

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



**EMPLEO DEL COMPUTADOR PERSONAL PARA EL DIAGNÓSTICO
AUTOMATIZADO DE LA TARJETA SENSORA DE CARGA Y MONITORA DE
ALARMAS ERICSSON**

TESIS

PRESENTADA A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

ALFREDO DONATO RUIZ SERRANO
AL CONFERÍRSELE EL TÍTULO DE
INGENIERO MECÁNICO ELECTRICISTA

GUATEMALA, MAYO DE 1999.

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley orgánica de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de tesis titulado:

Empleo del computador personal para el diagnóstico automatizado de la tarjeta electrónica sensora de carga y monitora de alarmas Ericsson.

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica-Eléctrica, con fecha 21 de abril de 1,998.

Alfredo Donato Ruiz Serrano.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	ING. HERBERT RENÉ MIRANDA BARRIOS
VOCAL 1o.	ING. JOSÉ FRANCISCO GÓMEZ RIVERA
VOCAL 2o.	ING. CARLOS HUMBERTO PÉREZ RODRÍGUEZ
VOCAL 3o.	ING. JORGE BENJAMÍN GUTIÉRREZ QUINTANA
VOCAL 4o.	BR. DIMAS ALFREDO CARRANZA BARRERA
VOCAL 5o.	BR. JOSÉ ENRIQUE LÓPEZ BARRIOS
SECRETARIA	INGA. GILDA MARINA CASTELLANOS DE ILLESCAS

**TRIBUNAL QUE PRACTICÓ EL EXAMEN
GENERAL PRIVADO**

DECANO	ING. HERBERT RENÉ MIRANDA BARRIOS
EXAMINADOR	ING. CARLOS CHICOJAY COLOMA
EXAMINADOR	ING. GUSTAVO BENIGNO OROZCO
EXAMINADOR	ING. FRANCISCO TZIRÍN
SECRETARIA	INGA. GILDA MARINA CASTELLANOS DE ILLESCAS

Guatemala, 7 de Agosto de 1998.

Ingeniero
Julio Solares Peñate
Coordinador de Area de Electrónica
Comunicaciones y Control
Escuela de Ingeniería Mecánica Eléctrica
Universidad de San Carlos de Guatemala

Señor Coordinador:

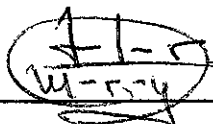
De manera atenta presentamos el trabajo de tesis elaborado por el señor Alfredo Donato Ruíz Serrano, titulado:

EMPLEO DEL COMPUTADOR PERSONAL PARA EL DIAGNÓSTICO AUTOMATIZADO DE LA TARJETA ELECTRÓNICA SENSORA DE CARGA Y MONITORA DE ALARMAS ERICSSON.

En calidad de Asesor le informo que he revisado el mencionado trabajo y considero que cumple con los objetivos planteados, por lo que recomiendo su impresión.

Por lo tanto, el autor de ésta tesis y yo, como su Asesor, nos hacemos responsables por el contenido y conclusiones de la misma.

Atentamente,

f.  _____

Ing. Héctor Amílcar Monroy Menéndez
ASESOR

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Guatemala, 9 de septiembre de 1,998

Señor Director
Ing. Roberto Urdiales Contreras
Escuela de Ingeniería Mecánica Eléctrica.


Señor Director,

Me permito dar aprobación al trabajo de tesis titulado: Empleo del computador personal para el diagnóstico automatizado de la tarjeta electrónica sensora de carga y monitora de alarmas Ericsson, desarrollado por el señor Alfredo Donato Ruiz Serrano, por considerar que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,

ID Y ENSEÑAD A TODOS


Ing. Julio César Solares Peñate
Coordinador Area Electrónica

JCSP/edem.

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de tesis del estudiante Alfredo Donato Ruiz Serrano, título: Empleo del computador personal para el diagnóstico automatizado de la tarjeta electrónica sensora de carga y monitora de alarmas Ericsson, procede a la autorización del mismo.

Ing. Roberto Urdiales Contreras
Director

Guatemala, 8 de enero de 1,999.



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



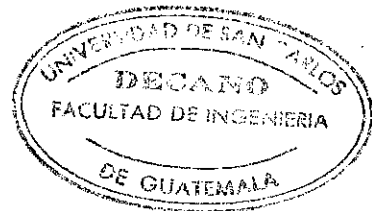
FACULTAD DE INGENIERIA

El Decano de la Facultad de Ingeniería, luego de conocer la autorización por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de tesis: Empleo del computador personal para el diagnóstico automatizado de la tarjeta electrónica sensora de carga y monitora de alarmas Ericsson, del estudiante Alfredo Donato Ruiz Serrano, procede a la autorización para la impresión de la misma.

IMPRIMASE:


u.
Ing. Herbert René Miranda Barrios
Decano

Guatemala, 9 de febrero de 1,999.



ACTO QUE DEDICO

A Dios todopoderoso, que siempre me ha acompañado, en cada actividad que he iniciado y llevado a cabo.

A mis padres: Celia Juventina Serrano Velásquez, y Donato Ruiz Soto; por su cariño y apoyo incondicional.

A mis abuelos: Héctor Serrano Muñoz (E.P.D.) y Petrona Velásquez López, por su cariño y buenos consejos.

A la Universidad de San Carlos de Guatemala y catedráticos en general, por los conocimientos que me han brindado.

A todos mis amigos, por los gratos momentos que hemos compartido.

ÍNDICE GENERAL

LISTA DE ILUSTRACIONES.....	V
GLOSARIO.....	VII
INTRODUCCIÓN.....	X
1. PUERTOS PARALELOS Y PUERTOS SERIE.....	1
1.1 Generalidades.....	1
1.2 El puerto paralelo en la transmisión de datos.....	2
1.2.1 Descripción de los registros del puerto paralelo.....	3
1.2.1.1 El primer registro.....	3
1.2.1.2 El segundo registro.....	4
1.2.1.3 El tercer registro.....	5
1.2.2 Control en la transmisión / recepción de datos.....	5
1.2.2.1 Entrada / salida simple.....	6
1.2.2.2 Mano a mano simple (Handshaking).....	6
1.2.2.3 Doble mano a mano en la transmisión de datos.....	7
1.2.4 Estructura y operación del puerto paralelo del PC.....	8
1.2.4.1 Descripción de la interfaz Centronics.....	9
1.2.4.2 Operación del puerto paralelo del PC.....	11
1.2.4.3 Consideraciones para el uso del puerto paralelo del PC.....	11
1.3 El puerto serie del computador.....	12
1.3.1 Transmisión serial de datos.....	12
1.3.1.1 Estándar para transmisiones RS-232C.....	13
1.3.1.1.1 Niveles de tensión.....	15
1.3.1.2 Problemas en las transmisiones seriales.....	15

1.3.1.2.1	Sincronización de bit.....	15
1.3.1.2.2	Sincronización de caracter.....	16
1.3.1.2.3	Sincronización de mensaje.....	16
1.3.2	RS-232C para interfaces a niveles TTL.....	17
2.	CONSIDERACIONES DE SOFTWARE.....	18
2.1	Generalidades.....	18
2.2	Lenguajes de máquina.....	18
2.3	Lenguajes ensambladores.....	19
2.4	Lenguajes de alto nivel.....	20
2.5	Manejo de puertos con el lenguaje ensamblador.....	21
2.5.1	Proceso de entrada.....	21
2.5.2	Proceso de salida.....	21
2.6	Manejo de puertos con el lenguaje Turbo Pascal.....	22
2.6.1	Proceso de entrada.....	22
2.6.2	Proceso de salida.....	22
2.7	Manejo de puertos con el lenguaje Visual Basic.....	23
2.7.1	Proceso de entrada.....	23
2.7.2	Proceso de salida.....	25
2.7.3	Parámetros de comunicación	26
2.7.4	Protocolo de comunicaciones.....	27
2.8	Librerías de acceso dinámico.....	28
2.8.1	Creación de librerías con el lenguaje Turbo Pascal.....	28
2.8.2	Uso de las librerías en Visual Basic.....	30
3.	DESCRIPCIÓN GENERAL DE LA TARJETA A DIAGNOSTICAR.....	32
3.1	Generalidades.....	32
3.2	Descripción mecánica.....	33



3.3	Descripción de funcionamiento.....	35
3.3.1	Fuente de alimentación.....	35
3.3.2	Monitor de tensión para sobre voltaje.....	36
3.3.3	Monitor de tensión para bajo voltaje.....	38
3.3.4	Temporizador de los ciclos de carga.....	40
3.3.4.1	Parámetros principales de funcionamiento.....	41
3.3.4.1.1	Frecuencia.....	41
3.3.4.1.2	Tiempo en estado alto.....	42
3.3.4.1.3	Tiempo en estado bajo.....	43
3.3.5	Etapa divisora de frecuencia.....	43
3.3.5.1	Descripción funcional de la etapa.....	44
3.3.5.2	Ciclos de carga.....	46
3.3.5.2.1	Tabla de los ciclos de carga.....	46
3.3.6	Etapa de desactivación en caso de sobre temperatura.....	47
3.4	Descripción de los puntos de prueba.....	49
3.5	Descripción de las terminales de entrada y salida de la unidad.....	50
4.	DESCRIPCIÓN DE LA INTERFAZ Y PRUEBAS A LA UNIDAD.....	51
4.1	Generalidades.....	51
4.2	Consideraciones generales de diseño de hardware.....	53
4.2.1	Protección del puerto de impresión.....	53
4.2.2	Protección de los buffers tri-estado de la interfaz.....	53
4.2.3	Etapas manejadoras de relevador.....	54
4.2.4	Acoples ópticos.....	54
4.2.4.1	Diodo emisor.....	55
4.2.4.2	Transistor.....	55
4.2.5	La señal de conocimiento del PC.....	56
4.2.6	Etapa del regulador de voltaje variable.....	56

4.3	Palabras de control de la interfaz.....	57
4.4	Prueba de los voltajes de alimentación.....	58
4.4.1	Fase 1.....	58
4.4.2	Fase 2.....	59
4.5	Prueba del temporizador.....	60
4.6	Prueba de los estados lógicos en condición de carga.....	61
4.7	Prueba de la respuesta a bajo voltaje y sobre temperatura.....	63
4.8	Prueba de la respuesta a un sobre voltaje.....	65
5.	DESCRIPCIÓN GENERAL DEL SOFTWARE UTILIZADO.....	68
5.1	Generalidades.....	68
5.2	Software utilizado para el control del puerto de impresión.....	69
5.3	Creación de la base de parámetros.....	71
5.4	Software utilizado para la prueba de tensiones.....	81
5.5	Software utilizado para la prueba del temporizador.....	83
5.6	Software para la prueba de los estados lógicos.....	85
5.7	Software para la simulación de bajo voltaje.....	86
5.8	Software para la simulación de un sobre voltaje.....	87
5.9	Procedimiento de retardo.....	88
5.10	Procedimiento efecto.....	88
	CONCLUSIONES.....	90
	RECOMENDACIONES.....	91
	BIBLIOGRAFÍA.....	92
	APÉNDICE	93

ÍNDICE DE ILUSTRACIONES

FIGURAS

No.	Título	Pág.
1	El primer registro del puerto paralelo	3
2	El segundo registro del puerto paralelo	4
3	Simple entrada / salida	6
4	Descripción mecánica de la tarjeta Ericsson	34
5	Etapas de alimentación de la unidad Ericsson	35
6	Diagrama esquemático del monitor de sobre voltaje	37
7	Diagrama esquemático del monitor de bajo voltaje	39
8	Diagrama esquemático del circuito oscilador	41
9	Diagrama esquemático de la etapa divisora y circuito de carga	44
10	Diagrama esquemático de la etapa para sobre temperatura	48
11	Diagrama pictórico de la conexión de prueba	52
12	Etapas utilizadas en el diseño de la interfaz	53
13	Etapas de prueba de las tensiones de alimentación	59
14	Etapas de lectura de la señal del temporizador	61
15	Etapas de prueba de los estados lógicos	62
16	Etapas de prueba de la respuesta a un bajo voltaje	63
17	Etapas de prueba de la respuesta a sobre voltaje	66
18	Configuración estable del CI 555	95
19	Configuración típica del CI LM317K	100

TABLAS

No.	Título	Pág.
I	Descripción del conector Centronics	10
II	Descripción del estándar RS-232C	14
III	Rangos de voltaje del RS-232C	15
IV	Protocolos de comunicación en Visual Basic	27
V	Duraciones de los ciclos de carga e inactivos	47
VI	Descripción de los puntos de prueba de la unidad	49
VII	Descripción de las terminales de entrada y salida	50
VIII	Palabras de control de la interfaz	57
IX	Descripción de las terminales del CI 555	94
X	Descripción de las terminales del CI LM311	97

GLOSARIO

- Axe-10** Se le denomina así al sistema de conmutación digital de las centrales Monte Verde III, Guarda Viejo III, El Carmen II, Vista Hermosa II, Lourdes II, Nimajuyú I, Centro IV, Parroquia II, Tívoli III, Monte Verde V y sus unidades remotas.
- Baudio** Unidad de medida de la tasa de transmisión de bits; es decir, un baudio puede ser 1, 1/2, ó 1/3 bit / segundo.
- Buffer** Dispositivo digital capaz de amplificar corriente y que en la salida entrega una señal que puede tener el mismo estado lógico de la entrada o el estado lógico de la entrada invertido
- Caracter** Conjunto de bits que conforman una palabra en el proceso de transmisión de información.
- Compilador** Software utilizado para traducir un programa escrito en lenguaje de alto nivel, a lenguaje de máquina.

- Chip** Estructura con características físicas estandarizadas, en donde se coloca un circuito integrado, al cual accesa el usuario por medio de las terminales que quedan en el exterior de dicha estructura después de su fabricación.
- Ericsson** Compañía sueca, con la que TELGUA, S.A. hizo un contrato para la instalación de centrales telefónicas digitales y equipo de soporte.
- Flanco** Rampa de voltaje que se presenta cuando ocurre una transición de estados lógicos.
- Flip-flop** Dispositivo digital compuesto, basado en el principio de realimentación de compuertas lógicas que le imprime la capacidad de retener estados lógicos de acuerdo a ciertas condiciones.
- Hardware** Circuitería, elementos y dispositivos electrónicos físicamente disponibles en un equipo electrónico.
- I/O** Entrada y salida (E/S) líneas o dispositivos, empleados para obtener o enviar información de o hacia el exterior por medio del computador.

- Interfaz** Interconexión, conexión común a dos sistemas distintos, circuito o dispositivo que enlaza apropiadamente dos sistemas para su correcta comunicación.
- Mnemónico** Abreviatura proveniente de palabras utilizadas en el lenguaje humano, para facilitar la escritura de programas en lenguaje ensamblador.
- Objeto** Elemento de software que tiene definida una cantidad determinada de propiedades, dichas propiedades poseen ciertas características que pueden ser modificadas según lo requiera el programador.
- Periférico** Elemento de hardware que interactúa con el CPU del computador para ejecutar ciertas tareas.
- ROM BIOS** Sistema operativo grabado en una ROM, utilizado para ejecutar tareas de verificación / inicialización del hardware presente en el PC al momento de encenderlo.
- Slot** Bahía de expansión, ranura situada sobre el circuito impreso de la tarjeta principal de un computador, para alojar alguna tarjeta que ha sido diseñada para el acceso a los buses del PC.

INTRODUCCIÓN

En el departamento de Diseño y Reparaciones Electrónicas de TELGUA, S.A. se reciben semanalmente varios tipos de tarjetas electrónicas para su reparación; entre dichas tarjetas destaca por su importancia la tarjeta de los rectificadores Ericsson, que se utiliza para el control de la carga de los bancos de baterías de las centrales telefónicas digitales del sistema Axe-10 (Ericsson). En dicho sistema se carece de suficiente equipo de soporte al momento de presentarse un problema, por lo que se corre el riesgo de suspender el servicio telefónico en alguna central del sistema. Algunas de las tarjetas poseen poca información detallada a nivel de circuito, a pesar de que ésta es necesaria para un análisis efectivo, a nivel de diagrama esquemático la información con la que se cuenta se enfoca más que todo al funcionamiento global, por lo que hay que invertir tiempo para estudiar las etapas y realizar pruebas laboriosas para localizar etapas defectuosas; entonces surge la inquietud de implementar procesos automatizados que agilicen y faciliten los diagnósticos de las tarjetas a reparar, por medio de un procedimiento asistido por computador, se puede reducir el tiempo de diagnóstico y por consiguiente el tiempo de reparación de dichas tarjetas.

Para realizar las pruebas de diagnóstico, se diseñó una interfaz de prueba (específicamente para la tarjeta electrónica sensora de carga y monitora de alarmas de los rectificadores Ericsson), la cual utiliza para su operación palabras de control de 8 bits que son enviadas por el puerto paralelo del PC (puerto de impresión) por medio de software.

La interfaz, como respuesta a la palabra de control, emite 5 bits, los cuales contienen información del estado de la etapa bajo prueba, seguidamente se procede a la lectura de dicho resultado para que vía software se realice una búsqueda en un archivo de parámetros eléctricos. Finalmente se muestra en pantalla el mensaje de diagnóstico respectivo. El programa de control de interfaz está escrito en lenguaje Visual Basic.

1. PUERTOS PARALELOS Y PUERTOS SERIE

1.1 Generalidades

La comunicación del computador con el exterior se puede realizar de dos formas: serial y paralela. La entrada / salida en paralelo se basa en la transmisión simultánea de n bits, siendo estos los componentes de la unidad básica de información: palabra o carácter. Las distintas palabras que conforma el bloque de información son transmitidas secuencialmente por las mismas vías, la entrada / salida serial es un caso particular de la anterior en la que $n=1$, efectuándose entonces el secuenciamiento de bits dentro de la palabra y de palabras dentro del bloque.

Regularmente, un computador incluye un puerto paralelo y un puerto serie, que pueden ser expandidos utilizando tarjetas de puerto según se requieran y siempre y cuando la tarjeta principal del computador tenga Slots libres para ser utilizados por dichas tarjetas. En un puerto paralelo, las vías de comunicación pueden ser unidireccionales o bidireccionales. Siendo las primeras aquellas en las que la información se traslada en un solo sentido, estando, permanentemente definidos los elementos que actúan como transmisor y los que lo hacen como receptor, el sentido hacia o desde el CPU depende de la característica del elemento periférico. Por otra parte, las líneas bidireccionales son aquellas en las que la información se transmite secuencialmente en ambos sentidos permitiendo, por tanto, un diálogo entre el computador y el periférico. En el estudio de un sistema de entradas / salidas en paralelo, para realizar aplicaciones, debe tomarse en cuenta una serie de aspectos que condicionan el tipo de líneas o circuitos terminales que deban utilizarse.

Dichos aspectos son: la velocidad de transmisión, distancia entre equipos, sentido de la comunicación, interferencia por ruidos externos o diafonía entre bits, número de bits componentes de información, señales de control, etc.

Se puede pensar que cuando la distancia entre los conjuntos que deban comunicarse sea reducida o bien cuando la velocidad de transmisión deba ser elevada *será preferido el puerto paralelo sobre el puerto serie.*

1.2 El puerto paralelo en la transmisión de datos

La transmisión paralela de datos involucra el traslado de información en n líneas simultáneamente, estableciéndose un protocolo apropiado para que la información se transmita a su destino confiablemente. Un computador puede soportar hasta tres puertos paralelos, porque hay reservadas tres zonas en el espacio de direcciones de entrada / salida, al inicializar el computador, las rutinas almacenadas en la ROM BIOS se encargan de direccionar estos puertos, según su orden de aparición como LPT1, LPT2 y LPT3. Las direcciones de éstos puertos son almacenadas en el segmento de variables de la BIOS que consta de un arreglo de cuatro palabras, por ejemplo el puerto de impresión (LPT1) se encuentra en la palabra de memoria que inicia por default en la dirección 0008h, siendo la dirección de puerto almacenada: 378h – 37Fh.

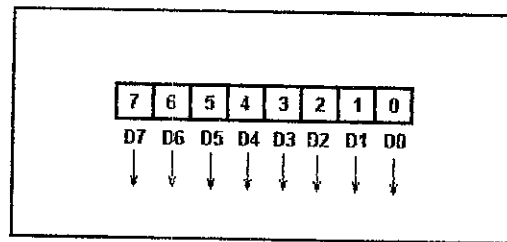
1.2.1 Descripción de los registros del puerto paralelo

Cada uno de los puertos paralelos posee un registro de interfaz unificada, que contiene a su vez tres registros, por ejemplo en el caso del puerto de impresión con su dirección 378h—37Fh, inician los registros en los valores: 378h, 379h y 37Ah respectivamente.

1.2.1.1 El primer registro

Es un registro de 1 byte, que opera con lógica positiva en todos sus bits, es importante tener en cuenta que éste registro es solamente de escritura, de manera que la interfaz solamente podrá leer de éstas ocho líneas de datos. En la figura 1, se ilustra dicho registro.

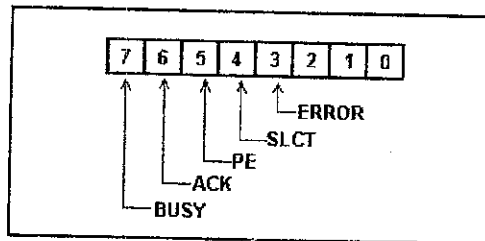
Figura 1: *El primer registro del puerto paralelo.*



1.2.1.2 El segundo registro

Este registro es de solo lectura, lo constituyen ocho bits, de los cuales se utilizan usualmente cinco por el computador para conocer el estado del impresor. El computador los utiliza básicamente para determinar si el impresor se encuentra ocupado (busy), si está en línea, si se ha terminado el papel o si se ha producido un error de transmisión de datos. El mencionado registro se ilustra en la figura 2.

Figura 2: *El segundo registro del puerto paralelo.*



De estas cinco líneas de estado, la línea de **BUSY** es la única que opera con lógica negativa, por lo que un estado bajo enviado por el impresor indicará al PC que se encuentra ocupado, la línea **ERROR** con un estado bajo indica que se ha producido un error en la transmisión de datos, un estado alto en la línea **SLCT** indica que el impresor se encuentra en línea, un estado alto en la línea **PE** indica que en ese instante el impresor se encuentra sin papel, un estado bajo en la línea **ACK** indica que el impresor está listo para recibir el siguiente carácter.

1.2.1.3 El tercer registro

Este registro es utilizado por el computador para indicarle al impresor la ejecución de ciertas funciones. Dichas funciones son:

- a) La puesta en marcha de la rutina de inicialización del impresor mediante la línea de control denominada INIT;
- b) Indicación al impresor para que envíe una línea de alimentación automáticamente después de cada retorno del carro (ASCII #13);
- c) Colocar en línea el impresor mediante la línea de control SLCT IN.

Este registro contiene un bit mediante el cual se puede hacer una interrupción de hardware, dicha interrupción se puede seleccionar por interruptores tipo DIP en la tarjeta, sin embargo raramente se tendrá que operar un puerto paralelo por el método de las interrupciones.

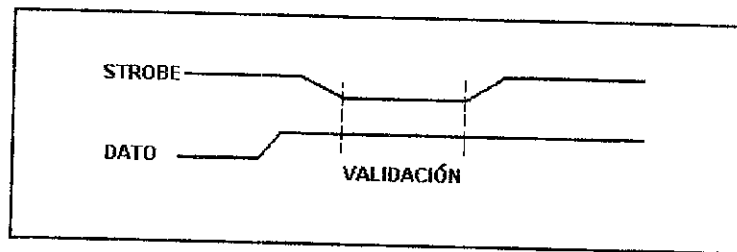
1.2.2 Control en la transmisión / recepción de datos

Existen diversas formas para la transmisión paralela de datos, siendo las más usuales, las que a continuación se describen.

1.2.2.1 Entrada / salida simple (I/O)

En este método, la información se transmite en un solo sentido, y cuando se quiere leer un dato en un bus o escribir en él, se debe validar la operación, por lo que se puede utilizar una señal que indique que el dato es válido en ese instante, una señal de este tipo puede ser por ejemplo la línea de control STROBE, el hecho de utilizar una línea de STROBE para validar un dato es de mucha utilidad en muchas aplicaciones, en la figura 3, se ilustra este proceso.

Figura 3: Simple entrada / salida



1.2.2.2 Mano a mano simple (Handshaking)

La transmisión de datos basada en la simple habilitación es un método que posee cierto control, pero existen inconvenientes sobre todo, cuando la velocidad de transmisión de datos se incrementa, ya que a mayor velocidad de transmisión se necesitan velocidades de conmutación altas, por lo que resulta en errores de lectura / escritura. Como una solución a este inconveniente se usan dos líneas para sostener una comunicación bilateral de mutuo entendimiento entre transmisor y receptor.

En el caso del computador, por ejemplo: El computador envía un dato y por lo tanto su línea STROBE es conmutada para indicar a cierto periférico que el dato está presente, el periférico envía una señal ACKNOWLEDGE para indicar que el dato que se encuentra en sus líneas paralelas ha sido leído, por lo que también el computador al notar la presencia de la señal ACKNOWLEDGE envía el siguiente dato al puerto paralelo, y es así como el proceso de transferencia de datos continúa.

1.2.2.3 Doble mano a mano en la transmisión de datos

El traslado de información puede requerir una forma más compleja de coordinación, enfocando esto en una mejor forma de comunicación entre el transmisor y el receptor. Se puede pensar, que las señales bilaterales llevan el doble de interpretación, la activación de cada una en determinado flanco señala un proceso, es decir en este caso el transmisor no escribe ningún dato sin antes solicitar al receptor una respuesta de si se encuentra o no listo para recibir el dato, cuando la respuesta del receptor es afirmativa, el transmisor puede escribir el dato.

Para tener la certeza de que dicho dato se ha interpretado exitosamente, el transmisor nuevamente pregunta al receptor si el dato ha sido interpretado, si esto ha ocurrido entonces el proceso de transferencia de datos continúa. Estas formas, de comunicación paralela, se establecen internamente entre muchos dispositivos de la "motherboard" y es susceptible de mantenerse entre ésta tarjeta principal y cualquier periférico siempre y cuando se definan, correctamente, el enlace bilateral y las líneas de control.

1.2.4 Estructura y operación del puerto paralelo del PC (tarjeta puerto del impresor)

El PC tiene como parte nominal de su equipo dos puertos, uno paralelo y uno serie, empleados para la comunicación con periféricos. Dichos puertos vienen incluidos en la tarjeta principal del computador.

El puerto paralelo del computador posee 8 líneas de datos para escritura solamente, esto es porque el puerto paralelo se diseñó para la transmisión de datos hacia el impresor, y el impresor no tiene porque enviar datos hacia el computador por éstas líneas. Los datos a ser impresos se envían al impresor en forma de caracteres en código ASCII (8 bits).

Sin embargo en el proceso de impresión, el computador solamente envía datos, en tanto que el impresor tiene que recibir los datos e imprimirlos por lo que llega un momento en que se llena el buffer de datos del impresor, en este instante el impresor tiene que indicar al computador que ya no siga enviando datos, porque de lo contrario se perdería información, es aquí en donde se utilizan las llamadas líneas de control, y el impresor puede utilizar una de ellas para indicar que se encuentra ocupado.

Se adoptó un estándar común para la transmisión de datos por el puerto de impresión hacia los impresores paralelos: El estándar paralelo Centronics.

1.2.4.1 Descripción de la interfaz Centronics

Los conectores Centronics tienen 36 pines. En la tabla I, se ilustra la descripción de los pines. Los pines 2..9 se emplean para la transferencia de datos (palabra de 8 bits) siendo el terminal 2 (D0) el del bit menos significativo.

El resto de los 36 pines del conector se utilizan en dos partes: Señales de control enviadas por el computador hacia el impresor, y señales de estado enviadas por el impresor hacia el computador. Las señales de control del impresor son INIT en el terminal 31, señal que indica al impresor que lleve a cabo su secuencia de inicio interna y STROBE, en el terminal 1, que da el aviso de que existe un caracter a salir del puerto paralelo para que el impresor se prepare.

Dos señales adicionales, el terminal 36 y el 14 se constituyen en señales que entran al impresor para mantener control y comunicación. Es recomendable conectar todos los puntos de tierra del conector a la referencia de la interfaz, unirlos al punto de tierra de la tarjeta, para minimizar corrientes de ruido que se induzcan en el conductor lógico de tierra.

Tabla I: Descripción del conector Centronics

Terminal Impresor	Terminal Comp.	Señal	Descripción
1	1	-STROBE	Indica que un caracter ha sido transmitido.
2	2	D0	Bit 0 del caracter.
3	3	D1	Bit 1 del caracter.
4	4	D2	Bit 2 del caracter.
5	5	D3	Bit 3 del caracter.
6	6	D4	Bit 4 del caracter.
7	7	D5	Bit 5 del caracter.
8	8	D6	Bit 6 del caracter.
9	9	D7	Bit 7 del caracter.
10	10	-ACK	Aproximadamente 5 μ Seg. Indica al PC que el caracter ha sido tomado.
11	11	-BUSY	Un valor en estado bajo indica al PC que no siga enviando datos porque el impresor está ocupado por: Buffer lleno, falta de papel, o error de transmisión.
12	12	PE	Error debido a la falta de papel en el impresor.
13	13	SLCT	Señal que indica al PC que el impresor se encuentra en línea.
14	14	-A. FEED	Un estado bajo en esta señal indica al impresor que corra una línea automáticamente después de cada retorno del carro.
32	15	-ERROR	Error de transmisión de datos.
31	16	-INIT	Un instante en estado bajo indica al impresor, que inicie su rutina de inicialización.
36	17	SLCT IN	Poner el impresor en línea.
19-30	18-25	GND	Masa.

1.2.4.2 Operación del puerto paralelo del PC

Se utilizan básicamente señales dispuestas en transmisión de doble mano a mano. Cuando el impresor ha sido inicializado, la señal BUSY está en estado bajo, indicando que el impresor está listo, aproximadamente 5 microsegundos después se activa la señal STROBE (se coloca en estado bajo), indicándole al impresor que un caracter ha sido enviado, luego el flanco de subida de STROBE, provoca que BUSY conmute. El dato puede considerarse válido 5 microsegundos después del flanco de subida de STROBE. Cuando el impresor está listo para recibir el siguiente caracter, provoca un bajo en la señal ACKNOWLEDGE (conocimiento), siendo el flanco de subida de ésta señal el que inicializa la señal BUSY y una indicación de que el puerto paralelo enviará el siguiente caracter.

1.2.4.3 Consideraciones para el uso del puerto paralelo

En la tabla I, se observan cuales son las líneas de datos y de control, la selección de las líneas a utilizar depende de la interfaz, de manera que al utilizar el conector Centronics en la interfaz se puedan seleccionar las líneas necesarias.

En realidad, la tarjeta de puerto paralelo insertada en el SLOT del computador transmite la información en código ASCII en los procesos de entrada / salida, utilizando la lógica TTL.

1.3 El puerto serie del computador

El puerto serie transmite información bit a bit, a determinada tasa o velocidad, este hecho hace que se pueda enviar información a gran distancia ya que no se necesitan muchas líneas de transmisión como en el caso del puerto paralelo, por lo que por ejemplo usando las líneas de las redes telefónicas, es posible enviar o recibir información a, o de casi cualquier parte del mundo.

1.3.1 Transmisión serial de datos

La transmisión de información entre dos sistemas digitales, por ejemplo un computador y un módem externo se realiza casi siempre caracter por caracter utilizando códigos binarios (ASCII, EBCDIC, BAUDOT). Otras veces la información que se transfiere no corresponde a ninguna codificación de caracteres numéricos o alfanuméricos si no que es puramente binaria.

Los sistemas de comunicación serie han alcanzado un alto grado de normalización, existiendo normas universalmente aceptadas que fijan completamente todos los detalles de comunicación incluyendo también los aspectos mecánicos.

Tabla II: Descripción del estándar RS-232C

Pin	Descripción de la señal	Dirección
1	Tierra de protección	Bidireccional
2	Transmisión de datos	Hacia el DCE
3	Recepción de datos	Hacia el DTE
4	Solicitud para enviar	Hacia el DCE
5	Borrar para enviar	Hacia el DTE
6	Dato listo	Hacia el DTE
7	Tierra de la señal	Bidireccional
8	Detección de portadora	Hacia el DTE
9	Reservado	
10	No asignado	
12	Detección de portadora (secundario)	Hacia el DTE
13	Borrar para enviar (secundario)	Hacia el DTE
14	Transmisión de datos (secundario)	Hacia el DCE
15	Reloj de transmisión	Hacia el DTE
16	Recepción de datos secundario	Hacia el DTE
17	Reloj de recepción	Hacia el DTE
18	No asignado	
19	Solicitud para enviar secundario	Hacia el DCE
20	Terminal de datos lista	Hacia el DCE
21	Detección de calidad de señal	Hacia el DTE
22	Indicador de timbrado	Hacia el DTE
23	Selección de la tasa de datos	Bidireccional
24	Reloj de transmisión	Hacia el DCE
25	No asignado	

1.3.1.1.1 Niveles de tensión

Los niveles de voltaje del RS-232C se ilustran a continuación en la tabla III.

Tabla III: Rangos de voltaje del RS-232C

Condición de la señal	Dato lógico	Rango de voltaje DC
Activa	0	+ 3 a + 25 Voltios
No activa	1	-3 a -25 Voltios

1.3.1.2 Problemas en las transmisiones seriales

1.3.1.2.1 Sincronización de bit

El receptor necesita conocer donde empieza y termina cada bit en la señal recibida para efectuar el muestreo de la misma en el centro de cada celda de bit. La información viaja en una sola línea, por lo que la presencia de varios bits iguales hace que la línea no efectúe ninguna transición y el receptor puede perder la secuencia de donde empieza y acaba cada bit. El problema se acrecienta conforme el número de bits repetidos crece. Es posible resolver este problema por medio de las siguientes formas:

- a) Enviar junto con cada bit transmitido en serie y por la misma línea información adicional que permita al receptor extraer la señal de reloj.

- b) Enviar por una línea independiente de la de datos una señal de reloj que indique el centro de las celdas de bits de la línea de datos.
- c) Lograr por medio de algún procedimiento que los relojes de transmisión y recepción se mantengan en fase continuamente.

1.3.1.2.2 Sincronización de caracter

La información serie se transmite bit a bit, pero la misma tiene sentido en palabras de 8, 16 ó 32 bits, debiendo permitir el sistema distinguir sin problemas dentro de una corriente de bits cuales son los 8, 16 ó 32 bits que forman la palabra. Para obtener sincronización de caracter pueden utilizarse diversos sistemas, unos se basan en la utilización de líneas adicionales a las de datos para enviar impulsos que indican el inicio de un bloque de caracteres, los sistemas de comunicación eficientes se basan en la transmisión síncrona o asíncrona.

1.3.1.2.3 Sincronización de mensaje

En un sistema de comunicaciones, generalmente la información se transmite en bloques de caracteres, teniendo que haber un mecanismo por el cual un conjunto de palabras es interpretado correctamente. Este problema normalmente no incumbe a los circuitos de codificación sino al procesador que lo utiliza.



El conjunto de reglas (protocolo) que permite interpretar correctamente los mensajes es controlado por una rutina de software que ejecuta el computador aunque, de hecho hay circuitos integrados programables vía hardware o software que efectúan algunas de éstas tareas.

1.3.2 RS-232C para interfaces a niveles TTL

La tarjeta RS-232C que contiene al puerto serie, utiliza dispositivos tales como el circuito integrado MC1488 para realizar la conversión de Lógica TTL utilizada por el USART 8251 a lógica ECL, la cual concuerda con el estándar RS-232C. El USART 8251 se puede considerar como el cerebro de la tarjeta RS-232C, ya que realiza operaciones de control sobre la tasa de bits a transmitir / recibir, pero no es compatible directamente con los niveles de señal del estándar RS-232-C, ya que utiliza para su operación la lógica TTL.

Es necesario entonces emplear otros componentes electrónicos para convertir los niveles RS-232C a niveles lógicos TTL, componentes como el circuito integrado MC1489, el cual es un manejador de bus que requiere una alimentación de +5 voltios DC para operar.

2. CONSIDERACIONES DE SOFTWARE

2.1 Generalidades

Se pueden accionar por medio del computador, a través de circuitos de decodificación y control de las líneas disponibles y los conectores ranura de la motherboard, los puertos, circuitería hardware correspondiente a interfaces externas. Es importante notar, que tales condiciones pueden ser programadas y predichas a requerimiento del operador del computador a través de instrucciones de programación en lenguajes de diferente nivel y generación de software, cada lenguaje de los mencionados posee instrucciones capaces de ejecutar secuencias de procesos de entrada / salida desde el teclado, con las ventajas que da el lenguaje y la versatilidad aritmético lógica del PC. Las mencionadas instrucciones son específicas de cada lenguaje, pero en muchas ocasiones existe similitud de un lenguaje a otro.

2.2 Lenguajes de máquina

Reciben el nombre de lenguajes de máquina debido a que el código es interpretado directamente por el computador sin necesidad de traducción previa. Es importante notar que cada computador puede interpretar solamente *su propio* lenguaje de máquina, se puede pensar que el lenguaje de máquina es una especie de lenguaje natural de un computador particular, el cual está íntimamente relacionado con el diseño de hardware adoptado.

El código del lenguaje de máquina consiste básicamente en cadenas de unos y ceros lógicos, que instruyen a los computadores para que ejecuten las instrucciones más elementales, una a la vez. Los lenguajes de máquina son dependientes de la máquina, es decir un lenguaje máquina particular puede ser utilizado en un solo tipo de computador. Es muy difícil para los humanos interpretar el lenguaje de máquina además los programadores tendrían que invertir demasiado tiempo en programas tediosos, los lenguajes de máquina presentan las siguientes desventajas:

- a) Es complicado depurar los programas.
- b) Poca fiabilidad.
- c) Los programas solo son ejecutables en el mismo procesador.

2.3 Lenguajes ensambladores

Debido a que los programas escritos en lenguaje de máquina son demasiado extensos y difíciles de interpretar, los programadores utilizan los llamados lenguajes ensambladores, los cuales se constituyen en abreviaturas similares a un idioma humano (por ejemplo el Inglés) y que se denominan mnemónicos. Esto ocasiona que el tamaño de los programas se reduzca, pero como un computador solo puede interpretar el lenguaje de máquina, es necesario utilizar una especie de "traductor" para convertir el código escrito en lenguaje ensamblador, en lenguaje de máquina, dichos traductores se denominan ensambladores, el programa original escrito en lenguaje ensamblador se denomina programa fuente, y el programa traducido se conoce como programa objeto.

Los lenguajes ensambladores se enfocan en aplicaciones de tiempo real, control de dispositivos electrónicos y procesos. Un ejemplo de lenguaje ensamblador lo constituye el turbo assembler.

2.4 Lenguajes de alto nivel

Los lenguajes de alto nivel son los más utilizados por los programadores alrededor del mundo. Están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes anteriores. Otra razón es que un programa escrito en un lenguaje de alto nivel es independiente de la máquina en la que se escribe, esto es, las instrucciones del programa del computador no dependen del diseño del hardware o de un computador en particular. En consecuencia, los programas escritos en lenguaje de alto nivel son portables y tienen la posibilidad de ser ejecutados con poca o ninguna modificación en diferentes tipos de computadores. El código de un lenguaje de alto nivel se constituye de instrucciones provenientes del lenguaje humano, los más utilizados en nuestro medio son: Turbo Pascal, Borland Pascal, Borland C, Borland C++, y Visual Basic. Análogamente al caso de un lenguaje ensamblador, un lenguaje de alto nivel necesita una especie de traductor, al cual se le denomina compilador. Las desventajas de utilizar programación en lenguajes de alto nivel radican en:

- a) El tiempo de ejecución es mayor, ya que un programa escrito en un lenguaje de alto nivel tiene que ser compilado cada vez que se ejecuta, resta tiempo y memoria que en ciertas aplicaciones son críticos e importantes.

- b) Los recursos internos de un computador no son explotados en su totalidad.

2.5 Manejo de puertos con el lenguaje ensamblador

2.5.1 Proceso de entrada

Es necesario guardar la información entrante al computador, en un registro para su posterior procesamiento, el mnemónico utilizado en este caso es IN, una instrucción en dicho código tiene la siguiente sintaxis:

IN <Registro interno> , <Dirección de puerto en hexadecimal>

2.5.2 Proceso de salida

En este caso se usa el mnemónico OUT, el valor que se desee enviar de un registro interno R, hacia un puerto de datos, debe presentar la sintaxis:

OUT <Dirección del puerto en hexadecimal> , <Registro interno>

2.6 Manejo de puertos con el lenguaje Turbo Pascal

Turbo Pascal, implementa dos arreglos unidimensionales predefinidos, a saber: Port y PortW, para el acceso a los puertos de datos de las UCPs 80X86.

2.6.1 Proceso de entrada

Cuando un elemento de Port o PortW es referido a una expresión, el valor se lee desde el puerto seleccionado.

Valor: = Port [< Dirección en hexadecimal >]

Valor: = PortW [< Dirección en hexadecimal >]

Para este caso valor significa una variable de tipo entero. La diferencia entre Port y PortW radica en la cantidad de bits asignada a la variable valor.

2.6.2 Proceso de salida

Cuando un valor es asignado a un elemento de Port o PortW, el valor es puesto físicamente en el puerto seleccionado;

Port [< Dirección de puerto en hexadecimal >] := Valor.

PortW [< Dirección de puerto en hexadecimal >] := Valor.

2.7 Manejo de puertos con el lenguaje Visual Basic

Este lenguaje de programación es orientado a objetos, dichos objetos se crean fácilmente con el ratón en la llamada interfaz de usuario, con el tamaño, colores, etc. deseados. Visual Basic posee una cantidad considerable de objetos, herramientas e instrucciones para realizar presentaciones bajo el sistema operativo Windows de una manera sencilla y con muy buena presentación. El manejo de los puertos con Visual Basic es bastante sencillo principalmente cuando se trata de puertos serie, porque Visual Basic posee un objeto denominado Microsoft Communications (MSCOMM), en donde se encuentran propiedades tales como la paridad, protocolo de comunicaciones y sus parámetros, y todas las instrucciones básicas para la comunicación entre dos elementos mediante el puerto serial. Cuando se trata de puertos paralelos Visual Basic no incluye instrucciones definidas para la comunicación, por lo que es conveniente entonces utilizar rutinas hechas en otro lenguaje que facilite este tipo de comunicación, por medio de una Librería (extensión de aplicación, DLL).

2.7.1 Proceso de entrada

Aquí se utiliza la propiedad INPUT de MSComm, la cual remueve y entrega una cadena de caracteres desde el buffer de recepción. Esta propiedad no está disponible en tiempo de diseño (es decir solo se puede leer en tiempo de ejecución). Se utiliza la sintaxis siguiente:

```
[form.]MSComm.Input
```

Ejemplo:

```
' Recuperar todos los datos disponibles en el buffer del puerto serie  
Comm1.InputLen = 0  
' Verificar la existencia de datos  
If Comm1.InBufferCount Then  
    ' Leer datos  
    InString$ = Comm1.Input  
End If
```

La propiedad `InputLen` determina el número de caracteres que son leídos por la propiedad `INPUT`. Colocando la propiedad `InputLen` a 0, da como resultado que la propiedad `INPUT` lea el contenido completo desde el buffer de recepción. También es posible utilizar la función denominada `ComInput`, la cual remueve una cadena de caracteres desde el buffer de recepción. La sintaxis utilizada es la siguiente:

```
ComInput(ByVal hWnd As Integer, lpData As Any, ByVal cbData As Integer) As Integer
```

Parámetro	Tipo	Descripción.
<code>HWnd</code>	<code>HWND</code>	Ventana de maniobra de control.
<code>lpData</code>	<code>LPSTR</code>	Puntero largo al inicio del buffer de datos.
<code>CbData</code>	<code>int</code>	La longitud de <code>lpData</code> dada en bytes.

Esta función es equivalente a la propiedad `INPUT`, el valor de retorno de la función `ComInput` es el número de bytes recibidos.

2.7.2 Proceso de salida

Se utiliza la propiedad OUTPUT, que escribe una cadena de caracteres en el buffer de transmisión. Esta propiedad no está disponible en tiempo de diseño. Se utiliza la sintaxis siguiente:

```
[form.]MSComm.Output[ = outString$]
```

El siguiente ejemplo muestra como enviar los caracteres que se teclean, al puerto serie:

```
Private Sub Form_KeyPress (KeyAscii As Integer)
    Comm1.Output = Chr$(KeyAscii)
End Sub
```

No está de más indicar que la función ComOutput escribe una cadena de caracteres en el buffer de transmisión, como se ilustra a continuación.

Sintaxis:

```
ComOutput(ByVal hWnd As Integer, lpData As Any, ByVal cbData As Integer)
As Integer
```

Parámetro	Tipo	Descripción.
HWnd	HWND	Ventana de maniobra de control.
lpData	LPSTR	Puntero largo al inicio del buffer de datos.
cbData	int	La longitud de lpData dada en bytes.

El retorno de la función ComOutput es el número de bytes enviados.

2.7.3 Parámetros de comunicación

La propiedad SETTINGS, coloca y retorna la tasa en baudios, la paridad, la cantidad de bits de datos, cantidad de bits de parada. Se utiliza la sintaxis siguiente:

```
[form.]MSComm.Settings[ = ParamString$]
```

Si la cadena ParamString\$ no está disponible cuando el puerto es abierto, el MSComm genera un error. La ParamString\$ está compuesta de 4 parámetros que tienen el siguiente formato: "Tasa en baudios, paridad, número de bits de datos, número de bits de parada", y tiene por default los parámetros siguientes: "9600,N,8,1". Los valores válidos para las tasas de transferencia son: 110, 300, 600, 1200, 2400, 9600 (default), 14400, 19200, 38400 (reservado), 56000 (reservado), 128000 (reservado), 256000 (reservado). Los valores válidos para la paridad se indican a continuación.

E	Even	Par
M	Mark	Marca
N	Default	Ninguna
O	Odd	Impar
S	Space	Espacio

Los valores para la cantidad de bits de datos son: 4, 5, 6, 7, 8 (default); los valores válidos para la cantidad de bits de parada son: 1 (default), 1.5 y 2.

2.7.4 Protocolo de comunicaciones

Se utiliza la propiedad Handshaking, la cual coloca y retorna el protocolo de transferencia de datos del hardware. Los protocolos válidos en Visual Basic se ilustran en la tabla IV y se utiliza la sintaxis siguiente:

```
[form.]MSComm.Handshaking[ = protocolo]
```

Tabla IV: Protocolos de comunicación en Visual Basic

Parámetro	Valor	Descripción
ComNone	0	(Default), sin mano a mano
ComXOnXOff	1	Transferencia con XON / XOFF
ComRTS	2	Transferencia con RTS / CTS, (solicitud para enviar / borrar para enviar)
ComRTSXOnXOff	3	Transferencia con ambas, solicitud para enviar y XON / XOFF

La transferencia mano a mano se refiere al protocolo interno de comunicaciones por medio del cual los datos son transferidos desde el puerto, hacia el buffer de recepción. Cuando un caracter de datos llega al puerto serie, el dispositivo de comunicaciones tiene que mover dicho caracter al buffer de recepción, entonces es posible leer el dato mediante software. Si no existiese buffer de recepción, y por medio de software se intenta leer cada caracter directamente desde el puerto, es muy probable que se pierda información, debido a que los caracteres pueden llegar muy rápidamente.

2.8 Librerías de acceso dinámico (Dynamic Link Libraries: DLLs)

En los ambientes Windows y modo protegido, las DLLs ofrecen muchas aplicaciones para compartir código y recursos. Una DLL es un módulo ejecutable (extensión DLL) que contiene código y recursos que son utilizados por otras DLLs y otras aplicaciones. Por ejemplo, en el lenguaje Pascal a diferencia de las rutinas hechas en las unidades (Units), las cuales son estáticamente enlazadas, el enlace de una rutina de una librería se hace presente cuando el programa invocador corre, y el enlace se realiza dinámicamente.

En Pascal, el encabezado Library indica al compilador de Borland Pascal que genere un archivo con extensión Dll en vez de uno con extensión Exe. Los procedimientos y / o funciones a ser exportados por una DLL deben ser compilados con la directiva Export.

2.8.1 Creación de librerías en el lenguaje Pascal

Una librería tiene la misma estructura que un programa común de Turbo Pascal, la diferencia consiste en que la librería se inicia con la sintaxis Library <Nombre Valido>; en lugar de utilizar Program <Nombre Valido> por ejemplo:

```
Library < Nombre Librería >;  
  Var
```

```

    < Declaración de Variables >;
Procedure < Nombre de Procedimiento >; Export;
    Var
    < Variables de Procedimiento >
Begin
    < cuerpo del procedimiento >
End
Function < Nombre función >;Export;
    Var
    < variables de la función >
Begin
    <cuerpo>
End;

Exports
    <nombre procedimiento> index 1,
    <nombre función>      index 2;
Begin
End.

```

La palabra clave Export prepara al compilador de turbo para crear la librería. Para realizar la escritura de una DLL en el lenguaje Pascal se tiene que tener el software Borland Pascal y compilar como una aplicación bajo el ambiente Windows, también es posible utilizar la siguiente línea de comando:

```
Bpc /m /cw <Nombre librería>
```

La palabra clave Index, puede sustituir si así se quiere, el nombre de la rutina en la librería, cuando se llama la misma desde alguna aplicación.

2.8.2 Uso de las librerías en Visual Basic

En Visual Basic es necesario utilizar un módulo de nivel (extensión BAS) para declarar las referencias a rutinas externas en una librería. Se recomienda utilizar la siguiente sintaxis:

```
[Public | Private ] Declare Sub NombreRutina Lib " Nombre librería " [Alias "Nombre alias" ]([Lista de argumentos])
```

o también;

```
[Public | Private ] Declare Function NombreRutina Lib " Nombre librería " [Alias " Nombre alias " ] ([lista de argumentos])[As type]
```

En donde el elemento Public es utilizado para declarar las rutinas que son disponibles para todos los procedimientos en los demás módulos. El elemento Private es utilizado para declarar rutinas que son disponibles solamente en el módulo en donde la declaración es hecha; el elemento Sub indica que la rutina no regresa ningún valor, pero el elemento Function indica que la rutina si retorna un valor que puede utilizarse por ejemplo en una expresión; Lib indica que una DLL contiene la rutina recién declarada, la cláusula Lib es necesaria para todas las declaraciones; Alias indica que la rutina llamada tiene otro nombre en la DLL, esto es útil cuando el nombre de la rutina externa es una palabra clave. Nombre alias es el nombre de la rutina en la DLL. La lista de argumentos, lista las variables y argumentos que son cedidos a la rutina cuando se llama. La sintaxis utilizada es la siguiente:

```
[Optional][ByVal | ByRef][Arreglo de parámetros] varname[( )][As type]
```

Opcional (optional), indica que el argumento no es requerido. Si se utilizan todos los argumentos subsecuentes en la lista de argumentos deben también ser opcionales y se deben declarar usando la palabra clave Optional. Todos los argumentos opcionales deben ser del tipo variante (Variant); Byval indica que el argumento es cedido por valor; Byref indica que el argumento es cedido por referencia; el arreglo de parámetros es usado solamente como el último argumento en la lista de argumentos para indicar que el final del argumento es un arreglo opcional de elementos variantes.

La palabra clave ParamArray, permite ceder un número arbitrario de argumentos, no deben utilizarse con Byval, Byref o opcional; y finalmente, el nombre de variable (Varname) representa la variable cedida a la rutina.

3. DESCRIPCIÓN GENERAL DE LA TARJETA A DIAGNOSTICAR

3.1 Generalidades

La tarjeta bajo estudio es utilizada para el control de los rectificadores que cargan los bancos de baterías de las centrales digitales Axe-10 Ericsson. Dicha tarjeta contiene las siguientes etapas principales:

- a) Una fuente de alimentación para sus circuitos.
- b) Un monitor de tensión para sobre voltaje.
- c) Un monitor de tensión para bajo voltaje.
- d) Un temporizador de carga y divisores de frecuencia.
- e) Un circuito para desactivar los rectificadores en caso de presentarse una sobre temperatura en las baterías.

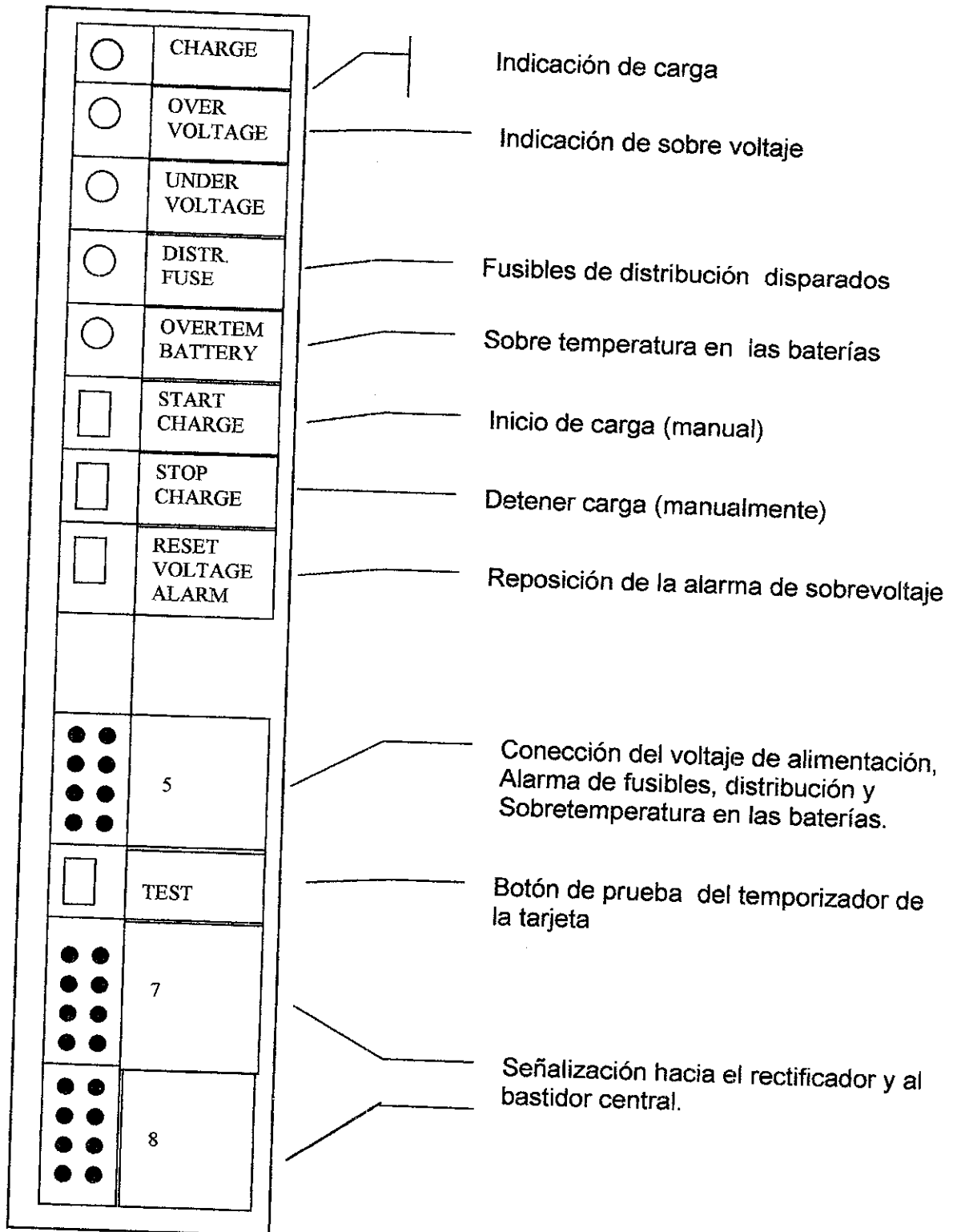
La tarjeta se puede alimentar según su tipo, con 2 tensiones básicas: -48 ó -24 voltios DC, que para el caso de las centrales digitales del sistema Axe-10 existentes en Guatemala, la tensión de operación de las unidades sensoras de carga y monitoras de alarmas adoptada es de -24 voltios. La tarjeta se conecta al bastidor central y a los rectificadores por medio de un conector PLUG IN disponible en la carátula de la misma. Por medio de dicho conector se alimenta la tarjeta y se obtienen las señales de control y monitoreo de los rectificadores, la unidad ofrece 14 puntos de prueba, numerados desde TP1 hasta TP14.

Dichos puntos de prueba están disponibles para las etapas de alimentación, temporización, divisores de frecuencia y monitores de tensión en una base doble en línea de 14 terminales existente sobre el circuito impreso de la misma.

3.2 Descripción mecánica

La unidad consiste de una tarjeta de circuito impreso con LEDs y botones de operación localizados en la carátula de la tarjeta, según se ilustra en la figura 4. La unidad también posee terminales de entrada y salida que se utilizan para el control de rectificadores, alimentación de la tarjeta, indicaciones de alarma de bajo y sobre voltaje.

Figura 4: Descripción mecánica de la unidad Ericsson

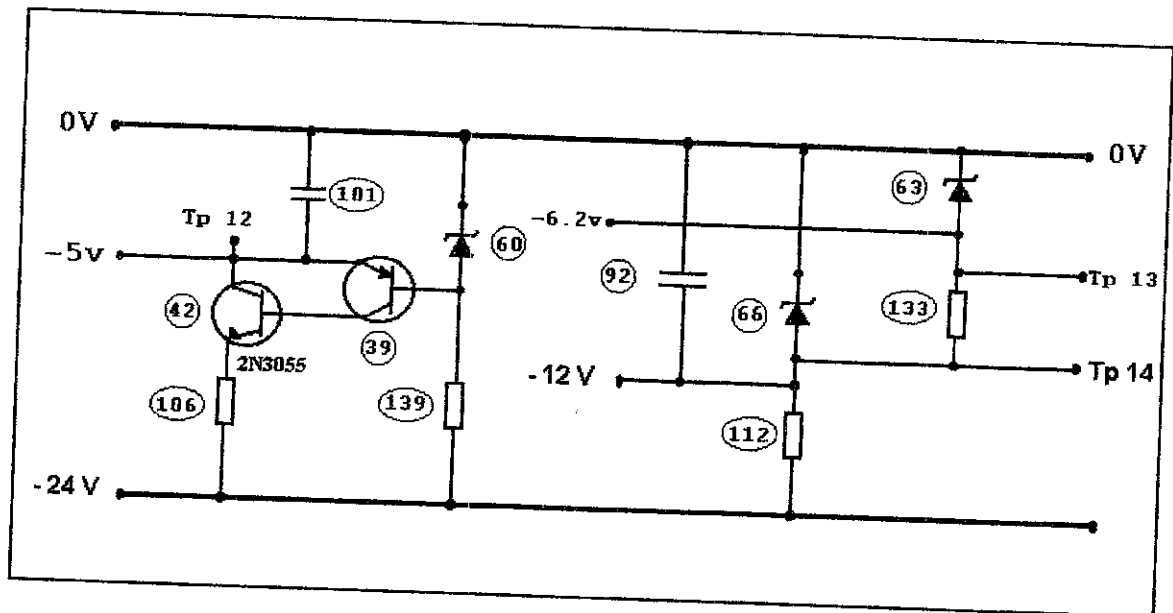


3.3 Descripción de funcionamiento

3.3.1 Fuente de alimentación

En la figura 5, se ilustra la etapa de alimentación de la unidad. La tensión de -24 voltios es la tensión nominal de alimentación de la unidad, las fuentes de -6 y -12 voltios utilizan diodos zener.

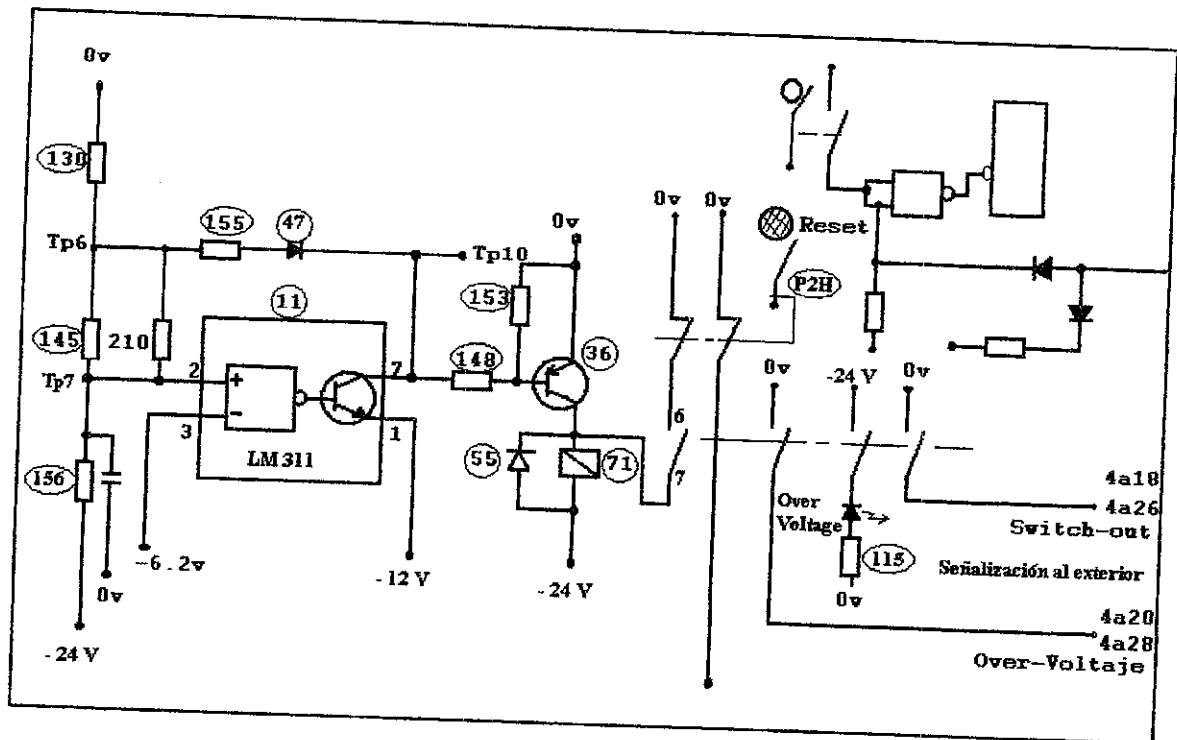
Figura 5: Etapa de alimentación de la unidad Ericsson



La fuente de alimentación de la unidad provee cuatro voltajes para la operación de las etapas, los valores de los voltajes y su aplicación se indican a continuación:

Dicha tensión ocasiona que en la salida (terminal 7) del comparador, aparezcan -12 voltios que es la tensión de alimentación de la etapa comparadora que lleva a saturación el transistor 36 que activa un relé 71 de funciones y alarmas. La etapa se ilustra simplificada en la figura 6.

Figura 6: Diagrama esquemático del monitor de sobre voltaje



El relé de funciones y alarmas ocasiona que las siguientes señales aparezcan:

- a) Por medio de un contacto normalmente abierto, se envía una tensión de 0 voltios (Over Voltage) hacia el bastidor central para que realice funciones de alarmas.

- b) Por medio de un contacto normalmente abierto, se envía una tensión de 0 Voltios (Switch Out), hacia los rectificadores, para apagarlos.
- c) Por medio de un contacto normalmente abierto, se enciende un Led de alarma situado en la carátula de la unidad.

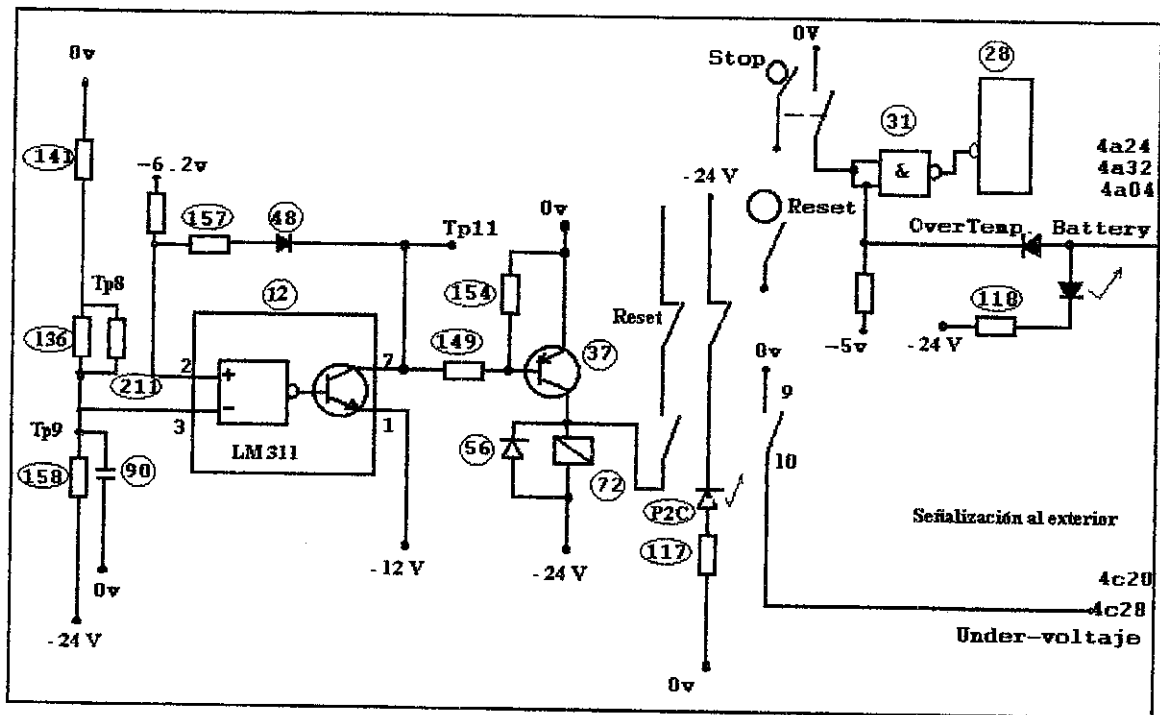
Las primeras dos señales antes mencionadas (incisos a y b) están disponibles en terminales ubicadas en la carátula de la unidad. Las tensiones altas se presentan generalmente cuando existen fallas en los rectificadores, por lo que es necesario apagarlos para evitar daños a las cargas de dichos rectificadores. Si por alguna razón la tensión retorna a su valor nominal, los rectificadores no se pondrán en marcha debido a que el relé de sobre voltaje utiliza uno de sus contactos para auto sujeción. Sin embargo, es posible la modificación de la opción de auto sujeción, para que los rectificadores inicien su operación una vez que la tensión se normalice, esto se puede realizar quitando un puente sobre el circuito impreso de la unidad. Si las cargas de los rectificadores incluyen bancos de baterías, la opción de auto sujeción es necesaria. La etapa ofrece 3 puntos de prueba: TP6, TP7 del divisor de tensión, y TP10 a la salida del LM311.

3.3.3 Monitor de tensión para bajo voltaje

La etapa está constituida por el circuito comparador integrado LM311. El voltaje de referencia para la entrada inversora se toma de una red divisora de tensión, que se alimenta con la tensión de entrada, ésta tensión en condiciones nominales de funcionamiento posee el valor de -24 voltios, para dichas condiciones el LM311 entrega en su salida un estado alto.

Dicho estado alto es de 0 voltios, por lo que la etapa manejadora no excita el relé 72 de bajo voltaje. En la entrada no inversora está presente una tensión de -6 voltios como referencia. El diagrama esquemático simplificado de la etapa se ilustra en la figura 7.

Figura 7: Diagrama esquemático del monitor de bajo voltaje



Cuando la tensión de entrada comienza a disminuir, hasta alcanzar el valor prefijado de -22.5 voltios, la salida del LM311 va a estado bajo, con lo cual se activa la etapa manejadora del relé 72 de bajo voltaje, que genera como resultados:

- a) Por medio de un contacto normalmente abierto, se activa un led de alarma ubicado en la carátula de la unidad.

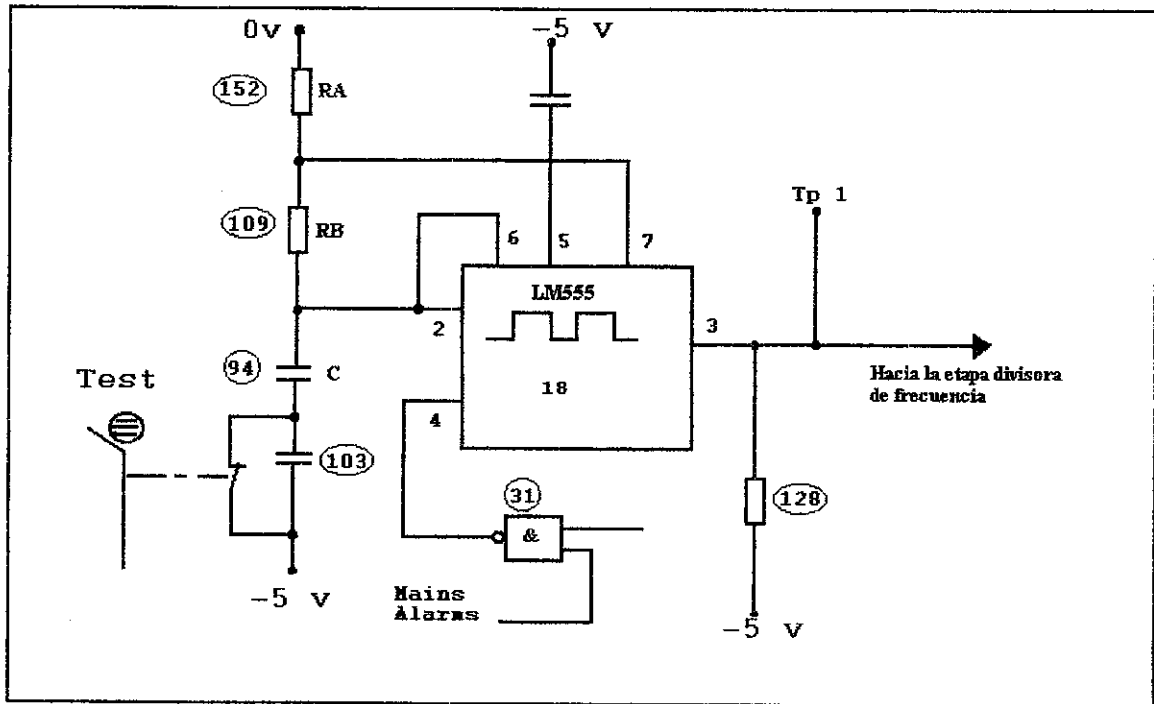
- b) Por medio de un contacto normalmente abierto, se envía a el bastidor central un voltaje de 0 voltios, como señal de bajo voltaje para que ejecute funciones de alarma.

La señal de alarma de bajo voltaje aparece en las terminales 4C20 y 4C28 situadas en la carátula de la unidad. Cuando la condición de bajo voltaje de entrada se hace presente, no se interrumpen los ciclos de carga normal. Si la tensión aumenta a -23 voltios, la condición de bajo voltaje se restablece automáticamente, desenergizándose el relé 72 de bajo voltaje. La etapa brinda 3 puntos de prueba: TP8, TP9; de la red divisora de tensión, TP11 en la salida del LM311 y entrada a la etapa manejadora de relé.

3.3.4 Temporizador de los ciclos de carga

Esta etapa utiliza el conocido circuito integrado 555, que en la tarjeta bajo estudio está configurado para operación estable. El temporizador ocasiona que se envíe una señal de arranque y paro a los rectificadores del sistema a determinados intervalos de tiempo; al intervalo comprendido entre la señal de arranque y la señal de paro se le denomina intervalo de carga, y al intervalo de tiempo comprendido entre dos intervalos de carga se le denomina intervalo inactivo, y durante este tiempo los rectificadores se encontrarán apagados. El diagrama esquemático del temporizador se ilustra en la figura 8.

Figura 8: Diagrama esquemático del circuito oscilador



3.3.4.1 Parámetros principales de funcionamiento

3.3.4.1.1 Frecuencia

La frecuencia de oscilación está dada por la relación:

$$f_{555} = 1/T = 1.44 / [(R_A + 2 * R_B) * C]$$

que para el caso de ésta unidad;

$$f_{555} = \frac{1.44}{(21.5k + 2 * 464k) * 10\mu F} = \frac{1.44}{9.495}$$

$$f_{555} = 0.15 \text{ Hz} = 150 \text{ mHz.}$$

No obstante, cuando el técnico reparador presiona el botón rotulado "test", abre un contacto normalmente cerrado y coloca un condensador de 470 pF, en serie con el condensador de 10 μ F, por lo que se obtiene una capacitancia resultante de:

$$C = \frac{10\mu\text{F} * 470\text{pF}}{10\mu\text{F} + 470\text{pF}} \cong 470\text{pF}$$

Entonces para éste caso, en condición de prueba manual, la frecuencia de los ciclos de carga se acelera por el factor:

$$f_{\text{PRUEBA}} / f_{\text{NOMINAL}} = 10\mu\text{F} / 470\text{pF} = 21000.$$

Este botón es muy útil cuando se está comprobando manualmente si la tarjeta realiza correctamente los ciclos de carga.

3.3.4.1.2 Tiempo en estado alto

El tiempo en estado alto está dado por la relación:

$$T_H = 0.693 * (R_A + R_B) * C$$

El cual, para el caso de ésta unidad es:

$$T_H = 0.693 * (21.5\text{k} + 464\text{k}) * 10\mu\text{F}$$

$$T_H = 3.36 \text{ Seg.}$$

3.3.4.1.3 Tiempo en estado bajo

El tiempo en estado bajo se calcula por la relación:

$$T_L = 0.693 * R_B * C$$

$$T_L = 0.693 * 464k * 10\mu F =$$

$$T_L = 3.22 \text{ Seg.}$$

3.3.5 Etapa divisora de frecuencia

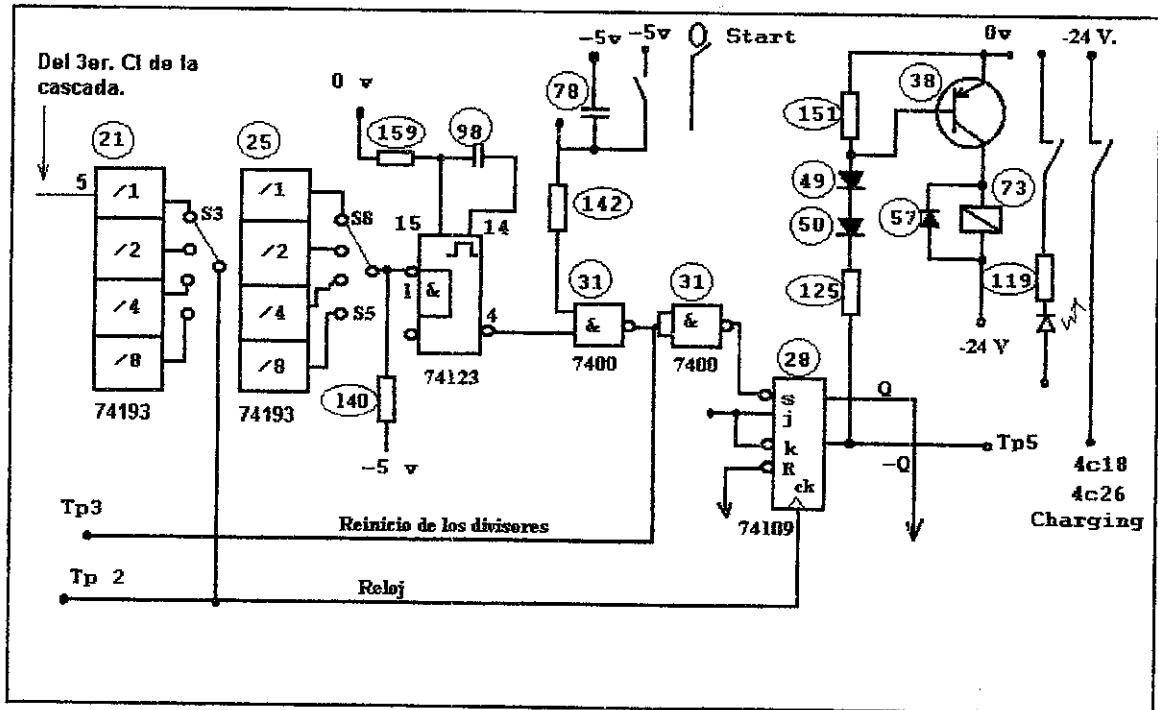
La etapa está formada por los siguientes elementos básicos:

- a) 5 Contadores síncronos de 4 bits (74193) conectados en cascada.
- b) Un multivibrador monoestable (74123).
- c) Un flip-flop \overline{JK} (74109).
- d) 3 compuertas NAND de 2 entradas (7400).

El 74193 es un contador síncrono binario de 4 bits (la información acerca de su operación se incluye en el apéndice de este trabajo de tesis). La cuenta se realiza ascendentemente. Al conectar los 74193 en cascada se puede dividir la frecuencia entre los múltiplos de 2, según se requiera.

En la figura 9 solo se muestran los últimos 2 CIs 74193 y los elementos principales de la etapa: 21, 25, 14, 28.

Figura 9. Diagrama esquemático del divisor de frecuencia y circuito de carga



3.3.5.1 Descripción funcional de la etapa

El primer CI 74193 de la etapa divisora obtiene la frecuencia base de la terminal 3 del CI 555 en la etapa temporizadora. Las diferentes salidas D_n de los 74193 se utilizan convenientemente en la unidad para obtener diferentes intervalos de carga e inactivos. El CI 74123 es un multivibrador monoestable, por medio del cual se logra fijar los intervalos inactivos.

El CI 74109 es un flip-flop Jk, conectado como flip-flop D, por medio del cual se activan los intervalos de carga. Al presionar el botón de inicio de carga, por medio de una compuerta NAND de dos entradas (31) se envía una señal de corta duración en estado alto a las terminales de reinicio de los CIs 74193, para reiniciar el conteo, dicha señal aparece en el TP3 y pasa por un inversor y se envía a la terminal \bar{S} del CI 74109, de manera que el flip-flop va al modo posicionar, activando el relé 73 de carga.

Cierto tiempo más tarde, después que se ha dividido la frecuencia un determinado número de veces en los CIs 74193, aparece en TP2 una señal en estado alto, que opera como señal de reloj para el flip-flop D, (cuya señal D está permanentemente en estado bajo) por lo que en la salida \bar{Q} del flip-flop y TP5 aparece un estado alto, y se desenergiza el relé de carga. Puesto que el temporizador sigue oscilando, antes de que finalice el intervalo inactivo (el cual es mucho mayor que el intervalo de carga) se presentarán varios pulsos de reloj para el flip-flop D, no obstante, no se energiza el relé 73 de carga porque la señal D tiene un estado bajo permanente.

Al finalizar el periodo inactivo el último CI 74193 en cascada (25) envía una señal de disparo en estado alto hacia el multivibrador monoestable, la señal pasa por la compuerta NAND de dos entradas y la salida de dicha compuerta se utiliza nuevamente como señal de reinicio para todos los divisores de frecuencia y la señal "posicionar" al flip-flop D, de manera que el ciclo se repite indefinidamente, en condiciones de funcionamiento normales.

3.3.5.2 Ciclos de carga

3.3.5.2.1 Tabla de los ciclos de carga

Los primeros 2 CIs 74193 se conectan como divisores de frecuencia entre 16, el tercer CI se conecta como divisor de frecuencia entre 8 y como divisor de frecuencia entre 16; el cuarto y quinto CI se conectan como divisores de frecuencia entre cualquiera de los valores: 2, 4, 8 ó 16.

En realidad, es posible construir una tabla útil para la duración de los intervalos de carga e inactivos para analizar la opción más adecuada para los sistemas, además también es útil cuando se utiliza el botón para la prueba manual del temporizador y los tiempos se reducen aproximadamente por un factor de 21000. Por ejemplo, para calcular la duración mínima de un intervalo de carga;

$$\begin{aligned} T &= 16 * T_{555} * 16 * 8 = \\ &= 16 * 6.6 \text{ Seg.} * 16 * 8 = 13517 \text{ Seg.} \\ &= 3.75 \text{ horas.} \end{aligned}$$

Y para calcular la duración mínima entre dos intervalos de carga;

$$\begin{aligned} T &= 16 * T_{555} * 16 * 16 * 16 * 2 \\ T &= 16 * 6.6 \text{ Seg.} * 16 * 16 * 16 * 2 = 240 \text{ horas} \\ T &= 10 \text{ días.} \end{aligned}$$

Las duraciones se pueden seleccionar, utilizando los puentes de la unidad, a continuación la tabla V muestra las duraciones y los puentes a utilizar (Straps).

Tabla V: Duraciones de los ciclos de carga e inactivos

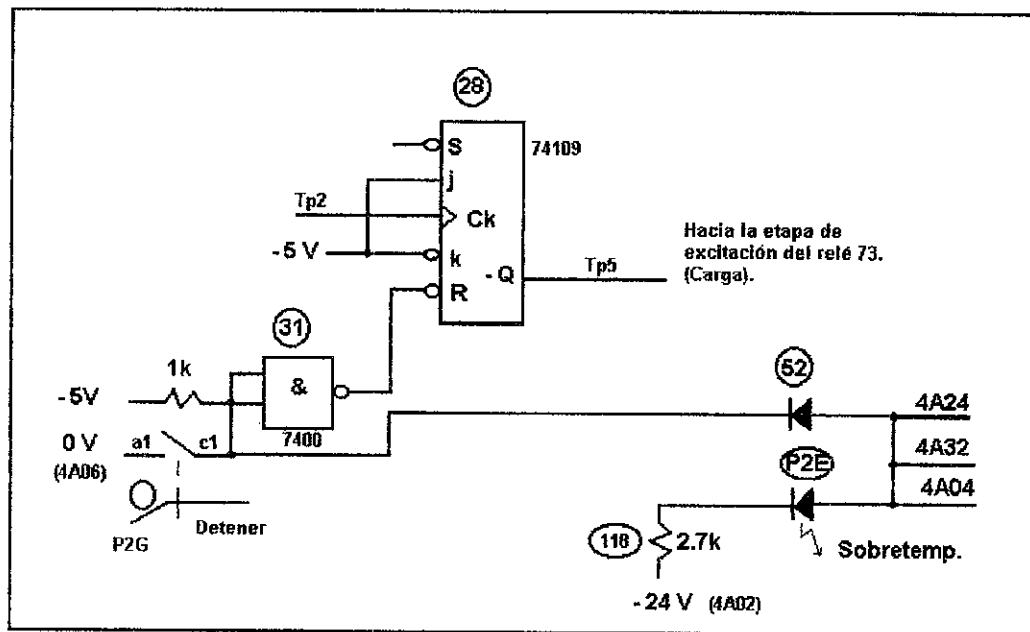
Puente	Función	Tiempo normal	Tiempo en condición de prueba (Aprox.)
S1	Intervalo de carga	30 horas	5 Seg.
S2	Intervalo de carga	15 horas	2.5 Seg.
S3	Intervalo de carga	7.5 horas	1.25 Seg.
S4	Intervalo de carga	3.75 horas	0.6 Seg.
S5	Intervalo inactivo	80 días	5.4 minutos
S6	Intervalo inactivo	40 días	2.7 minutos
S7	Intervalo inactivo	20 días	1.3 Seg.
S8	Intervalo inactivo	10 días	40 Seg.

3.3.6 Etapa de desactivación del ciclo de carga en condiciones de sobre temperatura

La etapa consta de una compuerta NAND de 2 entradas conectada como inversor, la señal de alarma de 0 voltios, proviene del exterior y se conecta a las terminales 4A24, 4A32, 4A04 en la carátula de la unidad, dicha señal, pasa por el diodo 52, activa el LED P2E, se invierte en la compuerta NAND y reposiciona el flip-flop 28, lo cual da por resultado la desactivación del relé 73 y por consiguiente que los rectificadores se apaguen.

Dicha condición se sostiene hasta que desaparece la señal de alarma (se normaliza la temperatura de las baterías). Esta etapa es muy simple, como se observa en la figura 10.

Figura 10: Diagrama de la etapa de sobre temperatura



En contraste a su simplicidad, esta etapa es muy importante y es recomendable revisar la operación del circuito, debido a que los sobrecalentamientos en las baterías disminuyen la vida útil de las mismas.

3.4 Descripción de la ubicación de los puntos de prueba

Los puntos de prueba (en total 14) son accesibles en una base tipo DIP en el circuito impreso de la unidad. En la tabla VI se describen los puntos de prueba.

Tabla VI. *Ubicación de los puntos de prueba en la unidad*

Pin base	Punto de prueba	Significado
1	TP14	Tensión de alimentación de los circuitos comparadores LM311 (-12 voltios).
2	TP8	Tensión en la red divisora de tensión para el comparador de bajo voltaje.
3	TP9	Tensión en la red divisora de tensión para el comparador de bajo voltaje.
4	TP10	Salida del comparador de sobre voltaje, los valores pueden ser -12 ó 0 voltios DC.
5	TP11	Salida del comparador de sobre voltaje, los valores pueden ser -12 ó 0 voltios DC.
6	TP13	Tensión de -6 voltios DC, utilizada como tensión de referencia para las etapas de bajo y sobre voltaje.
7	TP12	Tensión de -5 voltios (respecto a la tierra positiva), para la alimentación de los circuitos digitales.
8	TP5	Señal de salida -Q, del flip-flop D para la activación de la etapa manejadora del relé de carga.
9	TP1	Frecuencia de 0.15 Hz, proveniente de la terminal 3 del CI 555.
10	TP7	Tensión en la red divisora para la etapa de sobre voltaje.
11	TP6	Tensión en la red divisora para la etapa de sobre voltaje.
12	TP2	Estado de la señal de reloj para el flip-flop D.
13	TP3	Estado de la señal de reinicio de los divisores de frecuencia.
14	TP4	Estado de la señal de disparo para el multivibrador monoestable.

3.5 Descripción de las terminales de entrada y salida de la tarjeta de los rectificadores Ericsson

En la carátula de la unidad se puede acceder a las terminales de entrada y salida con un conector "Plug In". En la tabla VII se describe el significado de las terminales.

Tabla VII: Descripción de las terminales de entrada y salida en la unidad

Term.	Nombre	Significado
4C18 4C26	Charging	Señal de salida, de -24 voltios que controla los rectificadores.
4C24 4C32 4C04	Alarm Distr. Fuse	Señal de entrada de 0 voltios, que enciende un led de indicación en la carátula de la unidad.
4A24 4A32 4A04	Over- Temp. Battery	Señal de entrada de 0 voltios, que indica a la unidad que existe una sobre temperatura en las baterías, por lo que se deben apagar los rectificadores.
4A18 4A26	Switch- Out	Cuando se presenta un sobre voltaje, esta señal de salida apaga los rectificadores.
4A20 4A28	Over voltage	Señal de salida que, en una condición de sobre voltaje, se envía a el bastidor central.
4C20 4C28	Under- Voltaje	Señal de salida de 0 voltios, que en una condición de bajo voltaje, se envía a el bastidor unidad central.
4A06 4C06	0 V	Entrada de alimentación.
4A02 4C02	-24 V	Entrada de alimentación.

4. DESCRIPCIÓN DE LA INTERFAZ Y PRUEBAS A LA TARJETA ERICSSON

4.1 Generalidades

La interfaz utiliza circuitos digitales para el diagnóstico de la tarjeta de los rectificadores Ericsson. Por medio de software se envía al puerto paralelo del PC (de impresión) una palabra de control de interfaz de 8 bits de la cual se utiliza solo un bit habilitador para ejecutar la prueba. En el formulario principal de la aplicación Tarjeta.exe, se sugiere al usuario el orden de prioridad de la ejecución de las pruebas (primero la prueba de la etapa de alimentación), sin embargo, se pueden realizar las pruebas en el orden que se desee, activando con el ratón del PC el botón de comando deseado. La interfaz, en respuesta a la palabra de control genera un resultado, que aparece en las líneas ERROR, SLCT, PE, ACKN y BUSY.

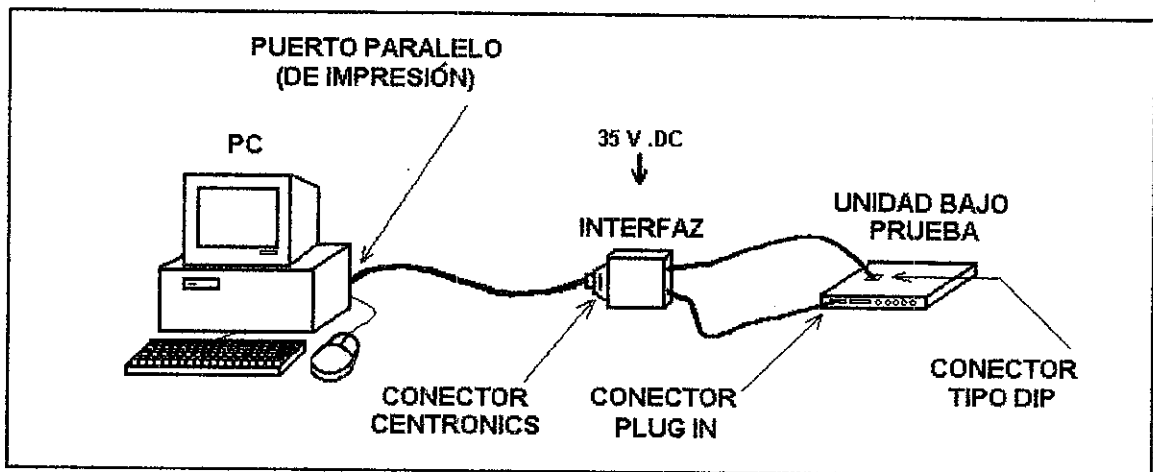
Como se ilustrará en el capítulo 5, dicho resultado se lee vía software, el cual realiza entonces una búsqueda en el archivo de parámetros eléctricos para interpretar el significado de la lectura y que finalmente aparezca en pantalla el mensaje de diagnóstico. Excepto en la prueba del oscilador, un estado lógico alto presente en las líneas: ERROR, SLCT, PE, ACK y BUSY, es interpretado por el software como resultado satisfactorio (dispositivo o etapa bajo prueba operando correctamente). La interfaz se alimenta con una tensión de 35 V DC, dicha tensión es suficiente para alimentar los circuitos de la interfaz y para alimentar la tarjeta bajo prueba, la interfaz se conecta al PC mediante un cable común de impresor. La unidad bajo prueba se conecta a la interfaz mediante un conector PLUG IN y un conector tipo DIP de 14 terminales.

El diagnóstico se realiza por medio de cinco pruebas:

1. Prueba de los voltajes de alimentación.
2. Prueba del temporizador.
3. Prueba de los valores lógicos en condición de carga.
4. Prueba de bajo voltaje y alarma de sobre temperatura.
5. Prueba de sobre voltaje.

La conexión de los distintos dispositivos se puede llevar a cabo según se ilustra en la figura 11.

Figura 11. Diagrama pictórico de la conexión de prueba

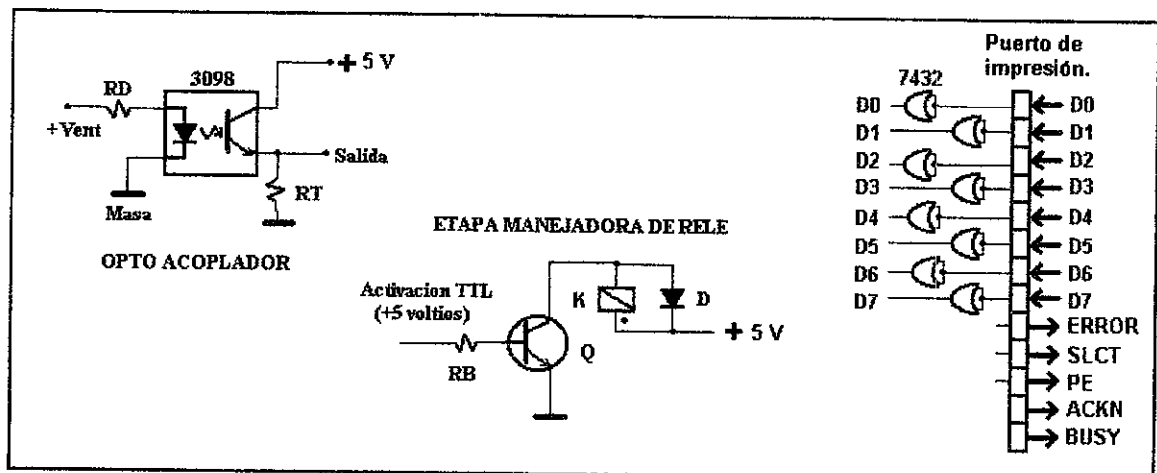


4.2 Consideraciones generales de diseño de hardware

4.2.1 Protección del puerto de impresión

Como se muestra en la figura 12, para proteger el registro de escritura del puerto de impresión contra señales dañinas tales como sobre tensiones o sobre corrientes, se ha aislado dicho puerto por medio de compuertas 7432.

Figura 12: Etapas utilizadas en el diseño de la interfaz



4.2.2 Protección de los buffers tri-estado de la interfaz

Se ha incluido en el diseño un relé que alimenta la etapa de buffers de la interfaz solamente por medio del software de control Tarjeta.exe, debido a que se han utilizado buffers tri-estado que se activan con un estado bajo aplicado a su terminal de habilitación.

Si por alguna razón el computador se reinicia cuando la interfaz se encuentra energizada y conectada al puerto, el computador por default envía una palabra de 8 bits que activa más de un conjunto de buffers tri-estado, lo cual ocasionaría que sus salidas se conectaran en corto circuito.

4.2.3 Etapas manejadoras de relé

En la figura 12 se ilustra la etapa manejadora de relé utilizada, el transistor Q, manejador del relé K es del tipo 123AP, el cual puede manejar nominalmente una corriente de colector de 600mA, y demanda una corriente de entrada mínima de 3 mA para su correcta operación. Los relés utilizados son de 5 voltios, los cuales consumen aproximadamente 100mA < 600mA, por lo tanto, los transistores no se sobrecargan. Para calcular el valor de la resistencia de base (R_B):

$$R_{B \text{ MÁX}} = [V_{CC} - V_{BE}] / I_{ENT \text{ mín}} = [5 - 0.6] / 0.003 = 1400\Omega \approx 1k\Omega,$$

Para proteger el transistor contra sobre tensiones originadas en la bobina del relé manejado al conmutar, se conecta el diodo D (tipo 1N4148) en paralelo a la bobina del relé.

4.2.4 Acoples ópticos

Para los acoples ópticos se utilizó la configuración ilustrada en la figura 12, con opto-aisladores del tipo 3098.

4.2.4.1 Diodo emisor

El diodo emisor soporta una corriente máxima de 60 mA, y se polariza con una tensión de 1.5 voltios, sin embargo, considerando que dicho diodo opera correctamente con una corriente de aproximadamente 7 mA, se calcula la resistencia R_D :

$$R_{D \text{ MÁX}} = [V_{\text{ENT}} - V_{\text{POLARIZACIÓN}}] / I_{\text{MÍN}} = [12 - 1.5] / 0.007 = 1500 \Omega$$

$R_D \approx 1.3 \text{ k}\Omega$, si la tensión V_{ENT} es de 24 voltios, se utilizará $R_D = 2.7 \text{ k}\Omega$.

La tensión inversa máxima para el diodo es de 5 voltios, pero en éste caso no puede existir inversión de polaridad debido a que, o bien el ánodo se conecta permanentemente a la referencia de 0 voltios (positiva), o el cátodo se conecta permanentemente a la terminal negativa de la alimentación (-24 voltios).

4.2.4.2 Transistor

La etapa del transistor puede manejar una carga de hasta 50 mA, sin embargo, la corriente de entrada de las compuertas TTL oscila entre 1 y 2 mA, siendo 2 compuertas el abanico máximo de entrada conectadas a un mismo opto acoplador, se calcula la resistencia R_T :

$$R_{T \text{ MÁX}} = [V_{\text{CC}} - V_{\text{UMBRAL}}] / I_{\text{MÍN}} = [5 - 0.6] / (2 * 0.002) = 1100 \Omega$$

$$R_T \approx 1 \text{ k}\Omega$$

4.2.5 La señal de conocimiento del PC

El computador, al enviar la palabra de control a la interfaz, no sabe en que momento se encuentra disponible el resultado de una prueba determinada, para proceder a su lectura. Si la tarjeta a diagnosticar posee únicamente compuertas lógicas, es posible calcular el retardo total de propagación e implementar un retardo utilizando por ejemplo un número determinado de buffers en cascada, sin embargo, como en este caso, la tarjeta posee dispositivos electromecánicos (relés), los cuales poseen tiempos de retardo relativamente más grandes de propagación de señales (dados en milisegundos), por lo que el problema es un poco diferente. Una solución a dicho problema podría ser: conectar relés para retardar la señal de conocimiento ya sea uno solo o en cascada (el anterior, por medio de un contacto activaría el posterior) dependiendo del retardo requerido. En este trabajo de tesis, no se utiliza señal de conocimiento para el PC, el software genera una pausa (1 segundo), la cual es suficiente para proceder a la lectura del resultado.

4.2.6 Etapa del regulador de voltaje variable

Esta etapa se ha diseñado para suministrar 3 tensiones: -24 , -22.5 , y -33 voltios, en función del valor de la resistencia de ajuste en curso. El origen de la ecuación utilizada para el diseño se encuentra ilustrado en el apéndice. Este regulador puede suministrar 1.5 amperios, que es suficiente para alimentar la tarjeta bajo prueba debido a que dicha tarjeta consume 1.3 amperios como máximo.

4.3 Palabras de control de la interfaz

La interfaz utiliza para cada una de las operaciones, una palabra de control, en la tabla VIII se puede observar la descripción de las palabras de control de interfaz.

Tabla VIII. *Palabras de control de la interfaz*

Palabra		Aplicación
Binario	Decimal	
11111110	254	La interfaz realiza la primera fase en la comprobación de los voltajes de la unidad.
11111101	253	La interfaz realiza la segunda fase en la comprobación de los voltajes de la unidad.
11111011	251	La interfaz verifica la señal temporizadora.
11110111	247	La interfaz comprueba los valores lógicos de la unidad Ericsson en estado de carga.
11101111	239	La interfaz realiza la prueba de bajo voltaje y alarma de sobre temperatura.
11011111	223	La interfaz realiza la prueba de sobre voltaje.
01111111	127	La interfaz apaga la unidad bajo prueba
10111111	191	La interfaz apaga sus buffers tri-estado.
00111111	63	La interfaz desenergiza sus buffers tri-estado, y la unidad bajo prueba.

4.4 Prueba de los voltajes de alimentación

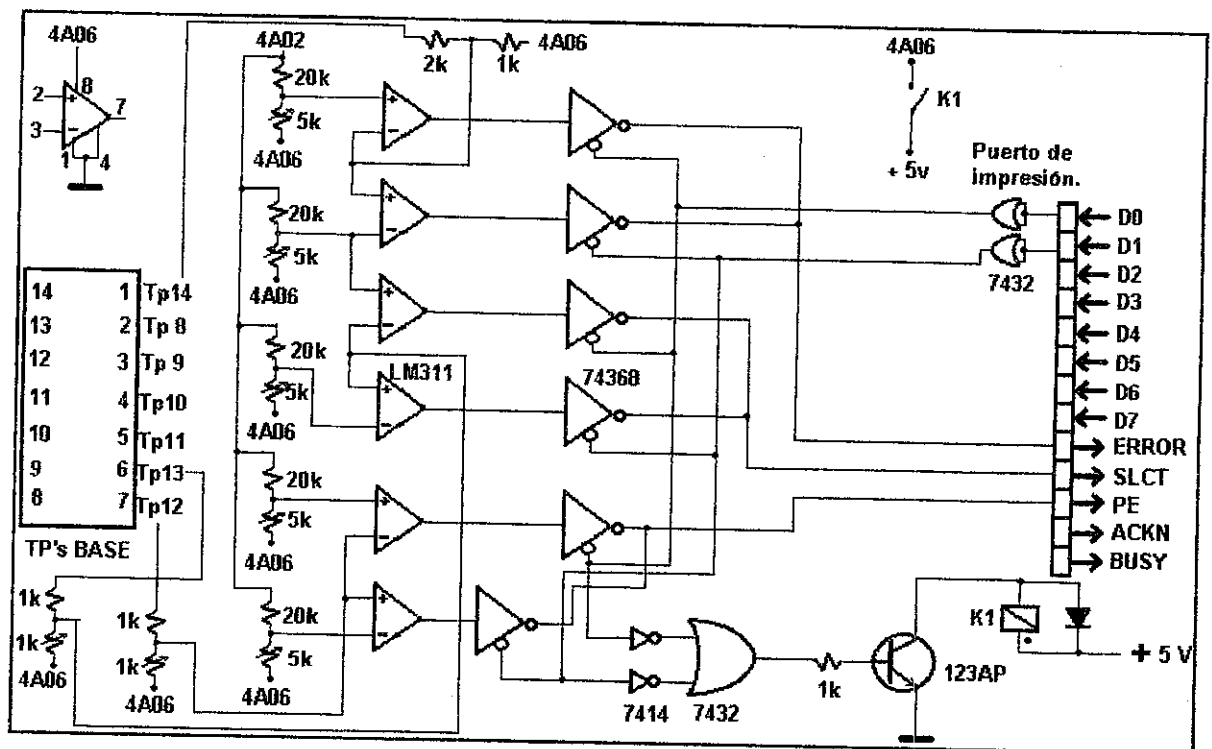
Esta prueba verifica que las tensiones de alimentación para las distintas etapas de la tarjeta sean correctas, para lograr esto la prueba se realiza en dos fases. A manera de ejemplo, si se quiere verificar que una fuente de +5 voltios está correcta, un criterio de prueba podría ser: utilizar un comparador de tensión para verificar que la tensión que está presente en su entrada no inversora sea mayor que 4.75 voltios (tensión de referencia baja), y otro para verificar que la tensión de la fuente, presente en su entrada inversora sea menor que 5.25 voltios (tensión de referencia alta); al cumplirse estas dos condiciones se tendrá bastante certeza, al afirmar que la fuente de alimentación está operando correctamente.

4.4.1 Fase 1

Inicialmente, se envía una palabra de control por software a la interfaz (254 decimal) la cual habilita tres "buffers" tri-estado, cada entrada de los buffers proviene de la salida de etapas comparadoras de tensión LM311. Existe una red divisora de tensión para que las etapas comparadoras tomen un voltaje de referencia y la otra entrada proviene de un divisor de tensión entre 3 para la tensión de -12 voltios, y divisores de tensión entre 2 para las tensiones de -6 y -5 voltios; la razón de dividir la tensión proveniente de la tarjeta Ericsson entre 3 y 2 es porque el suministro de las etapas comparadoras de la interfaz es de +5 voltios para que exista compatibilidad entre las salidas de dichas etapas y la tecnología TTL de los buffers.

Finalmente, se leen los tres bits entregados por los buffers en las líneas ERROR, SLCT y PE para ser interpretados vía software. Los elementos involucrados se ilustran en la figura 13.

Figura 13. Etapa de prueba de las tensiones de alimentación



4.4.2 Fase 2

Aproximadamente un segundo después, se envía la segunda palabra de control (253 decimal), la cual habilita los restantes tres buffers tri-estado, que toman sus entradas desde etapas comparadoras de tensión LM311.

Dichas etapas comparadoras toman sus tensiones de referencia y de prueba de divisores de tensión. Finalmente, se leen los tres bits de salida de los buffers en las líneas ERROR, SLCT, y PE. Cuando se cuenta con las dos lecturas, por medio de software, se realiza una búsqueda en el archivo de parámetros eléctricos con el fin de encontrar el significado de las lecturas obtenidas y determinar el mensaje de diagnóstico que aparecerá en la pantalla del PC.

En ambas fases, las palabras de control activan el relé K1, para que la referencia de las etapas LM311 de la interfaz sea conectada con la referencia de los monitores de tensión de la unidad.

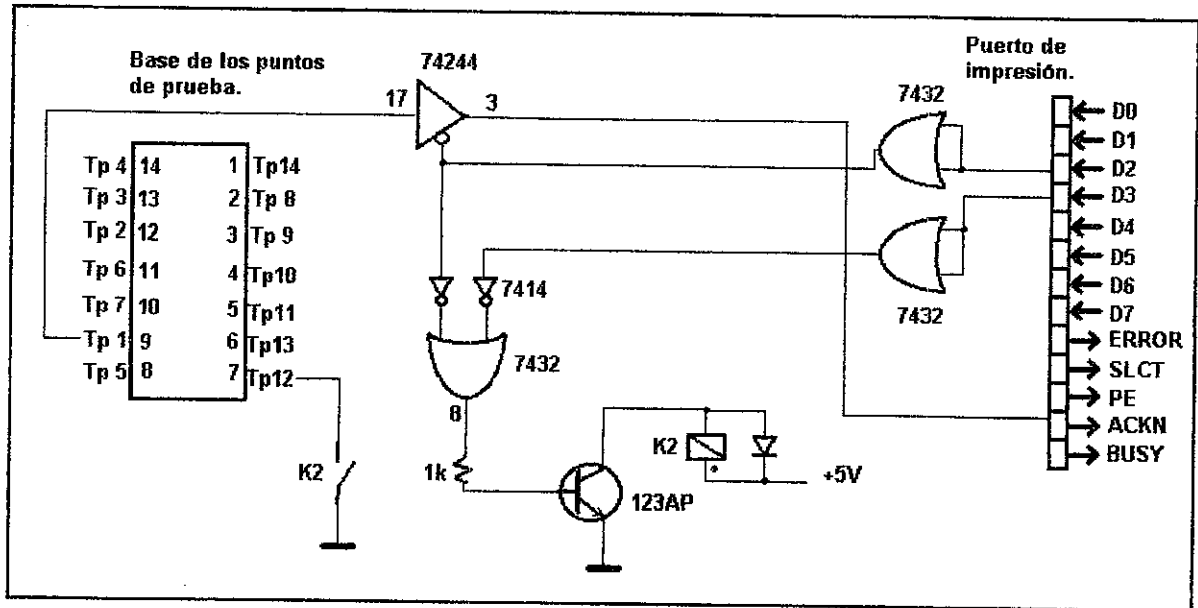
4.5 Prueba del temporizador

En esta prueba el software envía una palabra de control hacia la interfaz, para verificar la existencia de la señal temporizadora a la salida de la etapa del circuito integrado 555; la palabra de control (251 decimal), activa un buffer tri-estado, que tiene como entrada la señal proveniente del punto de prueba 1 (TP1) de la tarjeta Ericsson, la salida de dicho buffer se alambra a la línea ACKNOWLEDGE para su lectura vía software.

La palabra de control activa el relé K₂, que pasa la referencia de los circuitos lógicos de la interfaz con la referencia de los circuitos lógicos de la unidad bajo prueba (TP12).

En la figura 14 se ilustra la etapa de prueba del temporizador

Figura 14: Etapa de lectura de la señal del temporizador

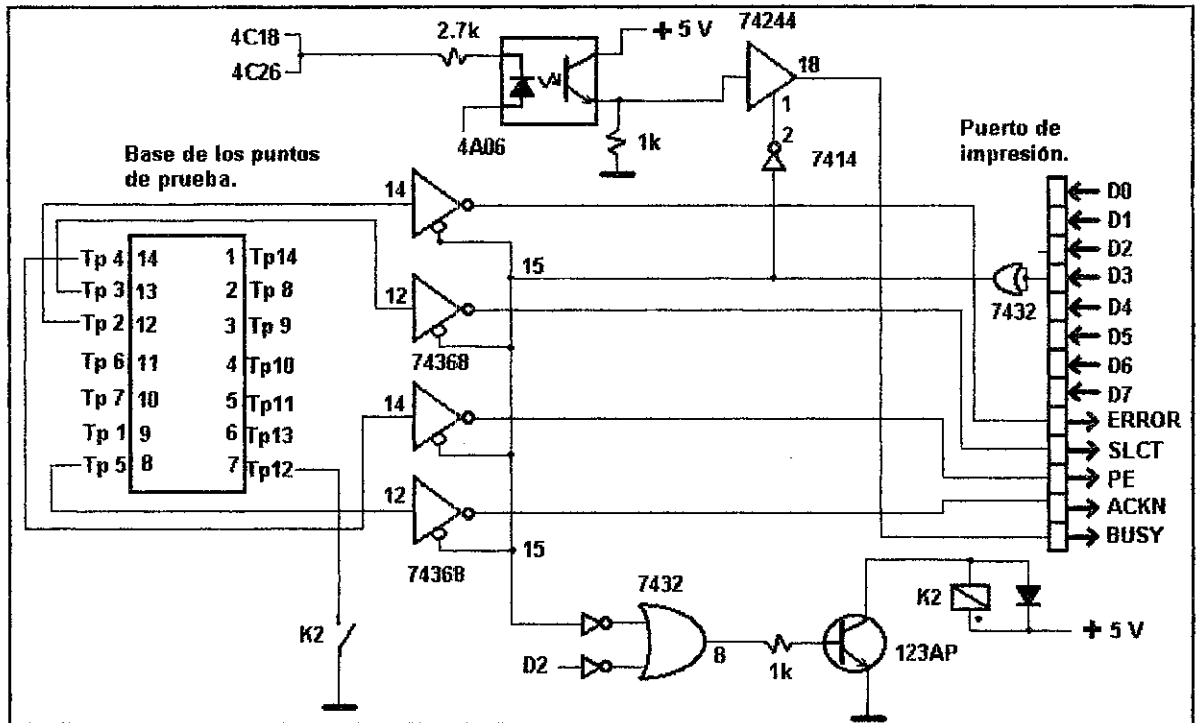


4.6 Prueba de los valores lógicos en carga

En ésta prueba se verifican los estados lógicos en la etapa divisora de frecuencia. La palabra de control (247 decimal), activa cinco buffers tri-estado en la interfaz, cuatro de ellos toman sus entradas de los puntos de prueba TP2, TP3, TP4 y TP5. El primer buffer entrega su salida a la línea ERROR, para verificar el estado lógico de la señal de reloj para el flip-flop D, que en condiciones normales de carga tiene un estado lógico bajo.

El segundo buffer entrega su salida a la línea SLCT y se utiliza para verificar el valor lógico de la señal de reinicio de los divisores de frecuencia, en condiciones normales de carga, se debe encontrar un estado lógico bajo. La etapa de prueba se muestra en la figura 15.

Figura 15. Etapa de prueba de los valores lógicos



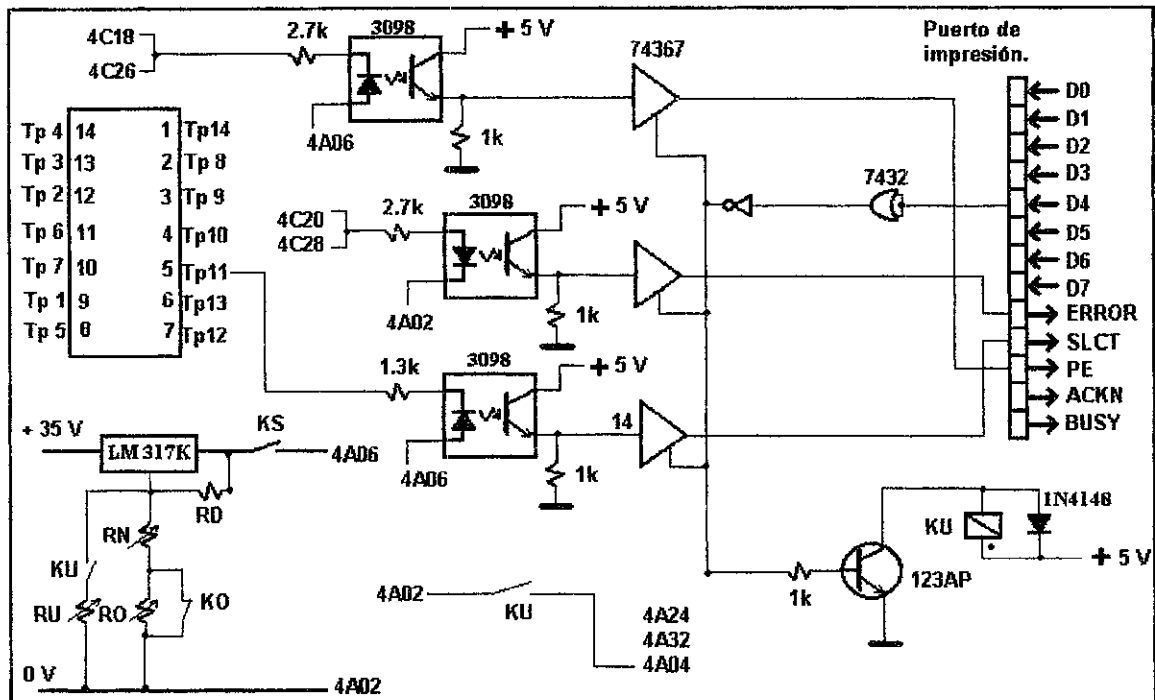
El tercer buffer sirve para verificar la presencia de un estado lógico bajo en la salida del último CI divisor de frecuencia, la salida se alambra a la línea PE; el cuarto buffer se utiliza para verificar que en la salida $-Q$ del flip-flop D se encuentre un estado lógico bajo el cual garantiza una excitación para la etapa de relé, la salida se lee en la línea ACK.

Finalmente, el quinto buffer toma su entrada desde la salida de un opto acoplador, para verificar que el relé de carga por medio de sus contactos, envíe la señal de -24 voltios, a las terminales 4C18 y 4C26 de la tarjeta bajo prueba, el resultado se lee en la línea BUSY. La palabra de control también activa el relé K₂ para conectar la referencia de los circuitos lógicos de la interfaz, con la referencia de los circuitos lógicos de la tarjeta bajo prueba.

4.7 Prueba de bajo voltaje y alarma de sobre temperatura

En la figura 16 se ilustra el diagrama esquemático de la etapa.

Figura 16: Etapa de prueba la respuesta a un bajo voltaje y sobre temperatura



La palabra de control (239 decimal) se envía hacia la interfaz, para que genere un voltaje de -22.5 voltios, para lograrlo la interfaz utiliza el regulador de voltaje variable LM317K (ver apéndice), el cual puede entregar tensiones desde 1.2 hasta 37 voltios DC.

En condiciones normales de funcionamiento, el resistor R_N está como la única resistencia de ajuste ya que por medio del contacto normalmente cerrado del relé K_O , el resistor R_O está en cortocircuito, y el resistor R_U está fuera del circuito debido a que está en serie con el contacto normalmente abierto de K_U . Por lo que la tensión de salida como función de la resistencia de ajuste y de derivación es:

$$V_{SAL}(R_{AJUSTE}, R_{DERIV}) = 1.2 * (1 + R_{AJUSTE} / R_{DERIV}),$$

normalmente se utiliza una resistencia de derivación de 240Ω ,

$$V_{SAL}(R_{AJUSTE}) = 1.2 * (1 + R_{AJUSTE} / 240) \text{ voltios.}$$

Para este caso, la resistencia de ajuste es R_N , que es un potenciómetro de $10 \text{ k}\Omega$ ajustado aproximadamente en 4560Ω , por lo que la tensión de salida es:

$$V_{SAL}(R_N) = 1.2 * (1 + 4560\Omega / 240\Omega)$$

$$V_{SAL}(R_N) = 24 \text{ V.}$$

En el instante en que se presenta la palabra de control 239, la interfaz energiza el relé K_U , el cual cierra sus contactos abiertos y conecta la resistencia R_U en paralelo con R_N . R_U es un potenciómetro de $100 \text{ k}\Omega$, ajustado aproximadamente a $64.7 \text{ k}\Omega$. Bajo ésta condición, la tensión de salida es:

$$V_{SAL} = 1.2 * (1 + [(4560\Omega * 64.7K\Omega / (4560\Omega + 64.7k\Omega)] / 240\Omega)$$

$$V_{SAL} = 22.5 \text{ voltios.}$$

Esta tensión alimenta la tarjeta bajo prueba, por lo que dicha tarjeta debe generar una señal de 0v como alarma de bajo voltaje en las terminales 4C20 y 4C28, dicha señal se detecta por medio de un opto acoplador, seguidamente a un buffer tri-estado para que el valor resultante se obtenga en la línea ERROR del puerto de impresión.

El comparador LM311 del monitor de bajo voltaje de la unidad debe enviar -12 voltios DC (detectado en TP11), que también pasa por un opto acoplador, un tri-estado, y luego a la línea SLCT; la salida del opto acoplador conectado a las terminales 4C18 y 4C26 (Charging) se alambra a un tri-estado y luego se lee en la línea PE, en esta prueba el relé de carga debe dejar de operar (terminales 4C18, 4C26 sin señal), debido a que por medio del relé K_U se envía a las terminales 4A24, 4A32, 4A04 una señal de 0 V para simular una sobretemperatura en los bancos de baterías.

4.8 Prueba de sobre voltaje

Se envía la palabra de control 223 hacia la interfaz para que genere una condición de sobre voltaje. En condiciones normales, la resistencia de ajuste para el regulador de voltaje LM317K es R_N .

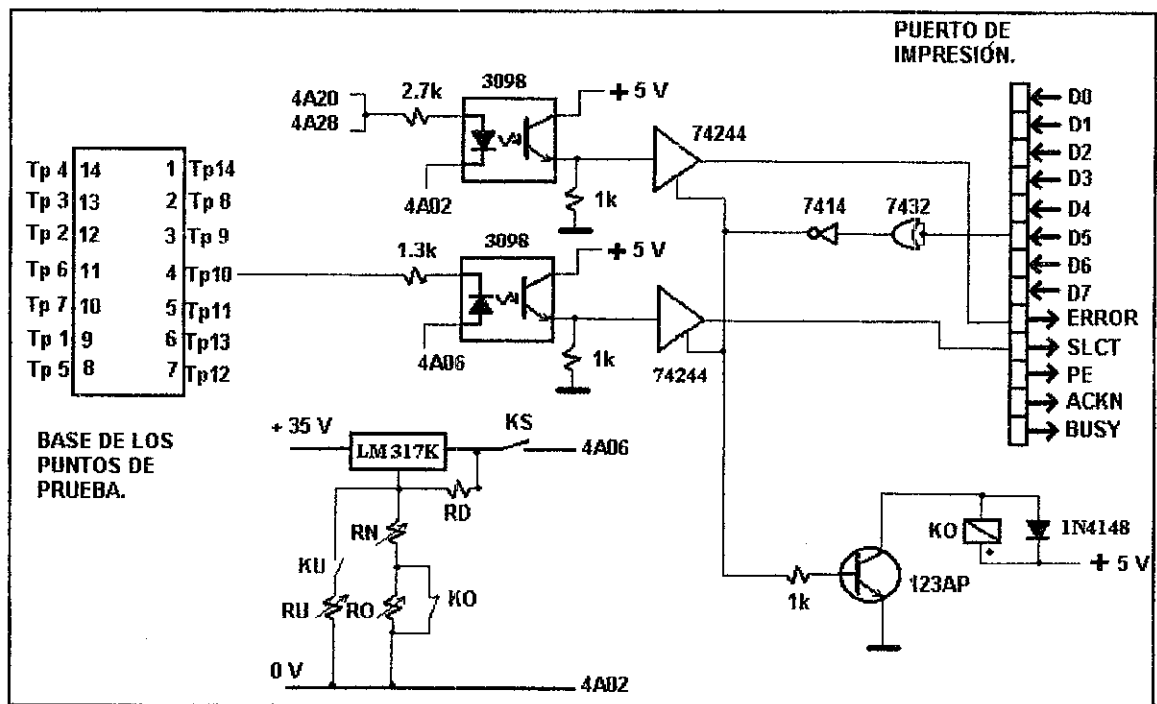
Al presentarse la palabra de control, según se ilustra en la figura 17, la interfaz energiza el relé K_O , el cual pone al resistor R_O en serie con R_N , siendo R_O un potenciómetro de $2\text{ k}\Omega$ ajustado aproximadamente en $1.8\text{ k}\Omega$, entonces la tensión entregada por el regulador de tensión es:

$$V_{SAL} = 1.2 * [1 + (R_N + R_O) / 240\Omega]$$

$$V_{SAL} = 1.2 * [1 + (4560\Omega + 1800\Omega) / 240\Omega]$$

$$V_{SAL} = 33\text{ voltios.}$$

Figura 17: Etapa de prueba de la respuesta a un sobre voltaje



Siendo ésta la tensión con la que se alimenta la tarjeta, el monitor de sobre voltaje de la unidad debe enviar una señal de 0 V como señal de alarma de sobre voltaje por las terminales 4A20 y 4A28 hacia los rectificadores, para apagarlos.

Esta señal es detectada por un opto acoplador, luego pasa por un buffer tri-estado, para luego aplicar la señal de salida a la línea ERROR. Para verificar la etapa comparadora, se utiliza el punto de prueba 10 (TP10), en el cual se deben presentar -12 voltios, que son entregados a un opto acoplador, a un buffer tri-estado y finalmente a la línea SLCT.

5. DESCRIPCIÓN GENERAL DEL SOFTWARE UTILIZADO

5.1 Generalidades

El control de los procesos de entrada / salida se realiza por un programa ejecutable escrito en lenguaje Visual Basic. Esta es una aplicación de 32 bits bajo el sistema operativo Windows, por lo que requiere como mínimo un computador 4x86 con 8 MB de memoria RAM para su correcta operación. Dicho programa se aplica para el control del puerto de impresión, utiliza una librería escrita en lenguaje Pascal (Tarjeta.pas), que contiene 2 rutinas principales: Enviar y Recibir. Al compilarse la librería Tarjeta.pas genera una extensión de aplicación Tarjeta.dll, a su vez ésta librería utiliza un módulo escrito en Turbo Assembler: Tarjeta.asm, que al compilarse genera el programa objeto Tarjeta.obj el cual se enlaza con la librería.

Con estas rutinas el programa principal puede enviar palabras de ocho bits, que controlan la interfaz de prueba, y la interfaz envía palabras hasta de cinco bits hacia el computador, que se pueden leer fácilmente vía software. El significado del resultado obtenido, se busca en un archivo de parámetros eléctricos para finalmente desplegar en pantalla el mensaje de diagnóstico.

Visual Basic permite realizar programas vistosos, sencillos, de fácil comprensión para el lector, por ésta razón, se incluyen a continuación los distintos procedimientos de software utilizados.

5.2 Software utilizado para el control del puerto de impresión

El programa Tarjeta.asm, escrito en Turbo Assembler es el encargado de manipular el puerto de impresión, realizando operaciones de entrada / salida. A continuación se incluye el listado del código de dicho programa:

```
;Representa la interfaz en ensamblador
;para el programa Tarjeta.pas y contiene rutinas para el acceso a puertos
DATA Segment Word Public          ;Segmento de datos de turbo
    Extrn PuertoEnt                ;Puerto de entrada como variable de turbo
    Extrn PuertoSal                ;Puerto de salida como variable de turbo
Data Ends                          ;Fin del segmento de datos

CODE Segment Byte Public          ;El segmento de código de turbo
Assume Cs:CODE, ds:DATA, es:Nothing, ss:Nothing
                                   ;Declaraciones Public de funciones internas

Public LeerB
Public EscribirB

;LeerB lee un byte del puerto de entrada, llamada desde turbo LeerB:byte;
LeerB proc near
    mov dx, PuertoEnt              ;Dirección del puerto en DX
    in al, dx                      ;Leer del puerto
    and al, 0F8h                   ;Apagar los bits 0-2
    ret                            ;retorno, resultado de la función en AL
LeerB endp

;EscribirB, envía un Byte al puerto de salida llamada desde turbo EscribirB(valor:byte);

EscribirB proc near
    Valor equ byte ptr[bp+4]       ;La variable pasada por TP
    push bp                        ;Permitir acceso a los argumentos
    mov bp, sp
```



```

        mov al, valor                ;Cargar el valor a AL
        mov dx, PuertoSal           ;Cargar puerto de salida a DX
        out dx, al                  ;Pasar el valor al puerto
        pop bp                      ;Restaurar BP
        ret 2                       ;Recoger pila y retornar
EscribirB endp

CODE Ends
END

```

A continuación se incluye el código de la librería Tarjeta.pas:

```

Library Rutinas_de_Puerto;
    Var PuertoSal, PuertoEnt:word;
        Valor:byte;

{$L Tarjeta.obj}                {Incluir el módulo en ensamblador}
Function LeerB:byte;External;    {Leer 5 bits desde el segundo registro del puerto de
                                {impresión}
Procedure EscribirB(Valor:Byte);External;    {Escribir 8 bits en el primer
                                                {registro del puerto de impresión}
Function EnviaByte(Valor:Byte):Boolean;
    Begin
        EscribirB(Valor);
    End;

Procedure Enviar(Valor:Byte);Export;
    Begin
        PuertoSal := MemW[$0040:8];
        EnviaByte(Valor);
    End;

```

```

Function Recibir:Byte;Export;
    Var Nibble:Byte;                                {Nibble recibido}
    Begin
        PuertoEnt:= MemW[$0040:8] + 1;
        Nibble:=(LeerB shr 3); {Utilizar solamente 5 bits}
        Recibir:=(Nibble);
    End;

Exports
    Enviar index 1,
    Recibir index 2;
Begin
End.

```

La librería anterior es utilizada por la aplicación Visual Basic, que controla todos los procesos de entrada y salida. A continuación se incluye la declaración de dicha librería:

```

Declare Sub Enviar Lib "Tarjeta" (ByVal Valor As Byte)
Declare Function Recibir Lib "Tarjeta" () As Byte

```

Estas líneas de código tienen que escribirse en un módulo de clase (extensión BAS).

5.3 Creación de la base de parámetros

Dependiendo de la prueba que se realice, se utilizarán de 1 a 5 bits como información del estado de la tarjeta Ericsson, por lo que se tiene entre 2^1 y 2^5 combinaciones posibles de bits, que tienen un significado particular.

Por ejemplo, si en una etapa manejadora de relé implementada con un transistor del tipo NPN se encuentra en la base de dicho transistor, un estado alto (+5V) proveniente de una compuerta TTL, una señal en el colector indicando saturación, y el relé no se activa, indicaría una posible falla en el relé y la combinación de bits recibida sería el binario 110 por lo que el mensaje hacia el usuario sería: "relé defectuoso". En adición, si se obtiene la siguiente palabra: 111, indicaría que la compuerta TTL funciona correctamente, que el transistor está operando y que el relé está activo, por lo que el mensaje podría ser: "etapa operando correctamente". Cada combinación de bits puede ser interpretada de igual forma para crear una base de datos con las diferentes posibilidades.

A continuación se incluye el código empleado para la escritura de la base de datos, los dispositivos se describen de la forma zener 23, relé 73, etc. ya que en el programa principal se incluye para cada prueba, por medio de un botón de comando, la opción de observar el diagrama esquemático de la etapa bajo prueba como imagen de mapa de bits (BITMAP), y allí aparecen los dispositivos en dicho orden numérico.

```
Private Sub Command2_Click()
Dim MyRecord As RegistroParametros
Open "c:\Parámetros" For Random As #1 Len = Len(MyRecord)
MyRecord.Medicion = "Tensiones"      ' Definir registro medición.
    MyRecord.Lectura1 = 16
    MyRecord.Lectura2 = 16
    MyRecord.Mensaje = "La tarjeta no está presente, falla grave "
    Put #1, , MyRecord      ' Escribir registro en el archivo.
MyRecord.Medicion = "Tensiones"
    MyRecord.Lectura1 = 23
    MyRecord.Lectura2 = 22
    MyRecord.Mensaje = "El Zener 66 está en corto circuito"
```

```
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 23
MyRecord.Mensaje = "Toda la etapa de alimentación está correcta"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 21
MyRecord.Mensaje = "El Zener 63 está en cortocircuito"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 19
MyRecord.Mensaje = "El Zener 60 está en cortocircuito"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 19
MyRecord.Lectura2 = 23
MyRecord.Mensaje = "El Zener 60 está abierto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 22
MyRecord.Lectura2 = 23
MyRecord.Mensaje = "El Zener 66 está abierto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 21
MyRecord.Lectura2 = 23
MyRecord.Mensaje = "El Zener 63 está abierto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 20
MyRecord.Mensaje = "Los Zener's 66 y 63 se detectaron en cortocircuito"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 20
MyRecord.Lectura2 = 23
```

MyRecord.Mensaje = "Los Zener's 66 y 63 se detectaron abiertos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 18
MyRecord.Mensaje = "Los Zener's 66 y 60 se detectaron en cortocircuito"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 18
MyRecord.Lectura2 = 23
MyRecord.Mensaje = "Los Zener's 66 y 60 se detectaron abiertos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 17
MyRecord.Mensaje = "Los Zener's 63 y 60 se detectaron en cortocircuito"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 17
MyRecord.Lectura2 = 23
MyRecord.Mensaje = "Los Zener's 63 y 60 se detectaron abiertos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 16
MyRecord.Mensaje = "Los Zener's 66,63,60 se detectaron en cortocircuito"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 16
MyRecord.Lectura2 = 23
MyRecord.Mensaje = "Los Zener's se detectaron abiertos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 21
MyRecord.Lectura2 = 22
MyRecord.Mensaje = "El Zener 66 está en corto y el Zener 63 abierto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 19

```

MyRecord.Lectura2 = 18
MyRecord.Mensaje = "El Zener 66 está en corto y el Zener 60 abierto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 22
MyRecord.Lectura2 = 21
MyRecord.Mensaje = "El Zener 63 está en corto y el Zener 66 abierto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 18
MyRecord.Lectura2 = 19
MyRecord.Mensaje = "El Zener 60 está en corto y el Zener 66 abierto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 17
MyRecord.Lectura2 = 22
MyRecord.Mensaje = "El Zener 66 está en corto y Los Zener's 63,60
abiertos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 18
MyRecord.Lectura2 = 21
MyRecord.Mensaje = "El Zener 63 está en corto y Los Zener's 66,60  abiertos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 20
MyRecord.Lectura2 = 19
MyRecord.Mensaje = "El Zener 60 está en corto y Los Zener's 66,63 abiertos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 22
MyRecord.Lectura2 = 17
MyRecord.Mensaje = "El Zener 66 está abierto y los Zener's 63,60 en corto"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 21
MyRecord.Lectura2 = 18
MyRecord.Mensaje = "El Zener 63 está abierto y Los Zener's 66,60 en corto"
Put #1, , MyRecord

```

MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 19
MyRecord.Lectura2 = 20
MyRecord.Mensaje = "El Zener 60 está abierto y los Zener's 66,63 en corto"
Put #1, , MyRecord

MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 16
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "La Tarjeta no está conectada: Main Failure"
Put #1, , MyRecord

MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 0
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,31,25,28,38 ó 73 defectuosos"
Put #1, , MyRecord

MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 24
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,31,25,38 ó 73 defectuosos"
Put #1, , MyRecord

MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 8
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,31,25 defectuosos"
Put #1, , MyRecord

MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 20
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,31,28 defectuosos"
Put #1, , MyRecord

MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 4
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,31,28,38 ó 73 defectuosos"
Put #1, , MyRecord

MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 28
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,31,38 ó 73 defectuosos"

```
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 12
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,31 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Tensiones"
MyRecord.Lectura1 = 18
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,25,28 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 2
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,25,28,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 26
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,25,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 10
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,25 en mal estado"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 22
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,28 en mal estado"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 6
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21,28,38 ó 73 en mal estado"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 30
MyRecord.Lectura2 = 0
```



```
MyRecord.Mensaje = "21,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 14
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "21 defectuoso"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 17
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "31,25,28 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 1
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "31,25,28 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 25
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "31,25,28,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 9
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "31,25 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 21
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "31,28 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 5
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "31,28,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 29
```

```

MyRecord.Lectura2 = 0
MyRecord.Mensaje = "31,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 13
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Compuerta de CI 31 defectuosa"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 19
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "25,28 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 3
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "25,28,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 27
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "25,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 11
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Contador 25 defectuoso"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 23
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Flip-Flop 28 defectuoso"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 7
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "28,38 ó 73 defectuosos"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"

```

```

MyRecord.Lectura1 = 31
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Transistor 38 o Relé 73 defectuoso"
Put #1, , MyRecord
MyRecord.Medicion = "Lógicos"
MyRecord.Lectura1 = 15
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "todo está correcto"
Put #1, , MyRecord
MyRecord.Medicion = "UnderVol"
MyRecord.Lectura1 = 16
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Tarjeta no conectada: falla grave "
Put #1, , MyRecord
MyRecord.Medicion = "UnderVol"
MyRecord.Lectura1 = 20
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Comparador 12 y compuerta 31 en mal estado"
Put #1, , MyRecord
MyRecord.Medicion = "UnderVol"
MyRecord.Lectura1 = 18
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Transistor 37 o relé 72 en mal estado"
Put #1, , MyRecord
MyRecord.Medicion = "UnderVol"
MyRecord.Lectura1 = 22
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Compuerta 31 defectuosa, relé 72 o transistor 37 defectuoso"
Put #1, , MyRecord
MyRecord.Medicion = "UnderVol"
MyRecord.Lectura1 = 17
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Tarjeta no conectada: Main Failure"
Put #1, , MyRecord
MyRecord.Medicion = "UnderVol"
MyRecord.Lectura1 = 21
MyRecord.Lectura2 = 0
MyRecord.Mensaje = "Transistor 37 o Relé 72 defectuoso"
Put #1, , MyRecord

```

```

MyRecord.Medicion = "UnderVol"
  MyRecord.Lectura1 = 19
  MyRecord.Lectura2 = 0
  MyRecord.Mensaje = "Todo está correcto"
  Put #1, , MyRecord
MyRecord.Medicion = "OverVol"
  MyRecord.Lectura1 = 16
  MyRecord.Lectura2 = 0
  MyRecord.Mensaje = "Tarjeta no conectada: Main Failure"
  Put #1, , MyRecord
MyRecord.Medicion = "OverVol"
  MyRecord.Lectura1 = 18
  MyRecord.Lectura2 = 0
  MyRecord.Mensaje = "Etapa manejadora 36 o rele 71 en mal estado"
  Put #1, , MyRecord
MyRecord.Medicion = "OverVol"
  MyRecord.Lectura1 = 17
  MyRecord.Lectura2 = 0
  MyRecord.Mensaje = "Relé 71 enclavado o su etapa manejadora 36 en corto"
  Put #1, , MyRecord
MyRecord.Medicion = "OverVol"
  MyRecord.Lectura1 = 19
  MyRecord.Lectura2 = 0
  MyRecord.Mensaje = "Todo está correcto"
  Put #1, , MyRecord
Close #1                                ' Cerrar archivo.

```

5.4 Software utilizado para la prueba de tensiones

cuando el usuario activa el botón de comando rotulado “Probar etapa de alimentación”, se emplea el procedimiento llamado Tensiones, el cual envía las palabras de control hacia la interfaz para que realice las dos fases de la prueba de los voltajes de alimentación.

```

Public Sub Tensiones()
    Form1.Hide                'Esconder formulario principal
    Load Form2                'Cargar en memoria formulario de tensiones
    Form2.Show                'Mostrar formulario de tensiones
    Call Enviar(254)           'Enviar palabra de control fase 1 a la interfaz
    Form2.label2.Caption = "Comunicando con el Hardware..."
    Call retardo(1)           'Retardo de 1 segundo
    lect1 = recibir            'Obtener lectura1 desde el segundo registro del puerto
    Call Enviar(253)           'Enviar palabra de control fase 2
    Call retardo(1)           'Retardo de 1 segundo
    lect2 = recibir            'Obtener lectura 2 desde el segundo registro del puerto
    Call Enviar(63)            'Deshabilitar la tarjeta bajo prueba
    Form2.label2.Caption = "Consultando base de datos.."
    Dim MyRecord As RegistroParametros
    Open "C:\Parámetros" For Random As #1 Len = Len(MyRecord)
    Do
        Contador = Contador + 1
        Get #1, Contador, MyRecord
        If (MyRecord.Lectura1 = lect1) And (MyRecord.Lectura2 = lect2)
            And(MyRecord.Medicion = "Tensiones    ") Then
                Message = MyRecord.Mensaje
                found = True                'Parámetro encontrado
            End If
        Loop Until (EOF(1) Or found)
        Close #1                        'Cerrar base de datos
        Contador = 0                    'Restaurar variables
        found = False
        Form2.label2.Caption = "Resultados:"
        Form2.list1.Visible = True       'Mostrar lista de texto
        Form2.list1.AddItem Message      'Desplegar el mensaje del diagnóstico.
    End Sub

```

5.5 Software utilizado para la prueba del temporizador

Al activar el botón de comando "Verificar temporizador", se utiliza el procedimiento listado abajo, el cual envía la palabra de control a la interfaz para que detecte la presencia de la señal de temporización de 0.15 Hz.

```
Public Sub Frecuencia()  
    Form1.Hide           'Esconder formulario principal  
    Load Form3          'Cargar en memoria formulario frecuencia.  
    Form3.Show           'Mostrar formulario frecuencia  
    Form3.label2.Caption = "Espere unos Instantes..."  
    Call Enviar(251)     'Enviar palabra de control a la interfaz  
    start = Timer        ' Iniciar temporizador para timeOut.  
    Do  
        DoEvents        'Permitir realizar otros procesos  
        lectura = recibir 'Leer desde el puerto de impresión.  
    Loop Until ((Timer > start + 15) Or (lectura = 16)) 'Estado bajo o timeout?  
    If Timer > start + 15 Then 'Si Timeout  
        lectura0 = "No se interrogó exitosamente"  
    Else  
        start = Timer    'Iniciar temporizador para timeOut2  
        Do  
            DoEvents  
            Do  
                lectura = recibir  
            Loop Until ((lectura = 24) Or (Timer > start + 15)) 'Estado alto o timeout?  
            tiempo0 = Time 'iniciar tiempo en estado alto  
            Do While (lectura = 25) 'Mientras estado alto  
                lectura = recibir 'Leer desde el puerto  
                tiempo1 = Time 'Incrementar con el tiempo  
        Loop Until ((lectura = 16) Or (Timer > start + 15)) 'Hasta bajo o timeout  
        If Timer > start + 15 Then
```

```

lectura00 = "Error de Timeout, se desconectó la tarjeta"
Else
    'Determinar tiempo aproximado en estado alto
    If ((Second(tiempo0) > 50) And (Second(tiempo1) < 10)) Then
        tiempoalto = 60 - Second(tiempo1) - Second(tiempo2)
        Control = 1
    End If
    If ((Second(tiempo1) > 50) And (Second(tiempo0) < 10)) Then
        tiempoalto = Second(tiempo0) - Second(tiempo1) + 60
        Control = 1
    End If
    If Control <> 1 Then
        tiempoalto = Second(tiempo1) - Second(tiempo0)
    End If
    Control = 0
    frequency = CInt((0.96 / tiempoalto) * 1000)    'Calcular y redondear
End If
End If
Form3.label2.Caption = "Resultados:"
Form3.list1.Visible = True                        'Mostrar cuadro de texto
Form3.list1.AddItem lectura0                      'Mostrar valor de lectura0
If (frequency > 100) And (frequency < 200) Then
    Form3.list1.AddItem frequency & " mHz , Es considerado Correcto"
Else
    Form3.list1.AddItem frequency & "mHz, se han detectado"
End If
Call Enviar(63)                                  'Desenergizar tarjeta bajo prueba.
End Sub

```

5.6 Software para la prueba de los estados lógicos

Este procedimiento envía la palabra de control a la interfaz para que ejecute la prueba de los valores lógicos en condición "Charging", procesa las lecturas y genera mensajes.

```
Public Sub Lógicos()  
    Form1.Hide           'Ocultar formulario principal.  
    Load Form4          'Cargar a memoria formulario "Lógicos"  
    Form4.Show          'Mostrar formulario de valores lógicos  
    Form4.label2.Caption = "Comunicando con el Hardware..."  
    Call Enviar(247)     'Enviar palabra de control a la interfaz  
    retardo (1)         'Retardo de 1 segundo  
    lectura = recibir   'Leer desde el puerto de impresión.  
    Dim MyRecord As RegistroParametros  
    Open "C:\parámetros" For Random As #1 Len = Len(MyRecord)  
    Do  
        Contador = Contador + 1  
        Get #1, Contador, MyRecord  
        If (MyRecord.Lectura1 = lectura) And (MyRecord.Medicion = "Lógicos") Then  
            Message = MyRecord.Mensaje  
            found = True  
        End If  
    Loop Until (EOF(1) Or found)  
    Close #1  
    found = False  
    Form4.label2.Caption = "Resultados:"  
    Form4.list1.Visible = True  
    Form4.list1.AddItem Message  
    Call Enviar(63)  
End Sub
```


5.7 Software para la simulación de bajo voltaje

Este procedimiento al activarse por parte del usuario el botón de comando respectivo, envía la palabra de control a la interfaz, para que genere una condición de bajo voltaje (22.5 voltios), procesa las lecturas y genera mensajes.

```
Public Sub BajoVoltaje()  
    Form1.Hide           'Esconder formulario principal  
    Load Form5          'Cargar a memoria formulario bajo voltaje  
    Form5.Show          'Mostrar formulario bajo voltaje  
    Form5.label2.Caption = "Espere unos Instantes..."  
    Call Enviar(239)     'Enviar palabra de control  
    retardo (1)         'Retardo de 1 segundo  
    lectura = recibir   'Leer desde el segundo registro del puerto de impresión  
    Dim MyRecord As RegistroParametros  
    Open "C:\parámetros" For Random As #1 Len = Len(MyRecord)  
    Do                  'Buscar en la base de parámetros  
        Contador = Contador + 1  
        Get #1, Contador, MyRecord  
        If (MyRecord.Lectura1 = lectura) And (MyRecord.Medicion = "UnderVol") Then  
            Message = MyRecord.Mensaje  
            found = True  
        End If  
    Loop Until (EOF(1) Or found)  
    Close #1  
    found = False  
    Form5.label2.Caption = "Resultados:"  
    Form5.list1.Visible = True  
    Form5.list1.AddItem Message  
    Call Enviar(63)     'Desenergizar tarjeta Ericsson  
End Sub
```

5.8 Software utilizado para la simulación de sobre voltaje

Este procedimiento envía la palabra de control para que la interfaz produzca una condición de alto voltaje (33 voltios), manipula las lecturas para desplegar mensajes.

```
Public Sub AltoVoltaje()  
    Form1.Hide           'Ocultar formulario principal  
    Load Form6          'Cargar formulario Sobre Voltaje  
    Form6.Show          'Mostrar formulario para sobre voltaje  
    Form6.label2.Caption = "Espere unos Instantes..."  
    Call Enviar(223)     'Enviar palabra de control  
    Call efecto(6)      'Retardo por medio de un efecto visual  
    lectura = recibir   'Leer desde el puerto de impresión.  
    Dim MyRecord As RegistroParametros  
    Open "C:\parámetros" For Random As #1 Len = Len(MyRecord)  
    Do                  'Buscar en la base de parámetros  
        Contador = Contador + 1  
        Get #1, Contador, MyRecord  
        If (MyRecord.Lectura1 = lectura) And (MyRecord.Medicion = "OverVol") Then  
            Message = MyRecord.Mensaje  
            found = True  
        End If  
    Loop Until (EOF(1) Or found)  
    Close #1            'Cerrar archivo de parámetros  
    found = False      'Iniciar centinela  
    Form6.label2.Caption = "Resultados:"      'Mostrar etiqueta  
    Form6.list1.Visible = True                'Cuadro de texto visible  
    Form6.list1.AddItem Message              'Incluir mensaje de diagnóstico  
    Call Enviar(63)                          'Desactivar buffers y tarjeta Ericsson  
End Sub
```

5.9 Procedimiento de retardo

Este procedimiento retarda un proceso un número entero de segundos.

```
Public Sub retardo(Segundos)
    start = Timer                ' Iniciar temporizador.
    Do While Timer < start + Segundos ' Establecer ciclo
        DoEvents                ' Permitir ejecutar otros procesos.
    Loop
End Sub
```

5.10 Procedimiento Efecto

Este procedimiento muestra un efecto visual con 2 figuras diferentes que se alternan consecutivamente.

```
Public Sub efecto(veces)        ' Rutina en función del número de veces
    Do
        DoEvents                ' Permitir ejecutar otros procesos
        i = i + 1                ' Variable de control
        calza = i Mod 2          ' Solo pares
        If calza = 0 Then        ' Si pares
            Form6.picture1.Visible = True ' Mostrar imagen 1
        Do
            DoEvents
            suma = suma + 1        ' Variable de retardo
        Loop Until suma > 5000    ' Establecer limite
        Form6.picture1.Visible = False ' Ocultar imagen 1
        End If
        If calza <> 0 Then        ' Si impares
            Form6.picture2.Visible = True ' Mostrar imagen 2
```

```

Do
DoEvents
suma1 = suma1 + 1
Loop Until suma1 > 5000
Form6.picture2.Visible = False
End If
suma = 0                                'Iniciar variables de retardo
suma1 = 0
Loop Until i >= veces
Form6.picture1.Visible = False
Form6.picture2.Visible = False
End Sub

```

Las siguientes líneas del código se refieren a la declaración del registro de parámetros eléctricos de la tarjeta Ericsson, el registro medición. Estas líneas de código deben teclearse en un módulo de clase (extensión BAS).

```

Type RegistroParámetros                ' Crear tipo definido por el usuario
Medición As String * 20                ' Definir elementos del tipo de datos
Lectura1 As Integer                    ' Lectura 1 obtenida del puerto
Lectura2 As Integer                    ' Lectura 2 obtenida del puerto
Mensaje As String * 70                 ' Tamaño de la cadena del mensaje
End Type

```


RECOMENDACIONES

1. Al crear un proyecto que utilice como medio de control un PC, es conveniente aprovechar al máximo la capacidad software del PC, de manera que se tenga un control con buena presentación en pantalla, y de fácil operación para el usuario.
2. En los procesos repetitivos, en los que entran en acción tarjetas electrónicas, equipo de cómputo, líneas de transmisión de datos, módems, etc., es conveniente analizar los protocolos que se manejan, con el fin de implementar software adecuado para la manipulación de puertos (paralelos o serie) y automatizar dichos procesos.
3. En una aplicación, en función de su complejidad, se deben tomar en cuenta el uso de los puertos internos del PC, aprovechando las ventajas de la normalización, ya que una tarjeta de puerto con diseño propio se orienta a ejecutar tareas específicas; si embargo, dicha tarjeta tendría que residir en el interior del PC.
4. Generalizar el procedimiento descrito en este trabajo de tesis, para diseñar la interfaz y software de prueba necesarios para los dispositivos más importantes a diagnosticar en el departamento de Reparaciones Electrónicas de TELGUA, S.A.

BIBLIOGRAFÍA

1. HOGAN. **Programmers PC source Book**. USA: Editorial Microsoft Press, 1988.
2. Catapult & Microsoft Corporation. **Guía para el programador en Visual Basic 4.0**. Editorial Mc Graw-Hill, 1995
3. MORRIS, Mano. **Diseño Digital**. Editorial Prentice Hall, 1987.
4. ECG, **Semiconductor Replacement Guide**.
5. Motorola, **Fast and LS TTL Data Book**.
6. TISCHER. **PC Interno**. Editorial Prentice Hall, 1983.
7. Luces Faulkenberry. **Introducción a los Amplificadores Operacionales con aplicaciones a CI lineales**. Editorial Limusa, 1990.
8. MILLMAN, Jacob. **Electrónica integrada**. Editorial Hispano Europea, 1995.

APÉNDICE

**INFORMACIÓN DE LOS CIRCUITOS INTEGRADOS LM555, LM311, 74193 Y
LM317K.**

1. El circuito integrado 555

El LM555, es un dispositivo sumamente estable, para generar retrasos u oscilaciones con precisión. Este dispositivo posee terminales adicionales para ciertas aplicaciones, como por ejemplo disparar o reajustar, según se requiera. En la modalidad de retraso, el tiempo se controla con precisión mediante un condensador y una resistencia conectados en el exterior del CI. En la operación estable como oscilador, la frecuencia de corrimiento libre y el ciclo de trabajo se controlan mediante dos resistencias y un condensador externos. El circuito puede dispararse y reajustarse sobre formas de onda descendentes y el circuito de salida puede proporcionar una corriente de hasta 200mA. Este circuito puede encontrarse integrado en un paquete de metal, o encapsulado doble en línea.

1.1 Descripción de las terminales

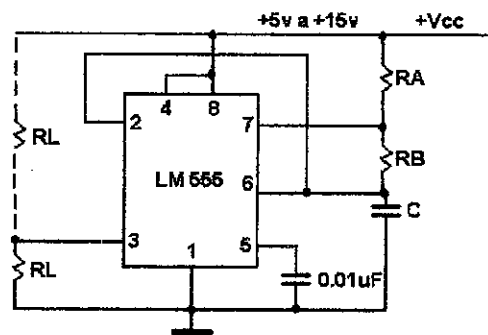
Tabla IX: *Descripción de las terminales del LM555*

Pin	Nombre	Descripción
1	Tierra	Masa o tierra del circuito integrado
2	Descarga	Terminal conectada a la resistencia de descarga
3	Disparador	Terminal de disparo del flip-flop interno
4	Reajuste	Terminal utilizada para la habilitación
5	Voltaje de control	Terminal utilizada para conectar la tensión de referencia
6	Umbral	Fija la tensión de umbral
7	Descarga	Terminal conectada a la resistencia de descarga
8	+Vcc	Terminal de alimentación del CI (voltaje positivo)

1.2 Operación estable

Si el circuito se conecta según se ilustra en la figura 18 (pines 2 y 6 alambreados juntos), se activará a sí mismo y correrá libremente como un multivibrador. El condensador exterior se carga a través de la resistencia equivalente $R_A + R_B$ y se descarga a través de R_B . De manera que el ciclo de trabajo puede establecerse con precisión por la relación entre estas dos resistencias.

Figura 18: Configuración estable del CI 555



En ésta modalidad de operación, el condensador se carga y se descarga entre $1/3 V_{cc}$ y $2/3 V_{cc}$. Al igual que en la modalidad de disparo, los tiempos de carga y descarga, y por lo tanto la frecuencia, son independientes del voltaje de suministro.

El tiempo de carga (salida alta) viene dado por:

$$T_1 = 0.693 * (R_A + R_B) * C;$$

y el tiempo de descarga (salida baja) por:

$$T_2 = 0.693 * R_B * C.$$

Así que el período total es:

$$T = T_1 + T_2 = 0.693 * (R_A + 2 * R_B) * C$$

La frecuencia de oscilación se calcula con:

$$F = 1 / T = \frac{1.44}{(R_A + 2 * R_B) * C}$$

y el ciclo de trabajo está dado por:

$$D = R_B / (R_A + 2 * R_B).$$

2 El comparador de voltaje LM311

Este dispositivo se constituye en un comparador integrado, puede adquirirse en un encapsulado del tipo TO-5 (metálico), o en un encapsulado doble en línea (plástico o cerámico). El encapsulado TO-5 posee ocho terminales, pero para el encapsulado doble en línea existen de 8 y de 14 terminales. El LM311 es un comparador de voltaje cuyas corrientes de entrada son cientos de veces más bajas que las de dispositivos tales como el LM306 o el LM710C. Está diseñado también para funcionar con una gama más amplia de voltajes de suministro: desde los valores estándar de amp-op de +/- 15 voltios hasta los 5 voltios únicos utilizados para la lógica TTL. Su salida es compatible con los circuitos RTL, DTL, TTL, así como los MOS. Además, puede activar lámparas y relevadores, conmutando voltajes hasta de 40 voltios con corrientes de 50mA.

2.1 Características

- a) Puede funcionar con una fuente única de 5 voltios.
- b) Corriente máxima de entrada: 250 nA.

- c) Corriente máxima de desajuste: 50 nA.
- d) Gama de voltajes de entrada diferencial +/- 30 voltios.
- e) Consumo de potencia: 135 mW a +/- 15 voltios.

Tanto la entrada como la salida del LM311 se pueden aislar de la tierra del sistema y la salida puede impulsar cargas referidas a tierra, al suministro positivo o al suministro negativo. Ofrece posibilidades de control del desajuste y de habilitación. Aunque es más lento que el LM306 y el LM710C (tiempo de respuesta de 200 nSeg contra 40 nSeg.), Este dispositivo es mucho menos propenso a las oscilaciones espurias. El LM311 tiene la misma configuración de terminales que el LM306 y el LM710C.

2.2 Descripción de las terminales

Esta descripción es válida para los encapsulados TO-5 y dual en línea de 8 terminales.

Tabla X: Descripción de las terminales del CI LM311

Terminal	Nombre	Descripción
1	Tierra	Terminal de masa o tierra
2	Entrada 2	Entrada no inversora
3	Entrada 3	Entrada inversora
4	V ⁻	Alimentación con tensión negativa
5	Balance	Balance para ajustar
6	Habilitar	Terminal de habilitación. (No debe conectarse directamente a tierra)
7	Salida	Salida del comparador
8	V ⁺	Alimentación con tensión positiva

3. El circuito integrado 74193

El CI 73193 es un contador binario módulo-16, puede realizar la cuenta ascendente o descendientemente, teniendo para cada caso un reloj individual. La salida cambia de estado, cuando un flanco de subida se presenta en las transiciones del reloj de entrada. Las salidas separadas de conteo arriba o abajo, se pueden utilizar como señal de reloj para etapas subsecuentes, lo que simplifica diseños multietapas. Dicho CI se encuentra en un encapsulado doble en línea, de 16 terminales.

El circuito posee 4 flip-flops maestro / esclavo, con disparo interno y circuitería que provee reposición general de los flip-flops y preajuste individual. Cada flip-flop posee realimentación JK, del esclavo hacia el maestro tal que un flanco de subida en su entrada T, causa que el esclavo, y por lo tanto la salida Q cambien de estado. Un flanco de subida en la entrada de cuenta hacia arriba, incrementa el contador en uno, un flanco similar en la entrada de cuenta hacia abajo, decrementa el contador en uno. Mientras se realiza el conteo solo con una entrada de reloj, la otra debe ser conectada permanentemente a un estado alto, si no se toma en cuenta esta precaución, el conteo será inestable.

La terminal de conteo hacia arriba, y la terminal de conteo hacia abajo tienen salidas normalmente en estado alto. Cuando un circuito alcanza el estado de conteo máximo (15), la salida $-TC_U$ va a estado bajo, y se mantiene en dicho estado hasta que la entrada de reloj de cuenta hacia arriba (CP_U) va nuevamente a estado alto, similarmente la salida $-TC_D$ de cuenta descendente, estará en estado bajo cuando el circuito este en estado de cuenta cero y el reloj de conteo descendente va a estado bajo.

El circuito permite la carga paralela de datos, permitiendo con esto que el contador sea preajustado. Cuando la terminal de carga paralela (-PL) y la terminal de reposición general (MR) están en estado bajo, los datos que estén presentes en las terminales de datos P0..P3 son cargadas al contador y aparecen en las salidas sin importar las condiciones de reloj de entrada. Un estado alto en la terminal MR deshabilita las compuertas preajustadas, haciendo caso omiso de las entradas de reloj, por lo que las salidas Q's van a estado bajo. Si alguna de las entradas de reloj está en estado bajo durante o después de la reposición, el próximo flanco de subida se interpreta como señal legítima de conteo.

3.1 Características

- a) Baja potencia consumida: 95 mW (disipación típica).
- b) Velocidad alta: 40 Mhz, frecuencia típica de conteo.
- c) Conteo síncrono.
- d) Entradas individuales para reposicionar.
- e) Circuitos internos en cascada.

3.3 Tabla de modo de selección.

Tabla XI: *Modos de selección del CI 74193*

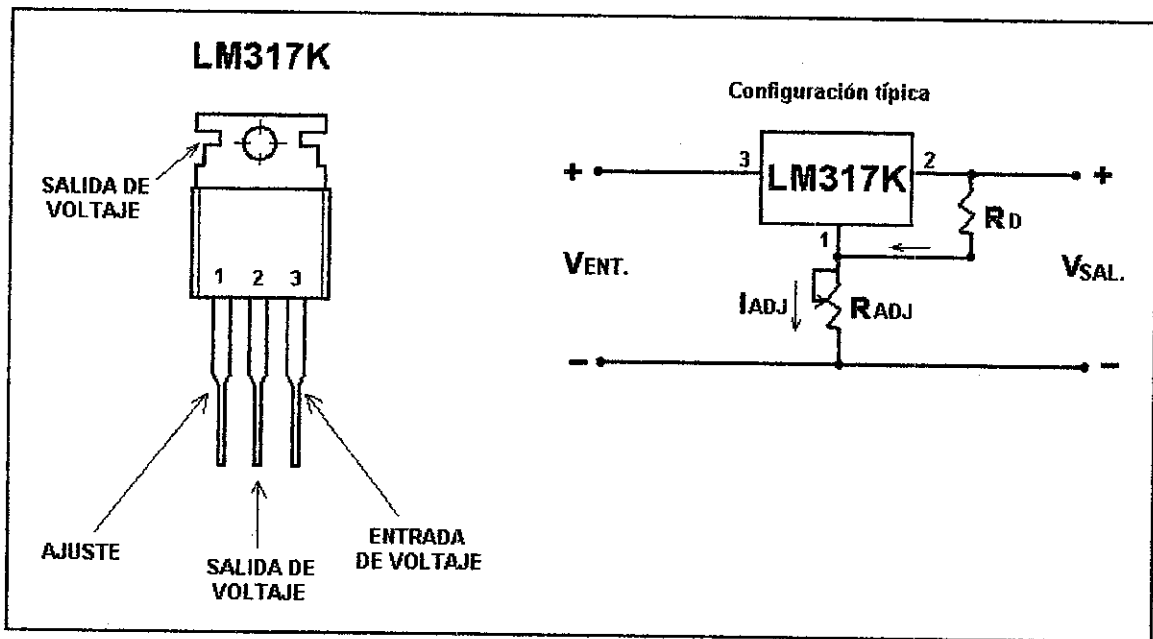
MR	-PL	CP _U	CP _D	Modo.
H	X	X	X	Reposición (asíncrona)
L	L	X	X	Preajuste (asíncrono)
L	H	H	H	Sin cambio
L	H	↑	H	Cuenta hacia arriba
L	H	H	↑	Cuenta hacia abajo

L = estado lógico bajo, H = estado lógico alto, X = Condición no importa, ↑ = flanco de subida.

4. El regulador de voltaje variable LM317K

Este dispositivo está diseñado para suplir un voltaje de salida entre el rango de 1.2 a 37 voltios DC. La selección del voltaje deseado se realiza por medio de una resistencia de ajuste, y una de derivación. Este dispositivo posee tres terminales, como se ilustra en la figura 19. El voltaje máximo de entrada, es de 40 voltios y el voltaje mínimo de entrada es de 3.7 voltios. Además puede suministrar una corriente máxima de 1.5 amperios.

Figura 19: Configuración típica del CI LM317K



Según se ilustra en la figura 19, la conexión de este dispositivo es sencilla, por default, la caída de tensión entre la terminal de salida 2 y la terminal de ajuste 1 es de 1.2 voltios, de manera que al seleccionar un valor para R_D de 240Ω , la corriente que circula por dicho resistor y por el resistor de ajuste es:

$$I_{ADJ} = 1.2 / 240, (A)$$

De manera que la tensión de salida V_{SAL} como una función de la resistencia de ajuste R_{ADJ} es:

$$V_{SAL}(R_{ADJ}) = 1.2 + I_{ADJ} * R_{ADJ}$$

$$V_{SAL}(R_{ADJ}) = 1.2 + 1.2 * R_{ADJ} / 240$$

$$V_{SAL}(R_{ADJ}) = 1.2 * (1 + R_{ADJ} / 240) \text{ Voltios DC.}$$

Así que la resistencia de ajuste en este caso, tiene que tener un valor en el rango de 0Ω a 7160Ω .