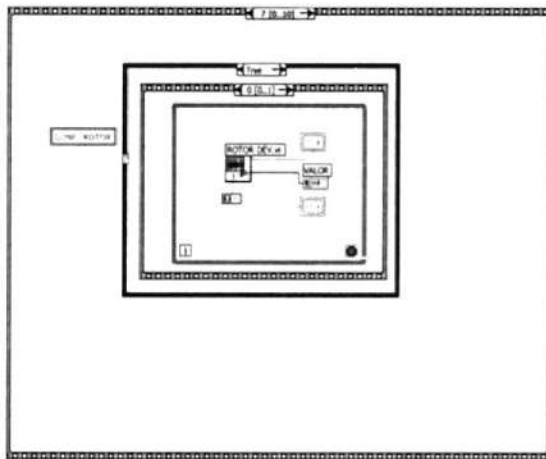
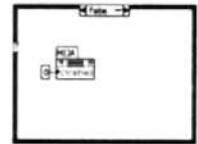
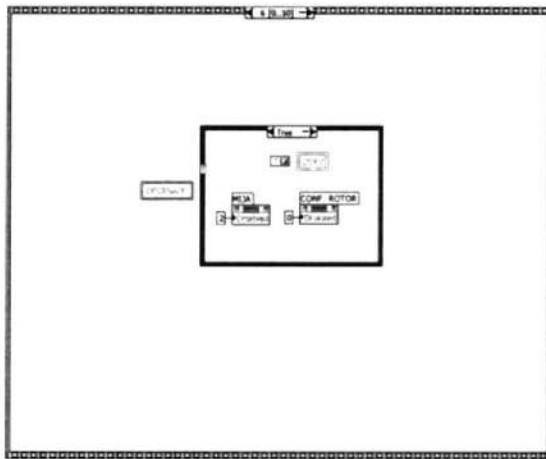
Connector Pane

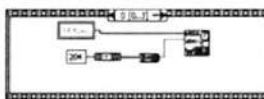
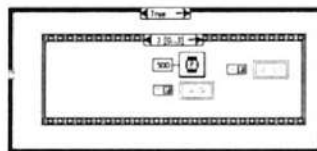
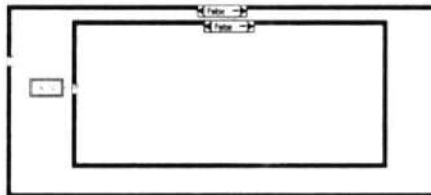
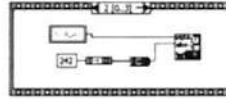
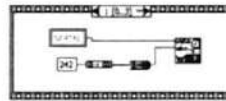
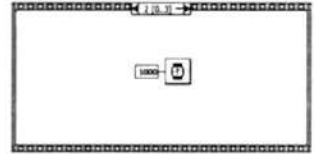
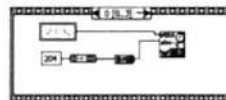
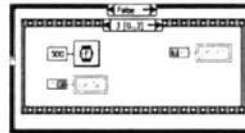
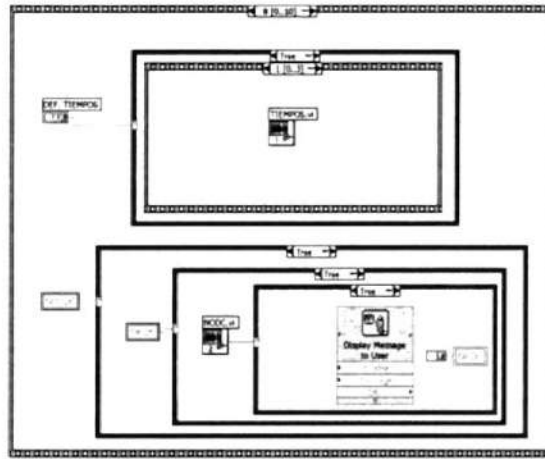


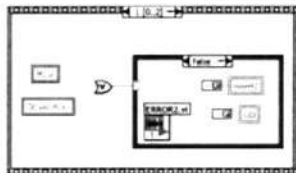
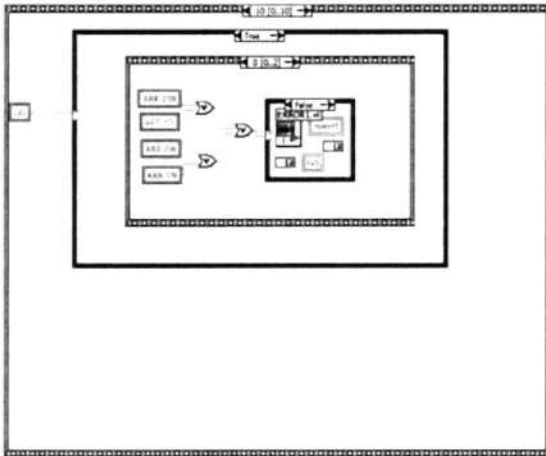
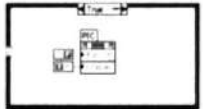
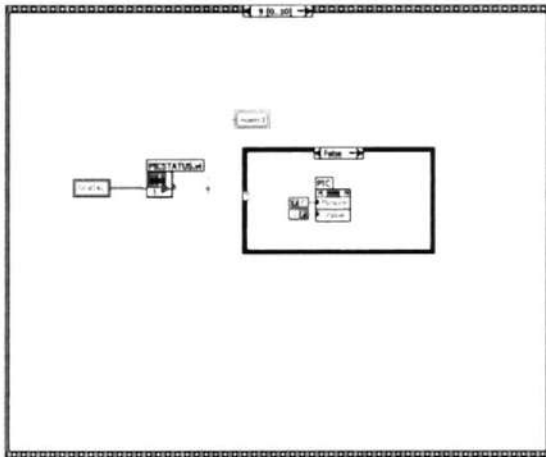
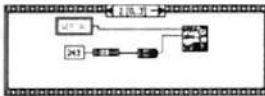
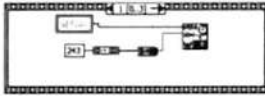


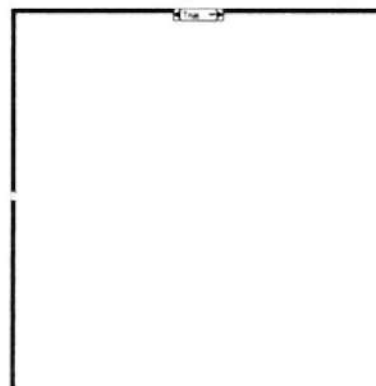
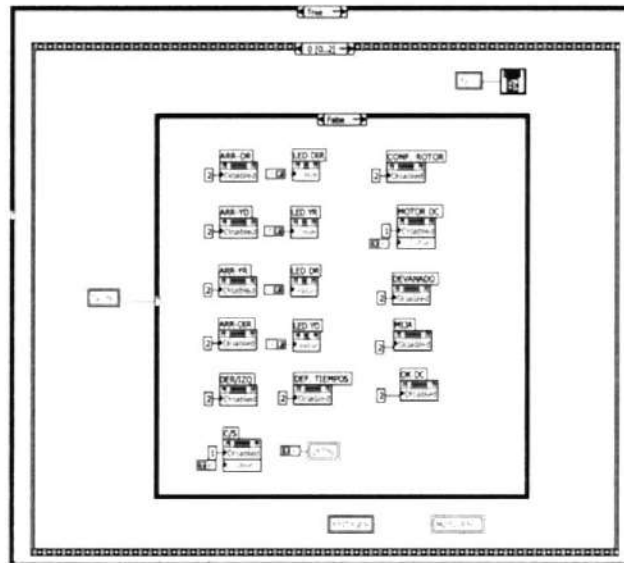
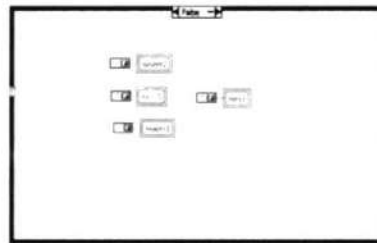
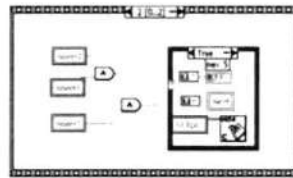
CIB - ESPOL

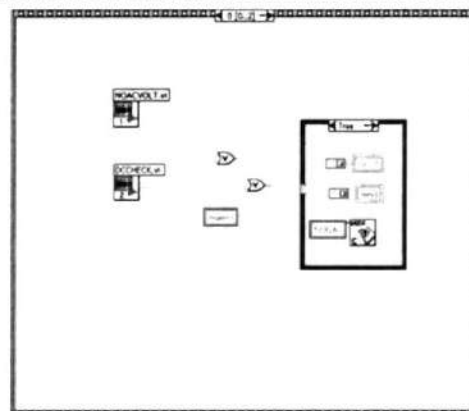
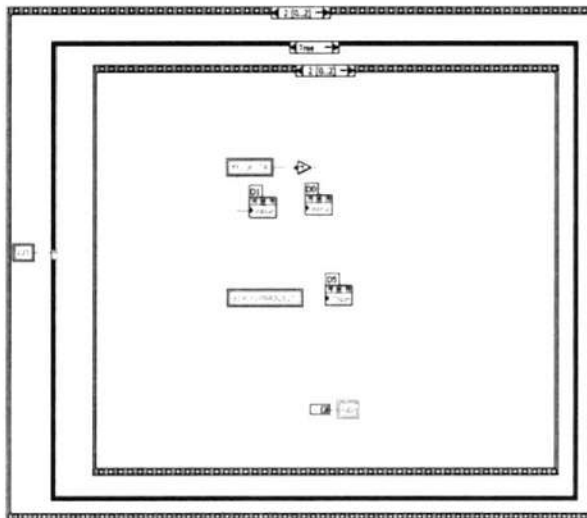
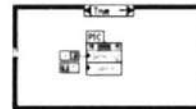
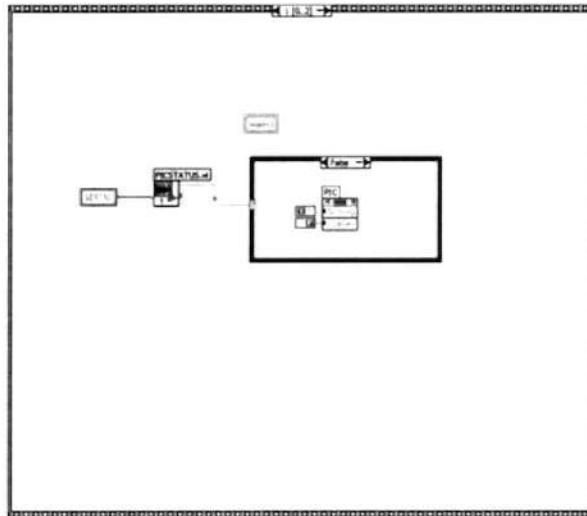


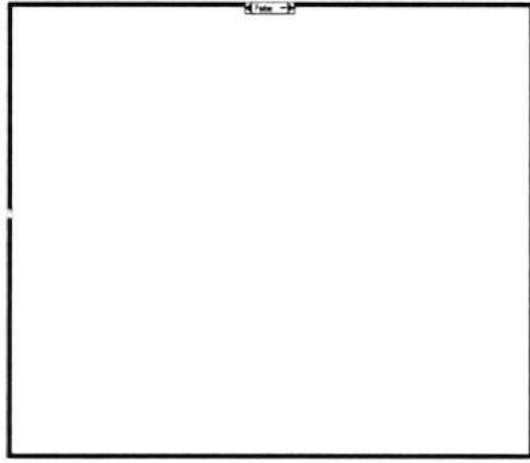
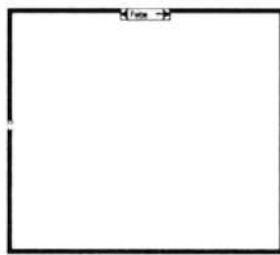
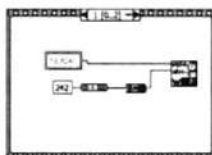
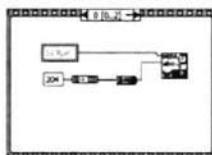
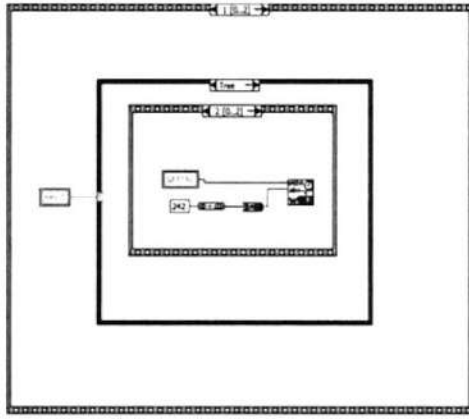
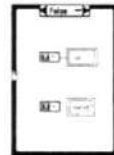














Display Message to User

Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

This Express VI is configured as follows:

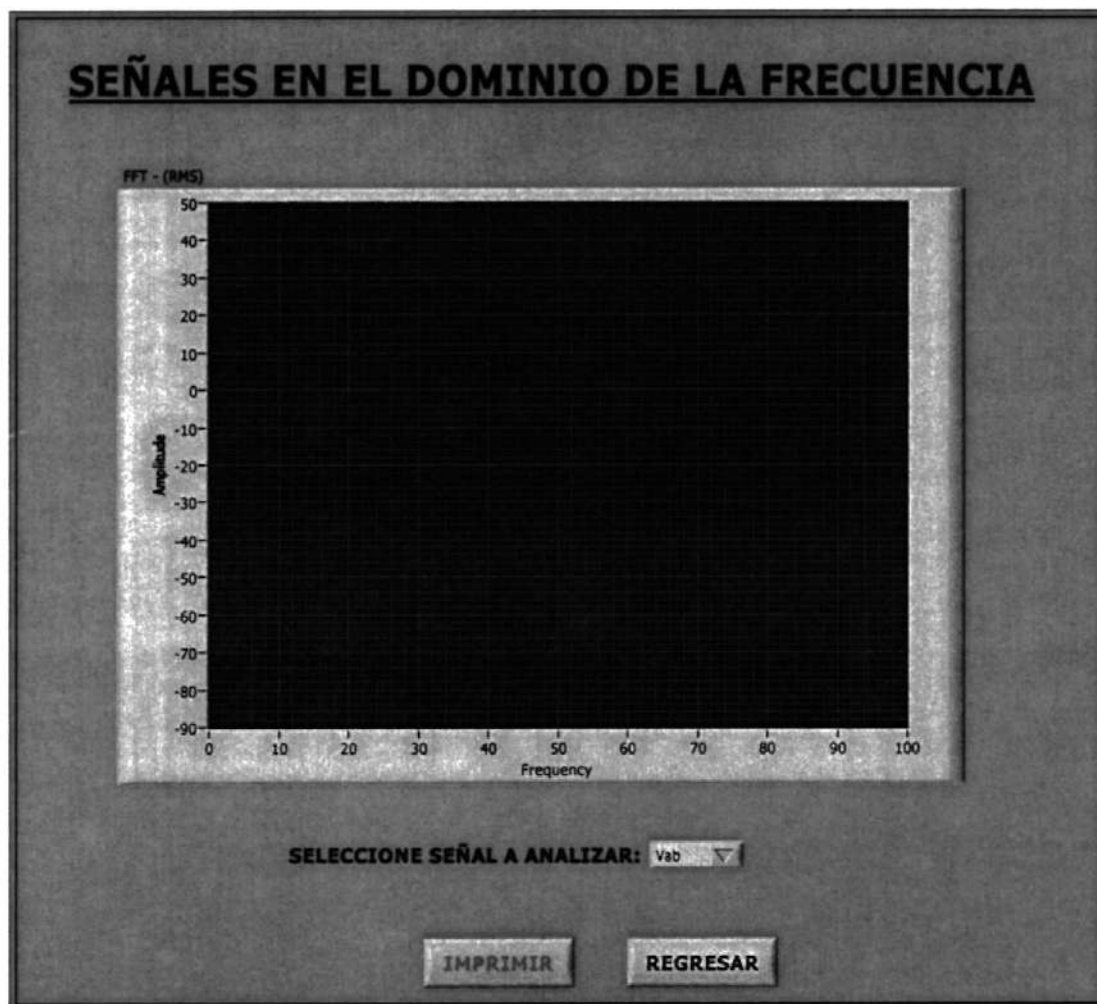
Message:

Connector Pane

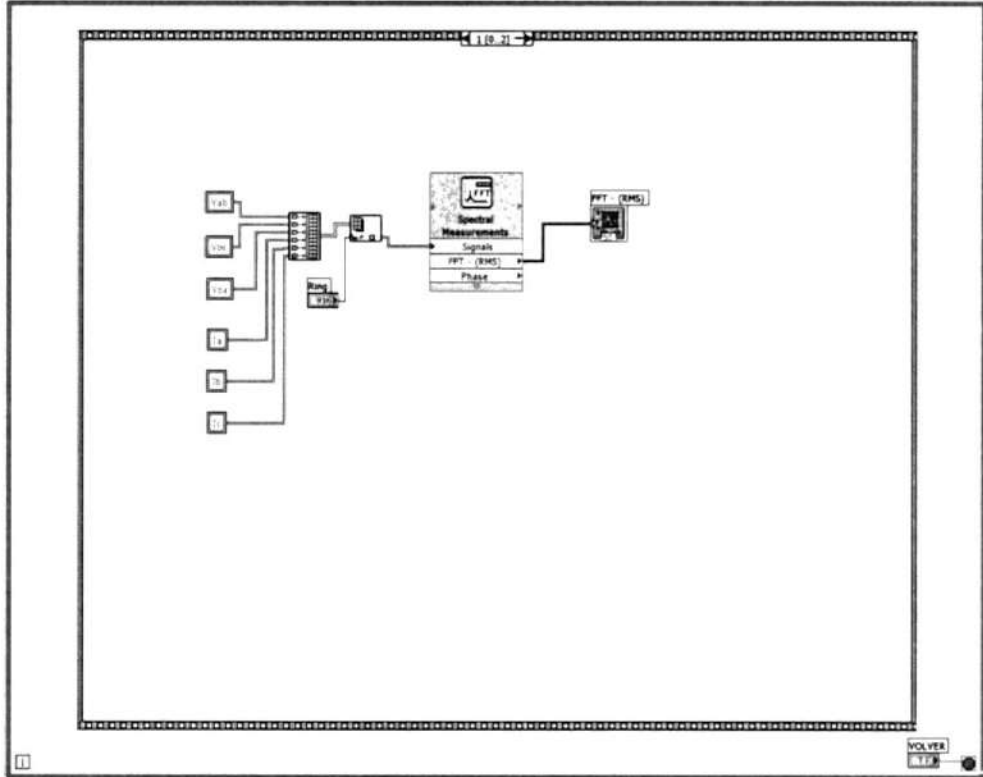


ESPECTROS.vi

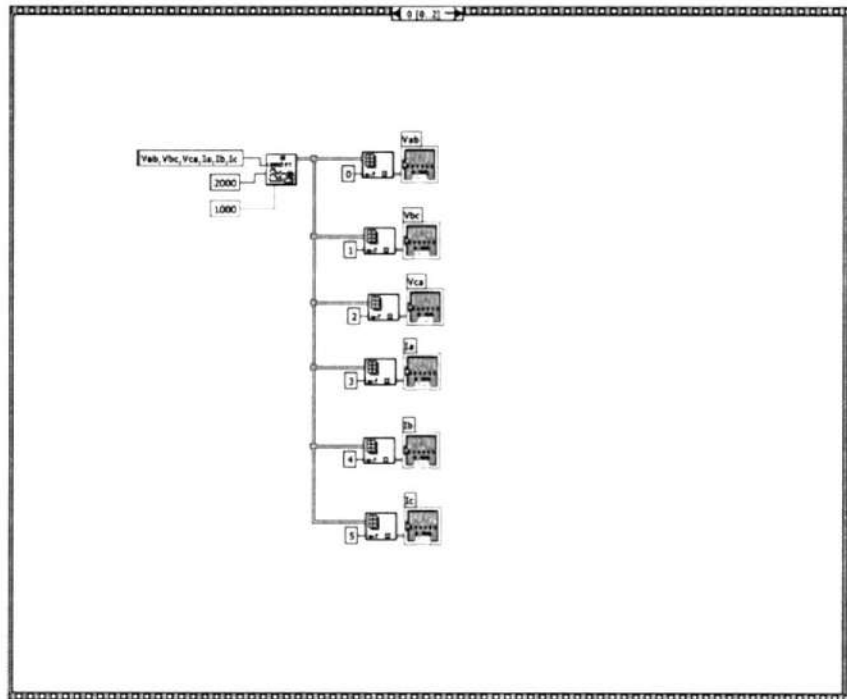
Front Panel

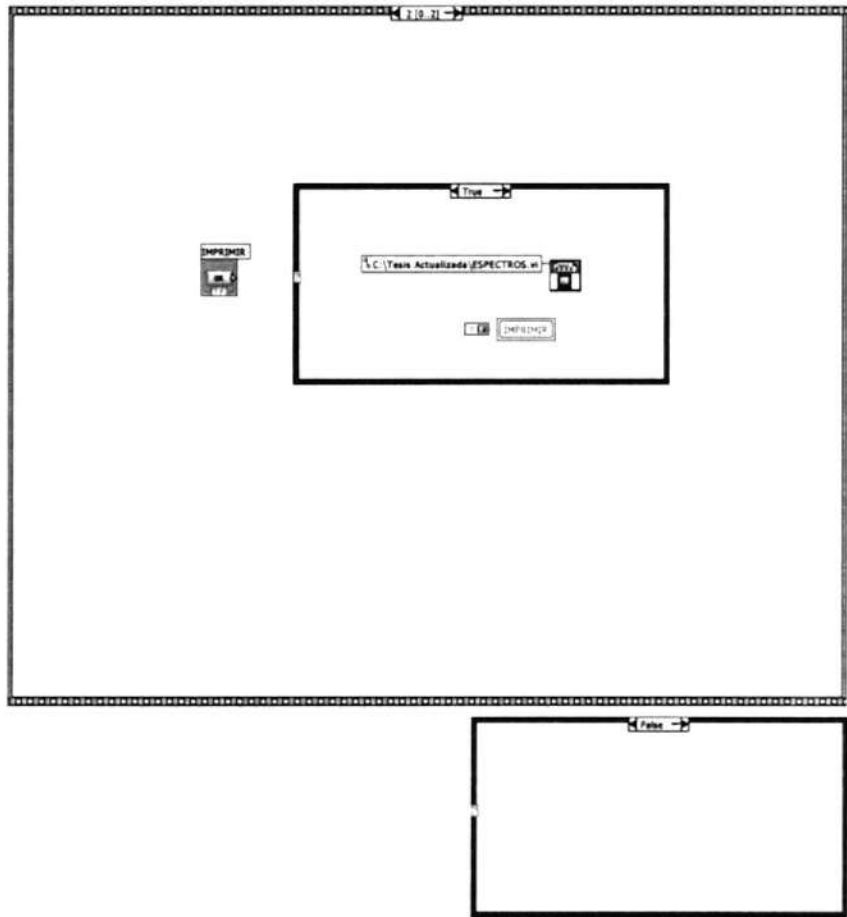


Block Diagram



16 200/200/10





Express VI Configuration Information



Spectral Measurements

Spectral Measurements

Performs spectral measurements, such as peak spectrum and auto-power spectrum, on a signal.

 This Express VI is configured as follows:

Selected Measurements: Magnitude (Peak)

View Phase: Wrapped and in Radians

Windowing: Flat Top

Averaging: None

Connector Pane



ERROR2.vi

Front Panel



Block Diagram



Connector Pane



ERROR1.vi

Front Panel



CIB -ESPOL

Block Diagram



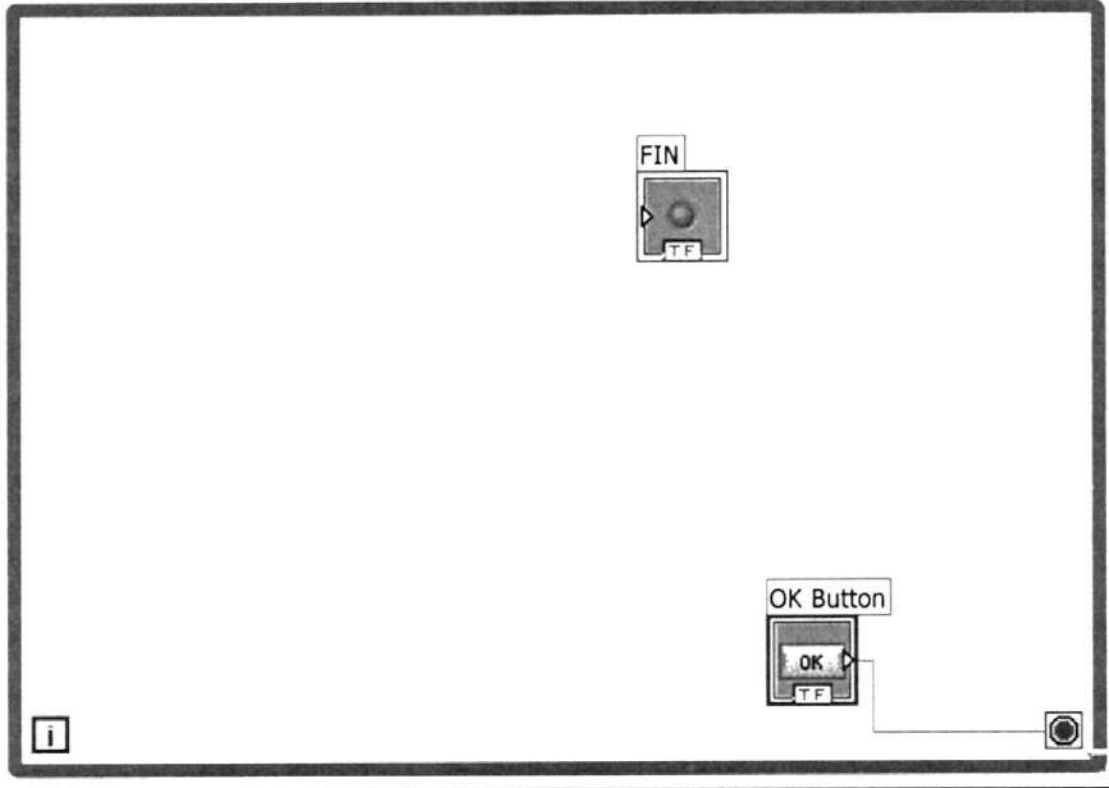
Connector Pane



Front Panel



Block Diagram



Connector Pane

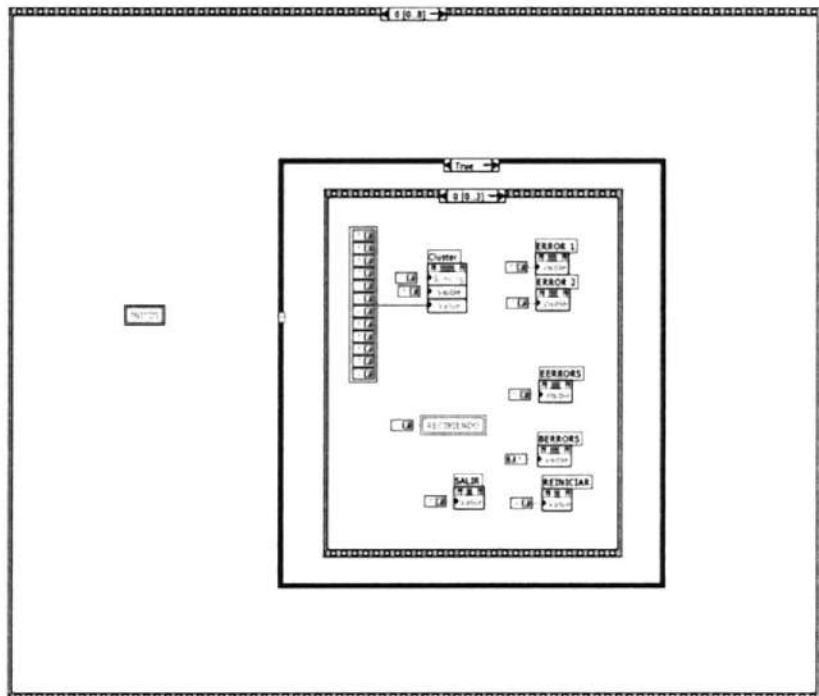
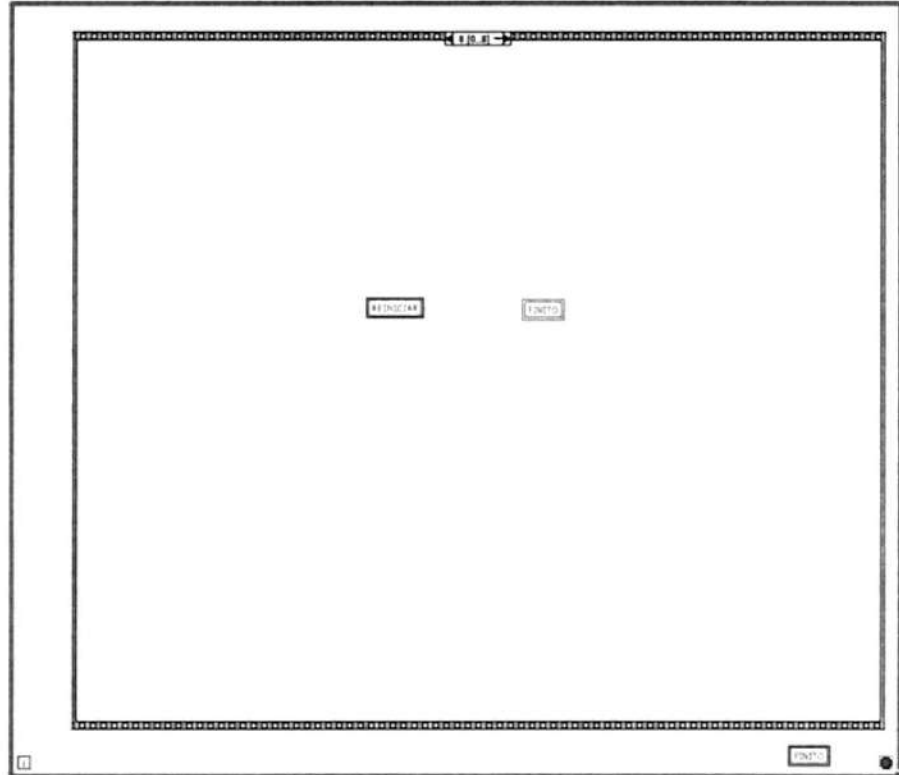


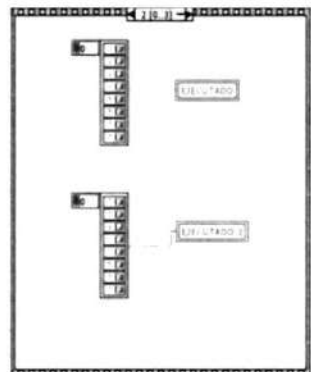
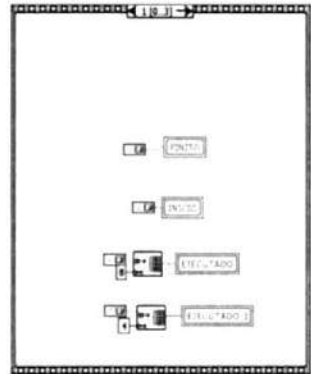
Front Panel

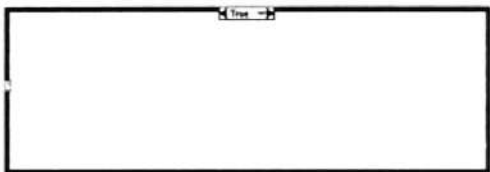
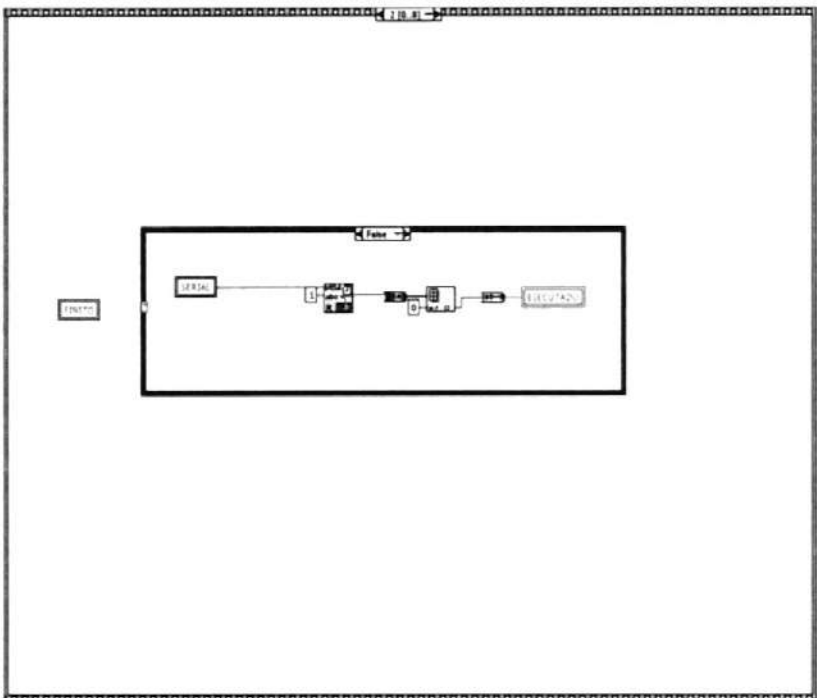
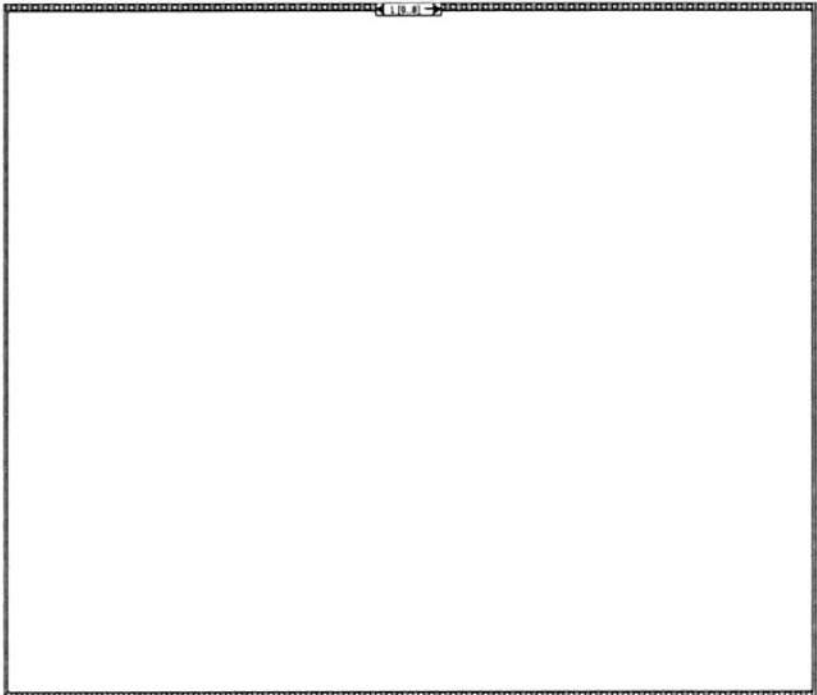


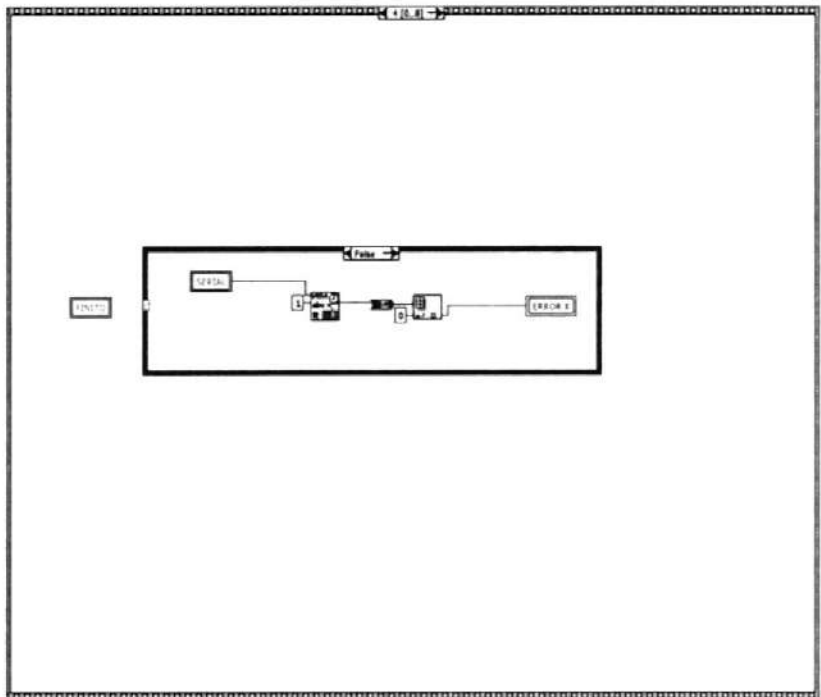
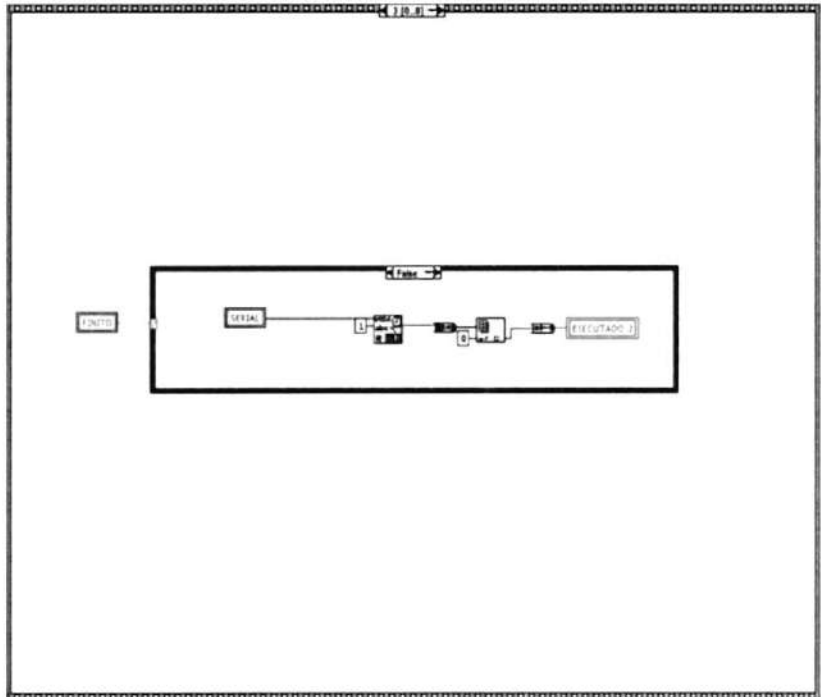
Block Diagram

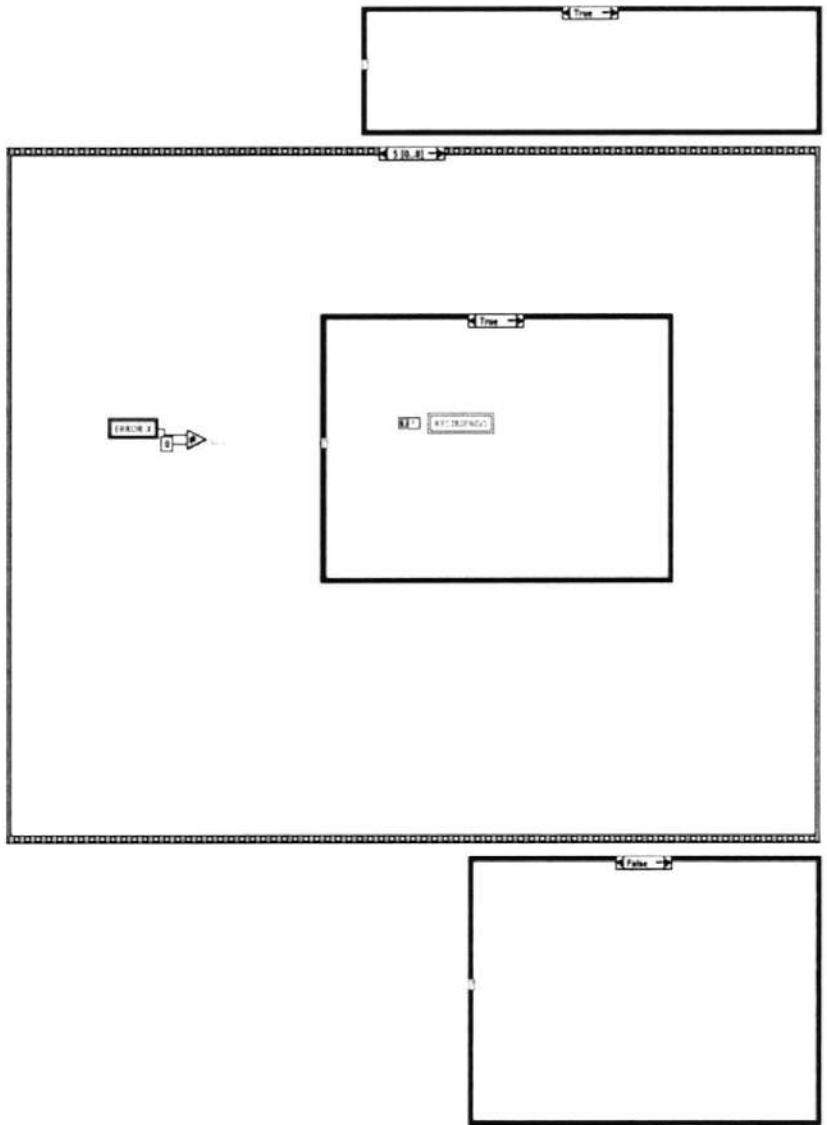
ORDENADO	177
ORDENADO 1	178
ORDENADO 2	179
TEST	180
ERROR 1	181
ERROR 2	182
RECIBIENDO	183
DATA IN	184
FALLA	185
PUNTO	186
SALIR	187
BERRORS	188
SEÑAL	189
EJECUTADA	190
ORDENADA	191
Charter	192
EJECUTADO	193
EJECUTADO 2	194
ENKCO	195
ERROR 1	196
ERROR 2	197
TITULO	198
REINICIAR	199
SERIAL	200
ERRORS	201
REINICIAR	202
ERRORS	203
REINICIAR	204
ERRORS	205
REINICIAR	206
ERRORS	207
REINICIAR	208
ERRORS	209
REINICIAR	210
ERRORS	211
REINICIAR	212
ERRORS	213
REINICIAR	214
ERRORS	215
REINICIAR	216
ERRORS	217
REINICIAR	218
ERRORS	219
REINICIAR	220
ERRORS	221
REINICIAR	222
ERRORS	223
REINICIAR	224
ERRORS	225
REINICIAR	226
ERRORS	227
REINICIAR	228
ERRORS	229
REINICIAR	230
ERRORS	231
REINICIAR	232
ERRORS	233
REINICIAR	234
ERRORS	235
REINICIAR	236
ERRORS	237
REINICIAR	238
ERRORS	239
REINICIAR	240
ERRORS	241
REINICIAR	242
ERRORS	243
REINICIAR	244
ERRORS	245
REINICIAR	246
ERRORS	247
REINICIAR	248
ERRORS	249
REINICIAR	250
ERRORS	251
REINICIAR	252
ERRORS	253
REINICIAR	254
ERRORS	255
REINICIAR	256
ERRORS	257
REINICIAR	258
ERRORS	259
REINICIAR	260
ERRORS	261
REINICIAR	262
ERRORS	263
REINICIAR	264
ERRORS	265
REINICIAR	266
ERRORS	267
REINICIAR	268
ERRORS	269
REINICIAR	270
ERRORS	271
REINICIAR	272
ERRORS	273
REINICIAR	274
ERRORS	275
REINICIAR	276
ERRORS	277
REINICIAR	278
ERRORS	279
REINICIAR	280
ERRORS	281
REINICIAR	282
ERRORS	283
REINICIAR	284
ERRORS	285
REINICIAR	286
ERRORS	287
REINICIAR	288
ERRORS	289
REINICIAR	290
ERRORS	291
REINICIAR	292
ERRORS	293
REINICIAR	294
ERRORS	295
REINICIAR	296
ERRORS	297
REINICIAR	298
ERRORS	299
REINICIAR	300

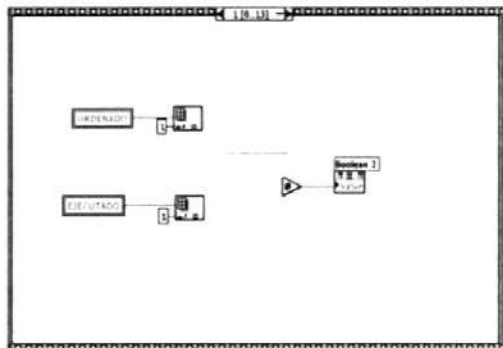
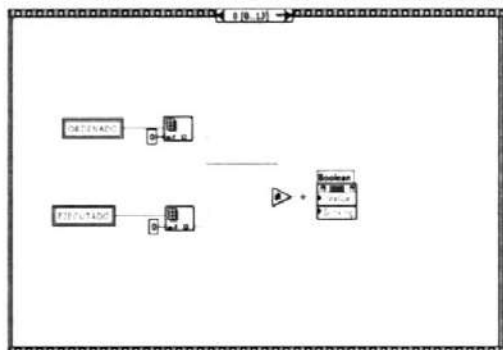
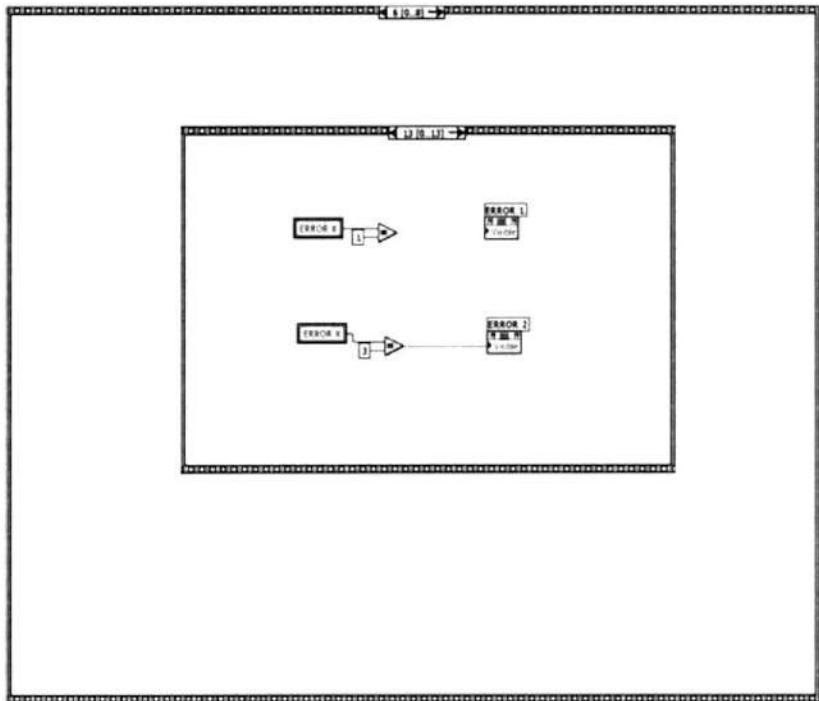


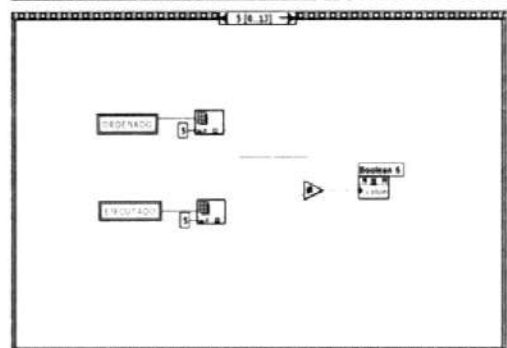
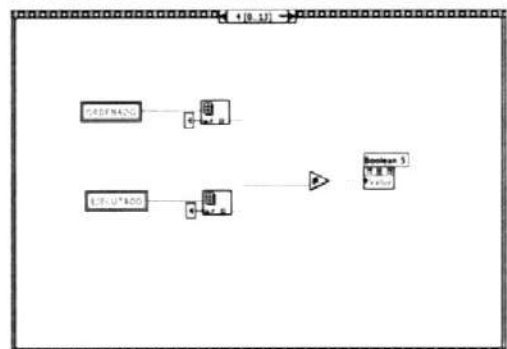
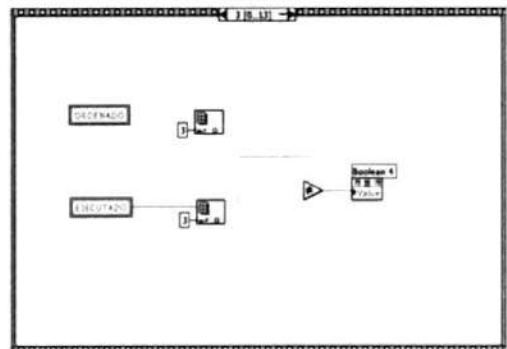
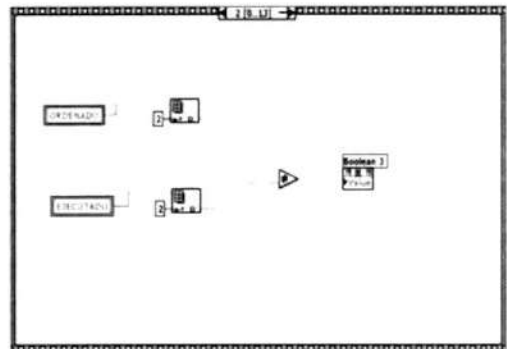


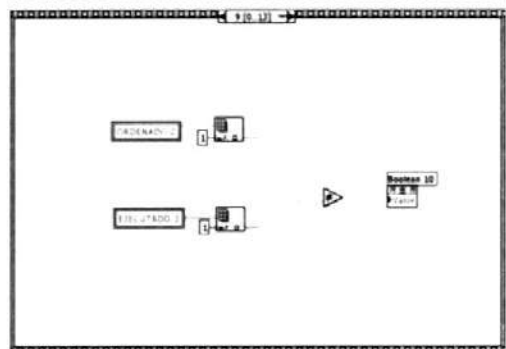
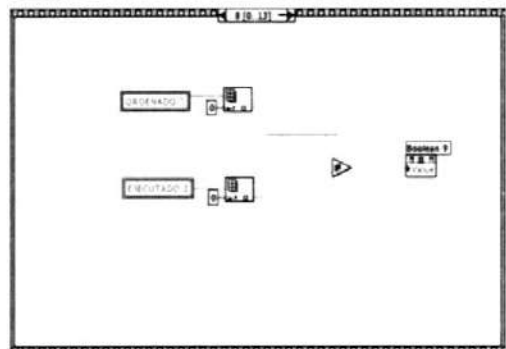
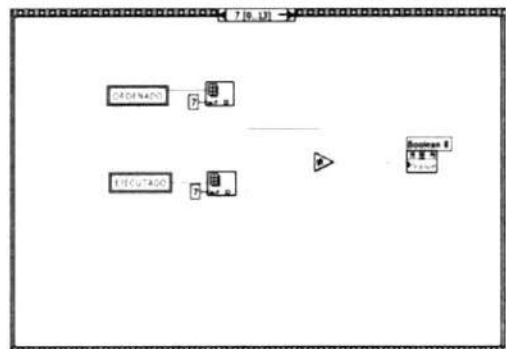
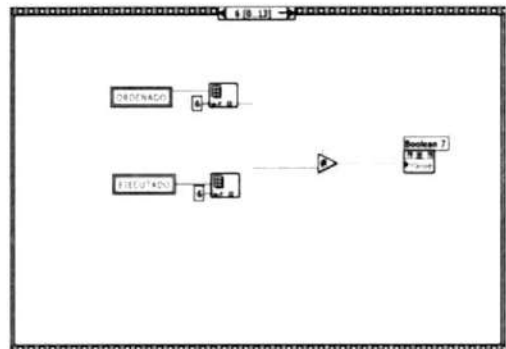


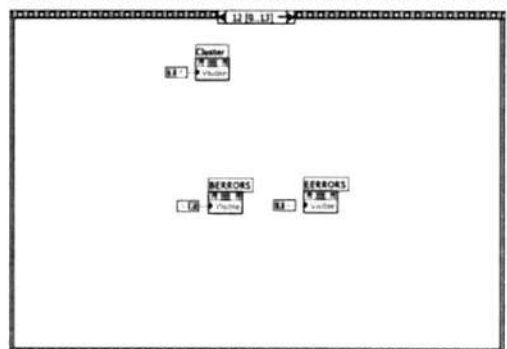
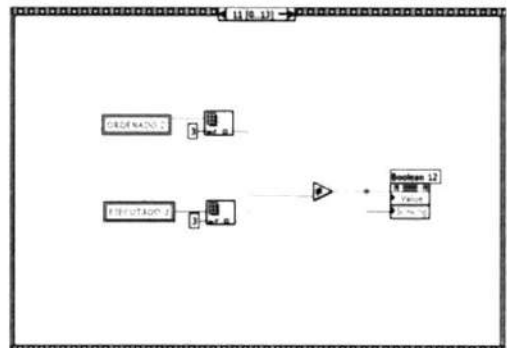
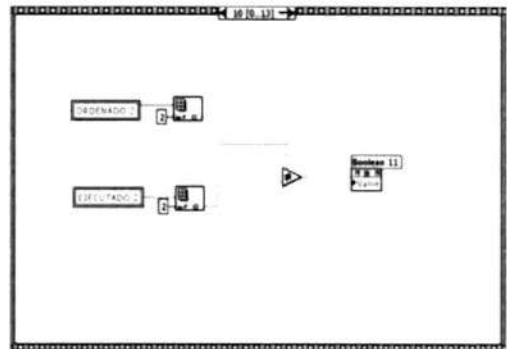


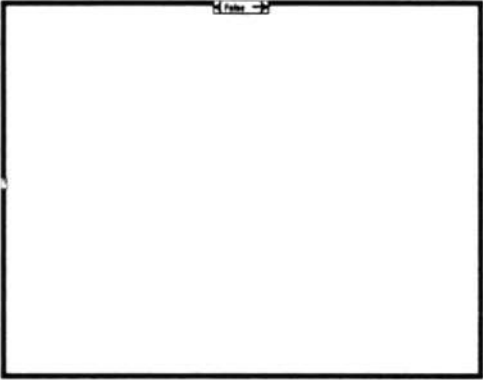
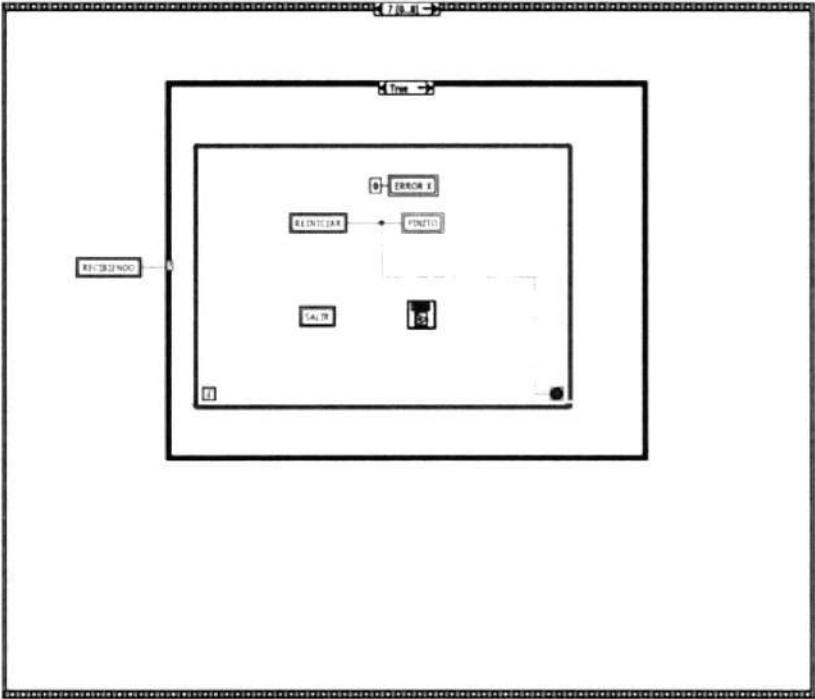










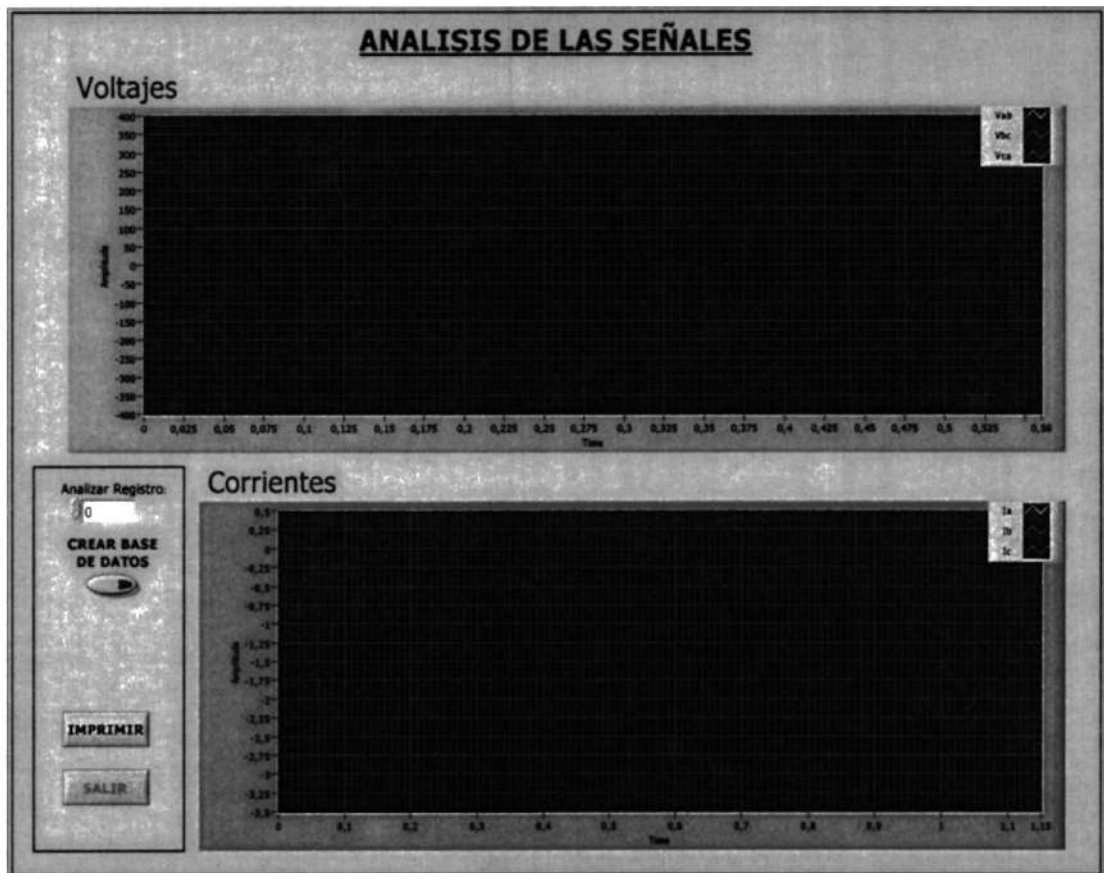


Connector Pane

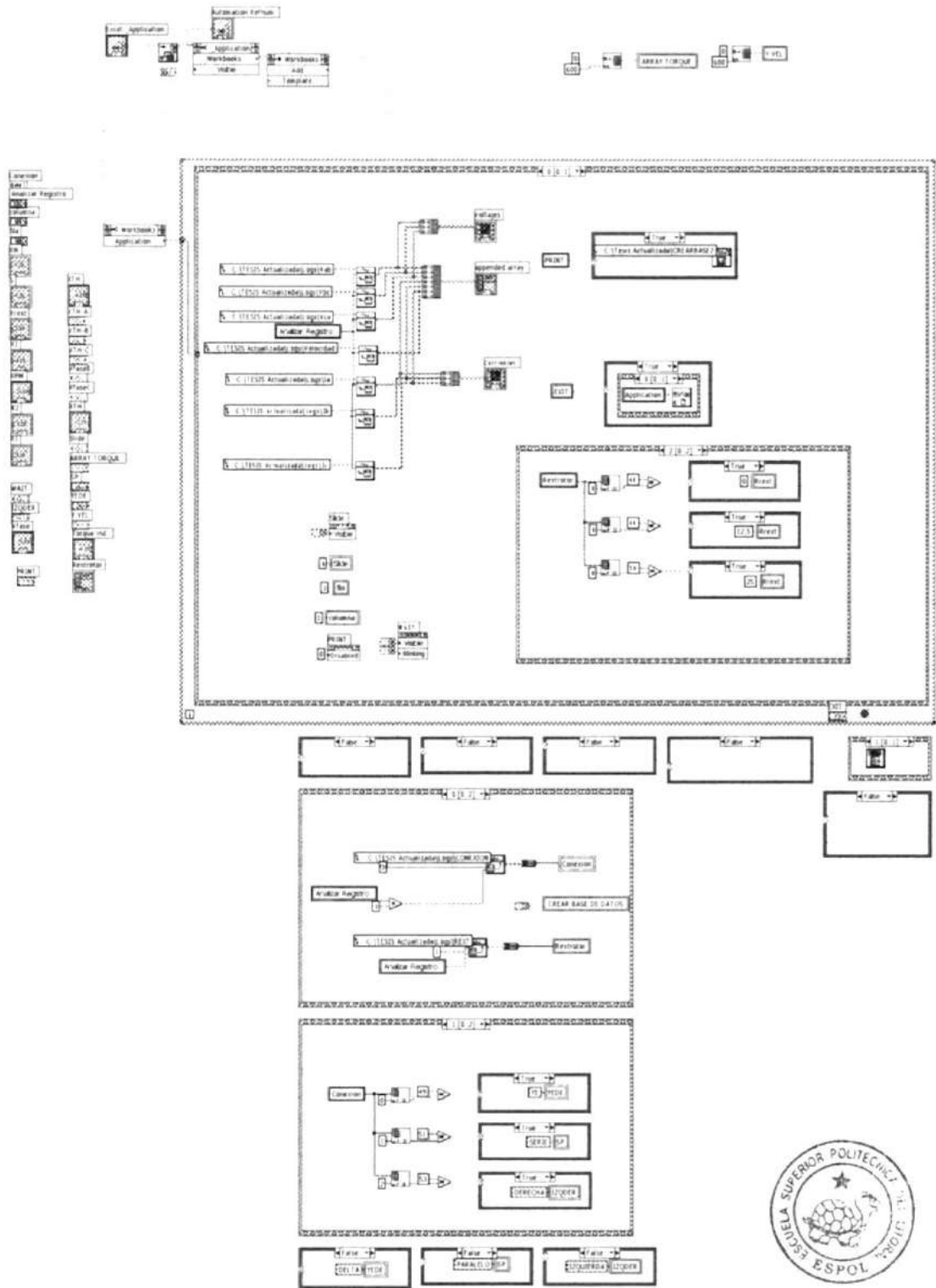


CREARBASE2.vi

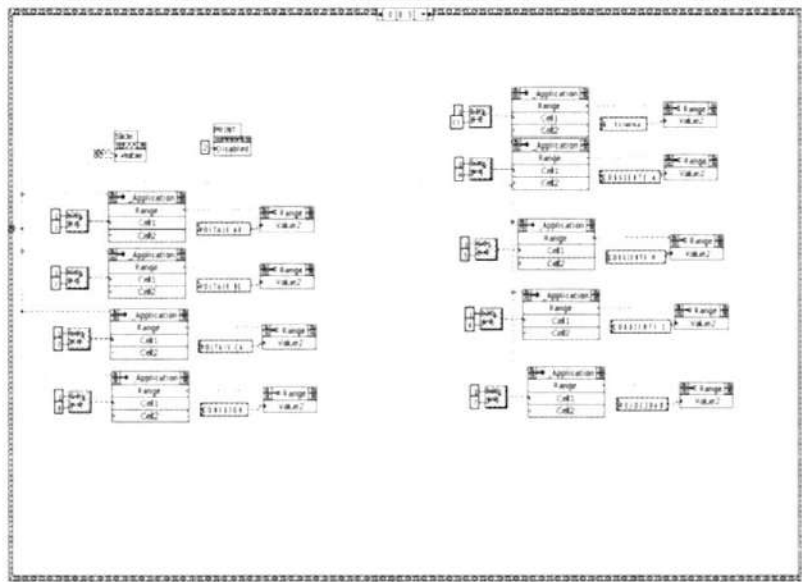
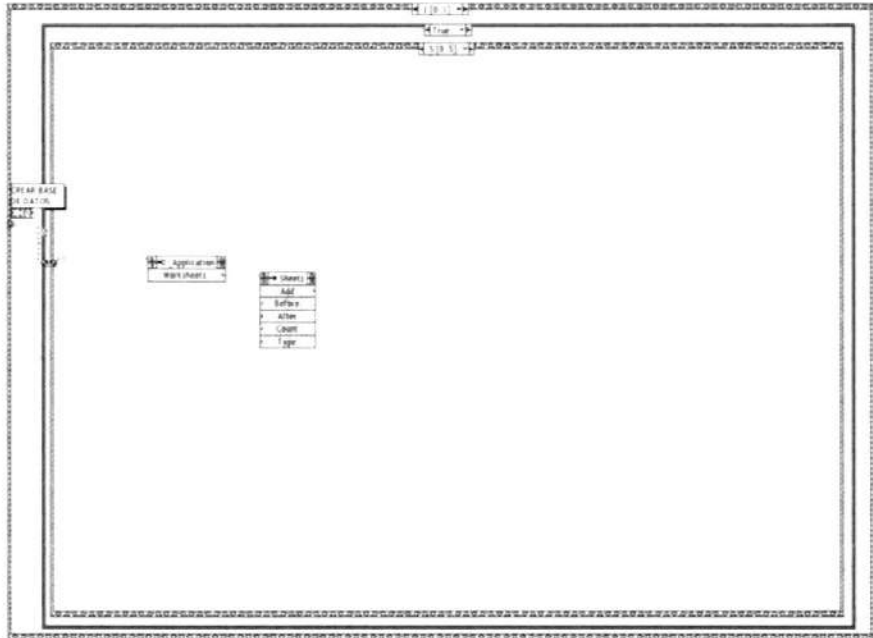
Front Panel

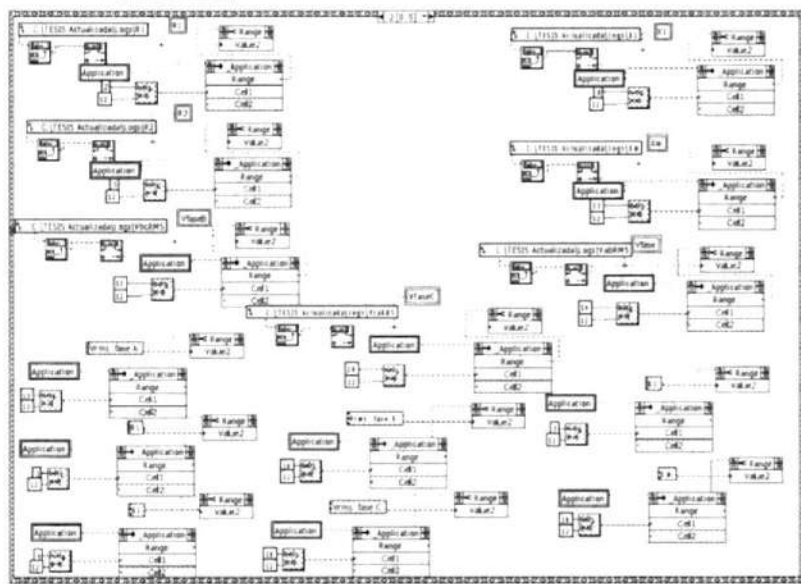
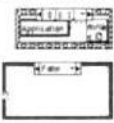
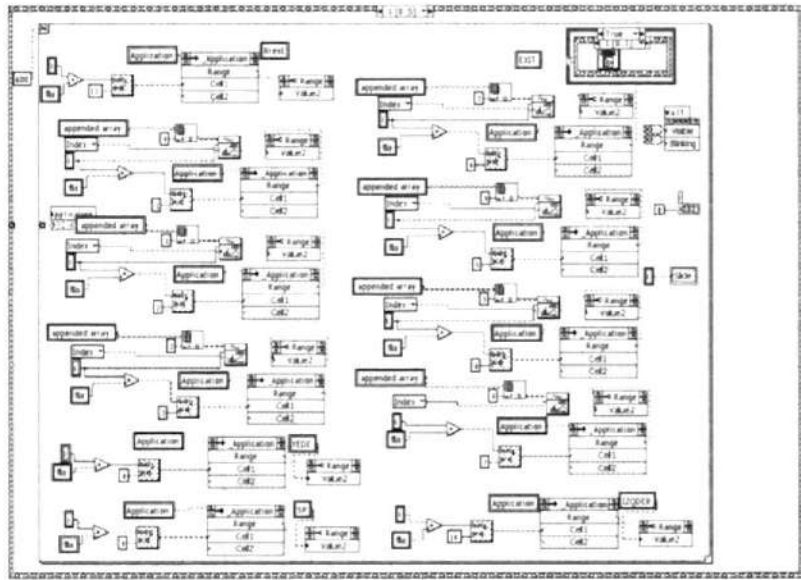


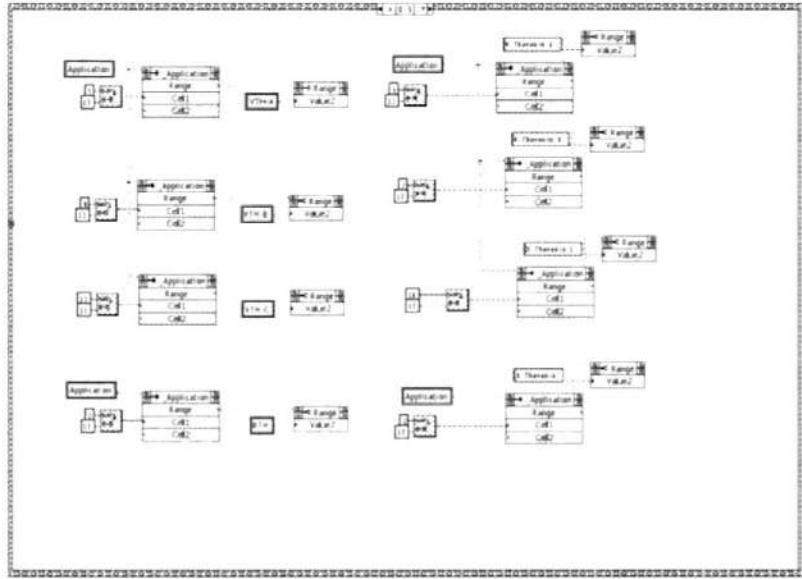
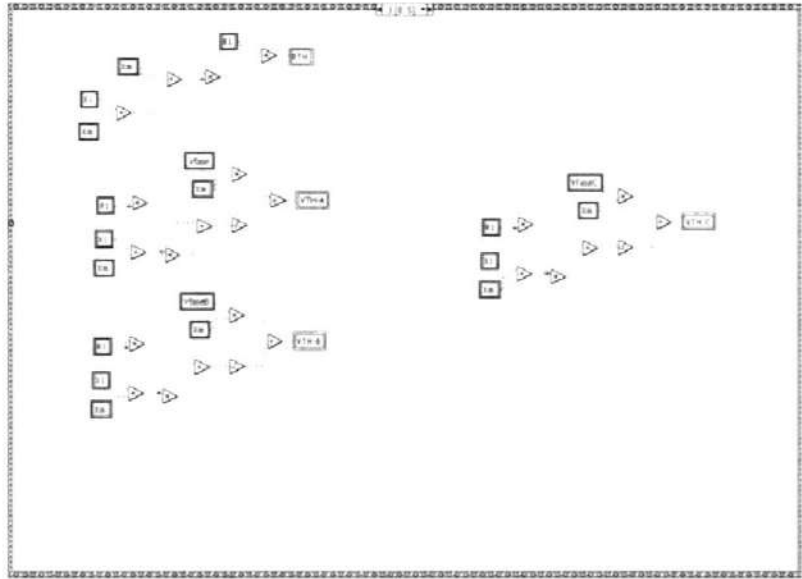
Block Diagram

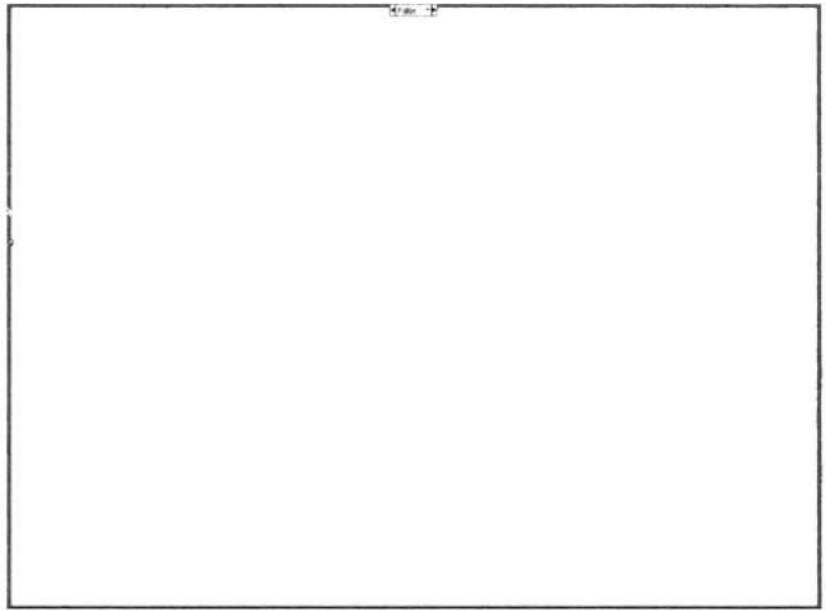


CIB - ESPOL









Connector Pane



PRUEBAS PARA EL MOTOR AC.vi

Front Panel

DATA CLUSTER

D0	D1	D2	D3	D4	D5	D6

PRUEBAS PARA EL MOTOR AC KATO

DETERMINAR RESISTENCIA DEL ESTATOR POR FASE

↓

PRUEBA DE ROTOR BLOQUEADO

↓

PRUEBA DE VACIO

↓

CIRCUITO EQUIVALENTE FINAL POR FASE

CONFIGURACIÓN DEL MOTOR:

YE SERIE 1. ARDILLA

EMPEZAR

SALIR

X28
0

OPCODE
0

0

Rc

0 SERIAL

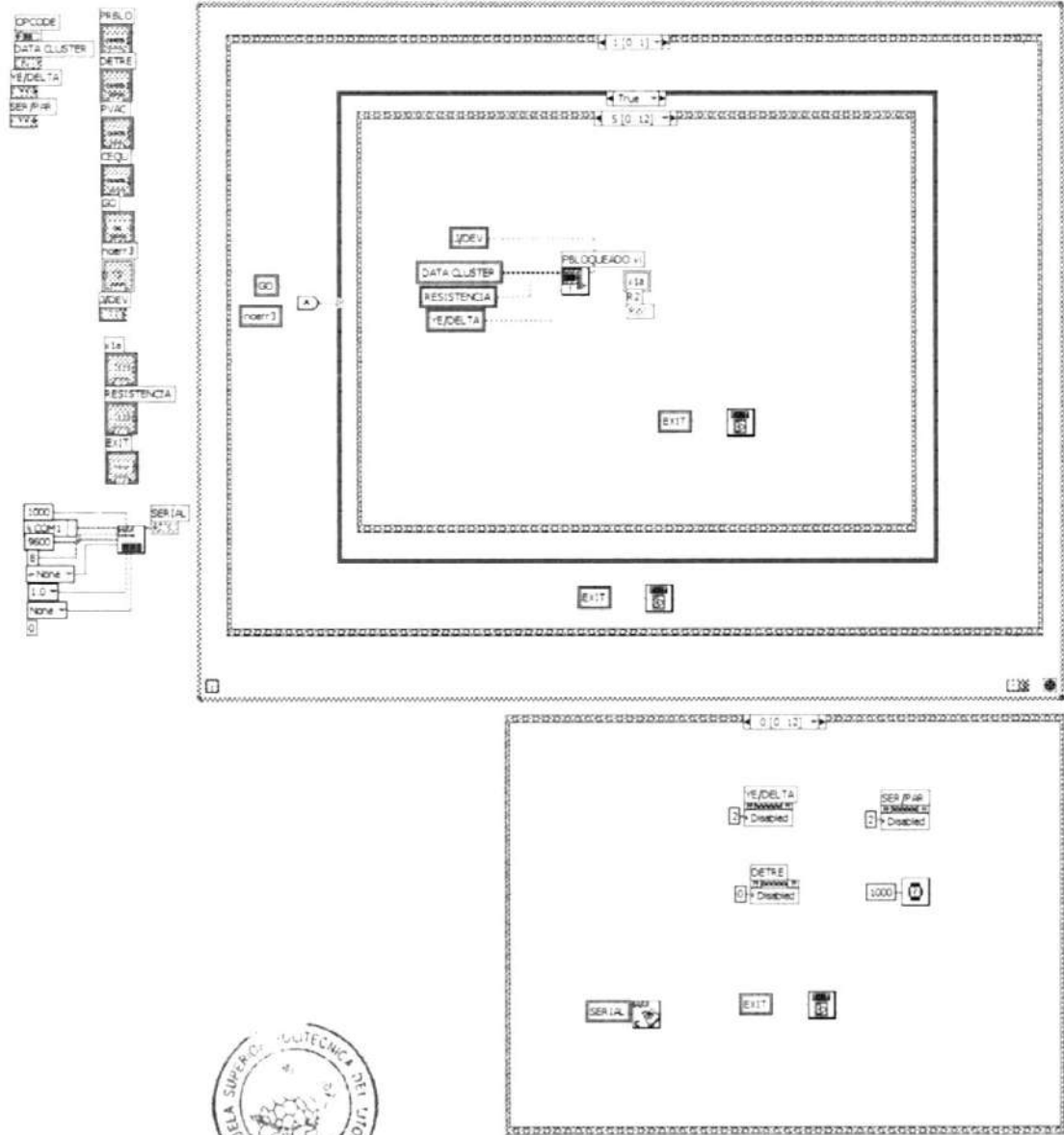
Xm

0

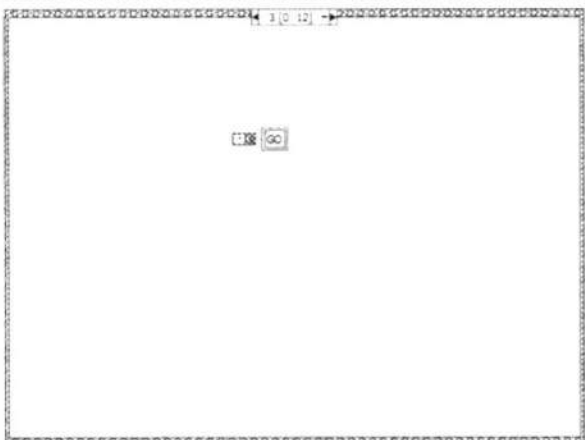
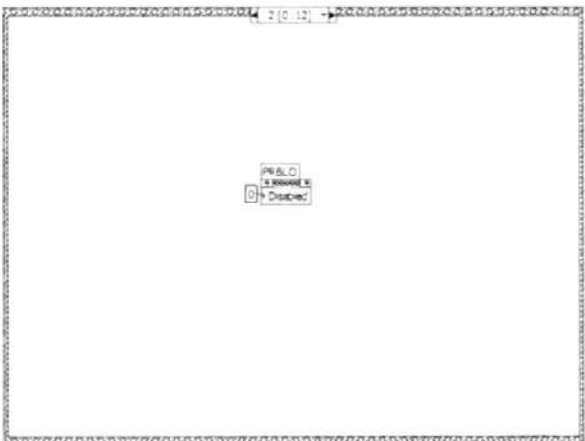
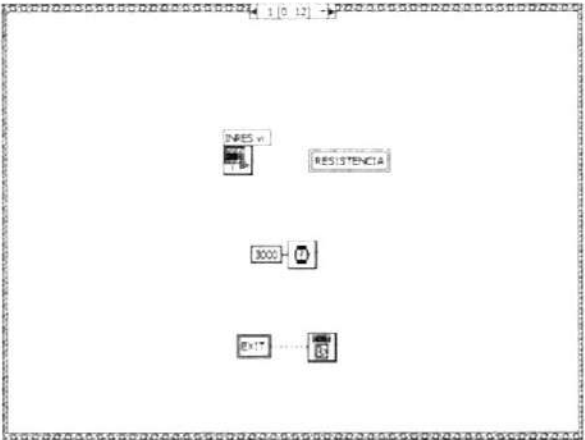
R2

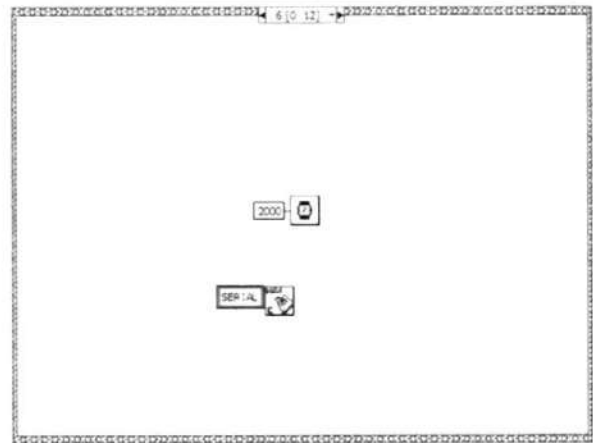
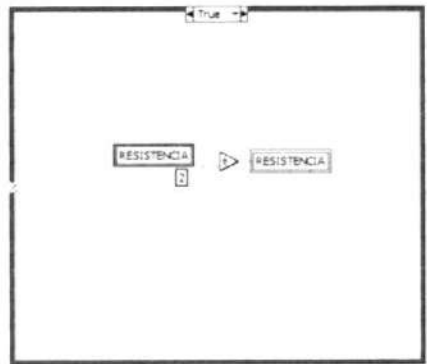
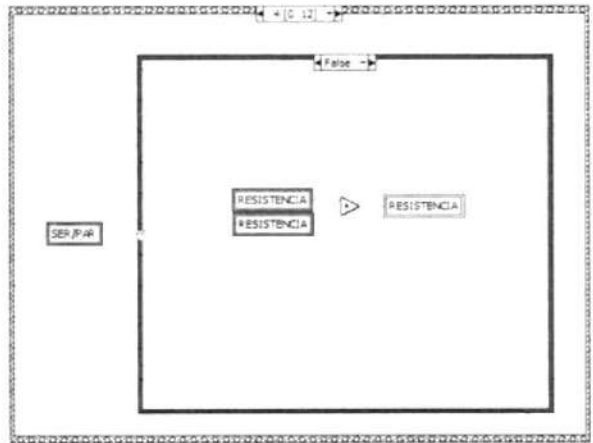
0

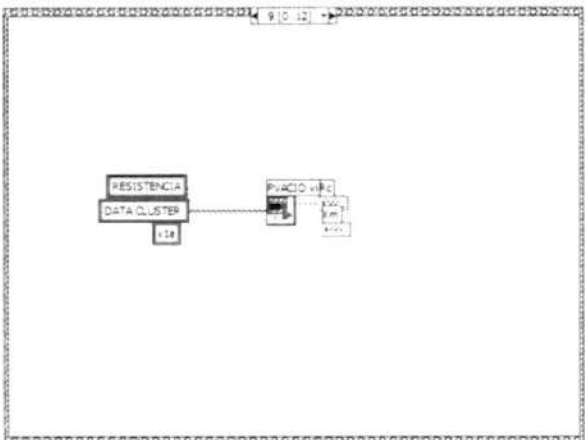
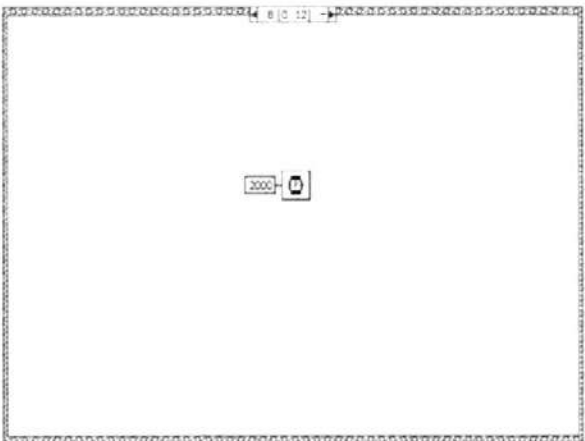
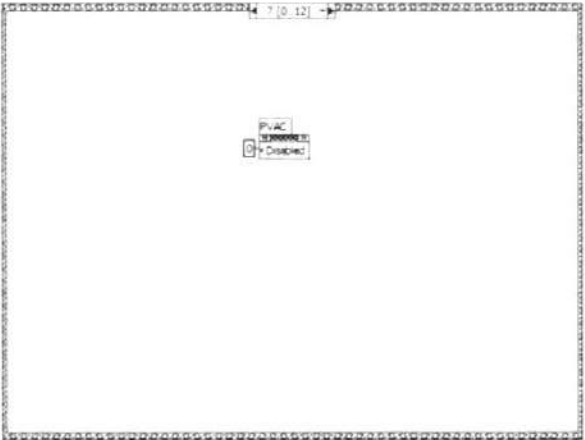
Block Diagram

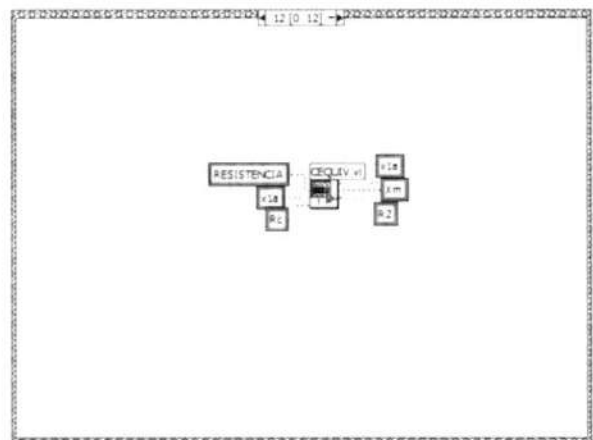
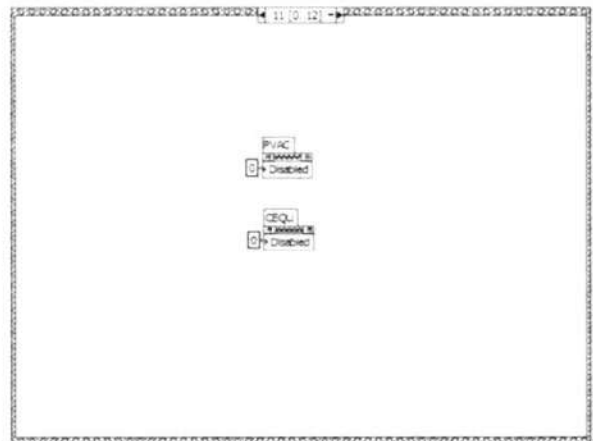
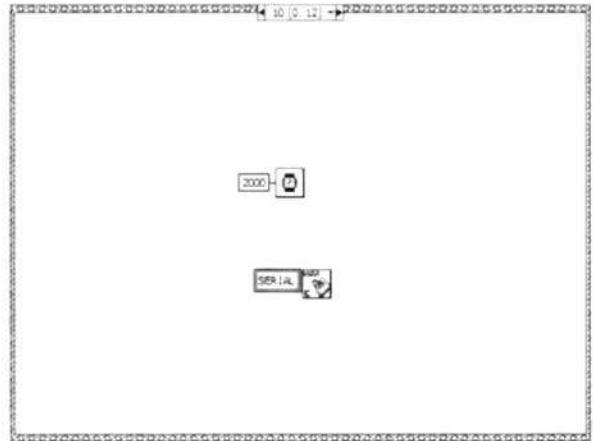


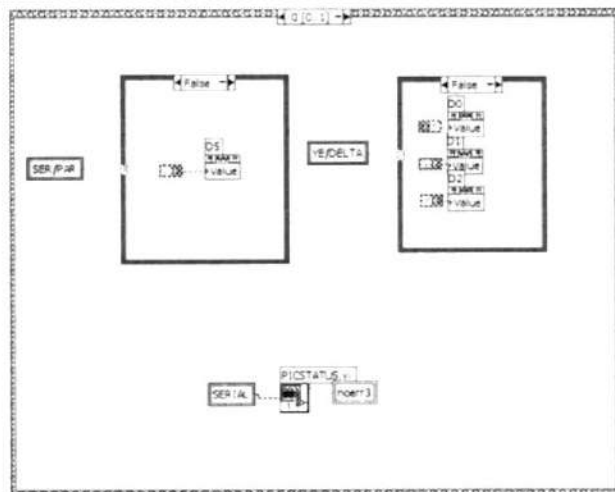
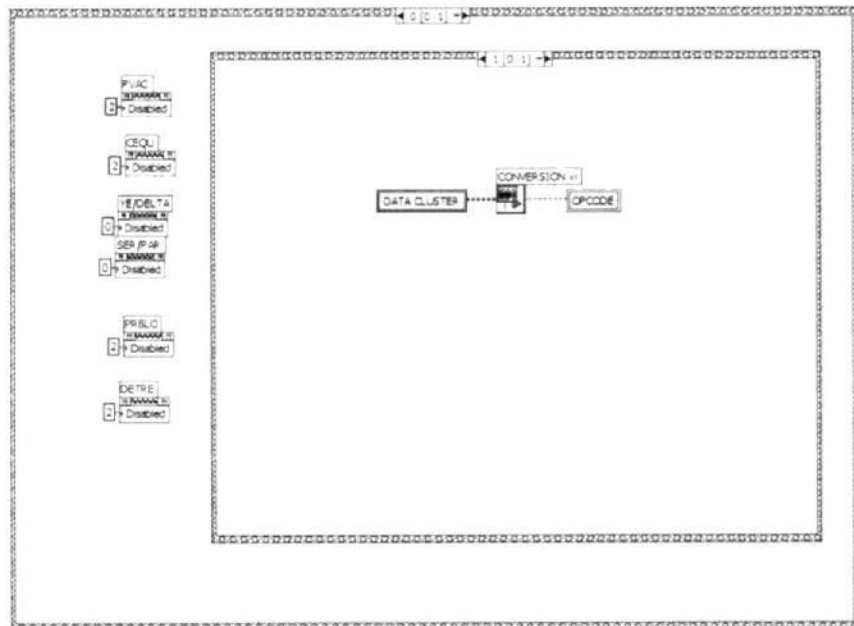
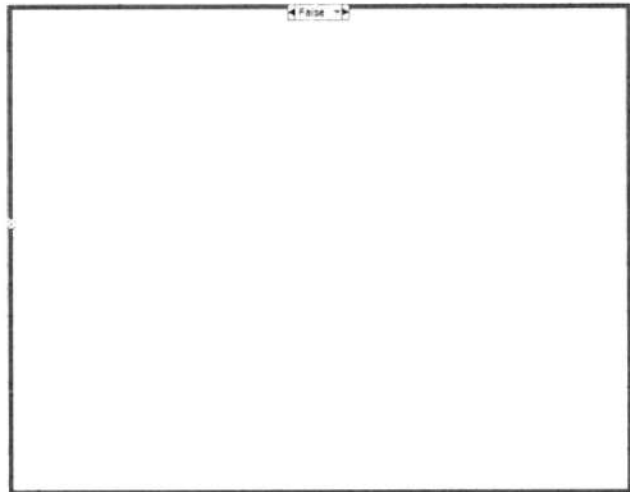
CIB-ESPOL

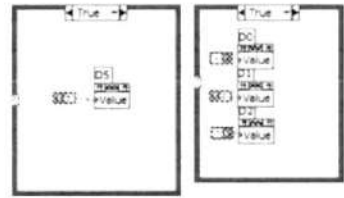




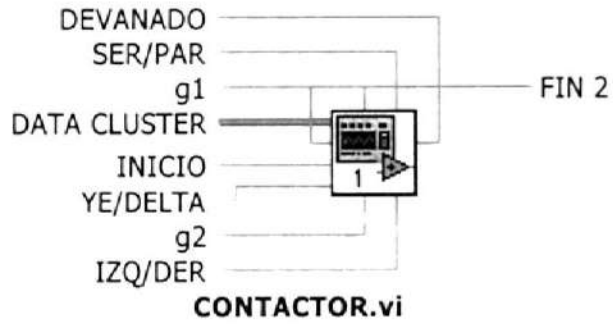




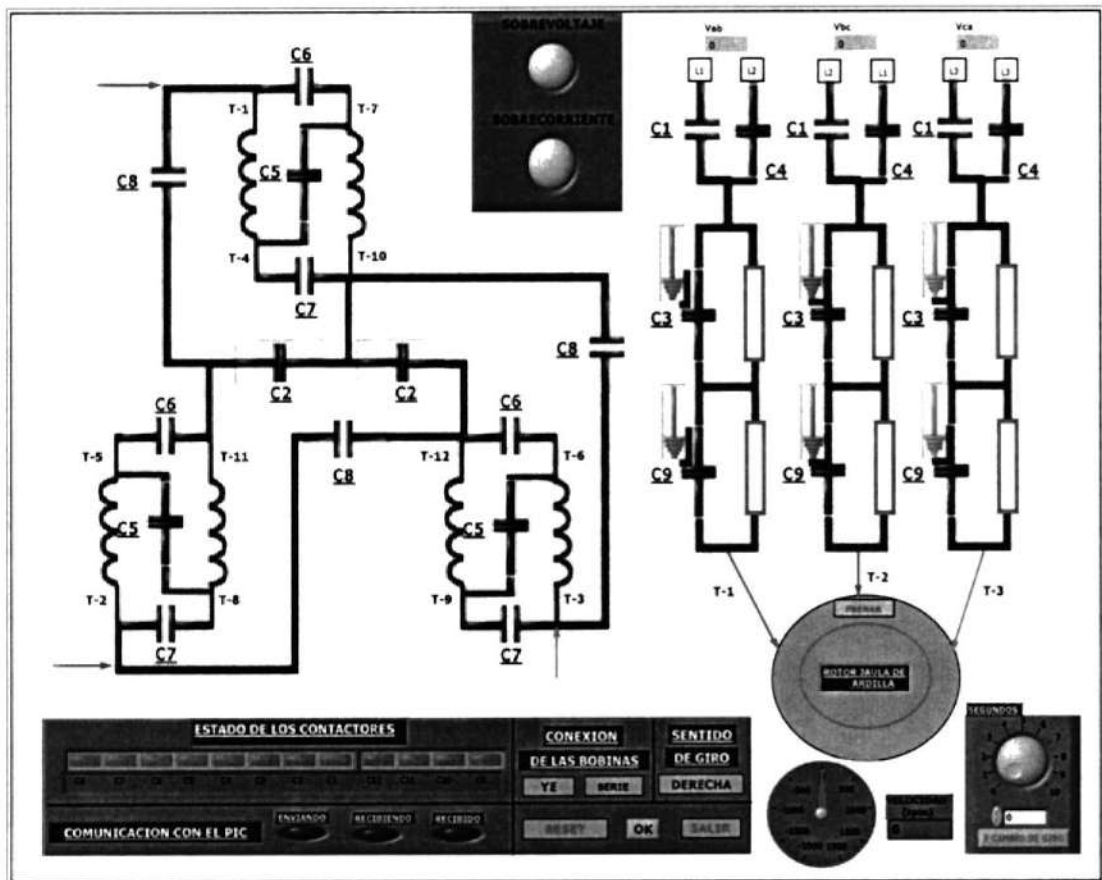


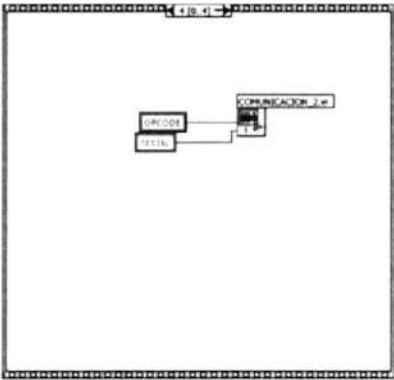
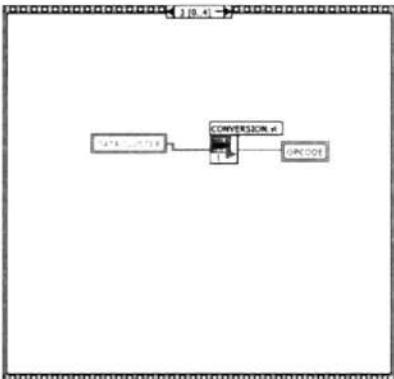
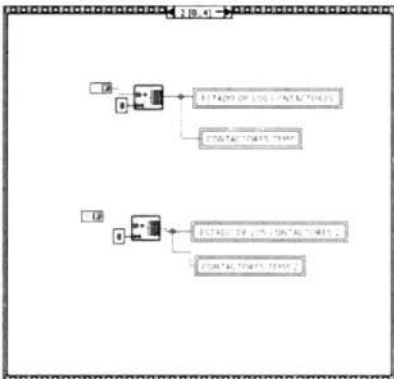
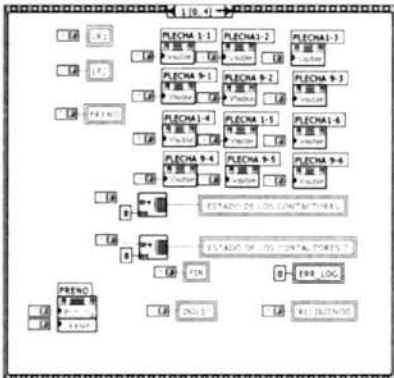


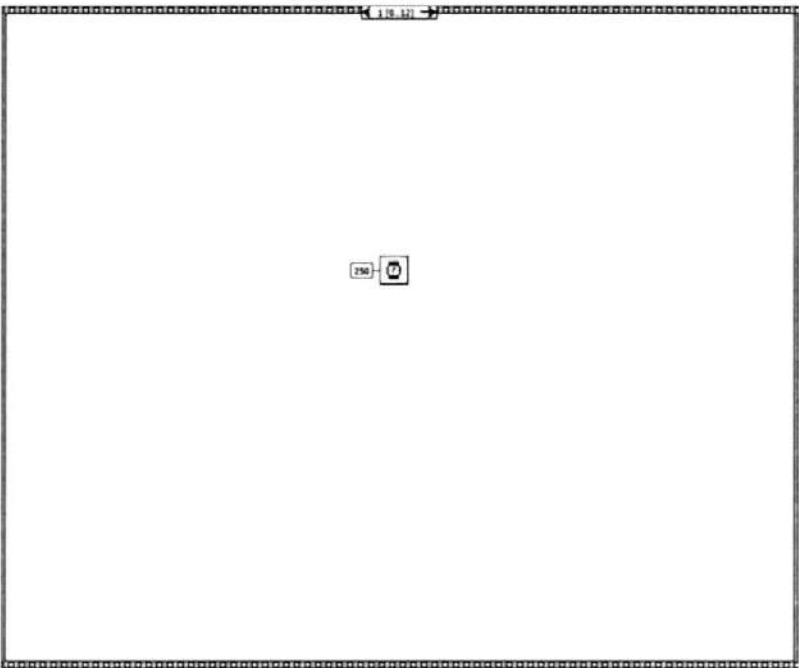
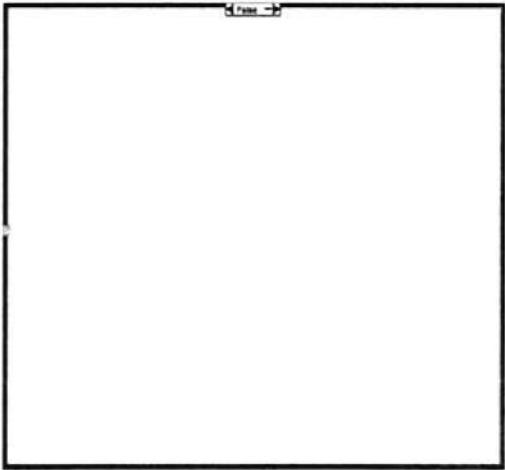
Connector Pane

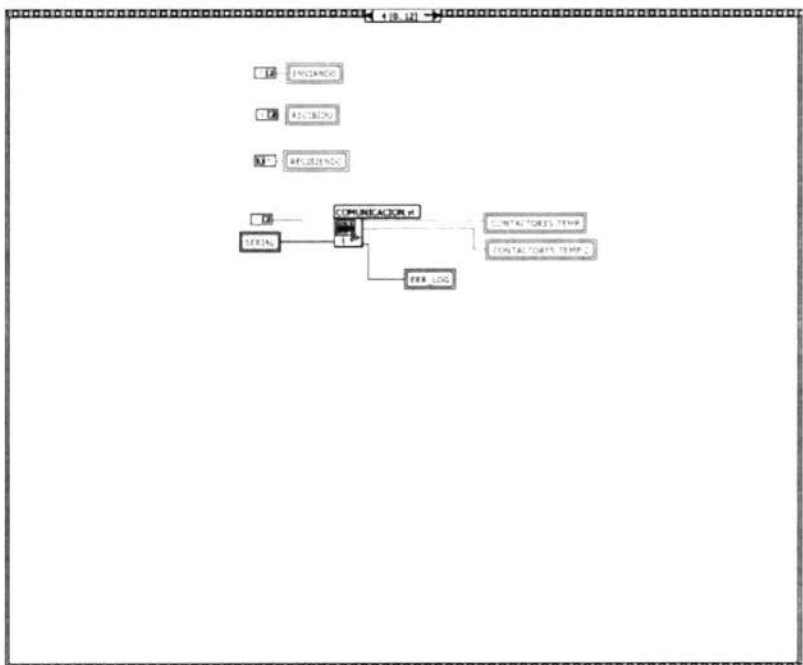
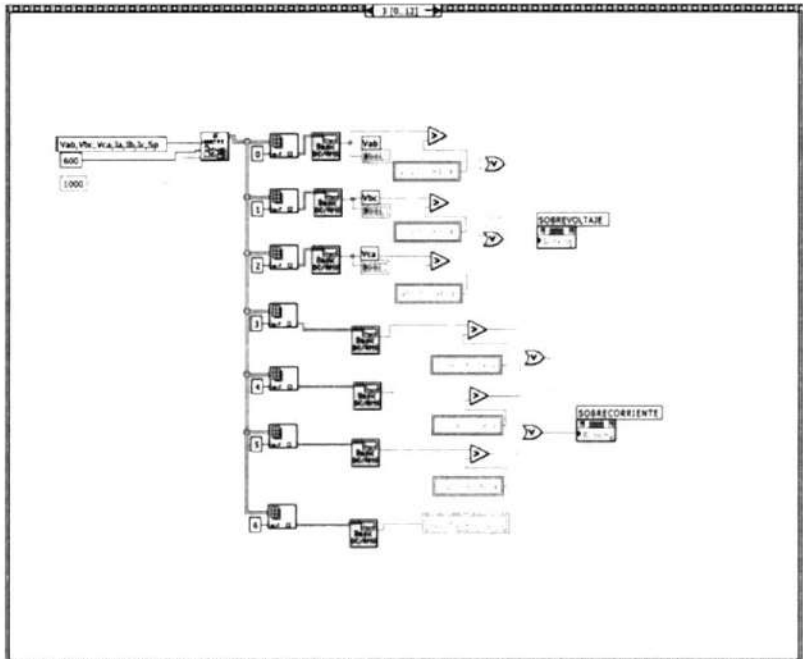


Front Panel









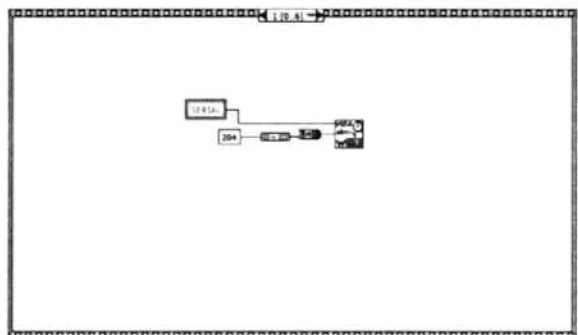
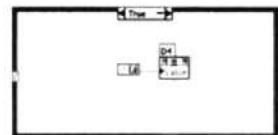
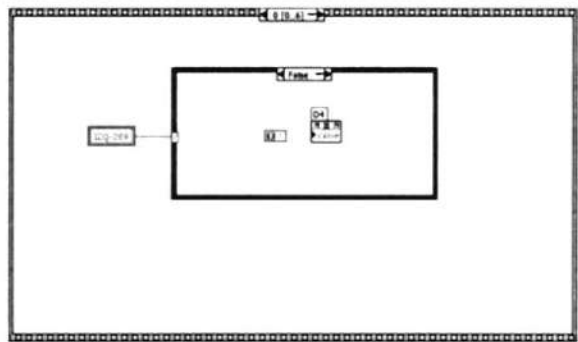
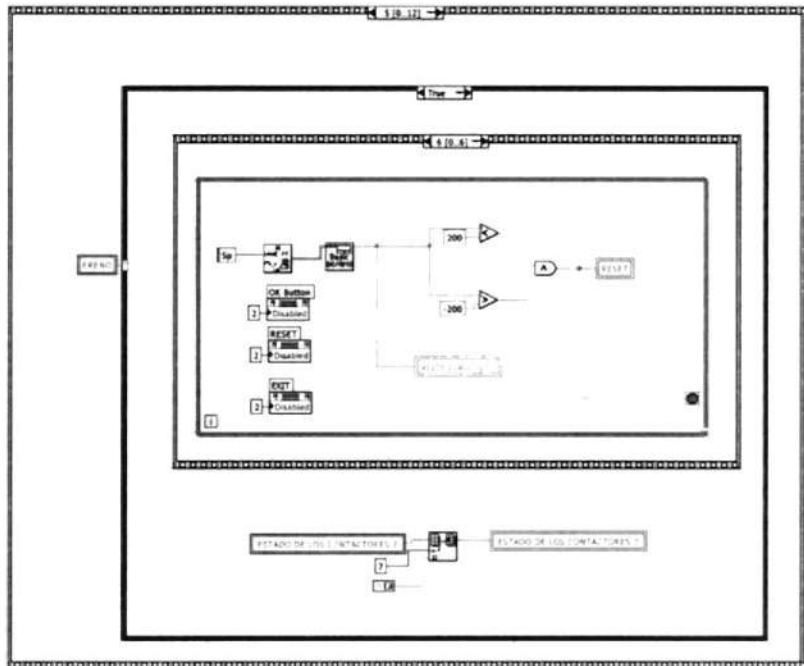


FIG. 61

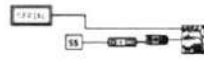
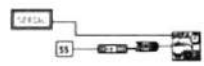


FIG. 62

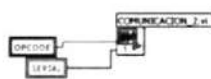


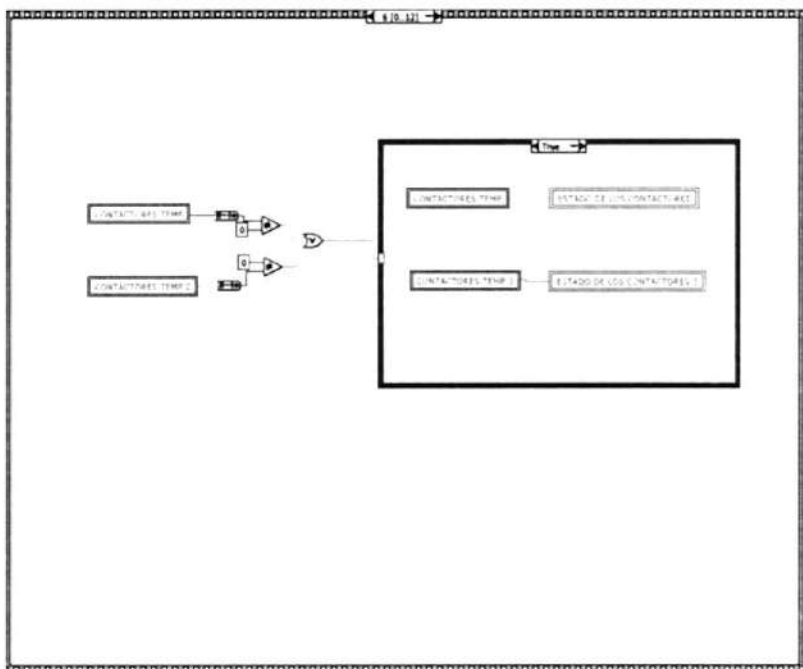
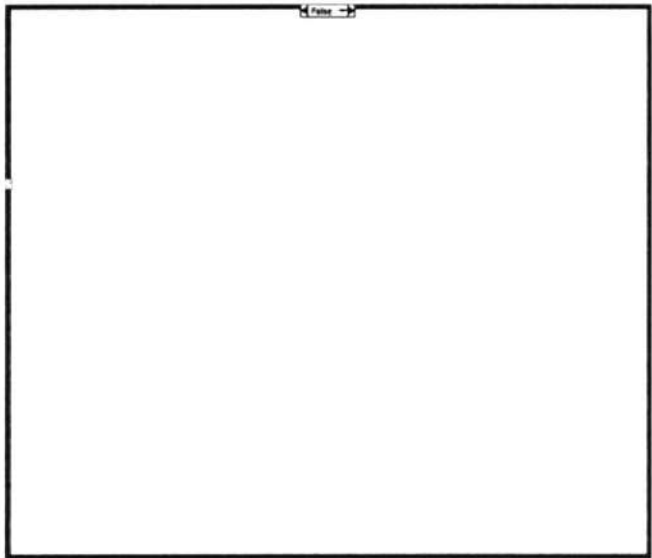
16 [RECORD]

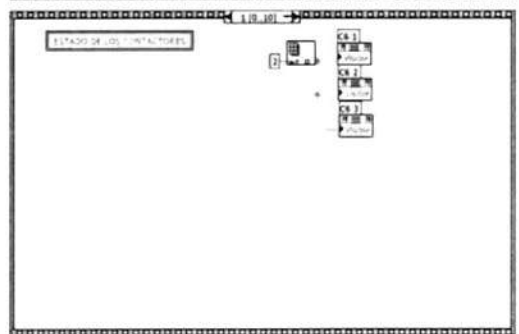
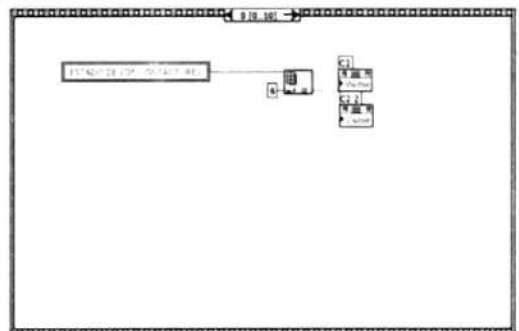
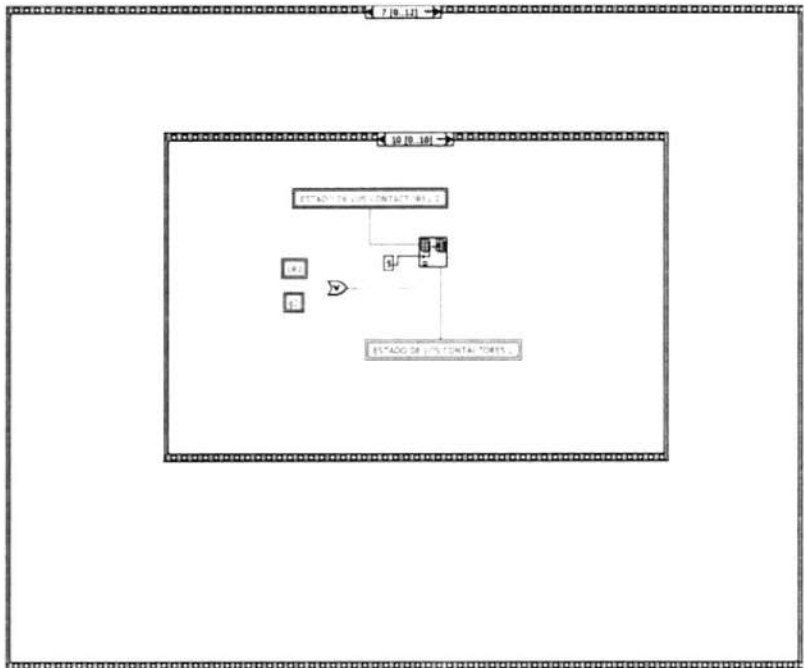
FIG. 63

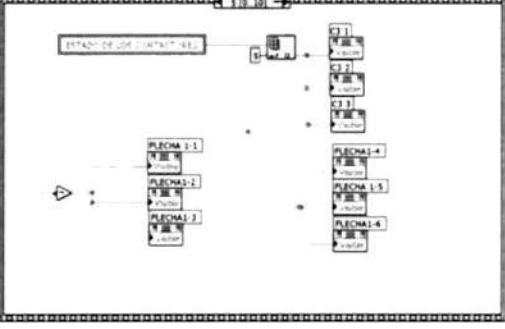
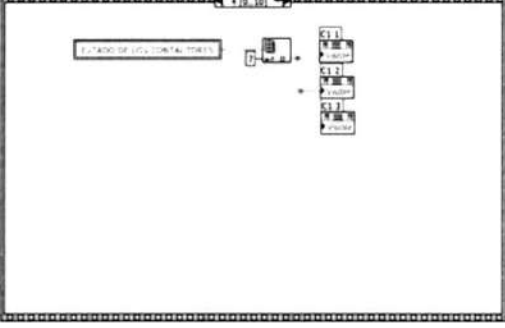
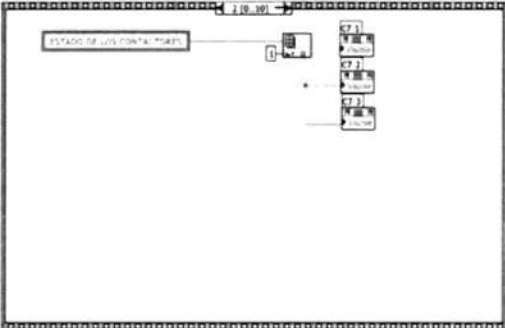


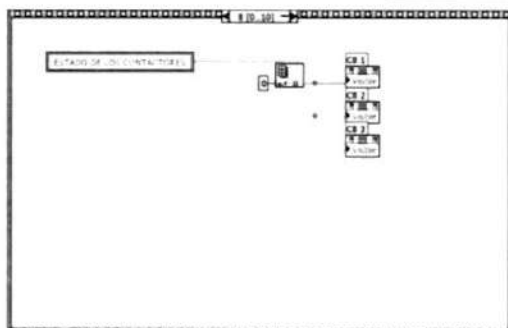
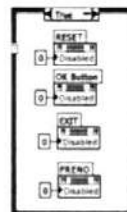
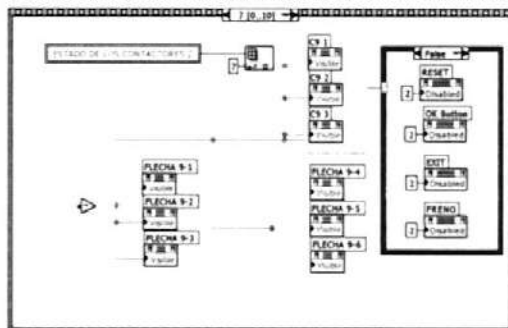
FIG. 64



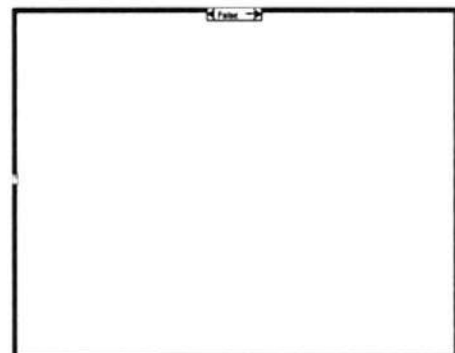
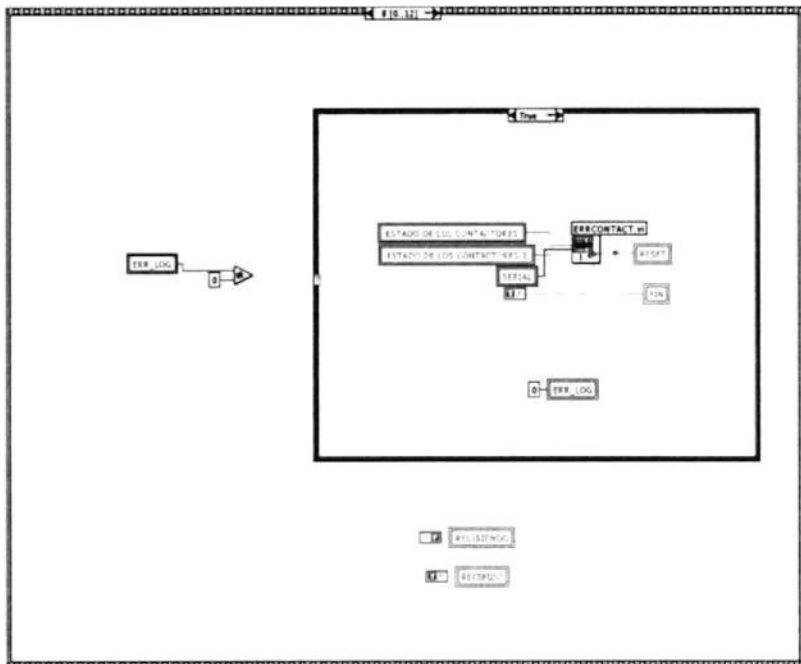




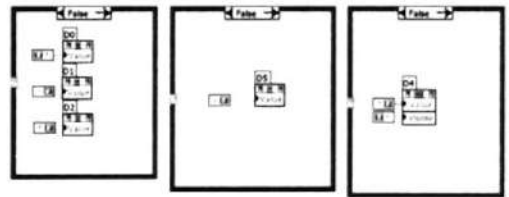
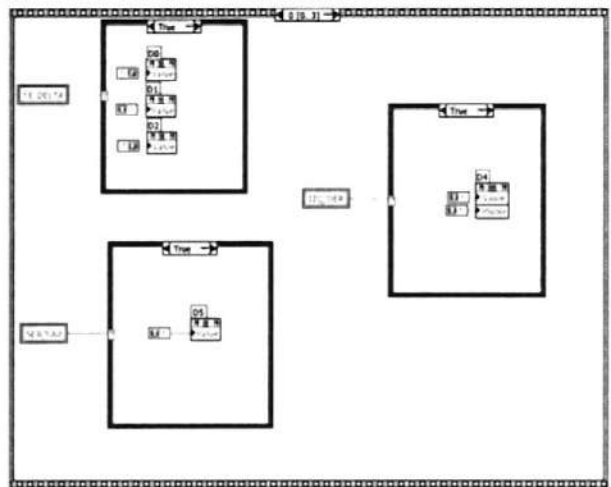
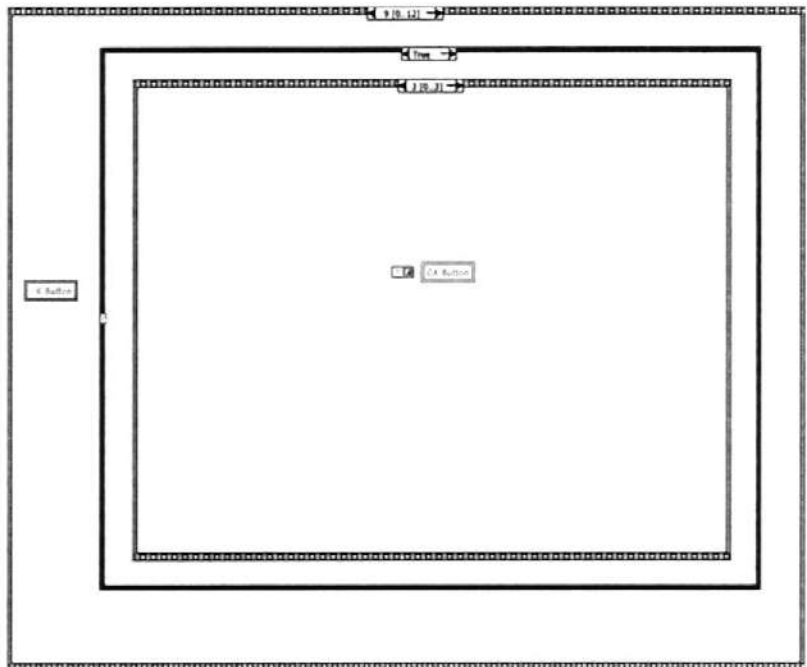


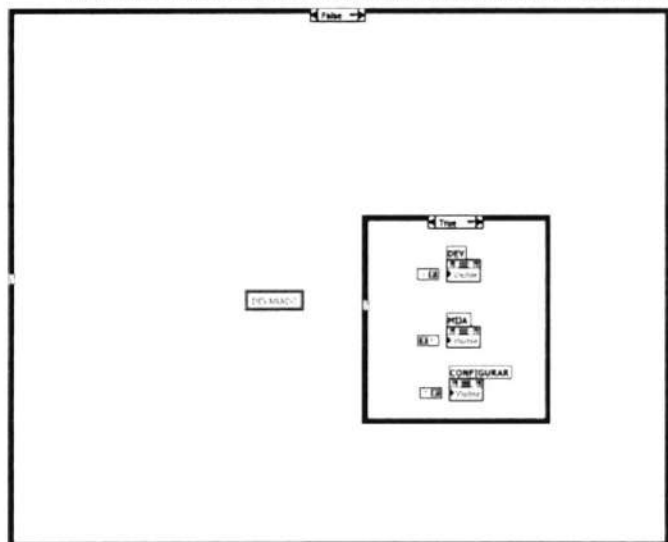
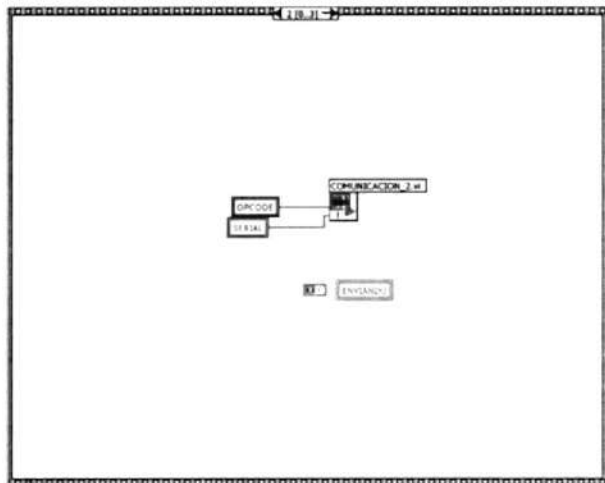
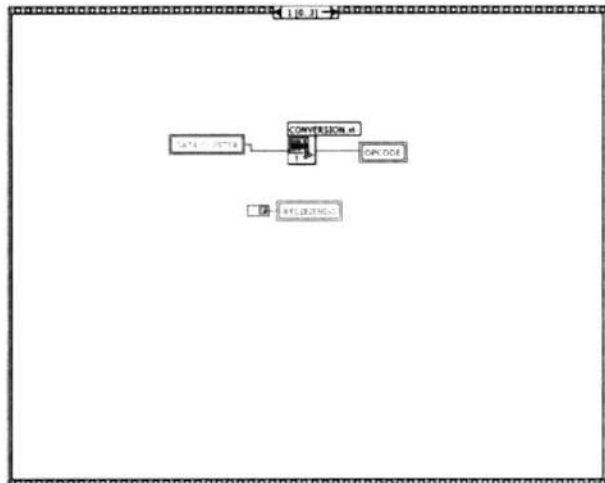


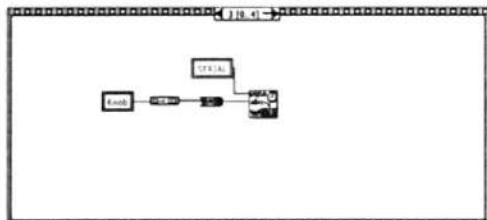
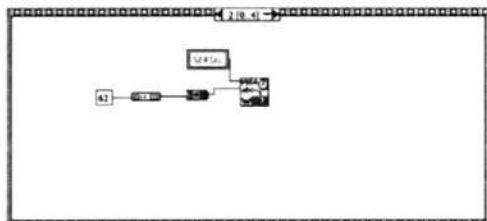
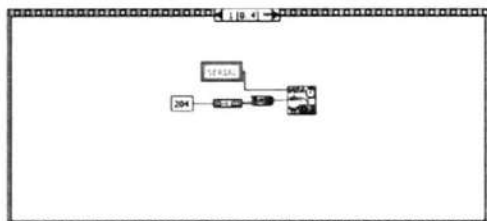
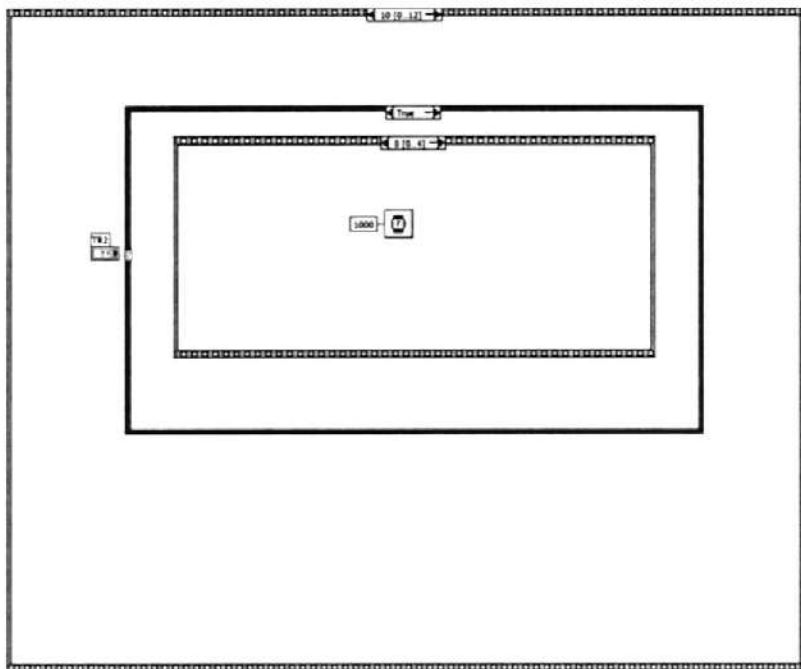
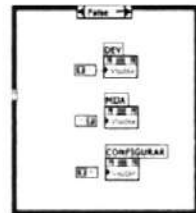
CIB -ESPOL

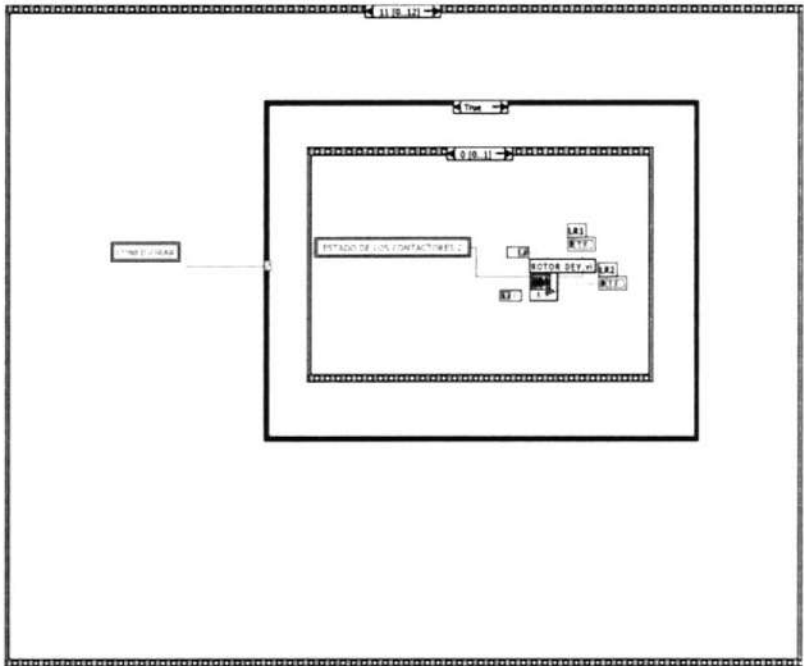
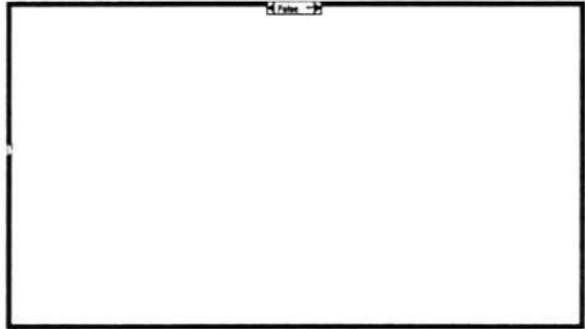
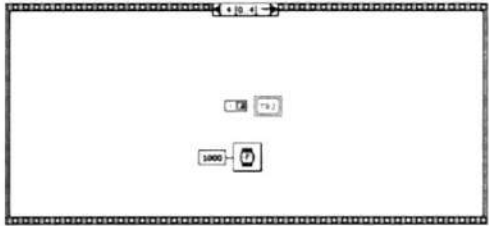


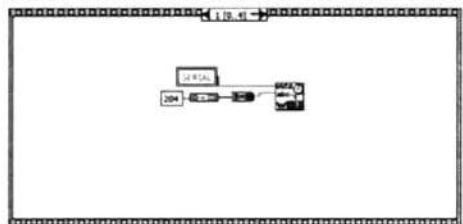
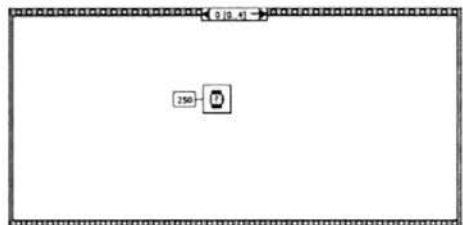
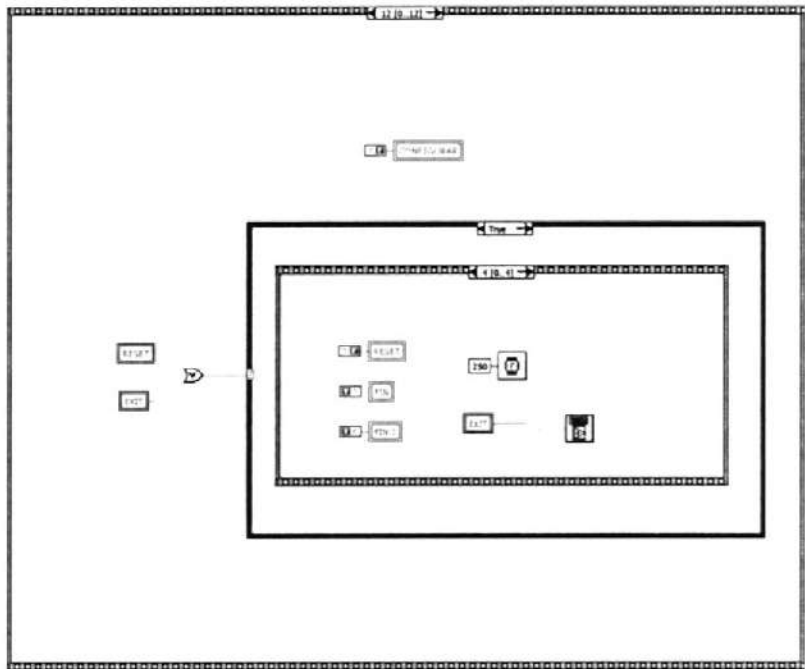
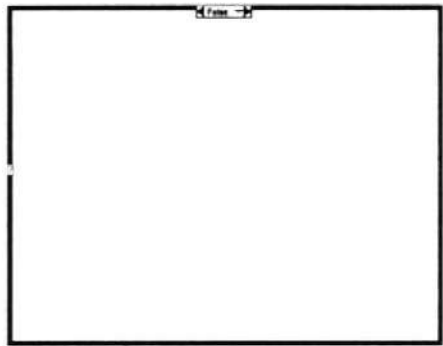
CIB -ESPOL

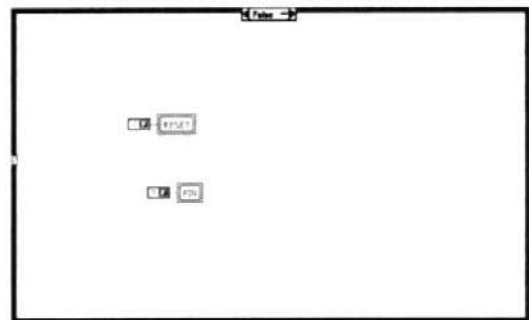
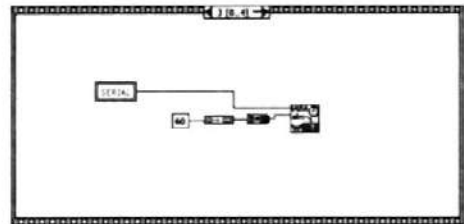
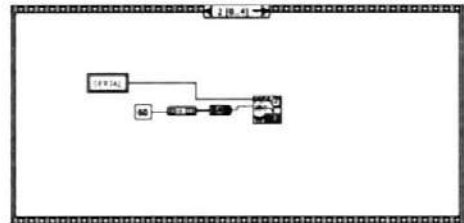




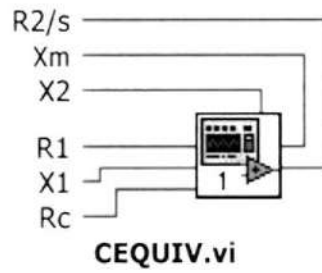




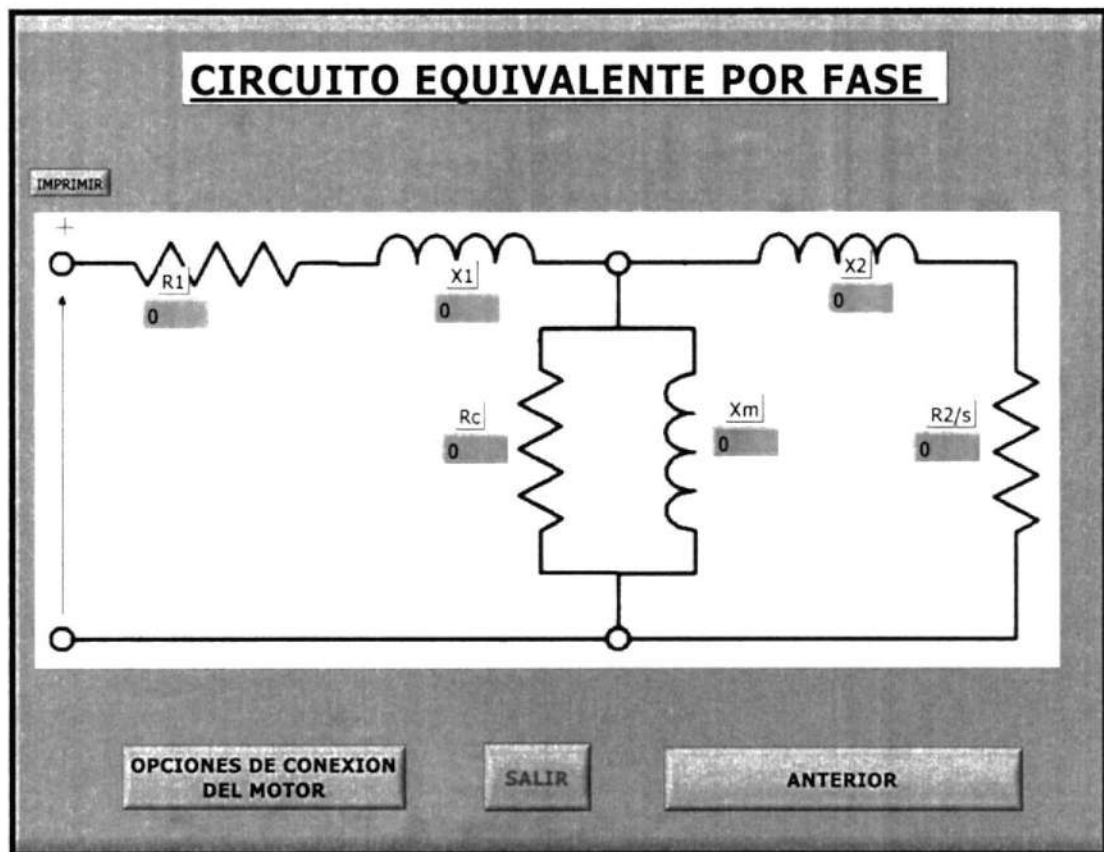




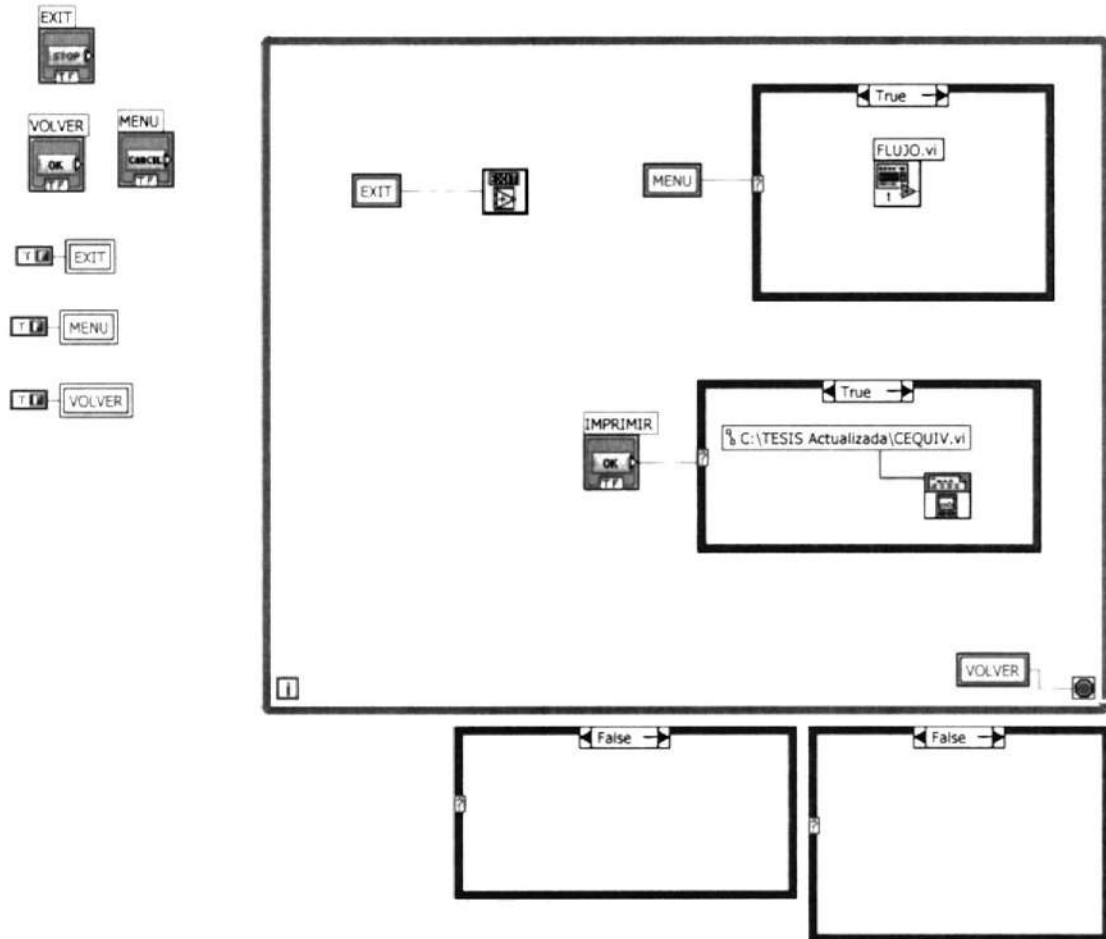
Connector Pane



Front Panel



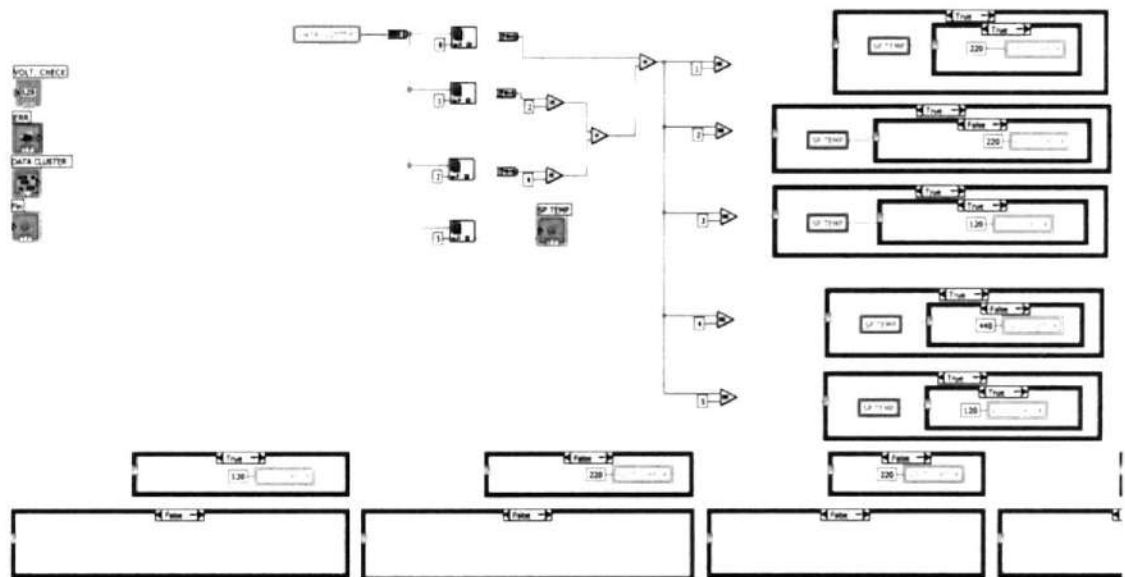
Block Diagram



Connector Pane



Block Diagram



Form with a small input field and a label.

Form with a small input field and a label.

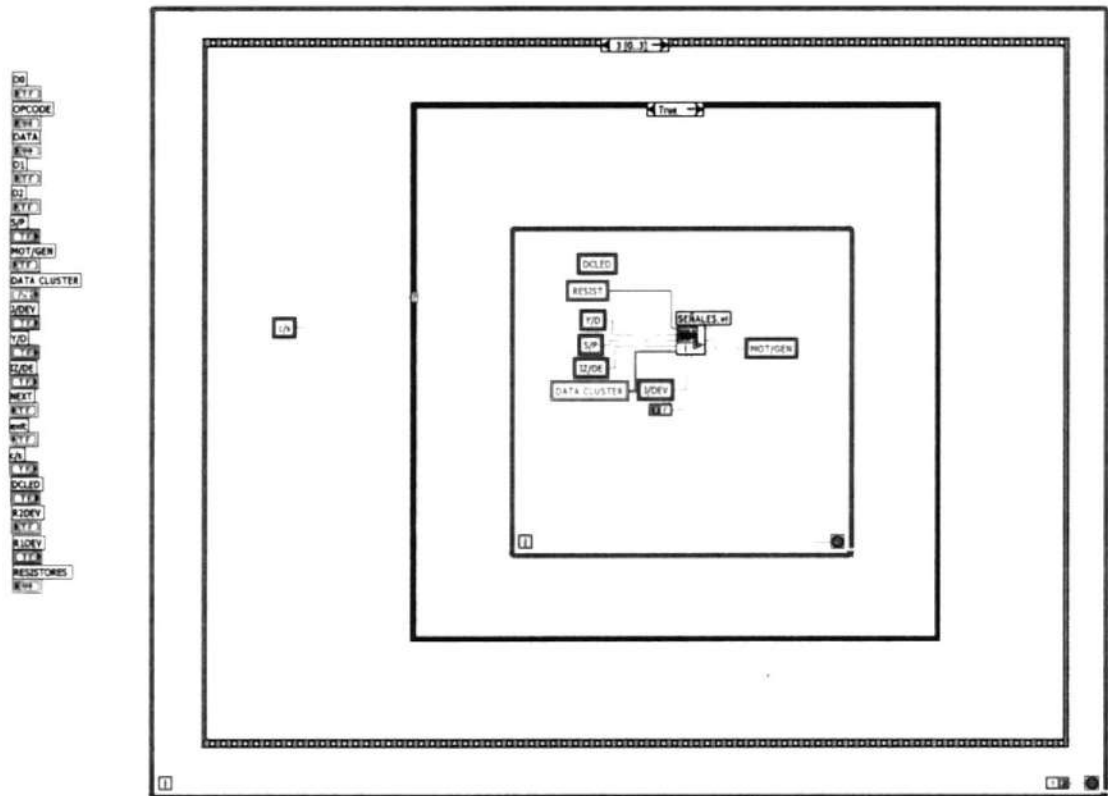
Form with a small input field and a label.

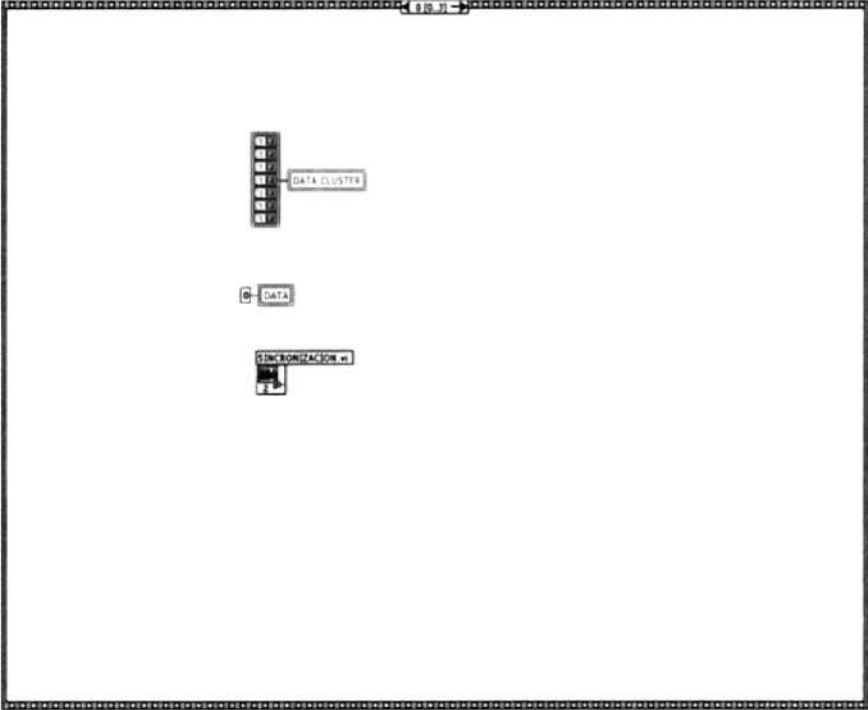
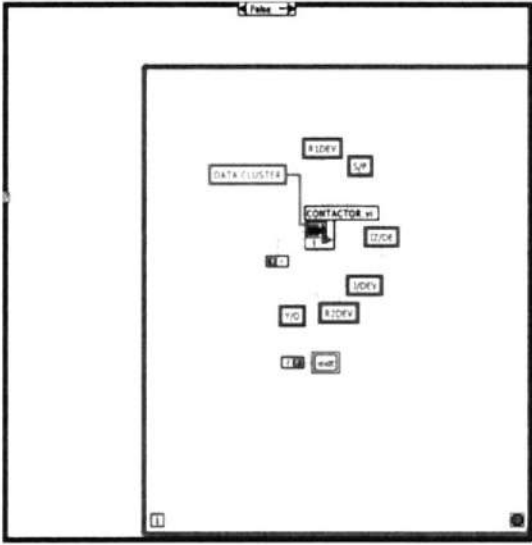
Form with a small input field and a label.

Connector Pane



Block Diagram





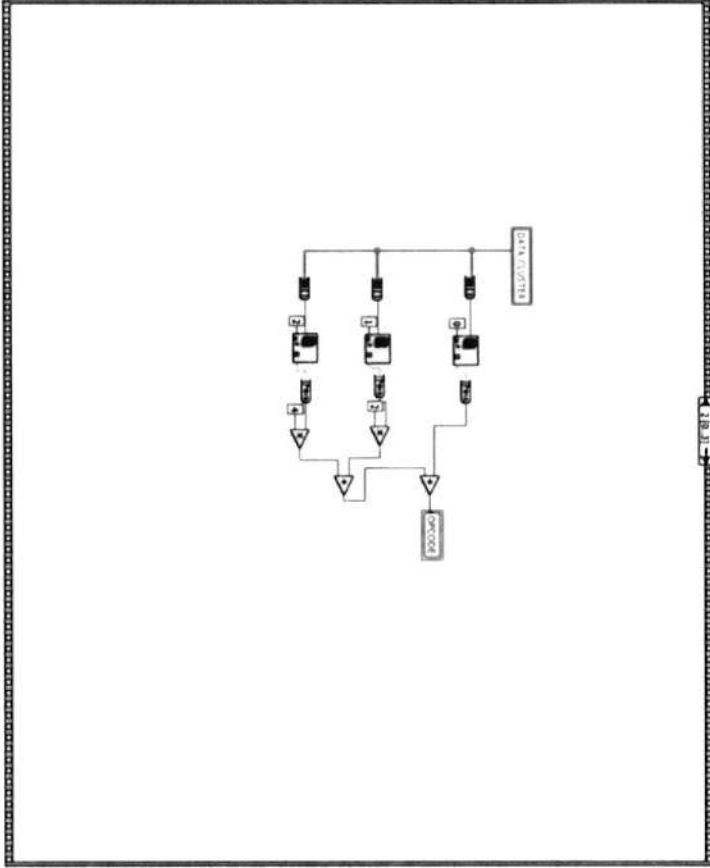


Figure 1

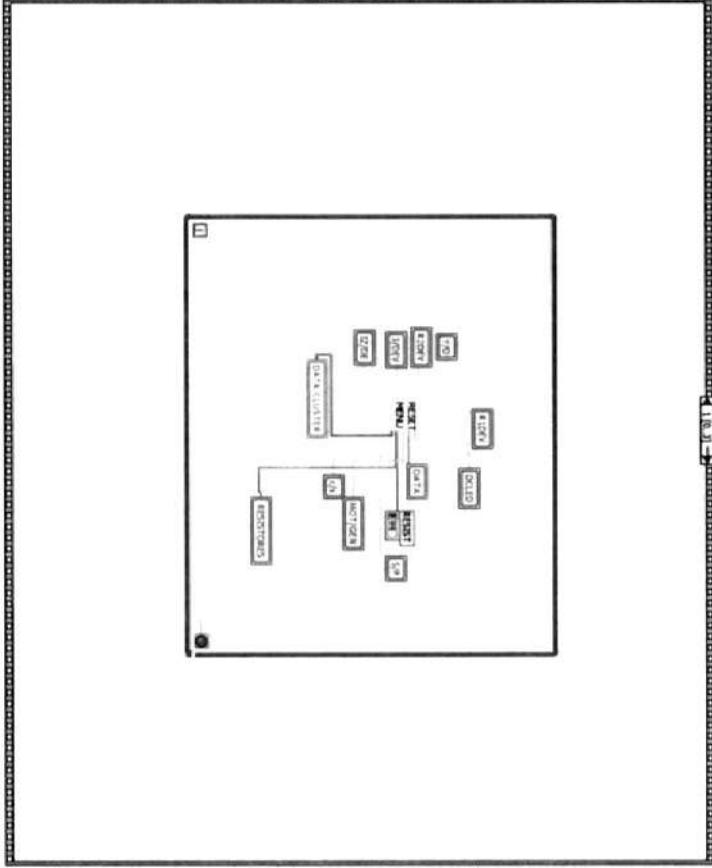
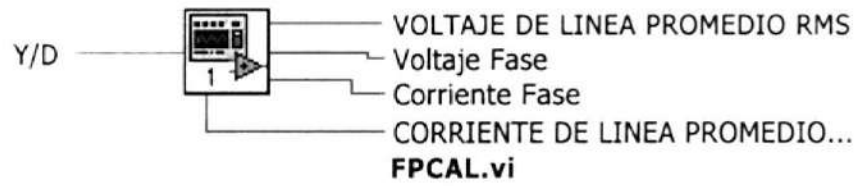
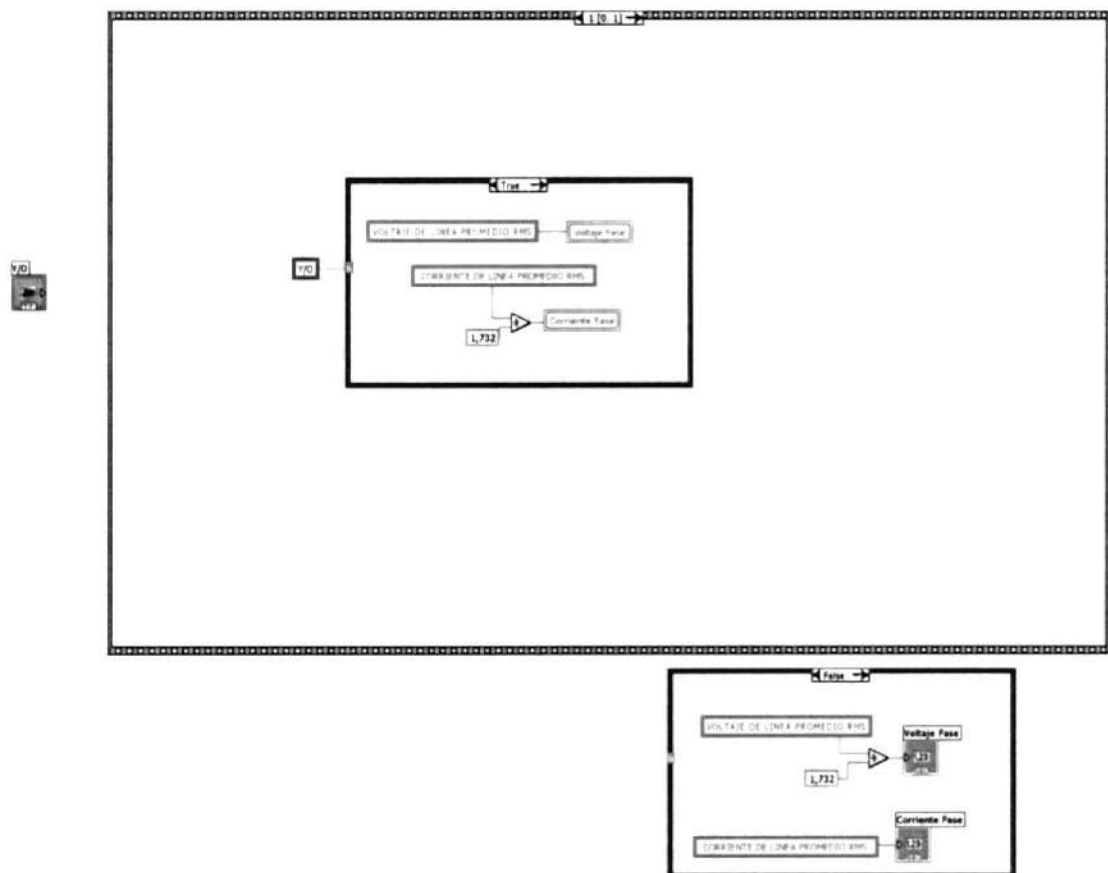


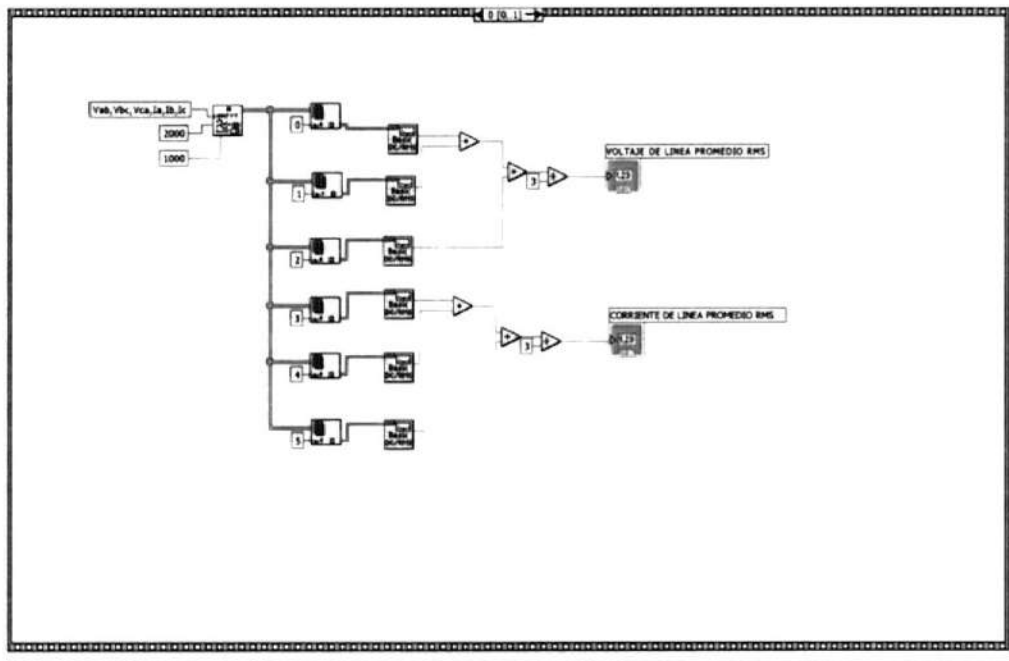
Figure 2

Connector Pane

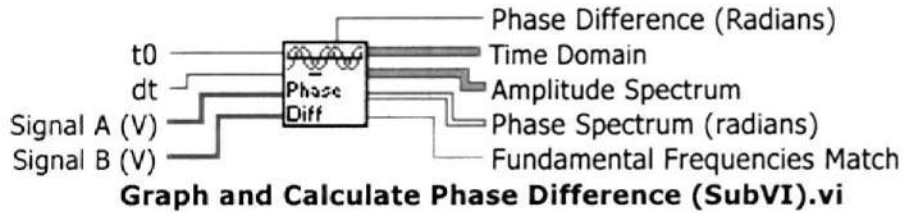


Block Diagram

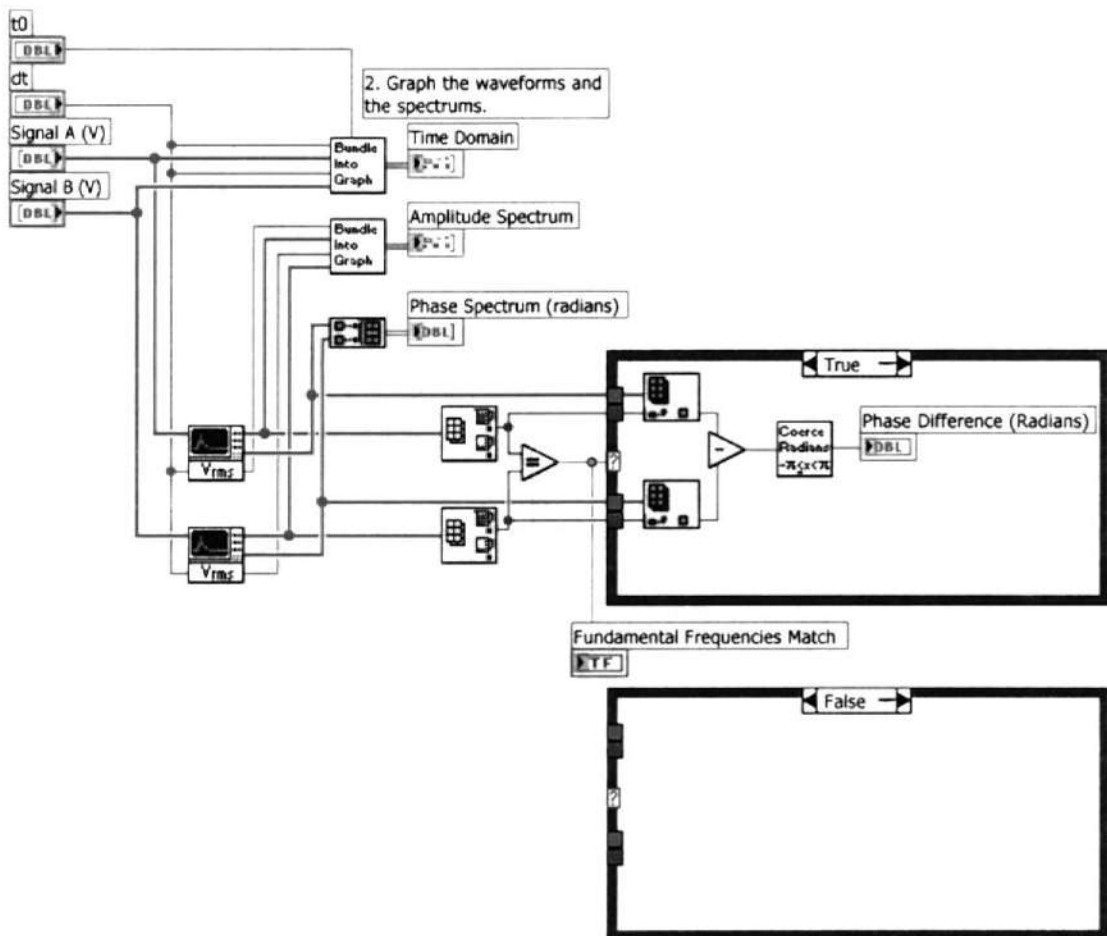




Connector Pane



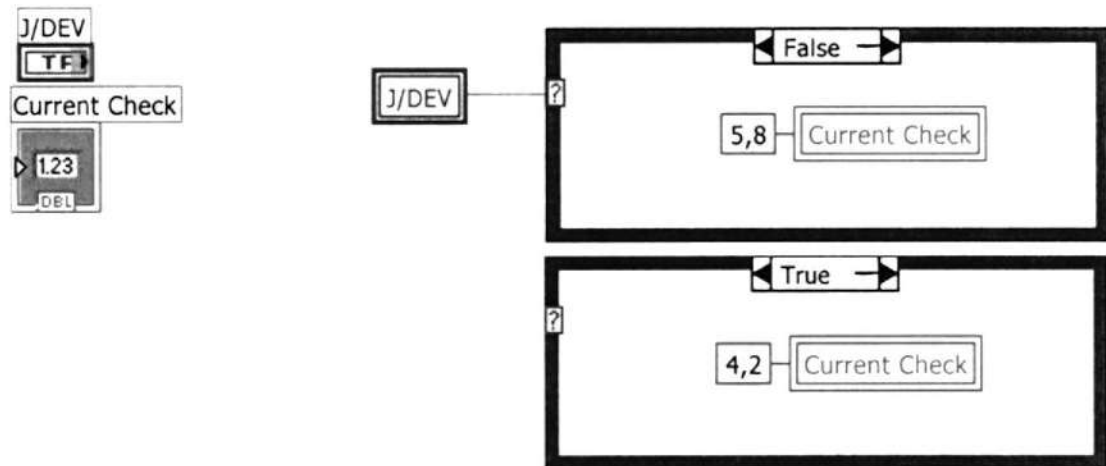
Block Diagram



Connector Pane



Block Diagram

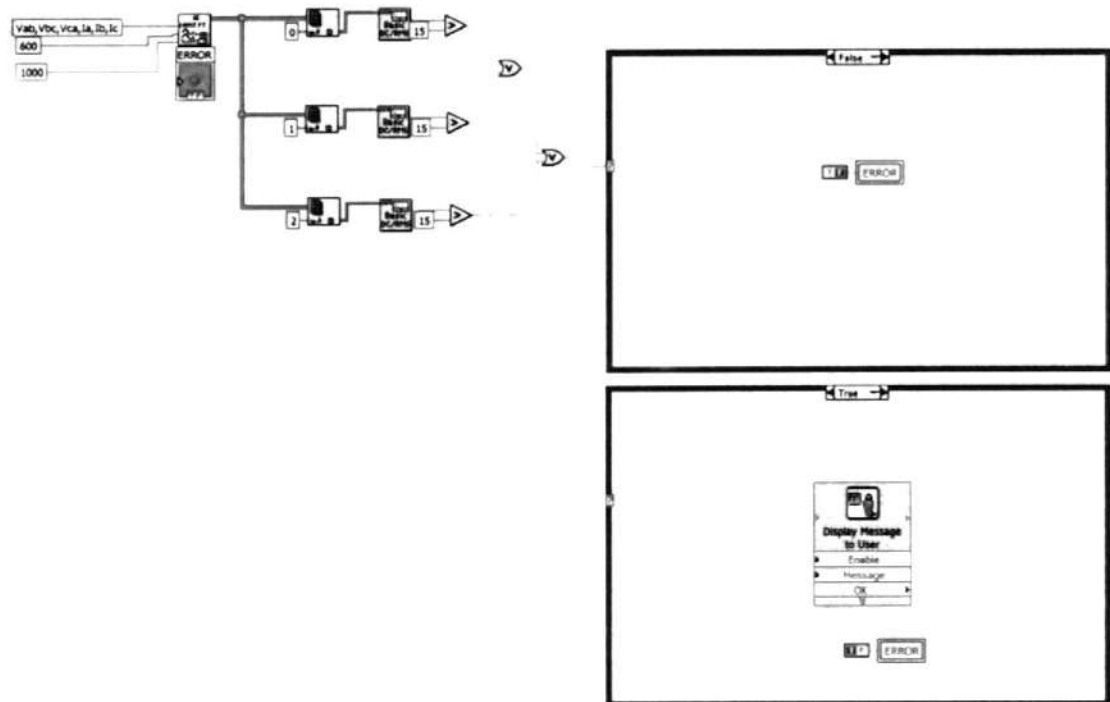


CIB - ESPOL

Connector Pane



Block Diagram



Express VI Configuration Information



Display Message to User

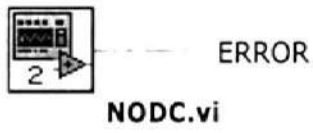
Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

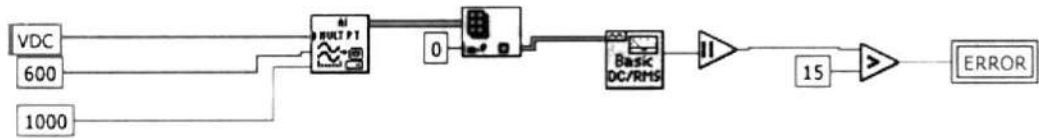
This Express VI is configured as follows:

Message:

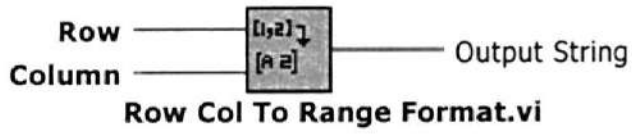
Connector Pane



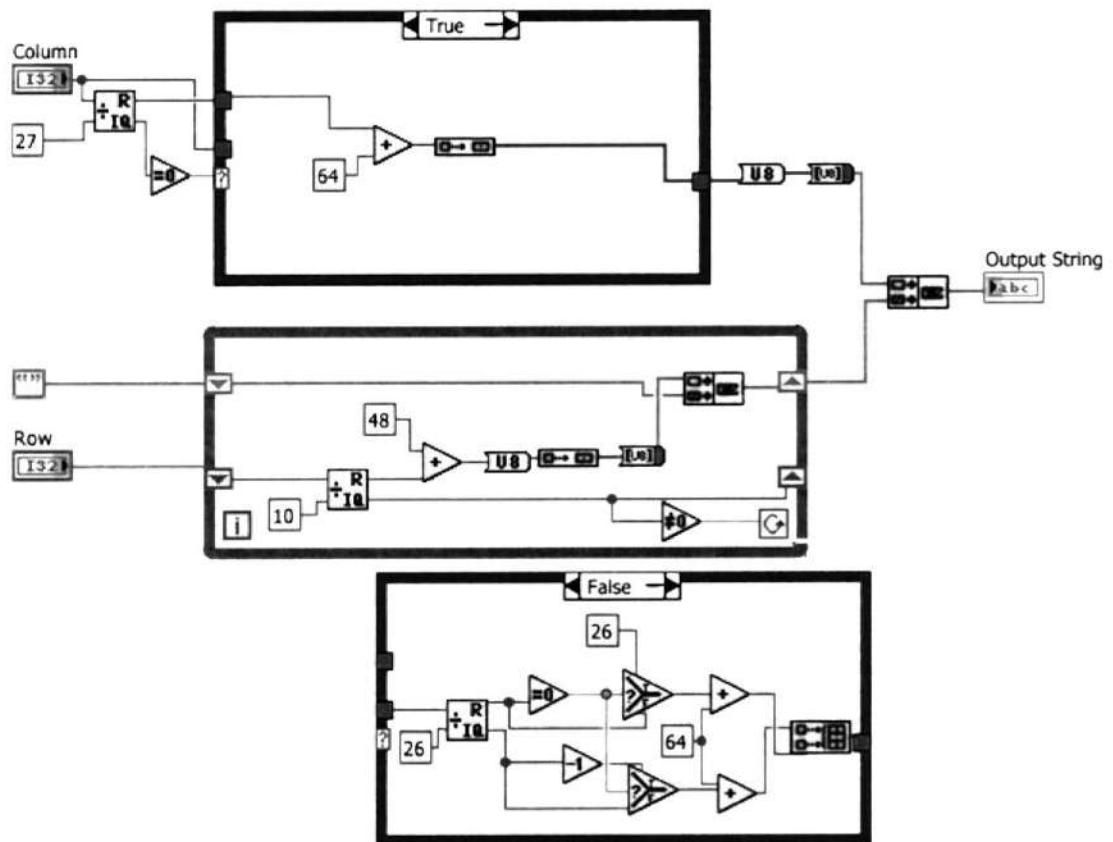
Block Diagram



Connector Pane



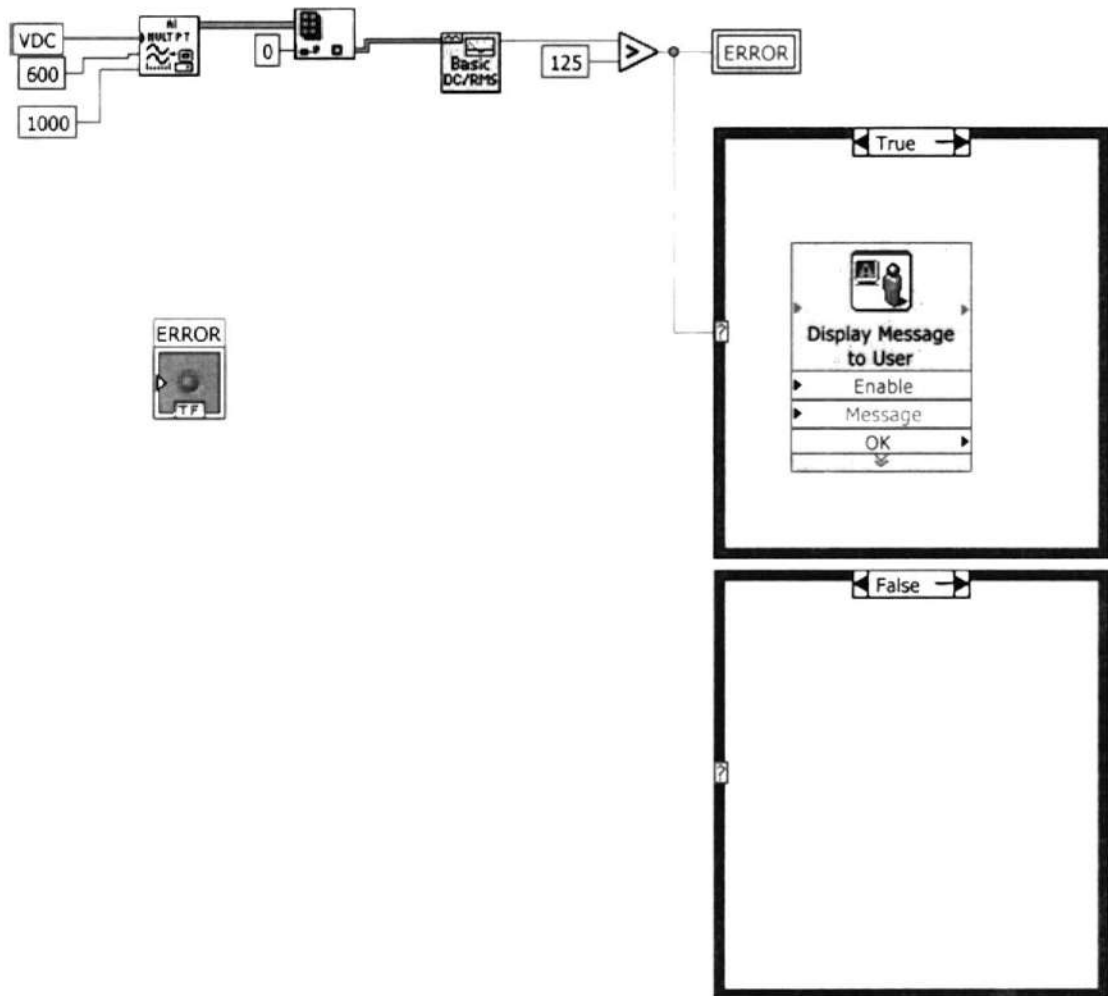
Block Diagram



Connector Pane



Block Diagram



Express VI Configuration Information



Display Message to User

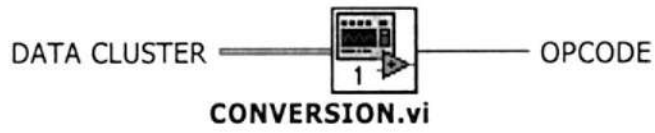
Display Message to User

Displays a standard dialog box that contains an alert or a message for users.

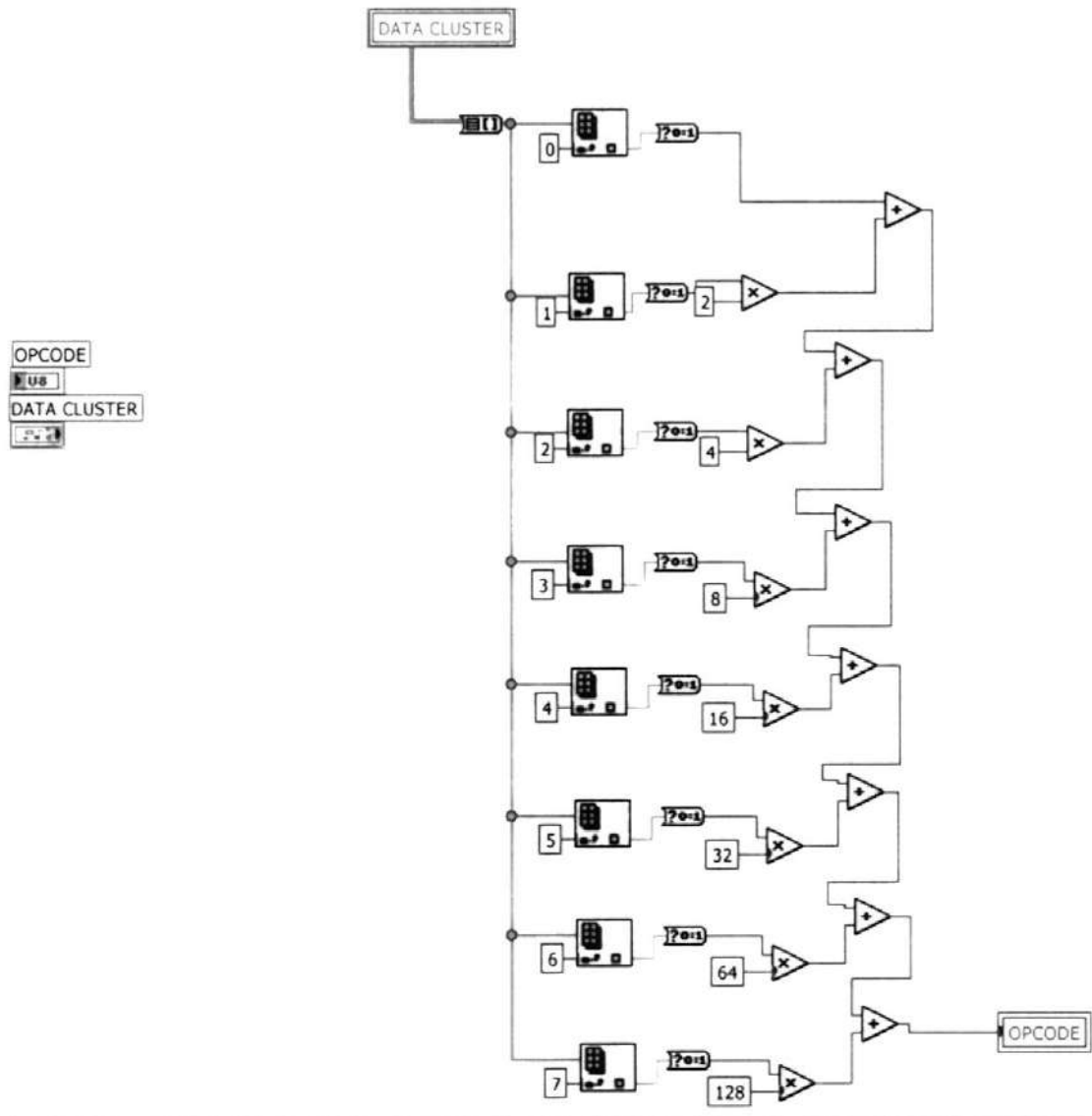
This Express VI is configured as follows:

Message:

Connector Pane



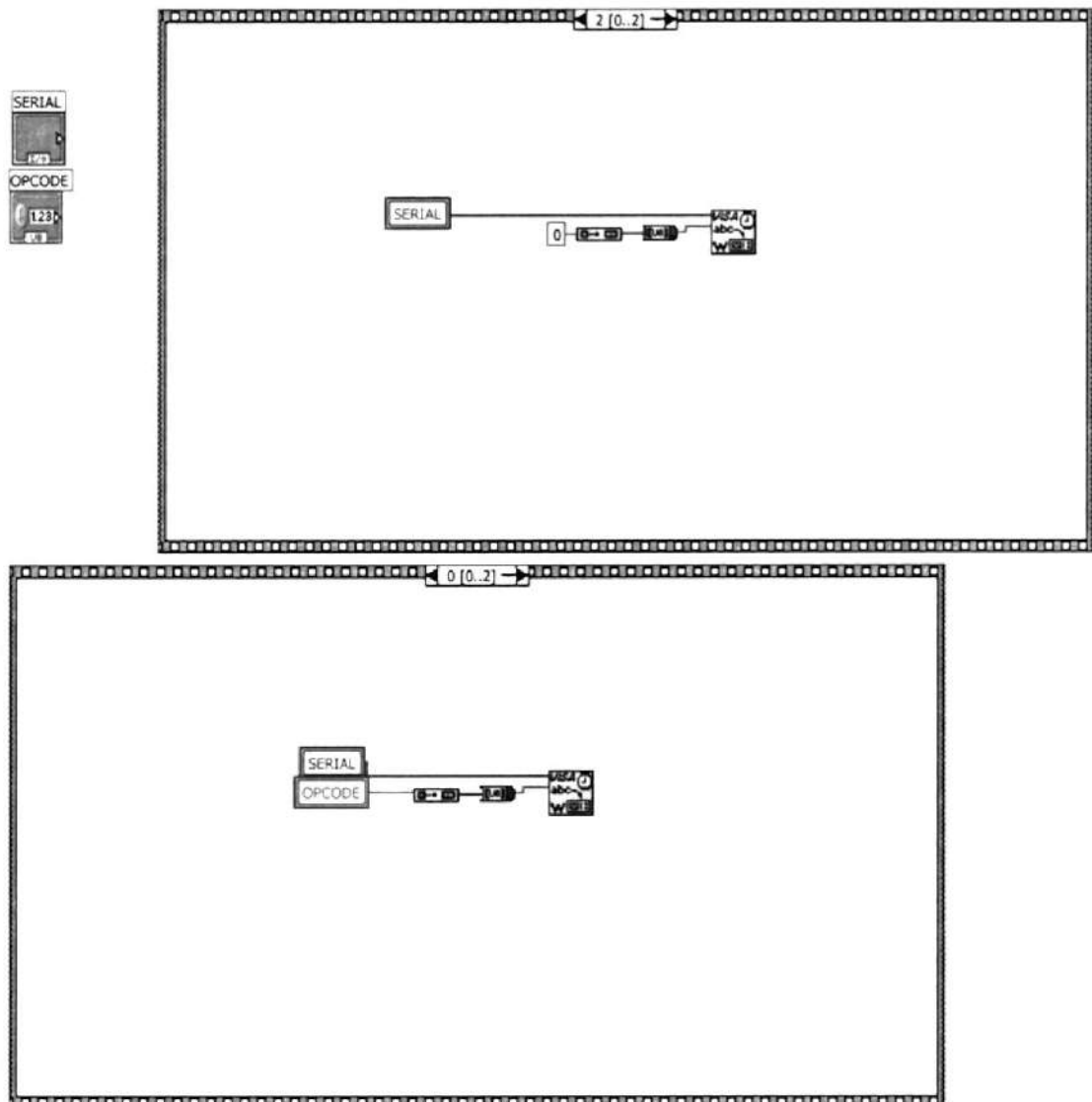
Block Diagram

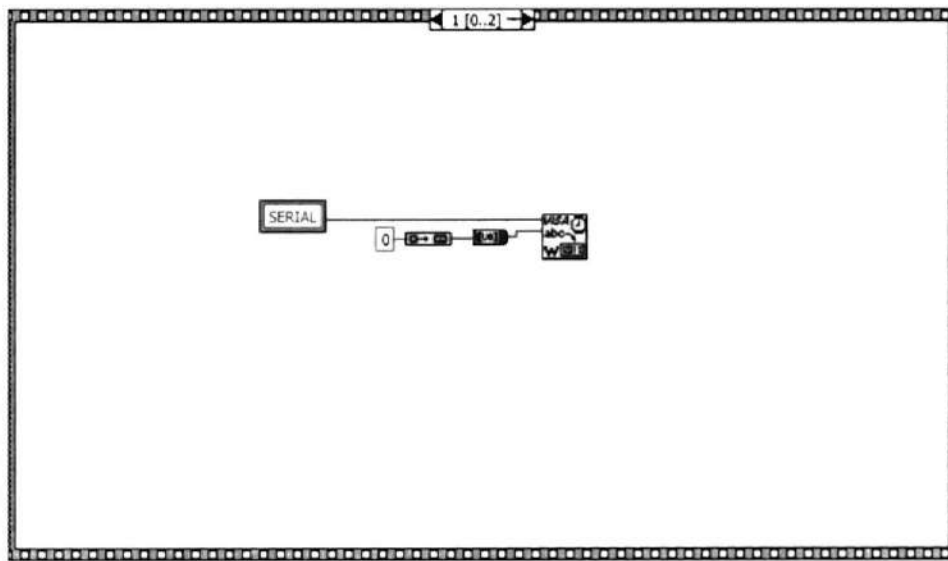


Connector Pane



Block Diagram

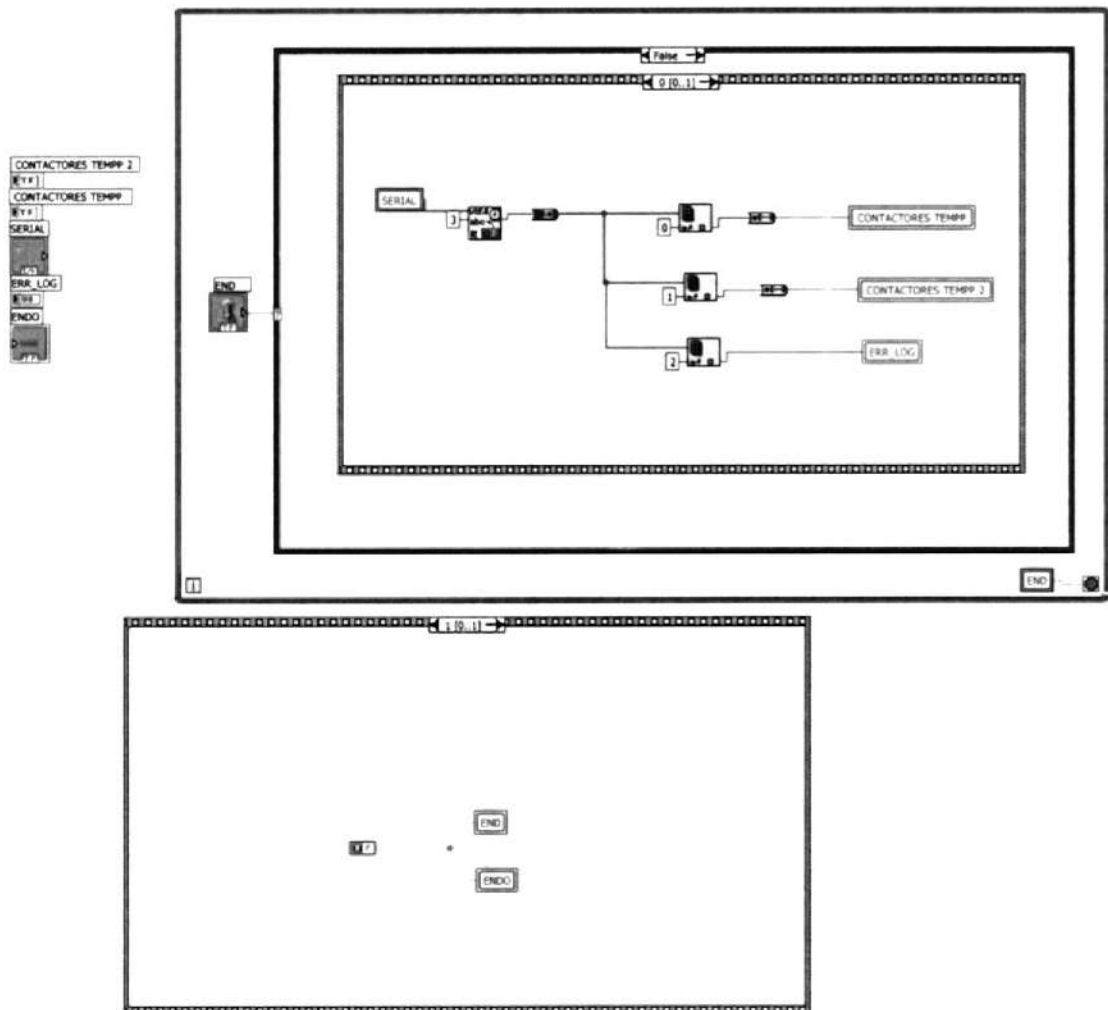


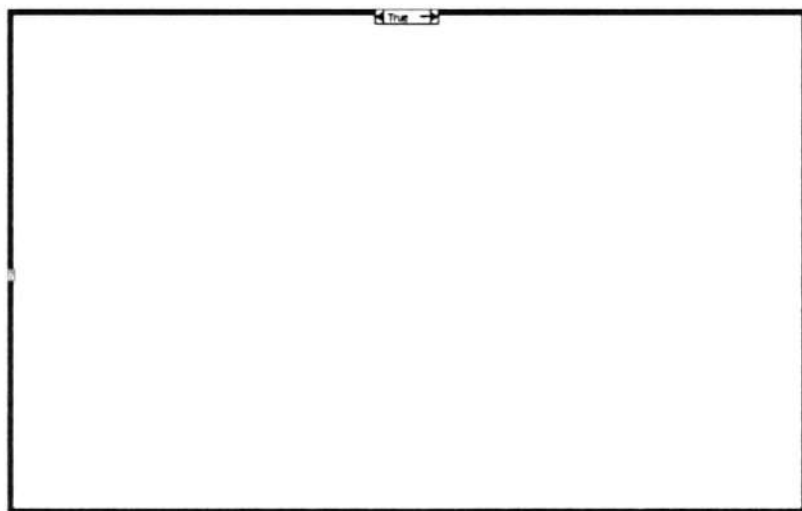


Connector Pane

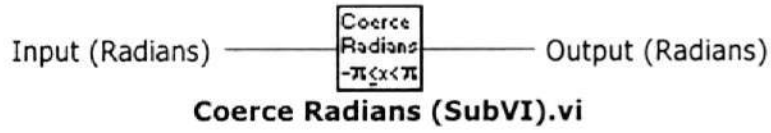


Block Diagram

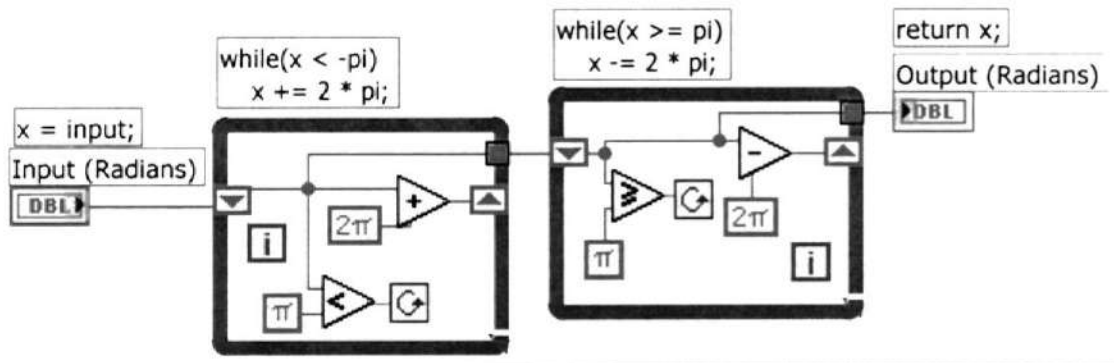




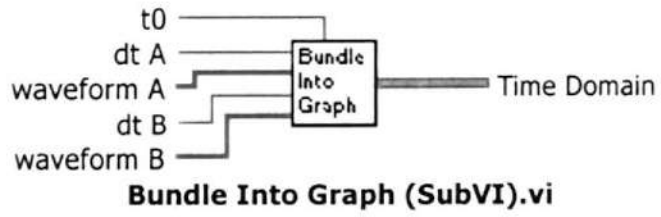
Connector Pane



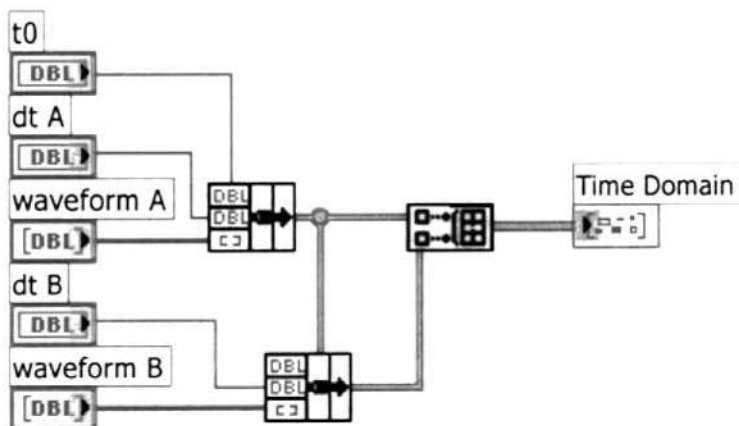
Block Diagram



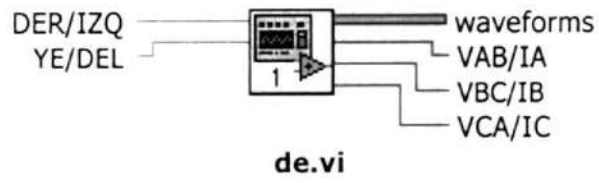
Connector Pane



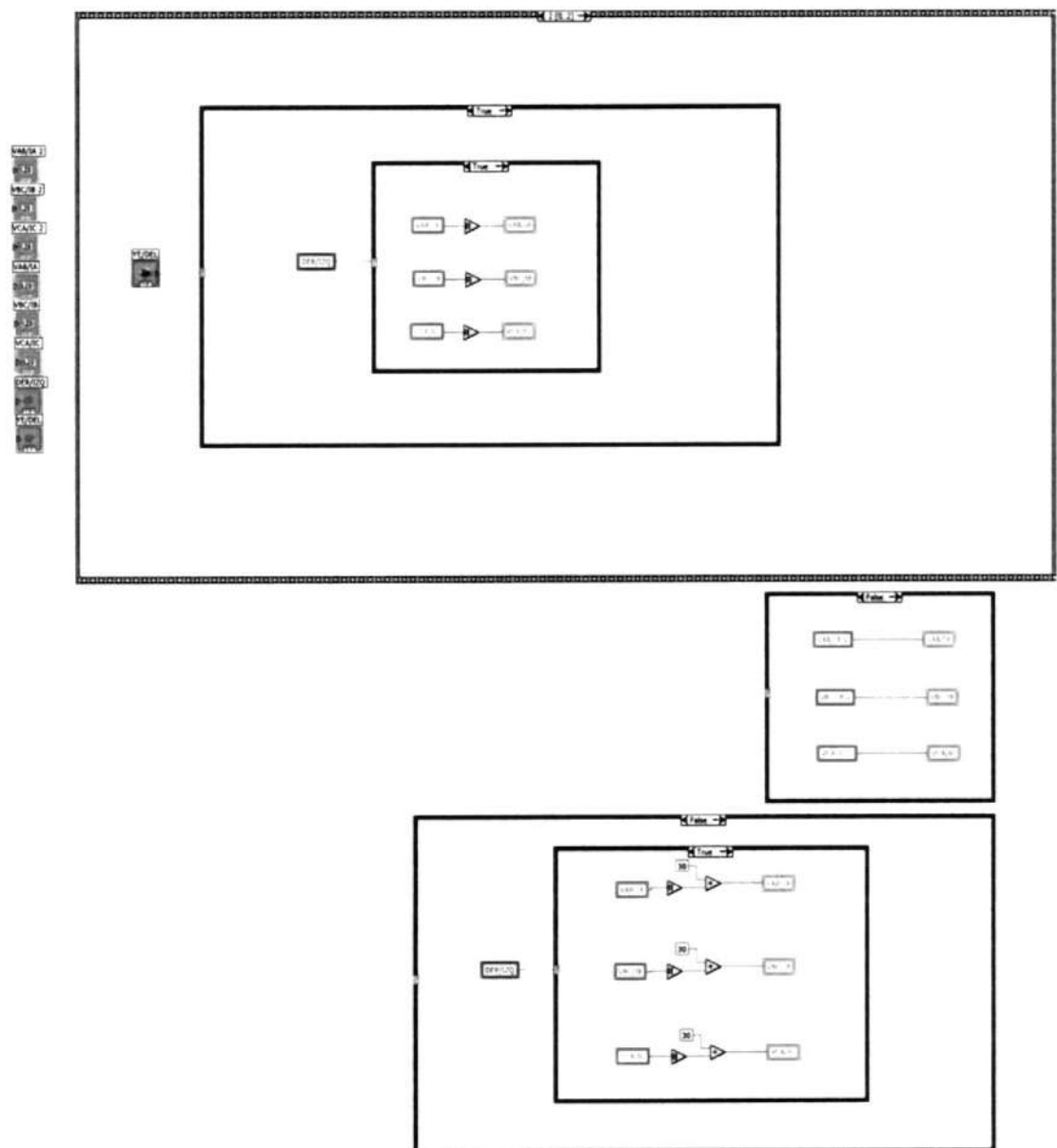
Block Diagram

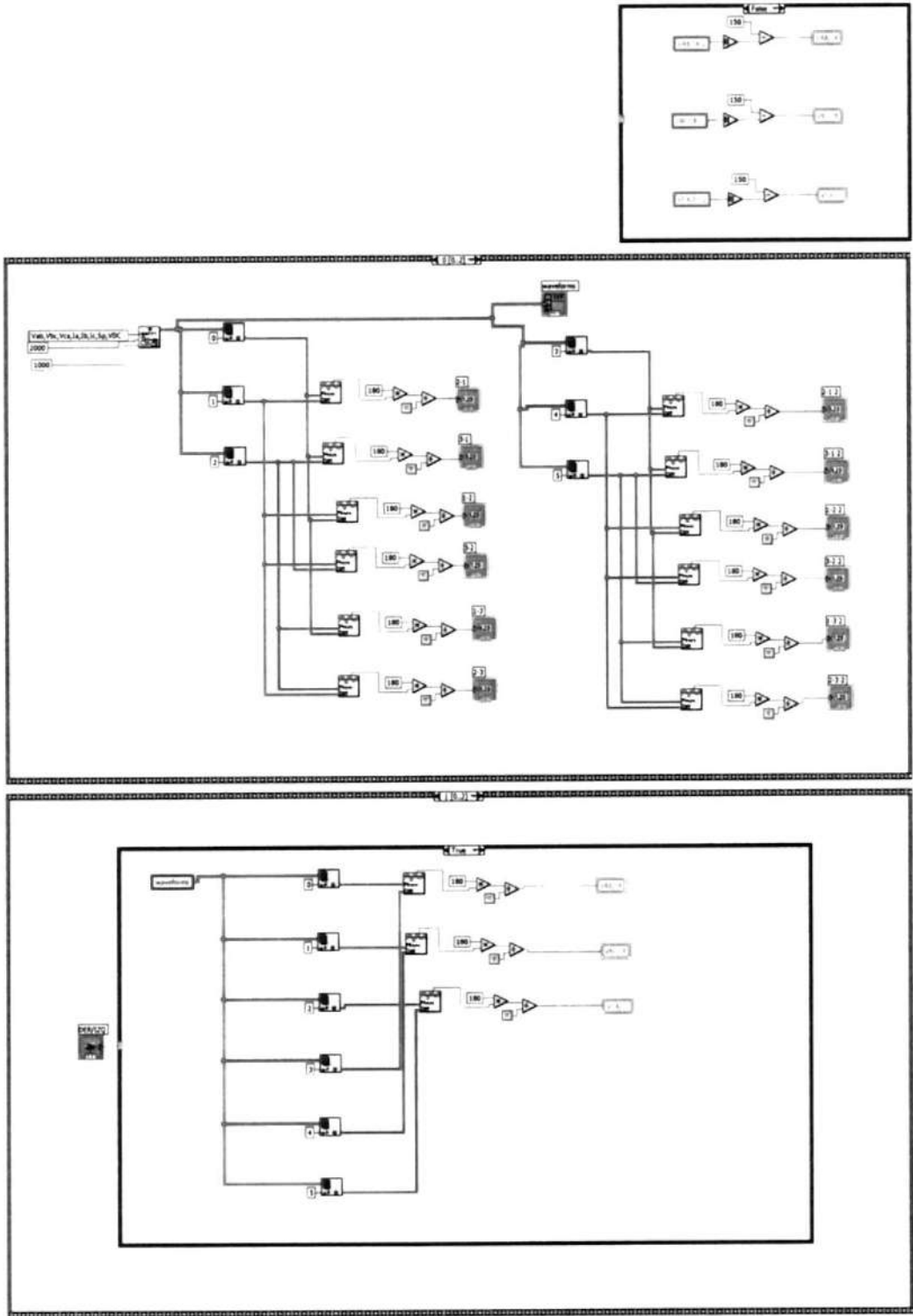


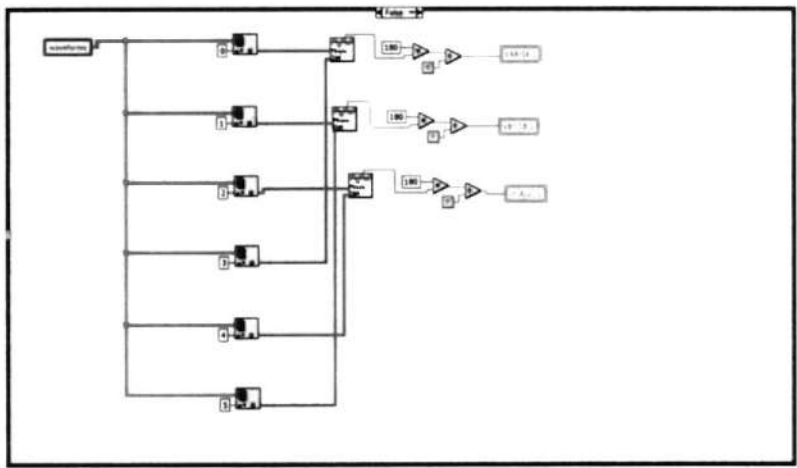
Connector Pane



Block Diagram









CIB -ESPOL

CONCLUSIONES Y RECOMENDACIONES

- 1.- El hardware de National Instruments provee gran precisión y velocidad para el acondicionamiento y adquisición de señales.

- 2.- LabView es una excelente herramienta de programación gráfica, muy versátil y adaptable a cualquier aplicación, permite acceso a los puertos del computador de una forma sencilla y confiable.

- 3.- Los microcontroladores pueden controlar confiablemente un sistema siempre y cuando su programación sea correcta y su circuito impreso sea de calidad, respetando las normas del diseño de circuitos impresos.

- 4.- La fase de prueba de un circuito en proto-board no debe alargarse demasiado, ya que el mismo proto-board introduce errores creando confusión y pérdida de tiempo en el desarrollo de un circuito.

5.- Los estudiantes se beneficiarán de este proyecto ya que podrán analizar profundamente la respuesta de las máquinas KATO, revisando las señales adquiridas y los circuitos equivalentes.

6.- Se aconseja que los estudiantes utilicen el sistema después de realizar las prácticas de forma manual, de esta forma tendrán más elementos de juicio para entender la información que muestran las pantallas de LabView.

7.- Se recomienda que los estudiantes utilicen los comandos de operación y de configuración de la tarjeta de control para realizar sus propios proyectos, en cualquier lenguaje de programación utilizando el puerto serial.

BIBLIOGRAFÍA

1. Chapman , Stephen, Máquinas Eléctricas, Segunda Edición, Grupo Editorial Mc. Graw-Hill.
2. Página oficial de National Instruments, [http//ni.com](http://ni.com)
3. Página oficial de Microchip, [http//microchip.com](http://microchip.com)

ANEXO A

HOJAS DE DATOS DE LOS INTEGRADOS

UTILIZADOS

MAXIM

+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

General Description

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

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

Applications

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

Features

Superior to Bipolar

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

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSF	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μ F)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	4.7/10	No		120	Ultra-low-power, industry standard pinout
MAX222	+5	2/2	4	0.1	Yes		200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 + receivers active in shutdown
MAX225	+5	5/5	0	-	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes		120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No		120	Standard +5/+12V or battery supplies, same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No		120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No		200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	-	No		120	No external caps
MAX233A	+5	2/2	0	-	No		200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No		120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	-	Yes		120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes		120	Shutdown, three state
MAX237 (MAX207)	+5	5/0	4	1.0 (0.1)	No		120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No		120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No		120	Standard +5/+12V or battery supplies, single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes		120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes		120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No		200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No		120	High slew rate
MAX245	+5	8/10	0	-	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	-	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	-	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

MAXIM

Maxim Integrated Products 1

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800

MAX220-MAX249

+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	696mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
V _{IN}	-0.3V to (V _{CC} + 0.3V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
R _{IN}	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	800mW
T _{OUT} (Note 1)	±15V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
Output Voltages		16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
T _{OUT}	±15V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
R _{OUT}	-0.3V to (V _{CC} + 0.3V)		
Driver/Receiver Output Short-Circuited to GND	Continuous	Operating Temperature Ranges	
Continuous Power Dissipation (T _A = +70°C)		MAX2_ _AC_ _ MAX2_ _C_ _	0°C to +70°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	MAX2_ _AE_ _ MAX2_ _E_ _	-40°C to +85°C
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	889mW	MAX2_ _AM_ _ MAX2_ _M_ _	-55°C to +125°C
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW	Storage Temperature Range	-65°C to +160°C
		Lead Temperature (soldering, 10sec)	+300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, $\overline{\text{SHDN}}$ or V_{CC} = 0V

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C₁-C₄ = 0.1μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Pull-Up/Input Current	Normal operation			5	40	μA
	$\overline{\text{SHDN}} = 0V$, MAX222/242, shutdown			±0.01	±1	μA
Output Leakage Current	V _{CC} = 5.5V, $\overline{\text{SHDN}} = 0V$, V _{OUT} = ±15V, MAX222/242			±0.01	±10	μA
	V _{CC} = $\overline{\text{SHDN}} = 0V$, V _{OUT} = ±15V			±0.01	±10	μA
Data Rate	Except MAX220, normal operation			200	116	kbits/sec
	MAX220			22	20	sec
Transmitter Output Resistance	V _{CC} = V ₊ = V ₋ = 0V, V _{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V _{OUT} = 0V		±7	±22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V _{CC} = 5V	Except MAX243 R _{2IN}	0.8	1.3		V
		MAX243 R _{2IN} (Note 2)	-3			V
RS-232 Input Threshold High	V _{CC} = 5V	Except MAX243 R _{2IN}		1.8	2.4	V
		MAX243 R _{2IN} (Note 2)		-0.5	-0.1	V
RS-232 Input Hysteresis	Except MAX243, V _{CC} = 5V, no hyst. in shdn.		0.2	0.5	1	V
	MAX243			1		V
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I _{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I _{OUT} = -1.0mA		3.5	V _{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V _{OUT} = GND		-2	-10		mA
	Sinking V _{OUT} = V _{CC}		10	30		mA
TTL/CMOS Output Leakage Current	$\overline{\text{SHDN}} = V_{CC}$ or $\overline{\text{EN}} = V_{CC}$ ($\overline{\text{SHDN}} = 0V$ for MAX222), 0V ≤ V _{OUT} ≤ V _{CC}			±0.05	±10	μA

+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

MAX220-MAX249
ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)
(V_{CC} = +5V ±10%, C1-C4 = 0.1μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

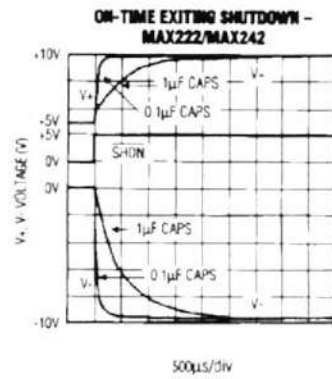
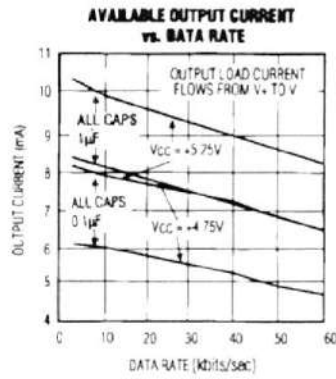
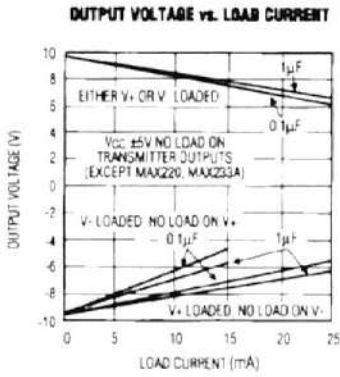
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS	
EN Input Threshold Low	MAX242			1.4	0.8	V	
EN Input Threshold High	MAX242		2.0	1.4		V	
POWER SUPPLY							
Operating Supply Voltage			4.5		5.5	V	
V _{CC} Supply Current (SHDN = V _{CC}), Figures 5, 6, 9, 19	No load	MAX220		0.5	2	mA	
		MAX222/232A/233A/242/243		4	10		
	3kΩ load both inputs	MAX220		12			
		MAX222/232A/233A/242/243		15			
Shutdown Supply Current	MAX222/242	T _A = +25°C		0.1	10	μA	
		T _A = 0° to +70°C		2	50		
		T _A = -40° to +85°C		2	50		
		T _A = -55° to +125°C		35	100		
SHDN Input Leakage Current	MAX222/242				±1	μA	
SHDN Threshold Low	MAX222/242			1.4	0.8	V	
SHDN Threshold High	MAX222/242		2.0	1.4		V	
AC CHARACTERISTICS							
Transition Slew Rate		C _L = 50pF to 2500pF, R _L = 3kΩ to 7kΩ, V _{CC} = 5V, T _A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/μs
		MAX220	1.5	3	30		
Transmitter Propagation Delay TLL to RS-232 (normal operation), Figure 1	I _{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	μs	
		MAX220		4	10		
	I _{PLHT}	MAX222/232A/233A/242/243		1.5	3.5		
		MAX220		5	10		
Receiver Propagation Delay RS-232 to TLL (normal operation), Figure 2	I _{PHLR}	MAX222/232A/233A/242/243		0.5	1	μs	
		MAX220		0.6	3		
	I _{PLHR}	MAX222/232A/233A/242/243		0.6	1		
		MAX220		0.8	3		
Receiver Propagation Delay RS-232 to TLL (shutdown), Figure 2	I _{PHLS}	MAX242		0.5	10	μs	
	I _{PLHS}	MAX242		2.5	10		
Receiver-Output Enable Time, Figure 3	I _{ER}	MAX242		125	500	ns	
Receiver-Output Disable Time, Figure 3	I _{DR}	MAX242		160	500	ns	
Transmitter-Output Enable Time (SHDN goes high), Figure 4	I _{ET}	MAX222/242, 0.1μF caps (includes charge-pump start-up)		250		μs	
Transmitter-Output Disable Time (SHDN goes low), Figure 4	I _{DT}	MAX222/242, 0.1μF caps		600		ns	
Transmitter + to - Propagation Delay Difference (normal operation)	I _{PHLT} - I _{PLHT}	MAX222/232A/233A/242/243		300		ns	
		MAX220		2000			
Receiver + to - Propagation Delay Difference (normal operation)	I _{PHLR} - I _{PLHR}	MAX222/232A/233A/242/243		100		ns	
		MAX220		225			

Note 2: MAX243 R_{2OUT} is guaranteed to be low when R_{2IN} is ≥ 0V or is floating.

+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243



4N25
4N37

4N26
H11A1

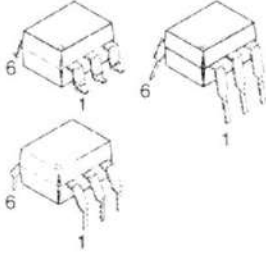
4N27
H11A2

4N28
H11A3

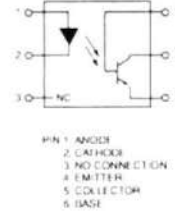
4N35
H11A4

4N36
H11A5

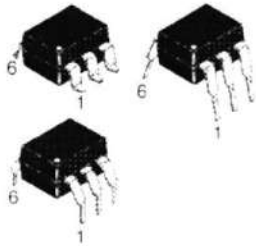
WHITE PACKAGE (-M SUFFIX)



SCHEMATIC



BLACK PACKAGE (NO -M SUFFIX)



DESCRIPTION

The general purpose optocouplers consist of a gallium arsenide infrared emitting diode driving a silicon phototransistor in a 6-pin dual in-line package.

FEATURES

- Also available in white package by specifying -M suffix, eg. 4N25-M
- UL recognized (File # E90700)
- VDE recognized (File # 94766)
 - Add option V for white package (e.g., 4N25V-M)
 - Add option 300 for black package (e.g., 4N25.300)

APPLICATIONS

- Power supply regulators
- Digital logic inputs
- Microprocessor inputs

4N25
4N37

4N26
H11A1

4N27
H11A2

4N28
H11A3

4N35
H11A4

4N36
H11A5

ABSOLUTE MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise specified)

Parameter	Symbol	Value	Units
TOTAL DEVICE			
Storage Temperature	T_{STG}	-55 to +150	$^\circ\text{C}$
Operating Temperature	T_{OPR}	-55 to +100	$^\circ\text{C}$
Wave solder temperature (see page 14 for reflow solder profiles)	T_{SOL}	260 for 10 sec	$^\circ\text{C}$
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	250 3.3 (non-M), 2.94 (-M)	mW
EMITTER			
DC/Average Forward Input Current	I_F	100 (non-M), 60 (-M)	mA
Reverse Input Voltage	V_R	6	V
Forward Current - Peak (300 μs , 2% Duty Cycle)	$I_F(\text{pk})$	3	A
LED Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	150 (non-M), 120 (-M) 2.0 (non-M), 1.41 (-M)	mW mW/ $^\circ\text{C}$
DETECTOR			
Collector-Emitter Voltage	V_{CEO}	30	V
Collector-Base Voltage	V_{CBO}	70	V
Emitter-Collector Voltage	V_{ECO}	7	V
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	150 2.0 (non-M), 1.76 (-M)	mW mW/ $^\circ\text{C}$



CIB - ESPOL

4N25
4N37

4N26
H11A1

4N27
H11A2

4N28
H11A3

4N35
H11A4

4N36
H11A5

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise specified)

INDIVIDUAL COMPONENT CHARACTERISTICS

Parameter	Test Conditions	Symbol	Min	Typ*	Max	Unit
EMITTER						
Input Forward Voltage	($I_F = 10\text{ mA}$)	V_F		1.18	1.50	V
Reverse Leakage Current	($V_R = 6.0\text{ V}$)	I_R		0.001	10	μA
DETECTOR						
Collector-Emitter Breakdown Voltage	($I_C = 1.0\text{ mA}$, $I_F = 0$)	BV_{CEO}	30	100		V
Collector-Base Breakdown Voltage	($I_C = 100\text{ }\mu\text{A}$, $I_F = 0$)	BV_{CBO}	70	120		V
Emitter-Collector Breakdown Voltage	($I_E = 100\text{ }\mu\text{A}$, $I_F = 0$)	BV_{ECO}	7	10		V
Collector-Emitter Dark Current	($V_{CE} = 10\text{ V}$, $I_F = 0$)	I_{CEO}		1	50	nA
Collector-Base Dark Current	($V_{CB} = 10\text{ V}$)	I_{CBO}			20	nA
Capacitance	($V_{CE} = 0\text{ V}$, $f = 1\text{ MHz}$)	C_{CE}		8		pF

ISOLATION CHARACTERISTICS

Characteristic	Test Conditions	Symbol	Min	Typ*	Max	Units
Input-Output Isolation Voltage	(Non '-M', Black Package) ($f = 60\text{ Hz}$, $t = 1\text{ min}$)	V_{ISO}	5300			Vac(rms)
	('-'M', White Package) ($f = 60\text{ Hz}$, $t = 1\text{ sec}$)		7500			Vac(pk)
Isolation Resistance	($V_{I-O} = 500\text{ VDC}$)	R_{ISO}	10^{11}			Ω
Isolation Capacitance	($V_{I-O} = 500\text{ VDC}$, $f = 1\text{ MHz}$)	C_{ISO}		0.5		pF
	('-'M' White Package)			0.2	2	pF

Note

* Typical values at $T_A = 25^\circ\text{C}$

4N25
4N37

4N26
H11A1

4N27
H11A2

4N28
H11A3

4N35
H11A4

4N36
H11A5

TRANSFER CHARACTERISTICS ($T_A = 25^\circ\text{C}$ Unless otherwise specified.)

DC Characteristic	Test Conditions	Symbol	Device	Min	Typ*	Max	Unit
Current Transfer Ratio, Collector to Emitter	$(I_F = 10 \text{ mA}, V_{CE} = 10 \text{ V})$	CTR	4N35 4N36 4N37	100			%
			H11A1	50			
			H11A5	30			
	4N25 4N26 H11A2 H11A3		20				
	4N27 4N28 H11A4		10				
	$(I_F = 10 \text{ mA}, V_{CE} = 10 \text{ V}, T_A = -55^\circ\text{C})$		4N35 4N36 4N37	40			
	$(I_F = 10 \text{ mA}, V_{CE} = 10 \text{ V}, T_A = +100^\circ\text{C})$		4N35 4N36 4N37	40			
Collector-Emitter Saturation Voltage	$(I_C = 2 \text{ mA}, I_F = 50 \text{ mA})$	$V_{CE(SAT)}$	4N25 4N26 4N27 4N28			0.5	V
	$(I_C = 0.5 \text{ mA}, I_F = 10 \text{ mA})$		4N35 4N36 4N37			0.3	
			H11A1 H11A2 H11A3 H11A4 H11A5			0.4	
AC Characteristic							
Non-Saturated Turn-on Time	$(I_F = 10 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)$ (Fig.20)	T_{ON}	4N25 4N26 4N27 4N28 H11A1 H11A2 H11A3 H11A4 H11A5		2		μs
Non Saturated Turn-on Time	$(I_C = 2 \text{ mA}, V_{CC} = 10 \text{ V}, R_L = 100\Omega)$ (Fig.20)	T_{ON}	4N35 4N36 4N37		2	10	μs

CD54/74HC240, CD54/74HCT240, CD74HC241, CD54/74HCT241, CD54/74HC244, CD54/74HCT244

High Speed CMOS Logic Octal Buffer/Line Drivers, Three-State

Features

- HC/HCT240 Inverting
- HC/HCT241 Non-Inverting
- HC/HCT244 Non-Inverting
- Typical Propagation Delay = 8ns at $V_{CC} = 5V$, $C_L = 15pF$, $T_A = 25^\circ C$ for HC240
- Three-State Outputs
- Buffered Inputs
- High-Current Bus Driver Outputs
- Fanout (Over Temperature Range)
 - Standard Outputs 10 LSTTL Loads
 - Bus Driver Outputs 15 LSTTL Loads
- Wide Operating Temperature Range . . . $-55^\circ C$ to $125^\circ C$
- Balanced Propagation Delay and Transition Times
- Significant Power Reduction Compared to LSTTL Logic ICs
- HC Types
 - 2V to 6V Operation
 - High Noise Immunity: $N_{IL} = 30\%$, $N_{IH} = 30\%$ of V_{CC} at $V_{CC} = 5V$
- HCT Types
 - 4.5V to 5.5V Operation
 - Direct LSTTL Input Logic Compatibility, $V_{IL} = 0.8V$ (Max), $V_{IH} = 2V$ (Min)
 - CMOS Input Compatibility, $I_I \leq 1\mu A$ at V_{OL} , V_{OH}

Description

The 'HC240 and 'HCT240 are inverting three-state buffers having two active-low output enables. The CD74HC241, 'HCT241, 'HC244 and 'HCT244 are non-inverting three-state buffers that differ only in that the 241 has one active-high and one active-low output enable, and the 244 has two active-low output enables. All three types have identical pinouts.

Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE
CD54HC240F3A	-55 to 125	20 Ld CERDIP
CD74HC240E	-55 to 125	20 Ld PDIP
CD74HC240M	-55 to 125	20 Ld SOIC
CD54HCT240F3A	-55 to 125	20 Ld CERDIP
CD74HCT240E	-55 to 125	20 Ld PDIP
CD74HCT240M	-55 to 125	20 Ld SOIC
CD74HC241E	-55 to 125	20 Ld PDIP
CD54HCT241F3A	-55 to 125	20 Ld CERDIP
CD74HCT241E	-55 to 125	20 Ld PDIP
CD74HCT241M	-55 to 125	20 Ld SOIC
CD54HC244F	-55 to 125	20 Ld CERDIP
CD54HC244F3A	-55 to 125	20 Ld CERDIP
CD74HC244E	-55 to 125	20 Ld PDIP
CD74HC244M	-55 to 125	20 Ld SOIC
CD54HCT244F	-55 to 125	20 Ld CERDIP
CD54HCT244F3A	-55 to 125	20 Ld CERDIP
CD74HCT244E	-55 to 125	20 Ld PDIP
CD74HCT244M	-55 to 125	20 Ld SOIC

NOTES

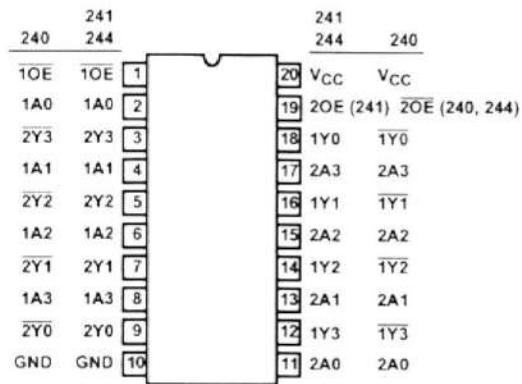
1. When ordering, use the entire part number. Add the suffix 96 to obtain the variant in the tape and reel.
2. Wafer and die for this part number is available which meets all electrical specifications. Please contact your local TI sales office or customer service for ordering information.



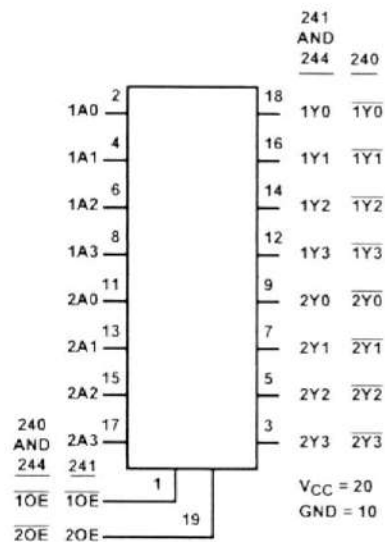
CIB - ESPOL

Pinout

CD54HC240, CD54HCT240, CD54HCT241,
 CD54HC244, CD54HCT244
 (CERDIP)
 CD74HC240, CD74HCT240, CD74HC241, CD74HCT241,
 CD74HC244, CD74HCT244
 (PDIP, SOIC)
 TOP VIEW



Functional Diagram



Absolute Maximum Ratings

DC Supply Voltage, V_{CC} -0.5V to 7V
 DC Input Diode Current, I_{IK}
 For $V_I < -0.5V$ or $V_I > V_{CC} + 0.5V$ $\pm 20mA$
 DC Output Diode Current, I_{OK}
 For $V_O < -0.5V$ or $V_O > V_{CC} + 0.5V$ $\pm 20mA$
 DC Drain Current, per Output, I_O
 For $-0.5V < V_O < V_{CC} + 0.5V$ $\pm 35mA$
 DC Output Source or Sink Current per Output Pin, I_O
 For $V_O > -0.5V$ or $V_O < V_{CC} + 0.5V$ $\pm 25mA$
 DC V_{CC} or Ground Current, I_{CC} $\pm 70mA$

Thermal Information

Thermal Resistance (Typical, Note 3) θ_{JA} ($^{\circ}C/W$)
 PDIP Package 125
 SOIC Package 120
 Maximum Junction Temperature 150 $^{\circ}C$
 Maximum Storage Temperature Range -65 $^{\circ}C$ to 150 $^{\circ}C$
 Maximum Lead Temperature (Soldering 10s) 300 $^{\circ}C$
 (SOIC - Lead Tips Only)

Operating Conditions

Temperature Range (T_A) -55 $^{\circ}C$ to 125 $^{\circ}C$
 Supply Voltage Range, V_{CC}
 HC Types 2V to 6V
 HCT Types 4.5V to 5.5V
 DC Input or Output Voltage, V_I, V_O 0V to V_{CC}
 Input Rise and Fall Time
 2V 1000ns (Max)
 4.5V 500ns (Max)
 6V 400ns (Max)

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

- 3. θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

DC Electrical Specifications

PARAMETER	SYMBOL	TEST CONDITIONS		V_{CC} (V)	25 $^{\circ}C$			-40 $^{\circ}C$ TO 85 $^{\circ}C$		-55 $^{\circ}C$ TO 125 $^{\circ}C$		UNITS
		V_I (V)	I_O (mA)		MIN	TYP	MAX	MIN	MAX	MIN	MAX	
HC TYPES												
High Level Input Voltage	V_{IH}	-	-	2	1.5	-	-	1.5	-	1.5	-	V
				4.5	3.15	-	-	3.15	-	3.15	-	V
				6	4.2	-	-	4.2	-	4.2	-	V
Low Level Input Voltage	V_{IL}	-	-	2	-	-	0.5	-	0.5	-	0.5	V
				4.5	-	-	1.35	-	1.35	-	1.35	V
				6	-	-	1.8	-	1.8	-	1.8	V
High Level Output Voltage CMOS Loads	V_{OH}	V_{IH} or V_{IL}	-0.02	2	1.9	-	-	1.9	-	1.9	-	V
			-0.02	4.5	4.4	-	-	4.4	-	4.4	-	V
			-0.02	6	5.9	-	-	5.9	-	5.9	-	V
High Level Output Voltage TTL Loads	V_{OH}	V_{IH} or V_{IL}	-6	4.5	3.98	-	-	3.84	-	3.7	-	V
			-7.8	6	5.48	-	-	5.34	-	5.2	-	V
Low Level Output Voltage CMOS Loads	V_{OL}	V_{IH} or V_{IL}	0.02	2	-	-	0.1	-	0.1	-	0.1	V
			0.02	4.5	-	-	0.1	-	0.1	-	0.1	V
			0.02	6	-	-	0.1	-	0.1	-	0.1	V
Low Level Output Voltage TTL Loads	V_{OL}	V_{IH} or V_{IL}	6	4.5	-	-	0.26	-	0.33	-	0.4	V
			7.8	6	-	-	0.26	-	0.33	-	0.4	V
Input Leakage Current	I_I	V_{CC} or GND	-	6	-	-	± 0.1	-	± 1	-	± 1	μA
Quiescent Device Current	I_{CC}	V_{CC} or GND	0	6	-	-	8	-	80	-	160	μA

DC Electrical Specifications (Continued)

PARAMETER	SYMBOL	TEST CONDITIONS		V _{CC} (V)	25°C			-40°C TO 85°C		-55°C TO 125°C		UNITS
		V _I (V)	I _O (mA)		MIN	TYP	MAX	MIN	MAX	MIN	MAX	
Three-State Leakage Current	I _{OZ}	V _{IL} or V _{IH}	-	6	-	-	±0.5	-	±0.5	-	±10	µA
HCT TYPES												
High Level Input Voltage	V _{IH}	-	-	4.5 to 5.5	2	-	-	2	-	2	-	V
Low Level Input Voltage	V _{IL}	-	-	4.5 to 5.5	-	-	0.8	-	0.8	-	0.8	V
High Level Output Voltage CMOS Loads	V _{OH}	V _{IH} or V _{IL}	-0.02	4.5	4.4	-	-	4.4	-	4.4	-	V
High Level Output Voltage TTL Loads			-6	4.5	3.98	-	-	3.84	-	3.7	-	V
Low Level Output Voltage CMOS Loads	V _{OL}	V _{IH} or V _{IL}	0.02	4.5	-	-	0.1	-	0.1	-	0.1	V
Low Level Output Voltage TTL Loads			6	4.5	-	-	0.26	-	0.33	-	0.4	V
Input Leakage Current	I _I	V _{CC} to GND	0	5.5	-	-	±0.1	-	±1	-	±1	µA
Quiescent Device Current	I _{CC}	V _{CC} or GND	0	5.5	-	-	8	-	80	-	160	µA
Additional Quiescent Device Current Per Input Pin: 1 Unit Load (Note 4)	ΔI _{CC}	V _{CC} -2.1	-	4.5 to 5.5	-	100	360	-	450	-	490	µA
Three-State Leakage Current	I _{OZ}	V _{IL} or V _{IH}	-	5.5	-	-	±0.5	-	±5	-	±10	µA

NOTE:

4. For dual-supply systems theoretical worst case (V_I = 2.4V, V_{CC} = 5.5V) specification is 1.8mA.

HCT Input Loading Table

INPUT	UNIT LOADS
HCT240	
nA0-A3	1.5
1OE	0.7
2OE	0.7
HCT241	
nA0-A3	0.7
1OE	0.7
2OE	1.5
HCT244	
nA0-A3	0.7
1OE	0.7
2OE	0.7

NOTE: Unit Load is ΔI_{CC} limit specified in DC Electrical Specifications table, e.g., 360µA max at 25°C.

Switching Specifications $C_L = 50\text{pF}$, Input $t_r, t_f = 6\text{ns}$

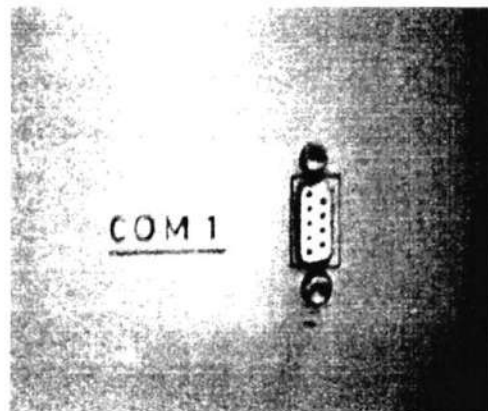
PARAMETER	SYMBOL	TEST CONDITIONS	V_{CC} (V)	25°C			-40°C TO 85°C			-55°C TO 125°C			UNITS
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
HC TYPES													
Propagation Delay Data to Outputs HC240	t_{PLH}, t_{PHL}	$C_L = 50\text{pF}$	2	-	-	100	-	-	125	-	-	150	ns
			4.5	-	-	20	-	-	25	-	-	30	ns
		$C_L = 15\text{pF}$	5	-	8	-	-	-	-	-	-	-	ns
		$C_L = 50\text{pF}$	6	-	-	17	-	-	21	-	-	26	ns
Data to Outputs HC241	t_{PLH}, t_{PHL}	$C_L = 50\text{pF}$	2	-	-	110	-	-	140	-	-	165	ns
			4.5	-	-	22	-	-	28	-	-	33	ns
		$C_L = 15\text{pF}$	5	-	9	-	-	-	-	-	-	-	ns
		$C_L = 50\text{pF}$	6	-	-	19	-	-	24	-	-	28	ns
Data to Outputs HC244	t_{PLH}, t_{PHL}	$C_L = 50\text{pF}$	2	-	-	110	-	-	140	-	-	165	ns
			4.5	-	-	22	-	-	28	-	-	33	ns
		$C_L = 15\text{pF}$	5	-	9	-	-	-	-	-	-	-	ns
		$C_L = 50\text{pF}$	6	-	-	19	-	-	24	-	-	28	ns
Output Enable and Disable Time	t_{THL}, t_{TLH}	$C_L = 50\text{pF}$	2	-	-	150	-	-	190	-	-	225	ns
			4.5	-	-	30	-	-	38	-	-	45	ns
			5	-	12	-	-	-	-	-	-	-	ns
			6	-	-	26	-	-	33	-	-	38	ns
Output Transition Time	t_{TLH}, t_{THL}	$C_L = 50\text{pF}$	2	-	-	60	-	-	75	-	-	90	ns
			4.5	-	-	12	-	-	15	-	-	18	ns
			6	-	-	10	-	-	13	-	-	15	ns
Input Capacitance	C_I	$C_L = 50\text{pF}$	-	10	-	10	-	-	10	-	-	10	pF
Three-State Output Capacitance	C_O	$C_L = 50\text{pF}$	-	-	-	20	-	-	20	-	-	20	pF
Power Dissipation Capacitance (Notes 5, 6)	C_{pD}	$C_L = 15\text{pF}$	5	-	38	-	-	-	-	-	-	-	pF
			5	-	34	-	-	-	-	-	-	-	pF
			5	-	46	-	-	-	-	-	-	-	pF
HCT TYPES													
Propagation Delay Data to Outputs HCT240	t_{PHL}, t_{PLH}	$C_L = 50\text{pF}$	4.5	-	-	22	-	-	28	-	-	33	ns
		$C_L = 15\text{pF}$	5	-	9	-	-	-	-	-	-	-	ns
Data to Outputs HCT241	t_{PHL}, t_{PLH}	$C_L = 50\text{pF}$	4.5	-	-	25	-	-	31	-	-	38	ns
		$C_L = 15\text{pF}$	5	-	10	-	-	-	-	-	-	-	ns
Data to Outputs HCT244	t_{PHL}, t_{PLH}	$C_L = 50\text{pF}$	4.5	-	-	25	-	-	31	-	-	38	ns
		$C_L = 15\text{pF}$	5	-	10	-	-	-	-	-	-	-	ns

ANEXO B

CABLE SERIAL, TRANSDUCTORES Y OTROS ELEMENTOS UTILIZADOS

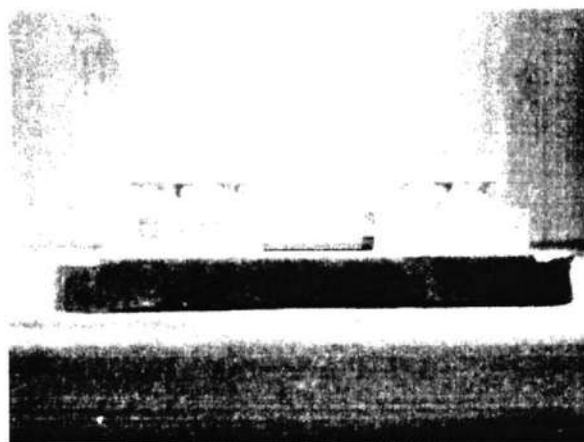
Cable serial: Existen diversas configuraciones o estructuras internas en los cables de comunicación serial, para el proyecto desarrollado se requiere que la estructura interna del cable sea compatible pin a pin entre el puerto de la PC y el puerto del tablero (COM1). La estructura correcta del cable es aquella que no es cruzada, es decir que el pin 1 de un extremo del cable corresponde al pin 1 del otro extremo y esto debe cumplirse para los 9 pines que tienen los conectores.

A continuación se muestra una foto del cable serial que se utiliza en el proyecto y el puerto del tablero del proyecto.

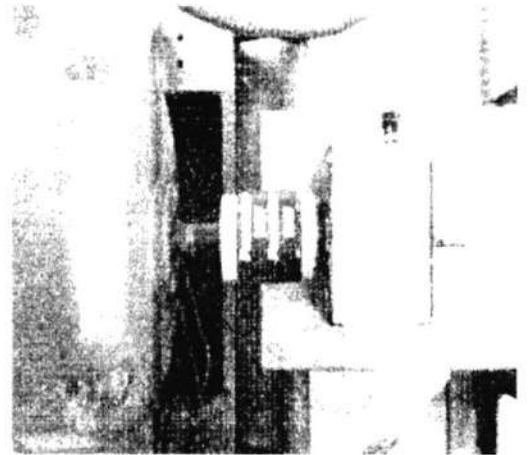
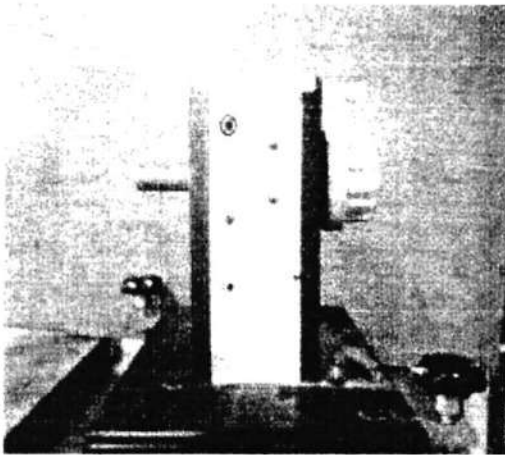


Transductores.- Los transductores que se utilizaron fueron: shunts para convertir la corriente de las fases en voltajes, un tacogenerador para medir la velocidad y el conector SCXI-1327 de NI para atenuar las señales de los voltajes de alimentación (el conector lo hace mediante divisores de voltaje).

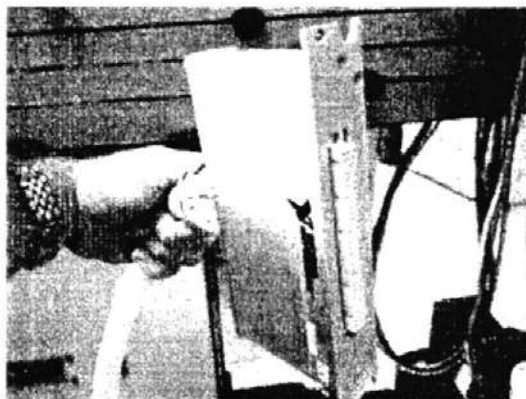
➤ **Shunts.-** Se utilizaron 3 shunts para convertir cada corriente de fase en voltaje y de esta forma poder medirla con el SCXI-1120 que solo acepta señales de voltaje. Los shunts son pequeñas resistencias conectadas en serie con las líneas de alimentación, de los cuales se toma el voltaje en los extremos de dichas resistencias. La relación de transformación de los shunts utilizados es de 1 voltio por 1 amperio y el rango de funcionamiento es hasta 5 amperios (=5 voltios). A continuación se muestra la foto de uno de los shunts utilizados.



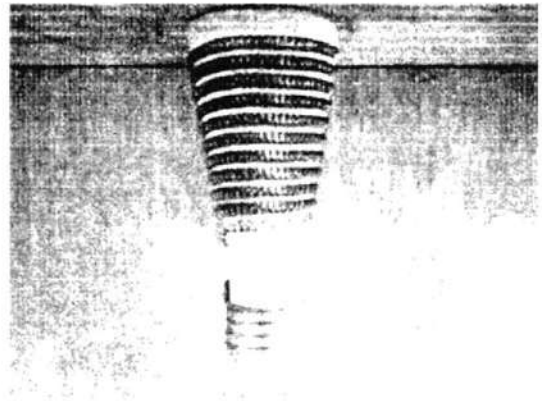
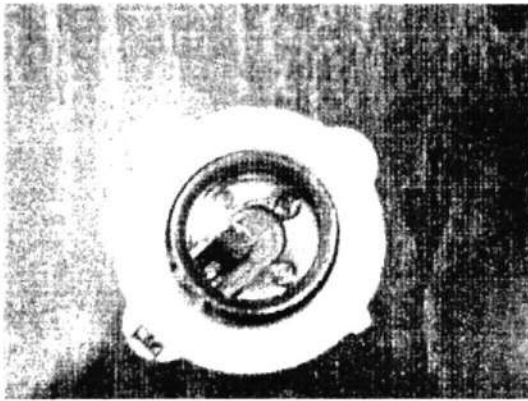
➤ **Tacogenerador.-** El tacogenerador utilizado se muestra a continuación y es el encargado de convertir la velocidad en una señal de voltaje. Su rango de voltaje es de -6 a $+6$ voltios correspondientes a 1800 revoluciones por minuto con giro a la izquierda o a la derecha.



➤ **Conector SCXI-1327.-** Este conector admite entradas de voltaje de hasta 250 Voltios RMS en cada canal. En su interior posee divisores de voltaje que atenúan las señales cien veces, estos divisores se habilitan para cada canal por medio de un interruptor por canal. A continuación se muestra una foto del conector mencionado.



➤ **Resistencias de poder.-** Se utilizaron 12 resistencias de poder para realizar los arranques con resistencias, cambios de giro y de velocidad en los rotores devanados. Cada resistencia es de 12 ohmios pero puede ser modificada cortando el alambre que contienen. Soportan hasta 10 amperios y fueron colocadas en boquillas de porcelana.



CIB -ESPOL

ANEXO C

TERMINALES Y TOMAS DEL TABLERO DE CONTROL

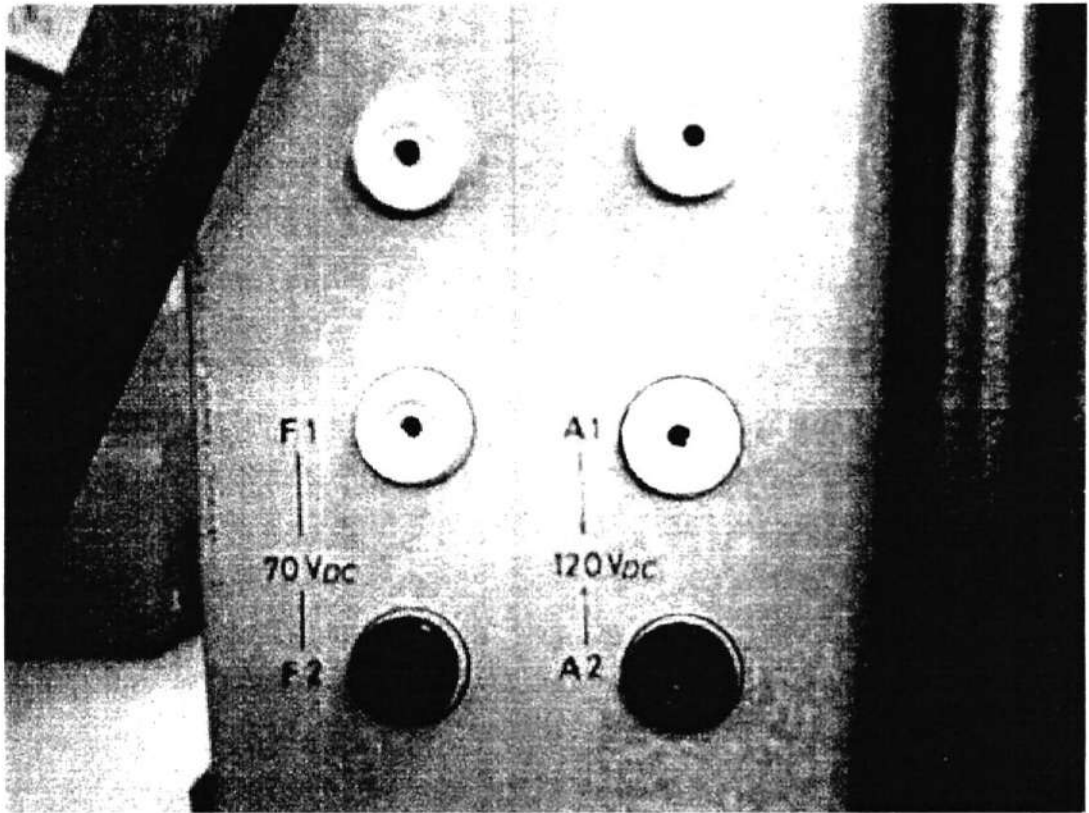
En este anexo se explicarán para qué sirven las tomas y terminales que se encuentran en un costado del tablero de control. Se presentarán fotos con la correspondiente explicación.

Conexión de bobinas y terminales DC.- En la siguiente foto se observan 6 terminales de diferentes colores, los cuales sirven para alimentar las bobinas de los contactores y alimentar el campo y el inducido de la máquina DC.

Los terminales celeste y blanco de la parte superior sirven para la alimentación de las bobinas (220 voltios), estos terminales deben conectarse siempre, de lo contrario los contactores no se energizarán nunca.

Los terminales blanco y negro del lado izquierdo, con las marcas F1 y F2 sirven para energizar el campo de la máquina DC con un voltaje de 70 Voltios DC. La conexión de estos terminales es opcional.

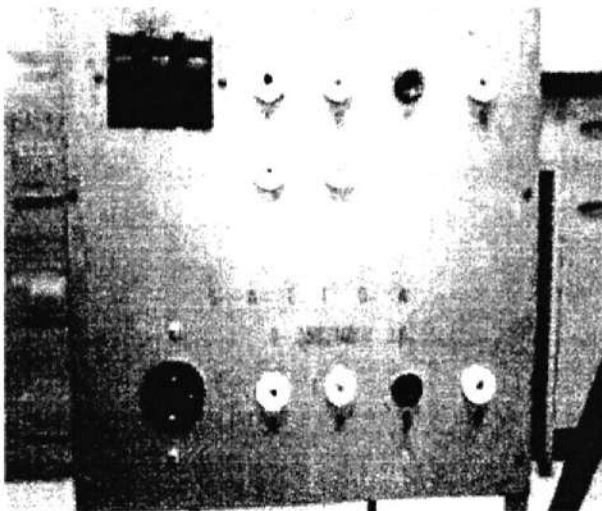
Los terminales rojo y negro del lado derecho, con las marcas A1 y A2 sirven para energizar la armadura (cuando funciona como motor) con 120 Voltios DC o para alimentar cargas externas (cuando funciona como generador). La conexión de estos terminales es opcional.



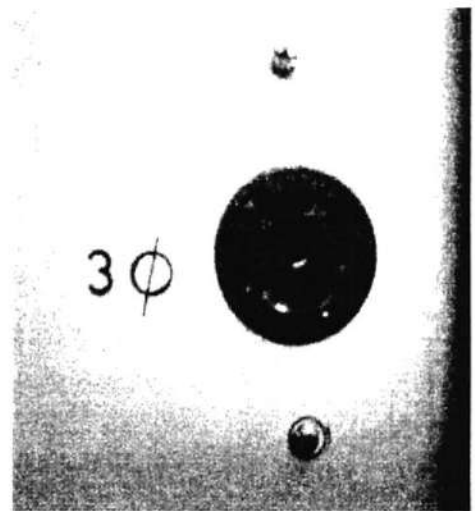
Terminales para las bobinas y la máquina DC

Tomas Trifásicas.- El sistema utiliza dos tomas trifásicas: una en el tablero y otra en un VARIAC. Las dos tomas sirven para conectar las 3 fases de alimentación a la máquina AC por medio de un cable trifásico.

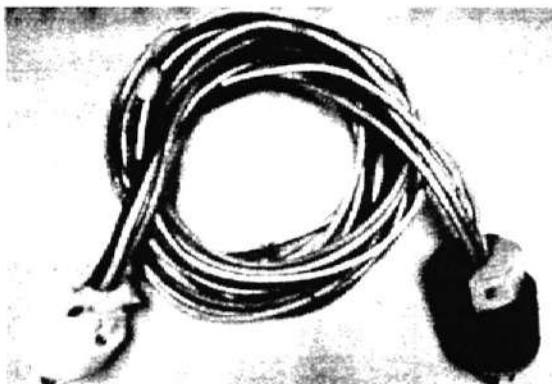
A continuación se muestran las tomas trifásicas y el cable utilizado.



VARIAC



Toma en el tablero



Cable trifásico



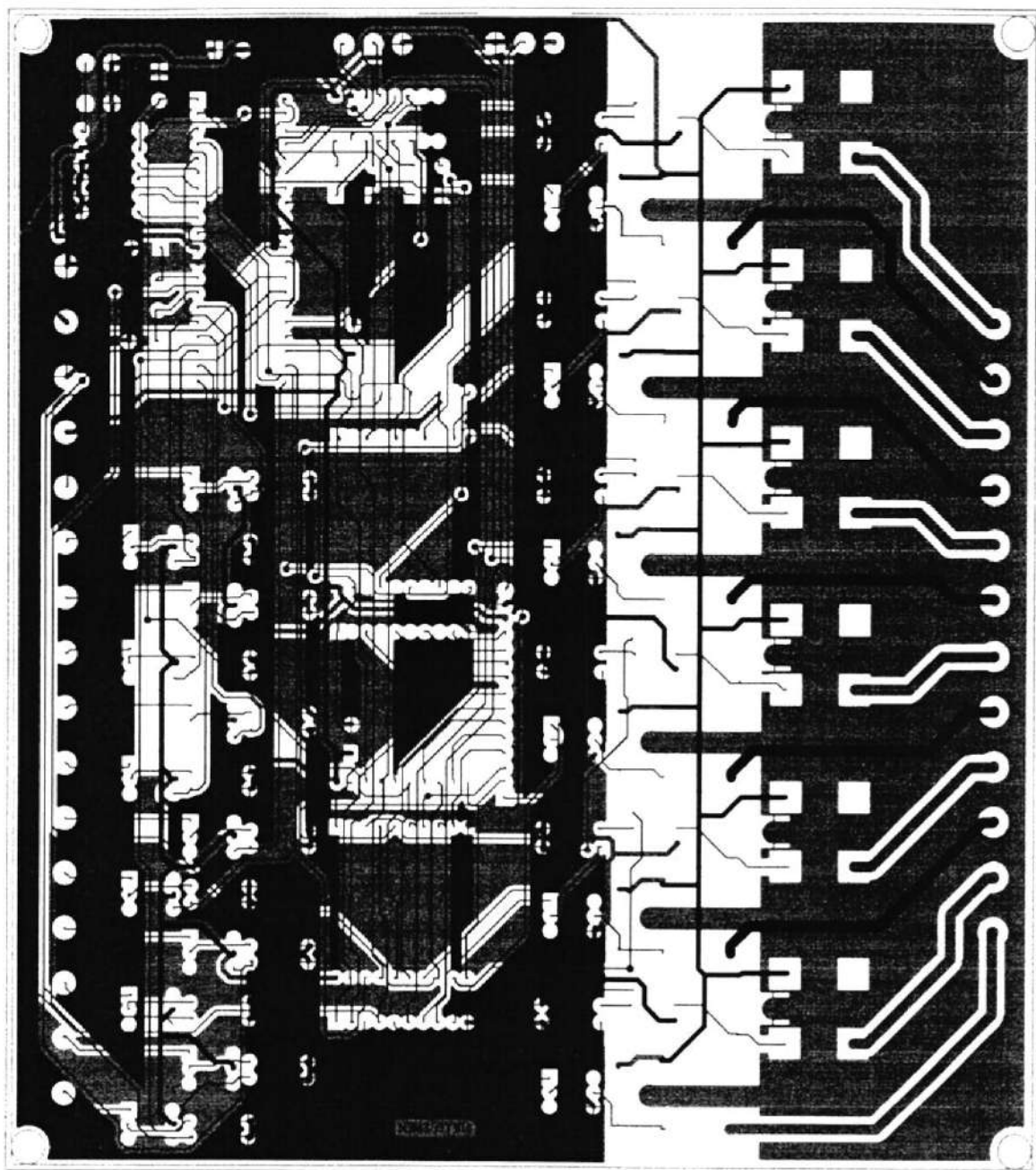
Conector trifásico

ANEXO D

CIRCUITO IMPRESO



CIB-ESPOL



A.F. 141944