

TRABAJO ESPECIAL DE GRADO

Analizador Dinámico de Funciones de Escala (A.F.E.)

TUTOR ACADÉMICO: Prof. José M. Milá de la Roca

Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Br. Walter E. Senesi L.
para optar al Título de
Ingeniero Electricista

Caracas, 2006.

CONSTANCIA DE APROBACIÓN

Caracas, 14 de Junio de 2006.

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Walter Eduardo Senesi Laguna, titulado:

“ANALIZADOR DINÁMICO DE FUNCIONES DE ESCALA (A.F.E.)”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

Prof. Pedro Pinto
Jurado.

Prof. Mary Power
Jurado.

Prof. José M. Milá de la Roca
Tutor académico.

DEDICATORIA

A mis Padres, mis hermanos y amigos, los cuales siempre han sabido estar allí y con los que siempre, de una u otra forma, he contado. Éste es un logro de todos nosotros.

RECONOCIMIENTOS Y AGRADECIMIENTOS

A todos los profesores de esta escuela que se sientan aludidos al haber contribuido en mi formación académica. Especialmente al Profesor José Milá de la Roca por haberme permitido la oportunidad de trabajar con él y compartir su sapiencia conmigo a lo largo de estos años y en el transcurso de este proyecto. A ustedes, Muchísimas gracias.

Senesi L., Walter E.

ANALIZADOR DE FUNCIONES DE ESCALA (A.F.E.)

Tutor Académico: José Milá de la Roca. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Electrónica. Institución: U.C.V. 2006. 99h + anexos.

Palabras Claves: Funciones de escala, Caracterización de funciones de escala, Representación gráfica, Automatización de procesos, Ajustes de escalas eléctricas.

Resumen. Se plantea el estudio, diseño y elaboración de un sistema prototipo capaz de realizar caracterizaciones gráficas a partir de la adquisición de datos de parámetros eléctricos que correspondan a la respuesta de una función de red o escala, con el fin de describir y trazar su comportamiento cuando éstas se encuentran conformadas por un parámetro de valor variable. El sistema en general permite contar con una herramienta que facilita el estudio de una red para realizar ajustes en el comportamiento eléctrico de la misma. El equipo fue concebido y diseñado para ser utilizado en las actividades del laboratorio de la asignatura Diseño de equipo electrónico, específicamente en las inherentes al estudio de funciones de red y funciones de escala.

ÍNDICE GENERAL

	Pág.
• CONSTANCIA DE APROBACIÓN	
• DEDICATORIA	
• AGRADECIMIENTOS	
• RESUMEN.....	v
• ÍNDICE GENERAL.....	vi
• ÍNDICE DE TABLAS.....	viii
• ÍNDICE DE FIGURAS.....	ix
• INTRODUCCIÓN.....	1
CAPÍTULO I	
• PLANTEAMIENTO DEL PROBLEMA.....	3
• OBJETIVOS.....	4
• MARCO REFERENCIAL.....	5
1. Funciones de Red y Funciones de Escalas Lineales.....	5
2. Motores de Paso.....	10
2.1. Historia.....	10
2.2. Clasificación.....	10
2.2.1. Motor de Reluctancia Variable.....	11
2.2.2. Motor de Cilindro Imantado.....	12
2.2.3. Motor Híbrido.....	13
2.2.4. Motor de Fase Simple.....	14
2.3. Características.....	14
2.4. Ecuaciones Características.....	17
2.5. Principios de Manejo a Lazo Abierto.....	18
CAPÍTULO II	
• PLANTEAMIENTO DEL SISTEMA PROTOTIPO.....	23
1. Hardware.....	23
1.1. Bloque Manejador de Motor de Paso.....	24
1.2. Bloque detector de Traba del motor de paso.....	25
1.3. Bloque Adecuador de la señal de entrada.....	29
1.4. Bloque sensor de temperatura ambiente.....	31
1.5. Bloque interfaz de comunicación entre TTL y RS-232.....	32
1.6. Microcontrolador.....	32
1.7. Bloque comparador de nivel de señal.....	35
1.8. Bloque selector de señal Tensión-Corriente.....	36
1.9. Bloque Referencias de Tensión.....	37
1.10. Bloque Convertidor PMW-VCO.....	37
1.11. Bloque extensión de memoria RAM.....	38
1.12. Bloque de Alimentación.....	40
1.13. Bloque Convertidor de tensión AC-DC.....	41

2. Software.....	42
2.1. Programación del microcontrolador.....	42
2.1.1. Módulo USART.....	43
2.1.2. Módulo convertidor analógico digital A/D.....	46
2.1.3. Módulo CCP1 (Captura, Comparación, PWM).....	47
2.1.4. Módulo CCP2 (Captura, Comparación, PWM).....	47
2.1.5. Puerto B.....	48
2.2. Programación del computador personal.....	49
CAPÍTULO III	
• CARACTERÍSTICAS Y COMPORTAMIENTO DEL PROTOTIPO CONSTRUIDO.....	51
1. Descripción física y funcionamiento general.....	51
2. Criterios y características del hardware construido.....	54
3. Características del software implementado.....	68
3.1. Programa del microcontrolador.....	68
3.2. Programa principal del computador.....	73
4. Resultados obtenidos.....	88
4.1. Precisión alcanzada.....	89
4.1.1. Precisión en la adquisición de Tensión A.C. y D.C.	89
4.1.2. Precisión en la adquisición de Corriente A.C. y D.C.	91
4.1.3. Precisión en la adquisición de Frecuencia.....	91
4.1.4. Precisión en la adquisición de Período.....	92
4.2. Exactitud alcanzada.....	93
• CONCLUSIONES.....	95
• RECOMENDACIONES.....	97
• BIBLIOGRAFÍA.....	100
• ANEXOS.....	101
1.Plano Electrónico del sistema A.F.E.....	101
2.Diagrama de conexión de la Tarjeta Prototipo construida	102
3.Diagramas de Flujo de funciones del Microcontrolador.....	103
4.Listado del programa del Microcontrolador.....	113
5.Listado del programa del Computador Personal.....	160
6.Lista de partes electrónicas de la tarjeta prototipo.....	216
7.Portadas de hojas de datos de los componentes utilizados.....	218

ÍNDICE DE TABLAS

Tablas	Pág
1. Secuencia de energización del motor de paso.	21
2. Comandos de petición de acción PC → μ Controlador.	44
3. Trama de datos provenientes de la conversión A/D y su significado, del μ Controlador al PC.	45
4. Trama de datos del período de señal y su significado, del μ Controlador al PC.	45
5. Trama de datos de avisos emergentes y su significado, del μ Controlador al PC.	46
6. Valores resumidos de ganancia del sistema de adquisición.	61
7. Valores resumidos de pedestal D.C. del sistema de adquisición.	62
8. Resumen de niveles de tensión de corrimiento D.C. del amplificador instrumental en función de la ganancia para modo de medición en tensión.	63
9. Resumen de niveles de tensión de corrimiento D.C. del amplificador instrumental en función de la ganancia para modo de medición en corriente.	64
10. Resumen de tensión D.C. de salida para controlar el V.C.O del generador de Funciones HP3311A.	86
11. Resultados estadísticos de la adquisición de una señal de tensión D.C.	90
12. Resultados estadísticos de la adquisición de una señal de tensión A.C.	90
13. Resultados estadísticos de comportamiento de la adquisición de una señal de Corriente D.C.	91
14. Resultados estadísticos de comportamiento de la adquisición de una señal de Corriente A.C.	91
15. Caracterización de precisión al adquirir datos de frecuencia.	92
16. Caracterización de precisión al adquirir datos de Período.	92
17. Precisión alcanzada con el modo de medición “# muestras en cada paso” para tensión D.C.	93
18. Precisión alcanzada con el modo de medición “# muestras en cada paso” para tensión A.C.	93

ÍNDICE DE FIGURAS

Figuras	Pág.
1. Pasos dados al energizar las 3 fases del motor de Reluctancia variable.	12
2. Secciones dentadas del rotor Multi-Stack.	12
3. Motor de Cilindro imantado y sección transversal.	13
4. Sección transversal de un motor Híbrido.	13
5. Oscilaciones del eje en un motor de una fase.	16
6. Diagrama de bloques de manejo a lazo abierto del motor de paso.	18
7. Energización de las bobinas para dar un paso.	20
8. Circuito simplificado del estator.	20
9. Configuración de puente H de 2 fases.	22
10. Configuración Darlington para estator de 4 fases unipolar.	22
11. Diagrama de bloques del sistema A.F.E.	23
12. Diagrama de bloques del dispositivo manejador del motor de paso.	24
13. Diagrama circuital del integrado MC3479-D.	25
14. Modelo circuital de las bobinas del estator del motor de paso.	26
15. Módulo detector de Traba de motor.	28
16. Bloque adecuador de la señal a caracterizar.	29
17. Bloque del amplificador instrumental INA121.	29
18. Diagrama de bloques del potenciómetro digital X9C503.	30
19. Bloque detector de cruce por cero.	31
20. Diagrama de bloques del medidor de temperatura TMP122.	31
21. Diagrama circuital de la interfaz RS232.	32
22. Diagrama de terminales del microcontrolador.	33
23. Diagrama del bloque de comparación de la señal.	35
24. Bloque selector de medición V-I.	36
25. Bloque proporcionador de las referencias de tensión.	37
26. Diagrama circuital del adecuador PWM-VCO.	38
27. Diagrama circuital de Bloque de RAM y Microcontrolador.	39
28. Circuito combinacional para el manejo de la expansión RAM.	39
29. Bloques de alimentación de 5V, 12V y -12V.	40
30. Bloque convertidor A.C.-D.C.	41
31. Aspecto del equipo A.F.E.	56
32. Detalle de la tarjeta prototipo con la placa deflectora de temperatura.	56
33. Diagrama de bloques del sistema analógico de adquisición.	60
34. Respuesta en frecuencia del rechazo a modo común del INA121.	66
35. Apariencia en ventana del menú principal.	75
36. Apariencia de la ventana “en accionamiento fijo”.	76
37. Ventana para realizar alteración del parámetro θ .	77
38. Ventana para seleccionar el modo de adquisición con θ fijo.	77
39. Ventana “Módulo Graficador”.	78
40. Ventana para la inclusión de la función teórica.	80
41. Aspecto de la ventana selectora de columnas para el cálculo de error.	82

42. Ventana para el manejo del motor de paso.	84
43. Ventana de configuración para potenciómetros digitales.	85
44. Aspecto de la ventana “ <i>A excitación externa</i> ”.	87
45. Aspecto de la ventana “ <i>Visor del Sistema</i> ”.	87
46. Integrado AD7731 como bloque de adquisición.	95
47. Aspecto e implementación física del disco codificado para detectar traba mecánica	97

INTRODUCCIÓN

Inmerso en el campo de la Ingeniería Eléctrica, específicamente en la especialidad de Electrónica, se encuentra que los estudios teóricos y conceptuales del Diseño de circuitos Electrónicos arrojarán verdaderas directrices que apuntan en dirección correcta y concreta hacia el desarrollo optimizado de nuevas tecnologías, formas de utilización, modernización en la metodología de procesos existentes y mejoras en los niveles electrónicos de integración, por nombrar alguno de ellos.

El estudio de circuitos bajo el enfoque del Diseño Electrónico en sí supone el poder describir, entre otras cosas, el comportamiento de una red eléctrica que pretende desempeñar una tarea específica en el mundo real. Esta realidad requiere de una cantidad de procedimientos a seguir y de consideraciones a tomar para lograr el correcto funcionamiento del equipo en cuestión. En general, existe la necesidad de poder describir el funcionamiento del circuito usando la ciencia exacta, basada en la Física y la Matemática. Es en este punto donde aparece la definición de Funciones de Red.

Las Funciones de Red son arreglos matemáticos que permiten plantear la dependencia de una variable eléctrica en función de otras variables físicas (eléctricas o no) tomando en cuenta los componentes que interactúan entre sí para formarla. La Función de Red pretende plantear un modelo matemático válido que represente el comportamiento del circuito sirviendo de puente entre el mundo real y el campo matemático para permitir la interacción, estudio, alteración y modificación de la misma hasta conseguir el comportamiento deseado en el circuito en estudio.

Tomando en consideración la relevancia inmersa en las Funciones de Red, es importante señalar entonces que el comportamiento del circuito será completamente descrito por la misma. Lógico es pensar que contando con un sistema capaz de realizar mediciones de las variables eléctricas más relevantes, (Tensión, Corriente, Frecuencia y Período), de un circuito genérico, se podrá trazar la característica de la

Función de Red de manera experimental para su estudio, análisis, comparación y modificación.

En la Escuela de Ingeniería Eléctrica de la Universidad Central de Venezuela, dentro de la especialidad de Electrónica, específicamente en la asignatura Diseño de Equipo Electrónico se planteó la posibilidad de desarrollar un sistema que fuese capaz de automatizar la caracterización de Funciones de Red, específicamente al realizar las actividades previstas en las prácticas inherentes a dicha materia.

Para la elaboración de este proyecto se hizo primeramente una investigación exhaustiva de la teoría envuelta en el problema, los requerimientos, métodos y procedimientos a utilizar; luego se realizó un estudio y posterior selección de bloques electrónicos y herramientas de software para su completo desarrollo según las necesidades encontradas, seguidamente se ejecutó la implementación de un prototipo con el propósito de realizar pruebas piloto que arrojarán resultados tendientes a dilucidar tanto el software como el hardware adecuado para, finalmente, construir el sistema definitivo.

El contenido de este trabajo está dividido en 3 capítulos, el capítulo I hace referencia al marco teórico inherente al proyecto. El capítulo II especifica el diseño de las etapas de software y hardware involucradas en la hechura del prototipo, luego de haber concluido la etapa investigativa. El capítulo III muestra los resultados obtenidos al hacer uso del sistema Analizador de Funciones de Escala (A.F.E.). Por último se presentan las recomendaciones y conclusiones para futuros proyectos con miras a la expansión del prototipo diseñado.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

Vista la importancia que tiene el estudio de las funciones descriptivas de un circuito Eléctrico (Funciones de Red) en el contexto práctico de la materia Diseño de Equipo Electrónico, se planteó que las actividades inherentes a la misma pudiesen ser realizadas de una forma automatizada por parte de los estudiantes mediante un sistema capaz de recolectar, procesar, graficar y almacenar todos aquellos datos eléctricos provenientes de un análisis práctico que, como respuesta, arroje un circuito específico. El principal propósito es el de minimizar el tiempo y el esfuerzo del estudiante en tomar mediciones mediante procedimientos tediosos, repetitivos y que en sí poseen poco contenido conceptual. Aunado a ésto último está el simple hecho que dichas mediciones realizadas de forma manual por parte del alumno están sujetas a errores operativos que pudieran alterar el resultado esperado de la experiencia realizada.

La idea de automatizar el procedimiento de caracterización de las Funciones de Red facilita, mejora y optimiza la realización de las actividades prácticas que se tengan previstas ya que, primeramente, se consigue minimizar el tiempo invertido en realizar las experiencias, lo que se traduce inmediatamente en tiempo disponible al estudiante con el que podrá contar para avocarse a la comprensión de la teoría involucrada en el proceso estudiado. Segundo, se contará con un sistema cuyo comportamiento al realizar la adquisición de datos será siempre el mismo, asegurando así la repetibilidad de las pruebas en las experiencias realizadas y minimizando la contribución de errores operativos por parte del estudiante. Tercero, dado que el proyecto planteado realizará la caracterización de la Función de Red y de Funciones de Escala de manera automática, el mismo será capaz de enlazar los datos recolectados adecuando los mismos a las formas y protocolos de estructuras de datos existentes en los computadores personales y herramientas de cálculo estándar permitiendo así la compatibilidad entre los sistemas involucrados.

Objetivo General

El presente trabajo tiene como objetivo diseñar y construir un prototipo de Analizador dinámico de Funciones de Escala (A.F.E.) para ser empleado en el laboratorio de la asignatura Diseño de Equipo Electrónico, Escuela de Ingeniería Eléctrica (E.I.E.) de la Universidad Central de Venezuela (UCV).

Objetivos Específicos

Los objetivos específicos son los siguientes:

1. Estudiar el funcionamiento y manejo de motores de paso.
2. Estudiar el funcionamiento y manejo de los sistemas bidireccionales de información.
3. Caracterizar el sistema actuador mecánico.
4. Diseñar la estructura en base a bloques específicos.
5. Programar el microcontrolador empleado para el correcto desempeño del prototipo.
6. Programar la comunicación entre el prototipo y el computador personal.
7. Programar la interfaz Hombre Máquina en el computador personal.
8. Implantar el hardware (tarjeta soldada).
9. Probar y depurar el prototipo.
10. Elaborar las ayudas para el usuario.

Marco Referencial

1. Funciones de Red y Funciones de Escalas Lineales, Generalidades:

Las Funciones de Red son simplemente expresiones algebraicas, pertenecientes al campo matemático con una ley de formación proveniente del mundo físico. Las mismas sirven para plantear un modelo ajustado a la realidad el cual toma en cuenta las variables involucradas en el fenómeno observado. En general, se tendrá a las funciones de red descritas por “n” variables independientes (siendo “n” el número de elementos distintos que la conforman) en un entorno de validez que estará delimitado por las características físicas de los elementos que la componen. Este enfoque permite entonces contar con herramientas matemáticas existentes para realizar un estudio que logre caracterizar la respuesta de un circuito eléctrico cuyo comportamiento está predicho por la teoría pertinente. A su vez, permite incluir, sopesar y minimizar el efecto de la tolerancia de los componentes de dicha función en la respuesta del circuito. Lo anterior es debido a que los componentes involucrados pertenecen al mundo real e irremediablemente, tenderán a alterar el valor del resultado deseado. Como modelo matemático de primer orden, válido en un entorno delimitado por la realidad física de los componentes, se puede plantear que en general las funciones de red tienen la forma siguiente:

$$f(X)\big|_{X=X_0} = f(X_0) \pm \Delta f(X_0) \quad (1)$$

con X escalar compuesto por X(a, b, c,...,n) elementos.

X₀ es el valor nominal de cada elemento que compone a X.

Desde el punto de vista matemático, la descripción algebraica que brinda la Función de Red no es más que un lugar n Dimensional en el Espacio, en donde cada variable involucrada en la fórmula es un plano. Ocurre que, realizando las intersecciones de estos n planos en el espacio se forma una traza que define un lugar geométrico único que es entonces la característica gráfica de la Función de

Red bajo estudio, delimitada únicamente por la validez del modelo planteado. Ahondando en este punto y visto de otro modo, la delimitación del modelo matemático coincidente con la realidad física queda establecida por la naturaleza de las variables involucradas en el arreglo.

El estudio matemático de las Funciones de Red hace factible la posibilidad de corregir el valor de una función de red en un punto de trabajo deseado mediante variaciones en los valores de los elementos que conforman la red circuital. Teóricamente, existen muchos métodos para lograr este objetivo entre los cuales figura el poder reemplazar a uno de los elementos del circuito estudiado por una nueva red que posea, entre otras cosas, un elemento de valor variable accionado mecánicamente. La adición de este elemento permitirá que el valor de la Función de Red converja a un punto deseado debido a la modificación mecánica conveniente del valor nominal de la nueva red incluida. Este procedimiento es conocido como Ajuste(s) de la Función(es) de Red por alteración de variable(s).

En cuanto a las funciones que representan Escalas Lineales se puede decir que éstas pueden ser tomadas como un caso particular de las Funciones de Red debido a que las mismas se encargan de definir el comportamiento de un equipo y a su vez, servir de vínculo físico entre el usuario y el sistema. Se plantea entonces que, dado que el usuario modifica un elemento variable perteneciente al circuito, el mismo altere su valor de tal manera que logra hacer que la Función(es) de Red pertinente(s) respondan adecuadamente a la solicitud hecha externamente al circuito con un cambio de respuesta. Desde otro ángulo se puede extrapolar el concepto de Ajuste de Función de Red al concepto de Escala Lineal ya que se observa que el punto de trabajo de una función de red se modifica dado un accionamiento mecánico que no es más que la alteración de variable(s) que describen el funcionamiento del equipo. Así mismo se observa que la Función de Escala altera su valor de respuesta dado, entre otras cosas, un accionamiento que varíe por lo menos uno de sus parámetros. Las Funciones de Red ajustadas

mediante el método de alteración de variable pueden ser vistas también como una Función de Escala.

En el caso particular de una Función de Escala Lineal, se puede plantear, de manera teórica, un modelo con la siguiente forma general:

$$f(X, \theta) = A(X) + B(X) \cdot \theta \quad (2)$$

donde:

$A(X)$: Valor mínimo ó máximo de la escala.

$B(X)$: Constante de proporcionalidad deseada.

θ : Accionamiento del elemento variable, su cambio puede ser de cualquier índole entre los que cabe la naturaleza mecánica, este cambio propicia la alteración en el valor de algún elemento del circuito.

Para el caso en que se desee caracterizar una Función de Escala se observa que la misma tendrá siempre un error o incertidumbre al compararla con la escala teórica pertinente, estas diferencias pueden ser justificadas por los siguientes motivos:

1. Error por construcción: Se manifiesta en la diferencia de valores entre las trazas teórica y práctica, por la asociación eléctrica que resulta entre los elementos que componen la red. Dado esto, las características esperadas en el comportamiento de la Función de Escala resulta alterada.
2. Error por tolerancias: Dado que los elementos constitutivos del circuito son reales, los mismos poseen un intervalo de posibles valores válidos que depende de las tolerancias involucradas y de la forma en que éstas afectan la función bajo estudio, en consecuencia, la traza que resulta del estudio práctico ya no será única sino que por el contrario existirá una región delimitada por el intervalo de validez de θ y por la familia de trazas resultantes al considerar todos los posibles valores que puedan

tomar todas las variables que forman la Función bajo análisis. En general, este error existe por las incertidumbres de $A(X)$ y $B(X)$.

3. Error por derivación: Esta fuente de error ocurre por la naturaleza real en la construcción del elemento variable que parametriza la Función de Escala con respecto de θ . Tiene que ver con la forma en que el elemento cambia su valor real al plantear un cambio en la variable θ . Ésto se conoce también como incertidumbre de derivación del elemento.

Las fuentes de error antes expuestas serán apreciadas en la caracterización de la Función de Escala de forma conjunta, las mismas serán tratadas por separado para lograr minimizar su efecto y así acercar lo más posible la función práctica a la teórica disminuyendo el error existente.

En el caso de las Funciones de Red y también para el caso de las Funciones de Escala se observa que la descripción matemática de la función bajo estudio es función de la variable de accionamiento θ , por lo que, en general, resulta factible y obvio el hecho de poder caracterizar la respuesta del sistema (en tensión, corriente, frecuencia o período) haciendo variaciones controladas del parámetro θ .

Poder caracterizar la respuesta de una red eléctrica mediante un barrido del parámetro θ (en los casos en los que la Función de Red posea dependencia de dicha variable) amerita ciertas consideraciones. Primero, al realizar la variación del parámetro θ debe asegurarse que el avance o cambio planteado sea constante, es decir, que siempre la variable θ sea incrementada en pasos fijos. Dado que físicamente se define a la variable θ como el valor normalizado de un ángulo mecánico válido cualquiera con respecto a su valor máximo de accionamiento, se requiere en consecuencia que el avance necesario en el sistema debe ser en ángulos por paso. Segundo, se debe garantizar que este cociente se mantenga constante a lo largo de la experiencia, garantizando así el valor de accionamiento θ y, por consiguiente, la correcta caracterización de la Función de Escala. Al tener asegurado el valor θ en cuestión se debe también tener la certeza de que en el transcurso de tiempo que dure la adquisición de los datos pertinentes en ese

punto de accionamiento, la variable θ no sufra modificaciones, es decir, se debe asegurar la estabilidad mecánica del elemento variable en cada paso dado para luego proceder a adquirir datos concernientes a la caracterización. Tercero, las variaciones de θ deben ocurrir de tal forma que siempre se asegure la monotonicidad (creciente o decreciente) en la caracterización de la Función de Red o en la Función de Escala debido a que solo así se puede garantizar que el análisis hecho en el circuito es verdaderamente representativo de la realidad. Obsérvese que si no se tiene garantizada la monotonicidad en el cambio de la variable θ se podría caracterizar erróneamente la Función bajo estudio, introduciendo variaciones en la traza real que en realidad son inexistentes, conduciendo la experiencia a conclusiones poco confiables e incorrectas.

2. Motores de Paso:

2.1. Historia:

La aparición práctica de los motores de paso remonta aproximadamente al año 1920 en donde fueron expedidas patentes de invención en el Reino Unido. Una de estas patentes fue otorgada a un ingeniero civil Escocés llamado C.L. Walker en Aberdeen. Este ingeniero presentó un motor de paso capaz de hacer avanzar su eje en pequeños pasos angulares debido al diseño de su rotor y estator, dado que éstos fueron proyectados como estructuras dentadas. En los Estados Unidos fue expedida otra patente a dos inventores llamados C.B. Chicken y J.H. Thain en la localidad de Newcastle en el año 1920. Estos inventores consiguieron diseñar un motor de paso que poseía gran torque mecánico por unidad de volumen geométrico del motor. Se consigue también que los motores de paso formaron parte importante de la maquinaria de defensa Británica durante la Segunda Guerra Mundial al ser empleados desde 1927 para posicionar cañones de barcos de guerra, así como para controlar el direccionador de torpedos por control remoto. Poco tiempo después las fuerzas navales Estadounidenses adoptaron sistemas similares con los mismos fines.

No obstante, sólo fue hasta después de 1950 que los motores de paso comenzaron a ser empleados comercialmente, extendiéndose su uso por su habilidad natural de posicionar su eje en un punto deseado sin muchas complicaciones. Más tarde, en la década del 60 comenzaron a ser usados como parte de maquinaria automatizada de lectura/escritura en sistemas de tarjetas perforadas. Los usos desde entonces han sido innumerables y muestra de ello es que aún en nuestros días los motores de paso aparecen en sistemas de posicionamiento mecánico, como por ejemplo, en las unidades de almacenamiento de datos (Disquetes, Discos Duros, CDROM/CDRW), Impresoras, Faxes, Posicionadores rotativos, Unidades de control numérico, entre otros.

2.2. Clasificación

Los motores de paso se clasifican según su construcción en dos grupos, el Motor de Reluctancia Variable y el Motor de Magneto Permanente. A su vez, el

Motor de Magneto Permanente posee 3 tipos conocidos como: de Cilindro Imantado, Motor Híbrido y por último el Motor de Paso de Fase Simple. A continuación se presenta una breve explicación de cada uno de ellos.

2.2.1. Motor de Reluctancia Variable:

Es el motor de paso que presenta la forma más básica de funcionamiento, el mismo posee un rotor hecho de acero dulce, el cual posee una alta permeabilidad magnética. El estator a su vez está compuesto por un número de polos electromagnéticos formados por bobinas arrolladas las cuales son activadas por una fuente magnetomotriz externa que, a su vez, es la encargada de generar el flujo magnético para que ocurra el fenómeno. Se le conoce como motor de reluctancia variable debido al cambio de valor que sufre este parámetro al ocurrir movimiento del rotor.

Analizando la dinámica del sistema se observa que cuando el rotor se encuentra alineado con uno de los polos del estator, el sistema consigue su punto de equilibrio y la reluctancia equivalente del circuito magnético en este punto es la mínima posible. Al ocurrir un cambio en la posición del rotor con respecto al polo energizado del estator ocurre que aumenta la reluctancia equivalente del circuito magnético y se produce un torque contrario al de la perturbación introducida que busca recuperar la posición de equilibrio perdida. Este fenómeno es conocido como Estrés de Maxwell. Es necesario comentar que de esta manera el motor de paso asegura la posición mecánica de su rotor. Para lograr el avance del rotor, en el motor se procede a excitar eléctricamente el polo contiguo a aquel que se encuentre actualmente energizado, esto produce una nueva circulación de flujo de campo magnético en otra dirección que tenderá a desequilibrar al rotor debido a que la reluctancia del circuito equivalente aumenta. Las líneas de campo magnético sometidas a estrés de Maxwell provocarán torque mecánico para lograr minimizar la reluctancia del sistema logrando mover el eje hasta la nueva posición de reposo, alcanzando así el equilibrio mecánico nuevamente. Obsérvese que de esta manera se ha logrado mover el rotor una cantidad conocida como paso del motor. La secuencia

de movimiento se muestra en la figura 1, la misma explica el movimiento del rotor al energizar las bobinas (fases) del estator.

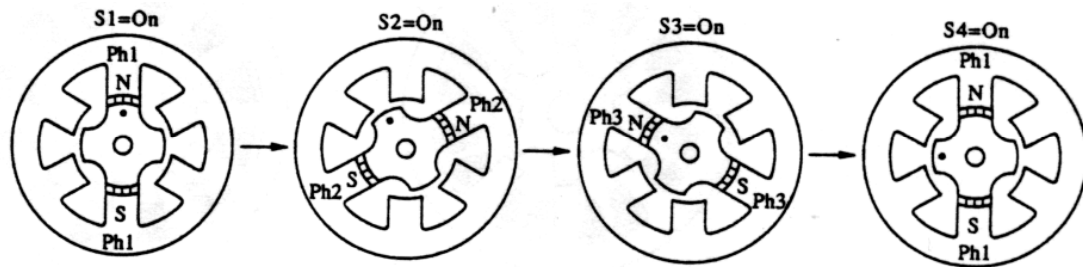


Figura 1. Pasos dados al energizar las 3 fases del motor de Reluctancia variable.

De los motores de reluctancia variable se encuentra también otras configuraciones geométricas de su rotor en donde se aumenta el número de secciones dentadas en un arreglo concéntrico y cuya alineación entre dientes es por lo general desplazada 1/3 de diente. Ésto con el fin de disminuir la cantidad de grados mecánicos por paso dado. A estos arreglos se les conoce como de tipo Multi-Stack. La figura 2 muestra un ejemplo de este tipo de rotores.

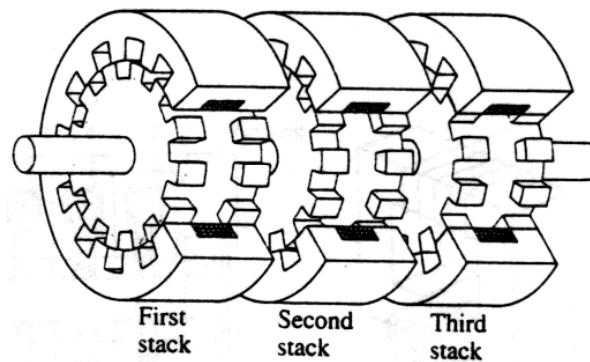


Figura 2. Secciones dentadas del rotor Multi-Stack.

2.2.2. Motor de Cilindro imantado:

El principio de funcionamiento de este motor es similar al que ocurre en los motores sincrónicos de corriente alterna. El estator del motor está compuesto de devanados en cada polo del mismo, al energizar el sistema, el rotor, que es un imán permanente, experimentará un torque que buscará orientarse convenientemente con los polos magnéticos del estator. Al ir energizando en secuencia a cada polo del

motor, el rotor gira, buscando alinearse en dirección del campo magnético provocado por los nuevos polos energizados, produciendo así el movimiento angular del eje. Este motor logra valores altos de torque en un espacio volumétrico reducido cuando se compara con el motor de reluctancia variable. La figura 3 muestra la sección transversal de este tipo de motor.

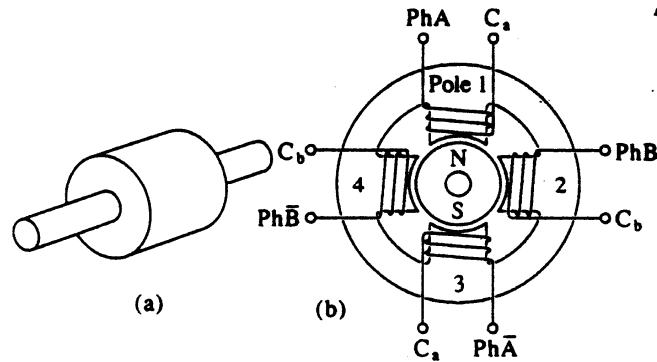


Figura 3. a) Cilindro imantado; b) Sección transversal del motor

2.2.3. Motor Híbrido:

Es otro tipo de motor de paso con rotor magnetizado. Las diferencias esenciales con respecto al motor de reluctancia variable son 2, la primera se debe a que la estructura de su rotor es dentada. Segunda, este motor logra un incremento en el torque entre paso y paso, debido al rotor imantado. Gracias a la combinación de estas dos características se le conoce con este nombre. El motor híbrido es el más usado entre todos los motores de paso. Ver figura 4.

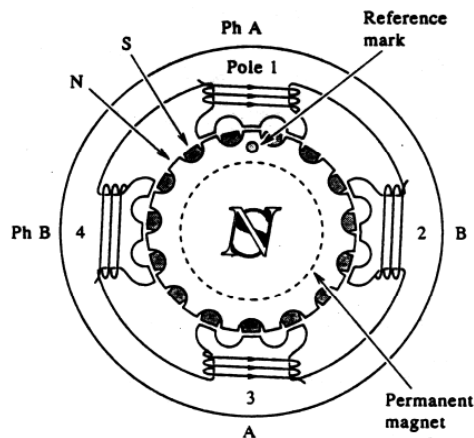


Figura 4. Sección transversal de un motor Híbrido.

2.2.4. Motor de Fase simple:

Este motor funciona basado en una diferencia geométrica entre los dientes del rotor y del estator, buscando distribuir el flujo magnético de tal forma de hacer girar el eje del motor al introducir una excitación magnetomotriz en sus polos. Este motor es conocido también como “Cyclonome”. Posee 3 polos, dos de ellos son los polos manejadores y el tercero es conocido como el polo de detención.

2.3. Características:

Entre las características favorables más relevantes de los motores de paso se enumeran las siguientes:

- Son sistemas que pueden operar a lazo abierto, siendo manejado por un secuenciador lógico y alimentado prácticamente de manera directa por una fuente de corriente directa (D.C.). No requieren de complicados sistemas de control para operar aunque esto restringe su rango de aplicabilidad debido a inestabilidades en cuanto al movimiento del eje con respecto a la velocidad de giro que puede alcanzarse.
- El cambio de posición angular que describe su eje es prácticamente el mismo, de esta forma se asegura que el paso dado corresponderá a un ángulo determinado por las características de construcción del motor usado.
- En cuanto a la estabilidad mecánica del rotor una vez transcurrida la dinámica del sistema al dar un paso, el motor asegura la nueva posición del eje debido a la concepción del funcionamiento natural del mismo con el fenómeno de reluctancia variable y el fenómeno de Estrés de Maxwell. En la práctica, esta característica es conocida como Par de Aguante (Holding Torque). En los casos en los que el motor usado tenga rotor magnetizado, el motor exhibe otra característica que es la de proporcionar una fuerza de restauración contraria al posible movimiento del eje aún cuando las bobinas de los polos se encuentran

desenergizadas. Esta propiedad es conocida como Par de Detención (Detent Torque).

- Poseen par mecánico que se encuentra en el rango de las milésimas hasta las decenas de N·m al avanzar de un paso a otro. Ésto permite el poder usar el motor con cargas acopladas al eje para realizar trabajo mecánico.
- Proporcionan gran exactitud en cuanto al error en la posición del eje luego de haber dado un paso. Este error puede ocurrir en cada paso siendo muy pequeño (en magnitud alcanza, a lo sumo, las décimas de grado) dada la geometría de construcción del rotor y del estator. La otra característica es que el error de posición es no acumulativo en este tipo de motores.
- El motor de paso asegura que su función de transferencia de pulsos de entrada vs. movimiento angular de salida, es una función monotónica.
- En general, los motores de paso poseen gran rapidez de respuesta al arranque y a las paradas mecánicas del eje. Este hecho asegura que el motor no pierde pasos en el tránsito planteado por el sistema de control que lo gobierne. Estas bondades se deben al cociente entre el par de fuerza del eje y su momento de inercia.
- Por su diseño, son robustos y duraderos.

Entre las desventajas de los motores de paso se encuentran las siguientes:

- Respuestas variables en la característica Par de Fuerza con respecto a la Velocidad de Giro. La misma presenta picos y valles en donde la característica mencionada se ve afectada, limitando así el uso de los motores de paso a zonas en donde la velocidad angular sea conveniente y a su vez proporcione suficiente torque a la aplicación.
- La posición del ángulo de salida con respecto al pulso que desencadena el avance es en general oscilatoria, mostrando la

respuesta de un sistema de, por lo menos, segundo orden ante el impulso, con oscilaciones transitorias hasta alcanzar la nueva posición de equilibrio. Este hecho deberá ser tomado en cuenta en el momento de querer asegurar un paso mecánico estable. Existen métodos para lograr disminuir estas oscilaciones que requieren de compensación mecánica. La figura 5 muestra el trazado del movimiento angular de un motor de paso con respecto al tiempo donde se observa el carácter oscilatorio del rotor al dar un paso (respuesta al pulso).

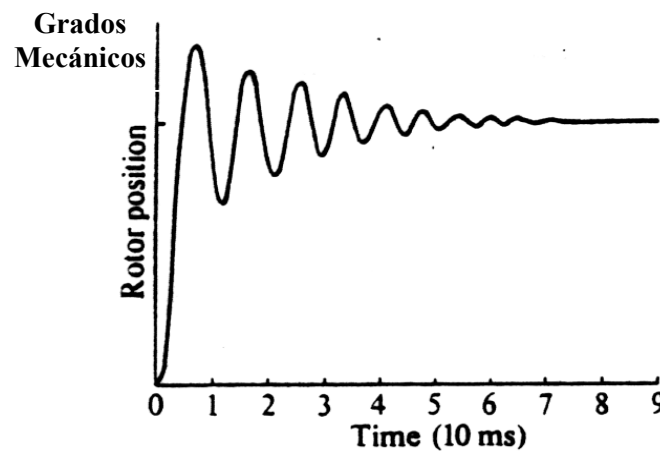


Figura 5. Oscilaciones del eje en un motor de una fase.

- Se debe prestar particular atención al diseño de la fuente de alimentación del manejador que energizará al motor de paso dado que éste presenta fuertes conmutaciones de corriente debido a las bobinas involucradas y al tipo de señal utilizada para hacer evolucionar el sistema. Ésto se traducirá en perturbaciones electromagnéticas a los circuitos físicamente cercanos y en ruido en las líneas de alimentación en general.
- En los casos en los que se requiera utilizar un sistema de control a lazo cerrado se deberá contar con sensores de posición muy sofisticados como lo son los codificadores digitales de posición. La inclusión de estos transductores encarece el sistema planteado de manera importante.

2.4. Ecuaciones características:

Las ecuaciones que rigen el comportamiento del motor son las siguientes:

- Torque en el eje:

$$\begin{aligned}\tau_a &= -p \cdot n \cdot \Phi_m \cdot i_a \cdot \text{Sen}(p \cdot \theta) \\ \tau_b &= p \cdot n \cdot \Phi_m \cdot i_b \cdot \text{Sen}(p \cdot (\theta - \lambda))\end{aligned}\quad (3)$$

Donde:

p = Número de pares de polos del motor.

n =Número de vueltas en el devanado.

Φ_m =Cantidad de flujo magnético.

i_a, i_b =Corriente circulante por las bobinas.

θ =Ángulo de salida.

λ =Desplazamiento físico entre polos y devanados.

- Fuerza Contraelectromotriz Inducida:

$$e_f = -n \cdot \frac{d\phi}{dt} = (n \cdot Nr \cdot \Phi_m \cdot \text{Sen}(Nr \cdot \theta) \cdot \dot{\theta}) \quad (4)$$

Donde:

Nr = Número de dientes del rotor.

- Ecuación diferencial del movimiento del rotor:

$$J \cdot \frac{d^2\theta}{dt^2} + D \cdot \frac{d\theta}{dt} + p \cdot n \cdot \Phi_m \cdot i_a \cdot \text{Sen}(p \cdot \theta) + p \cdot n \cdot \Phi_m \cdot i_b \cdot \text{Sen}(p \cdot (\theta - \lambda)) = 0 \quad (5)$$

Donde:

J =Momento de inercia vinculado al eje del motor.

D =Coeficiente de amortiguamiento, fricción del aire y corrientes de eddy.

- Transformada de Laplace de la relación posición Entrada Salida:

$$\frac{\Theta_o(s)}{\Theta_i(s)} = \frac{s^2 + \left(\frac{r}{Lp} + \frac{D}{J}\right)s + \left(\frac{r}{Lp} \cdot \frac{D}{J} + K_p \cdot \omega_{np}^2\right)}{s^3 + \left(\frac{r}{Lp} + \frac{D}{J}\right)s^2 + \left(\frac{r}{Lp} \cdot \frac{D}{J} + \omega_{np}^2 \cdot (1 + Kp)\right)s + \left(\frac{r}{Lp}\right) \cdot \omega_{np}^2} \quad (6)$$

Donde:

L= Inductancia propia de la Bobina en estudio.

M= Inductancia mutua en la Bobina bajo estudio.

$Lp = L - M$; Diferencia entre la Inductancia propia e inductancia mutua.

$$Kp = \frac{n \cdot \Phi_m}{Lp \cdot I_o} \cdot \frac{\text{Sen}^2\left(\frac{p \cdot \lambda}{2}\right)}{\text{Cos}\left(\frac{p \cdot \lambda}{2}\right)}$$

$$(\omega_{np})^2 = \frac{2 \cdot p^2 \cdot \Phi_m \cdot n \cdot I_o \cdot \text{Cos}\left(\frac{p \cdot \lambda}{2}\right)}{J}$$

- Torque en función de la fuerza contraelectromotriz:

$$\tau = -\frac{ef \cdot Ia}{\dot{\theta}} \quad (7)$$

2.5. Principios de manejo a lazo abierto:

Para lograr manejar un motor de paso convenientemente se debe contar con un sistema compuesto de al menos tres bloques eléctricos. La figura 6 muestra un diagrama básico de los bloques necesarios.

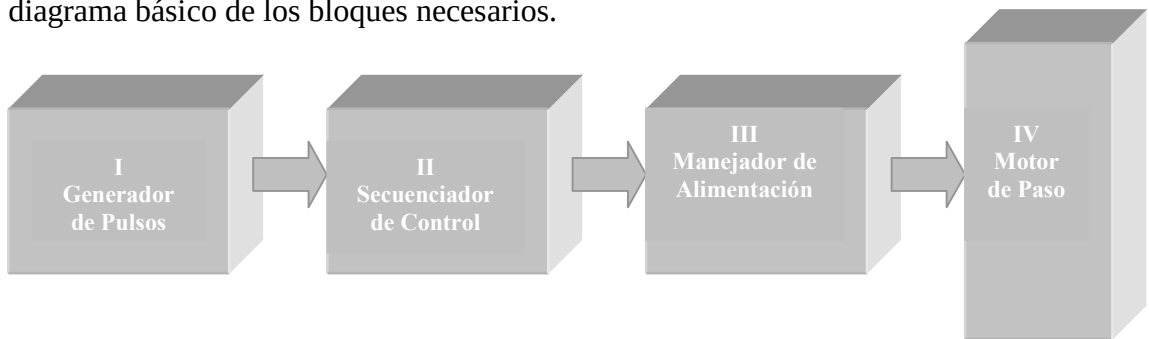


Figura 6. Diagrama de bloques de manejo a lazo abierto del motor de paso.

El primer bloque genera y maneja la secuencia y velocidad de pulsos que autorizarán el movimiento del eje del motor. El segundo bloque se encarga de establecer la secuencia lógica de alimentación para realizar el movimiento del eje en la dirección y ángulo por paso deseado. El tercer bloque, se encarga de energizar las bobinas del motor. Estos bloques se llamarán: Generador de Pulsos, Secuenciador de Control y Manejador de alimentación. El Generador de Pulsos es el encargado de autorizar el movimiento del eje del motor mediante una señal de amplitud y frecuencia adecuada. Esta señal será la entrada al Secuenciador de Control, que por lo general es un sistema digital, así que, la forma de onda proveniente del Generador de Pulsos es por lo general rectangular, con niveles lógicos correspondientes a los estándares de tecnología TTL¹ o CMOS². Generalmente, el Secuenciador de Control hace evolucionar su salida (al Manejador de Alimentación) al haber cambios en su entrada en una relación de ocurrencia de 1:1, por tanto se puede afirmar que al introducir un pulso adecuado mediante el Generador de Pulsos habrá evolución en todo el sistema hasta llegar a hacer cambiar la posición del eje del motor en un paso. Existen muchas formas de implementar el bloque Generador de Pulsos, entre ellas se destacan los microcontroladores como herramienta preferida.

El bloque Secuenciador de Control es el que establece el orden de ocurrencia en el tiempo y la selección apropiada de las bobinas que deben ser energizadas para lograr dar un paso en el eje del motor. Su trabajo es el de definirle la lógica coherente al Manejador de alimentación mediante señales que definen el estado de energización y de polarización de las bobinas en el estator del motor. Adicionalmente, es el encargado de poder controlar el sentido de rotación del eje de manera horaria (CW) o antihoraria (CCW). Como última observación, el bloque Secuenciador de Control también permite plantear que los pasos en avance del motor sean dados en pasos completos o en medios pasos por la acción de la secuencia en la energización de la (o las) bobina(s) del estator. Por lo general este bloque está compuesto de compuertas

¹ Por sus siglas en inglés: Transistor-Transistor-Logic.

² Por sus siglas en inglés: Complementary Metal Oxide Semiconductor.

que realizan funciones lógicas junto a Flip's-Flop's estableciendo Circuitos Secuenciales Sincrónicos, manejados por una señal de reloj, siendo ésta la proveniente del Generador de Pulsos.

Para poder establecer la lógica secuencial inherente al sistema basta con estudiar la forma en que el motor consigue dar un paso en avance. La figura 7 muestra un corte transversal del motor de paso estableciendo una secuencia que muestra como se logra dar un paso. Es necesario aclarar que la figura 7 es una simplificación ilustrativa en donde se muestra un motor de 3 fase, 6 polos, con un diente por cada uno de ellos. El rotor, a su vez, está compuesto de sólo 4 dientes en su periferia. La tabla 1 muestra una secuencia de energización de las bobinas para lograr dar una revolución completa al motor cuando el mismo es manejado con la técnica de paso completo. La figura 8 muestra un modelo de circuito simplificado del estator de un motor de paso de 2 fases.

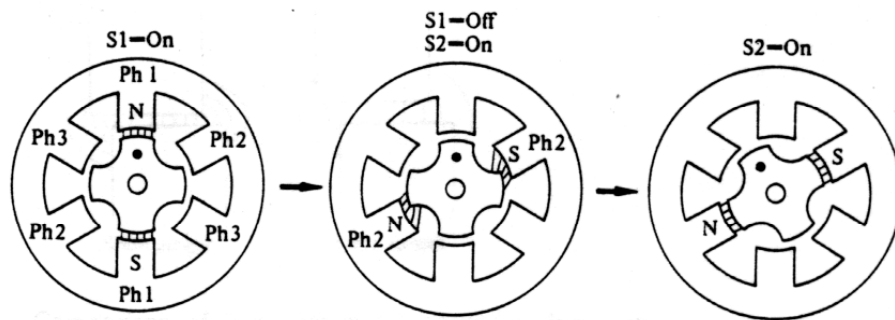


Figura 7. Energización de las bobinas para dar un paso.

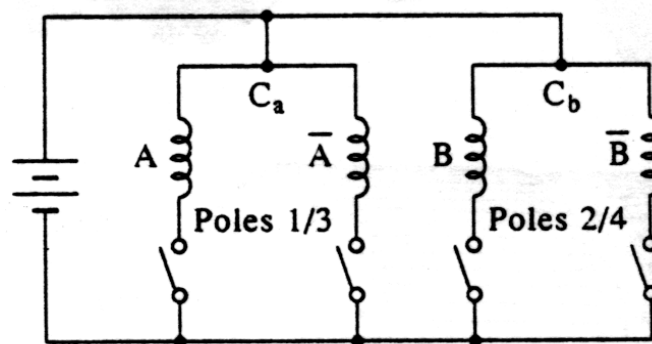
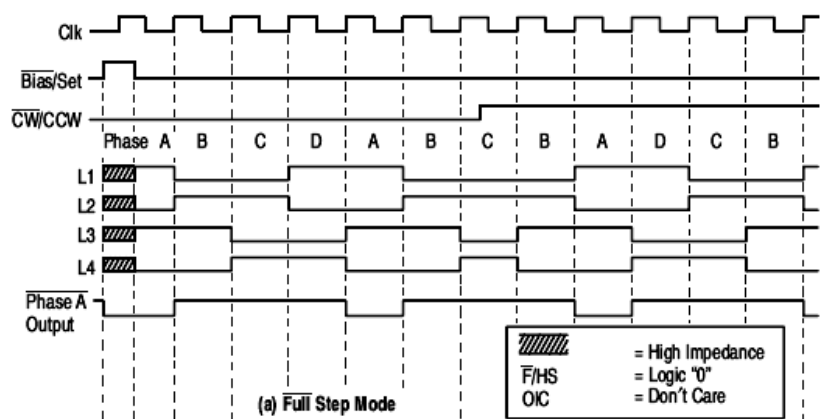


Figura 8. Circuito simplificado del estator.

Tabla 1. Posible secuencia de energización de un motor de paso.



El bloque Manejador de Alimentación tiene a su cargo canalizar la energía proveniente de las fuentes de alimentación a las bobinas del motor de paso. En esencia, este bloque está formado por un arreglo conveniente de transistores de tecnología BJT³ ó MOS⁴ capaces de manejar las corrientes requeridas por las bobinas del motor de paso a usar. Es necesario mencionar que, en la práctica, la resistencia eléctrica equivalente de estas bobinas se encuentra alrededor de los cientos de Ohm, lo que conduce a pensar que las corrientes exigidas a la fuente de alimentación tendrán una magnitud comprendida entre las decenas y los cientos de miliAmper. Se encuentran en el mercado de componentes electrónicos gran variedad de circuitos integrados destinados para este fin como lo son los manejadores en puente H y configuraciones de transistores Darlington unipolares. Ver figura 9 y figura 10 respectivamente.

³Por sus siglas en inglés Bipolar Junction Transistor (Transistor de unión Bipolar).

⁴Por Sus siglas en inglés Metal Oxide Semiconductor (Semiconductor Óxido Metal).

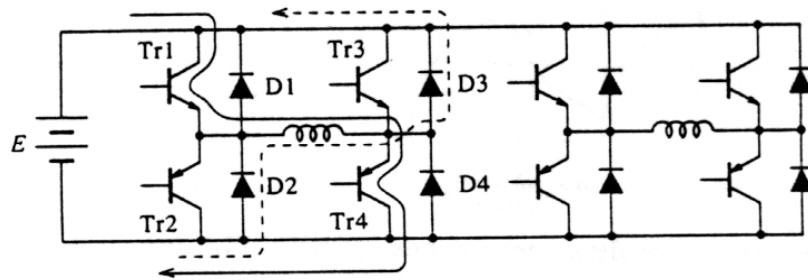


Figura 9. Configuración de puente H de 2 fases.

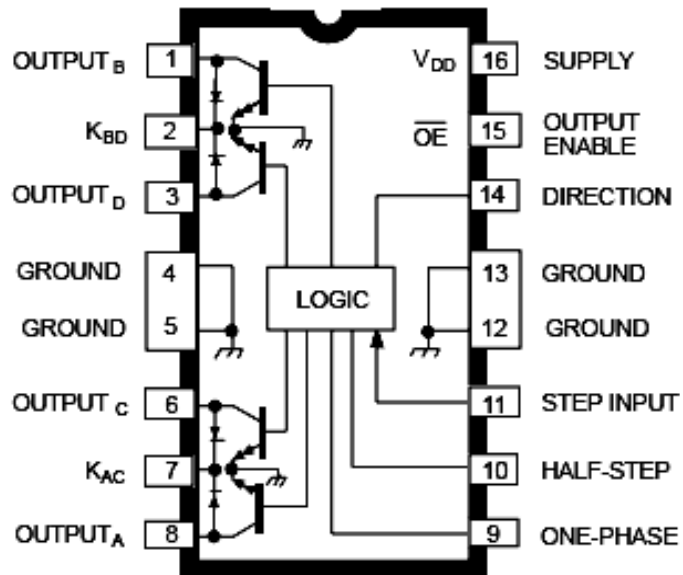


Figura 10. Configuración Darlington para estator de 4 fases unipolar y lógica de secuencia. Integrado A5804 (Allegro® Microsystems).

CAPÍTULO II

Planteamiento del sistema prototipo A.F.E.

1. Hardware:

La caracterización de las Funciones de Red o de las Funciones de Escala Lineales requiere de un sistema capaz de cumplir con dos requerimientos: Poder adquirir los datos eléctricos de la respuesta de la red bajo estudio en forma de Tensión, Corriente, Frecuencia o Período y accionar el parámetro θ de manera controlada, suponiendo que la función bajo estudio posee dependencia de esta variable. La figura 11 muestra el diagrama de bloques del sistema A.F.E.

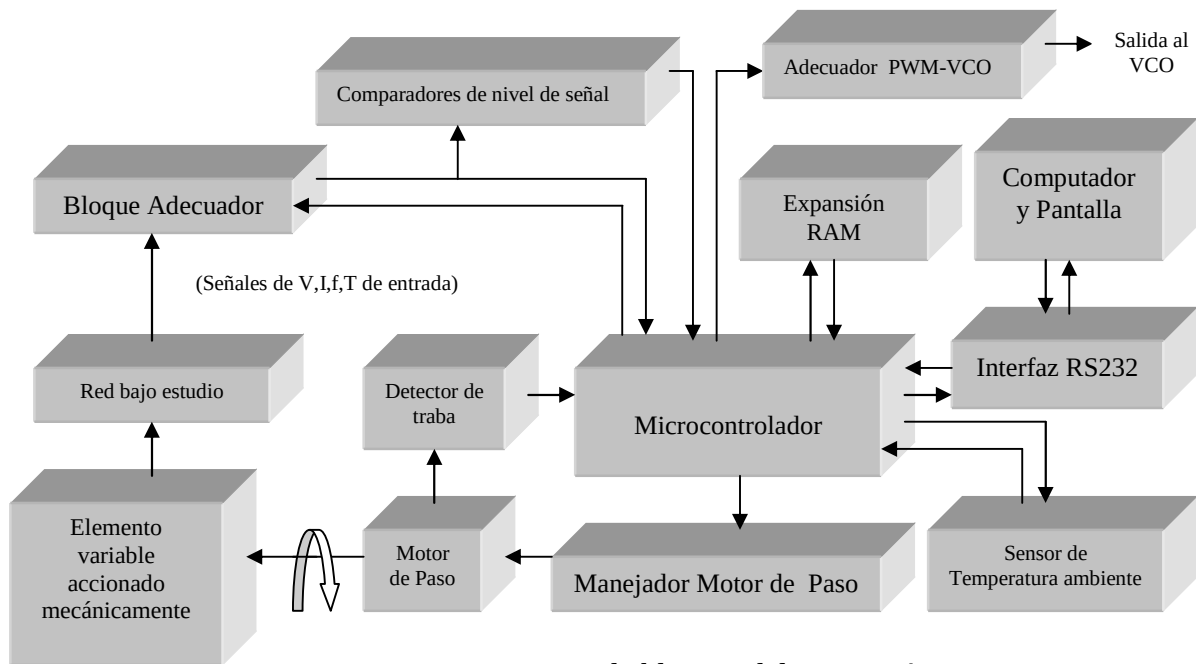


Figura 11. Diagrama de bloques del sistema A.F.E.

La dependencia queda planteada mediante cualquier elemento cuyo valor sea variable mecánicamente (potenciómetros, reóstatos, condensadores y bobinas ajustables) cuyo recorrido mecánico esté acotado por topes en su funcionamiento. Para realizar variaciones del parámetro θ se utilizará un motor de paso, aprovechando las características del mismo en cuanto a la versatilidad de su funcionamiento. El diagrama de bloques del sistema A.F.E. detalla los diferentes módulos con los que se cuenta para realizar el manejo del motor, la transmisión de datos de caracterización

del sistema al computador personal para su posterior representación gráfica, la acción del motor sobre el elemento variable de la red bajo estudio, dicha red y el detector de traba del motor al microcontrolador.

1.1. Bloque Manejador de motor de paso:

Para lograr adecuar el motor de paso al sistema planteado se utilizó un manejador integrado de Motorola®, específicamente el integrado MC3479-D que posee internamente los bloques Secuenciador Lógico y Manejador de Alimentación anteriormente descritos. El bloque Generador de Pulsos será implementado con un microcontrolador. El integrado MC3479-D tiene la posibilidad de manejar paso completo o medio paso, opción de ahorrar energía para cuando el motor esté en una posición fija, direccionamiento de rotación y manejador de impedancia de salida. La figura 12 muestra un diagrama de bloques representativo de la estructura interna y externa del dispositivo usado.

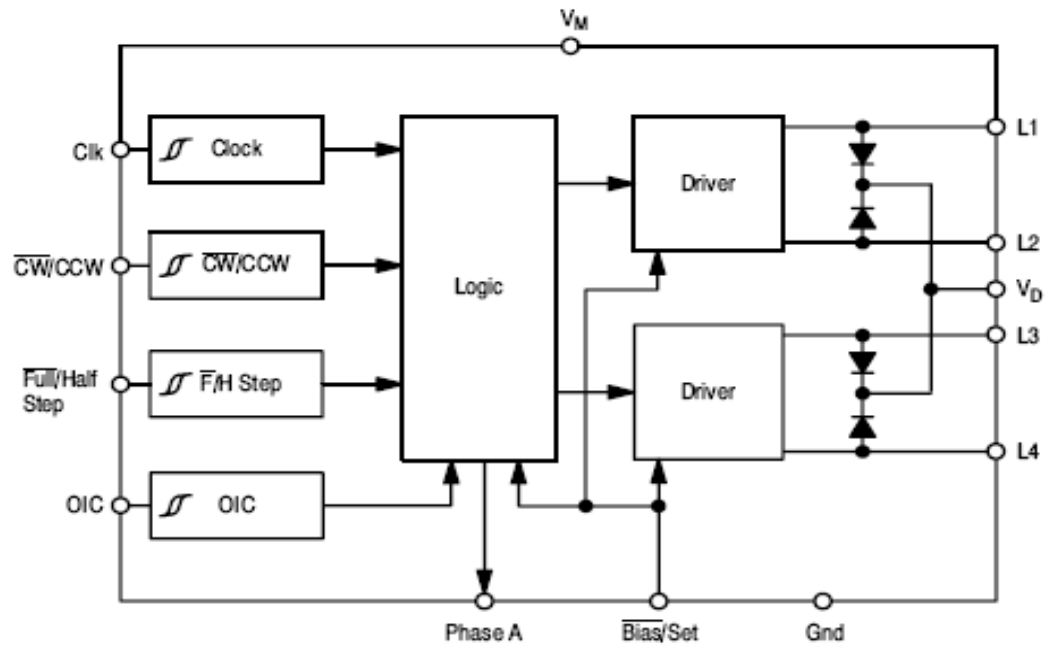


Figura 12. Diagrama de bloques del dispositivo manejador del motor de paso.

La figura 13 muestra el diagrama circuital implementado. Cabe destacar que el sistema manejará al motor de paso con la opción de medio paso.

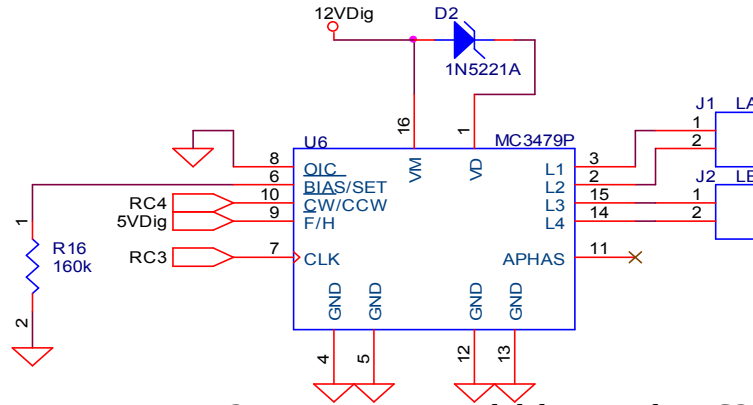


Figura 13. Diagrama circuital del integrado MC3479-D.

1.2. Bloque detector de traba del motor de paso:

Uno de los problemas técnicos a solventar en el proyecto planteado consistió en hacer posible que el recorrido de los elementos variables fuera detectado automáticamente por el sistema, obteniendo así el accionamiento máximo del mismo para realizar la caracterización. El fundamento teórico que avala el poder detectar la traba del motor de paso se deriva del estudio de las ecuaciones características que logran explicar el fenómeno que rige el comportamiento del motor de paso. En la figura 14 se muestra un circuito que modela eléctricamente la malla de excitación de una bobina del motor de paso usando una aproximación de primer orden. Suponiendo que la tensión de entrada es pulsante y aplicando la Transformada de Laplace a la ecuación de malla se obtiene:

$$V(s) + Nr \cdot n \cdot \phi_m \cdot \text{Sen}(Nr \cdot \theta) \cdot s \cdot \Theta(s) = Ia(s) \cdot (R + L \cdot s) + M \cdot s \cdot Ib(s) \quad (7)$$

Donde:

$n \cdot \Phi_m$ = Flujo máximo debido al imán permanente del rotor.

Nr = Número de dientes del rotor.

R = Resistencia en Ohm del devanado de la bobina.

L = Inductancia Propia de la bobina.

M = Inductancia mutua entre las bobinas del estator.

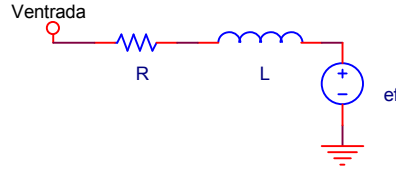


Figura 14. Modelo circuital de las bobinas del estator del motor de paso.

Al despreciar el efecto de la inductancia mutua en la ecuación se encuentra la dependencia de movimiento en función de la corriente:

$$\frac{\Theta(s)}{Ia(s)} = \frac{(R + s \cdot L) - V(s)}{Nr \cdot n \cdot \Phi_m \cdot Sen(Nr \cdot \theta) \cdot s} \quad (8)$$

El estudio del comportamiento de la fuerza contraelectromotriz deriva de la ecuación (3), adicionalmente la solución general en tiempo de la ecuación diferencial que describe la posición del eje del motor al dar un paso es:

$$\theta(t) = A \cdot e^{-\alpha \cdot t} + B \cdot e^{-\beta \cdot t} \cdot Cos(\omega \cdot t - \lambda) \quad (9)$$

Donde:

λ =Distancia angular entre dientes del estator en radianes.

Desarrollando la ecuación (9) en la ecuación (3) queda que la fuerza contraelectromotriz es:

$$ef = K1 \cdot (A \cdot \alpha \cdot e^{-\alpha \cdot t} + B \cdot e^{-\beta \cdot t} \cdot (\beta \cdot Cos(\omega \cdot t - \lambda) + \omega \cdot Sen(\omega \cdot t - \lambda))) \quad (10)$$

Estudiando la ecuación se concluye que para $t=0$ se consigue maximizar el efecto de la fuerza contraelectromotriz (al momento de inyectar un pulso de avance al sistema que está en reposo), quedando la siguiente expresión:

$$ef = K1 \cdot (A \cdot \alpha + B \cdot (\beta \cdot Cos(\lambda) - \omega \cdot Sen(\lambda))) \quad (11)$$

Se observa que la fuerza contraelectromotriz es proporcional a la aceleración angular del eje del motor. Este hecho muestra que es factible implementar un sistema que sea capaz de sensar las variaciones en la señal de corriente de una de las bobinas del motor para poder determinar el estado de movimiento del rotor y por consiguiente del tope mecánico.

Analizando la ecuación de malla (11) se observa que a medida que la velocidad angular del eje del motor aumente, es decir, a medida que la fuerza contraelectromotriz crezca, la corriente del circuito disminuye y viceversa. Por este fenómeno se decidió trabajar con un tren de pulsos provenientes del Generador de pulsos de aproximadamente 900 Hz haciendo más simple la detección de estado de traba en el eje del motor de paso (incrementando así el efecto de la tensión contraelectromotriz en la respuesta obtenida). Adicionalmente, se decidió integrar en el tiempo la señal de corriente de una de las bobinas del motor buscando calcular el promedio de dicha corriente entre pulsos (o pasos). Es de hacer notar que el área de dicha curva sufrirá cambios en su valor por el estado mecánico del eje del motor (Comparando los valores entre el giro a 900 r.p.m. y 0 r.p.m.), específicamente en las componentes de frecuencia más alta; por tanto se deberá además plantear una ventana de integración en el tiempo de tal manera de lograr integrar únicamente dichas componentes en la respuesta transitoria del circuito. Se determinó de manera experimental que la duración de la ventana de integración adecuada al sistema es de $63,73\mu s$.

El módulo detector de traba del motor está compuesto por un amplificador diferencial (U23A y R33) procesando la señal de corriente demandada por una de las bobinas del estator a través de la resistencia R41, esta señal es inyectada, en un intervalo de tiempo (ventana de integración controlada por el terminal RB2 en nivel bajo, proveniente del microcontrolador) a la entrada del integrador (U23B, C15, R38) mediante un interruptor analógico (S1), luego es procesada por un comparador a lazo cerrado (U13A, R34, R35, R36, R42). El comparador tiene además un condensador (C19) que retiene momentáneamente el nivel previo que tuvo la señal integrada, logrando en todo momento comparar el valor previo de salida del integrador con el valor actual de dicha salida. Al ocurrir cambios en la señal de corriente de la bobina del estator del motor de paso, ocurrirán cambios en la salida del comparador, estos cambios serán contabilizados por el microcontrolador para tomar una decisión en cuanto al sentido de rotación del motor o en cuanto a la detención del mismo

(dependiendo del sentido de giro que se esté efectuando). El circuito descrito se muestra en la figura 15.

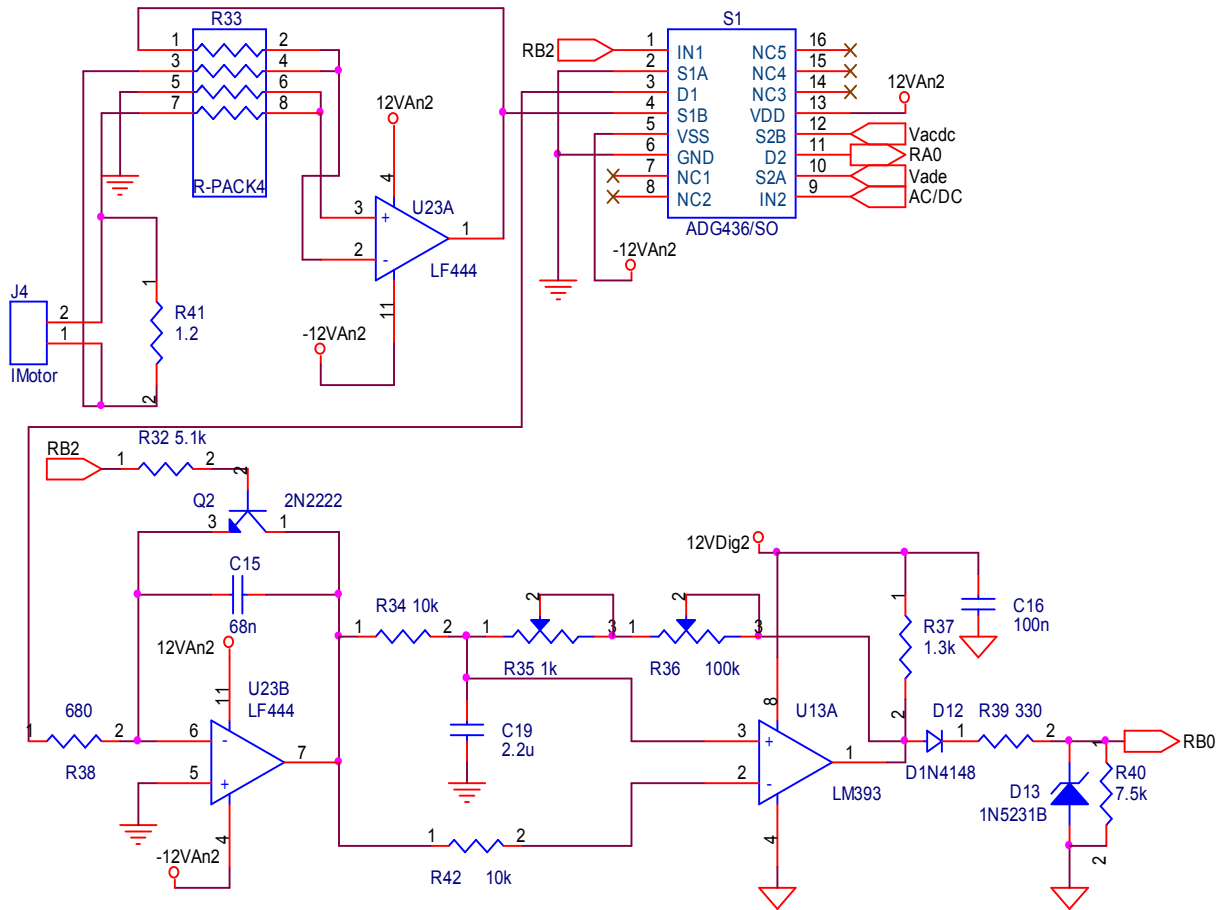


Figura 15. Módulo detector de Traba de motor.

Adicionalmente se tiene que el integrador será puesto nuevamente a cero para volver a calcular el área bajo la curva de la señal de corriente en cada paso dado, gracias a la acción del transistor Q2 (con RB2 en nivel alto). Los potenciómetros R35 y R36 ajustan el ancho de la ventana del comparador permitiendo graduar el nivel de sensibilidad del mismo para diferentes cargas mecánicas acopladas al eje del motor de paso que varíen el momento de inercia y el coeficiente de amortiguamiento del sistema mecánico. La salida del comparador está conectada al terminal de interrupción externa del microcontrolador (terminal RB0), dicha señal es un tren de pulsos que anuncian el estado de traba del eje.

1.3. Bloque adecuador de la señal de entrada:

Para poder adecuar las señales de tensión y corriente que se van a medir se utilizó el bloque mostrado en la figura 16.

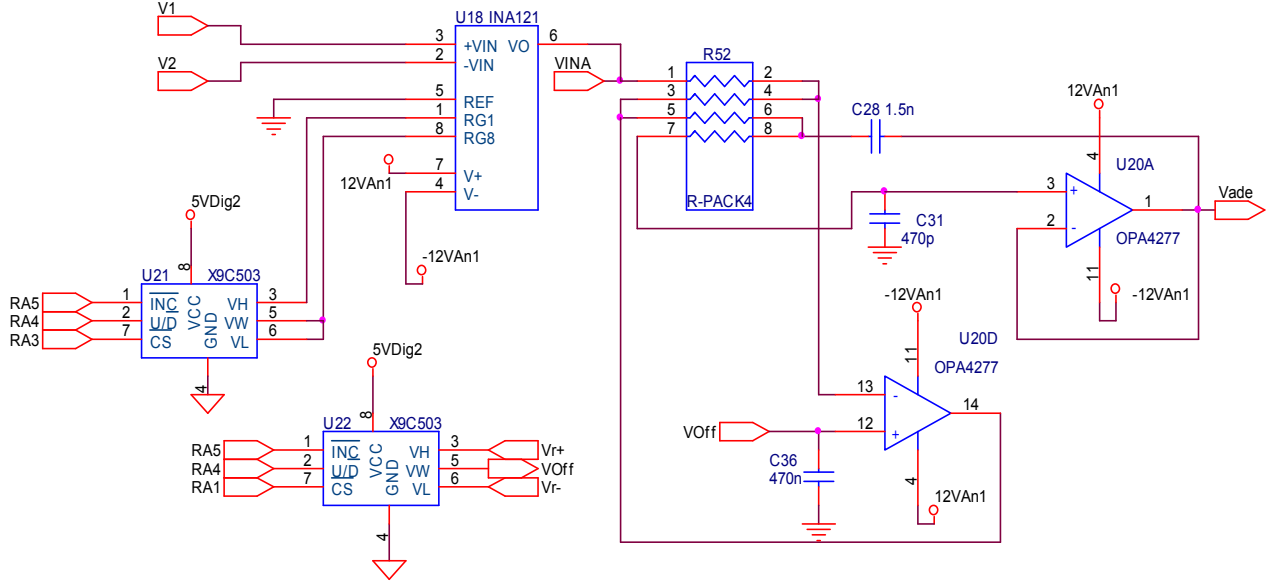


Figura 16. Bloque adecuador de la señal a caracterizar.

Este bloque está constituido por un amplificador instrumental (INA121) y un amplificador operacional de alta precisión (OPA4277) en paquete cuádruple, ambos de la casa Texas Instruments®. La ganancia del amplificador instrumental está controlada por un potenciómetro digital X9C503 (U21) de la casa Intersil®, el cual permite que la misma sea modificada mediante software. La figura 17 muestra el diagrama simplificado del amplificador instrumental. La figura 18 muestra un diagrama de bloques general de la estructura del potenciómetro digital.

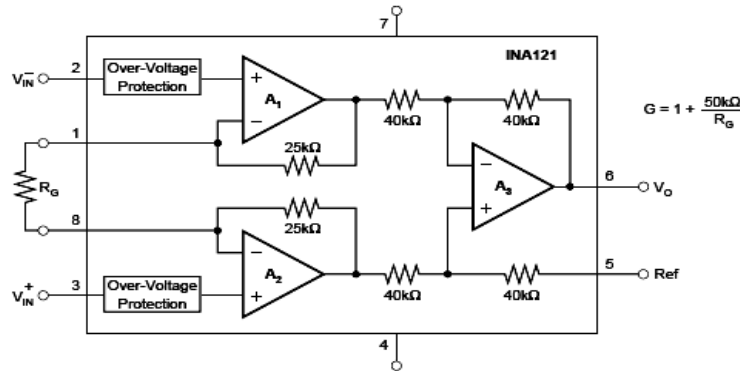


Figura 17. Bloque del amplificador instrumental INA121.

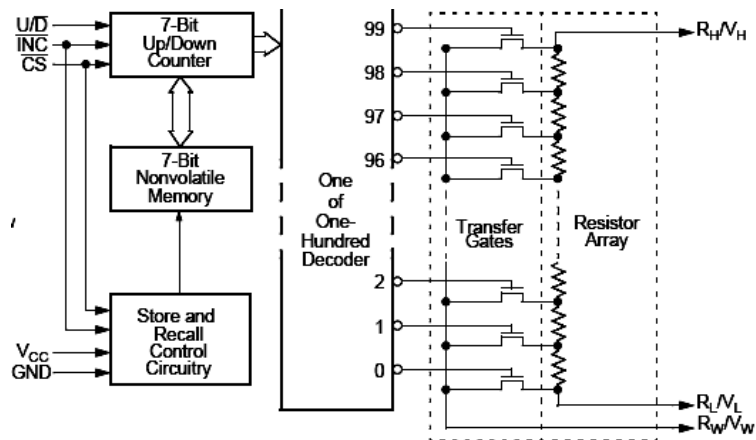


Figura 18. Diagrama de bloques del potenciómetro digital X9C503.

El amplificador OPA4277 (U20A) está conectado formando un filtro en configuración Sallen Key de segundo orden con frecuencia de corte de 10kHz pasa bajo máximo plano. Este filtro fue incluido para evitar el efecto de Aliassing en la conversión de los datos y a su vez limitar el ancho de banda de las señales a estudiar debido a que, a lo sumo, la frecuencia más alta de la señal de entrada a estudiar será de 10kHz. En el bloque mostrado también se observa el amplificador sumador (U20D) que es el encargado de mezclar la señal proveniente del amplificador Instrumental con un nivel de pedestal D.C. que adecue los niveles de tensión de la señal que será procesada por el microprocesador a través del convertidor Analógico Digital cuyo rango de trabajo va desde 0 a 5V. El nivel de pedestal D.C. es a su vez extraído de otro potenciómetro digital (U22) el cual está alimentado con tensión bipolar simétrica de $\pm 2,5V$. Para adquirir datos concernientes al período de una señal, el bloque adecuador incluye un detector de cruce por cero que bloquea el posible nivel D.C. de la señal proveniente del amplificador instrumental. El detector dará pulsos de salida en alto al sentir que el nivel de la señal está por encima de 0V. La salida del bloque detector es conectada al módulo de captura del microcontrolador con el fin de contabilizar el período de la señal estudiada. Este bloque detector de cruce por cero está compuesto por un comparador de lazo cerrado con un buffer que permite la implementación de un filtro pasa altos de primer orden (RC pasivo) con el polo ubicado aproximadamente en 1Hz. La figura 19 muestra el diagrama electrónico.

1.5. Bloque Interfaz de comunicación RS-232:

La interfaz utilizada para lograr la conexión entre el módulo USART⁵ del microcontrolador y el computador fue implementada con el integrado MAX232A de la casa Maxim®. Este integrado es un convertidor del estándar de comunicación RS232 a TTL o CMOS (y viceversa) que adecua los niveles de tensión en la información de transmisión y la de recepción, tanto del microcontrolador (TTL) como del computador (RS232). La figura 21 muestra la configuración circuital implementada.

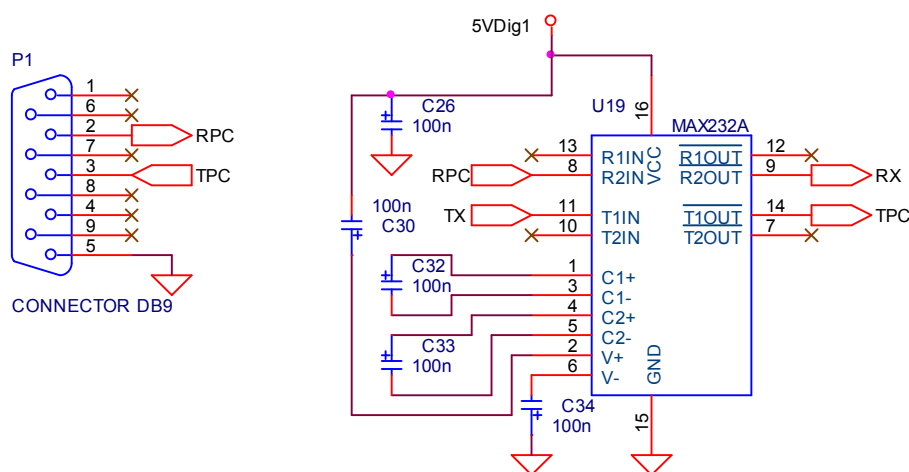


Figura 21. Diagrama circuital de la interfaz RS232.

1.6. Microcontrolador:

El núcleo del sistema fue implementado con un microcontrolador de la casa Microchip® específicamente de la familia 16Flash, definida para aplicaciones de rango medio. El microcontrolador utilizado es el PIC16F877A, de 8 bits, 8kBytes de memoria de programa, 368 bytes de memoria RAM⁶ interna, con oscilador externo de 16 MHz el cual permitirá que un ciclo de instrucción sea ejecutado en 250ns. El microcontrolador empleado posee, de manera integrada, un módulo de conversión analógica digital de 10 bits, un módulo de captura o comparación para realizar

⁵ Por sus siglas en inglés Universal Synchronus Asynchronous Recieve Transmit (Receptor-transmisor universal sincrónico y asincrónico).

6 Por sus siglas en inglés Random Access Memory. (Memoria de acceso aleatorio).

cuentas de intervalos temporizados, 15 interrupciones, 1 módulo PWM⁷, módulo de comunicaciones USART, 5 puertos de entrada salida (3 puertos de 8 bits, 1 puerto de 5 bits y 1 puerto de 3 bits). El diagrama de los terminales del microcontrolador se muestra en la figura 22.

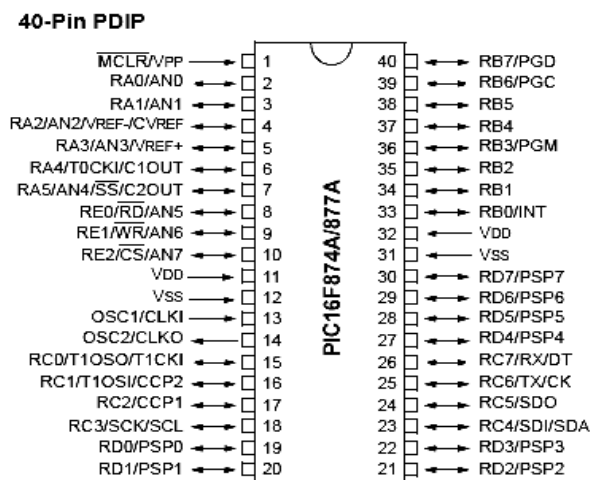


Figura 22. Diagrama de terminales del microcontrolador.

Las tareas asignadas al microcontrolador por parte del sistema son las siguientes:

- Proveer de las señales de control al Manejador integrado del motor de paso, específicamente las señales de reloj (Clock), sentido de rotación (CW*/CCW), y ahorro de energía (Bias*/Set).
- Proveer de las señales de control a los potenciómetros digitales controladores de ganancia y pedestal en el bloque Adecuador de Señal, específicamente las señales de reloj (Clock), sentido de incremento (Up/Down*) y selección del dispositivo (CS*).
- Proveer de las señales adecuadas que logren activar y desactivar el interruptor analógico para detectar el estado de traba, así como

⁷ Por sus siglas en inglés Pulse Width Modulation (Modulación por ancho de pulso).

* Para denotar que el control es activo con nivel de tensión bajo (TTL).

también el poder reiniciar dicho integrador en cada paso dado al motor (para el modo de búsqueda de tope mecánico).

- Establecer la comunicación entre el sistema y el computador personal a través del puerto USART de comunicación serial.
- Realizar la conversión analógica digital de las señales de tensión o corriente provenientes del bloque adecuador así como también obtener el período de una señal con el fin de obtener los datos de la caracterización de la función estudiada.
- Sincronizar convenientemente el movimiento del motor de paso con la adquisición de datos de la variable eléctrica a caracterizar.
- Conseguir, automáticamente, el tope mecánico planteado por el elemento variable que forma parte de la red eléctrica bajo estudio.
- Adecuar automáticamente el nivel de ganancia y de tensión de pedestal a los rangos del convertidor analógico digital (entre 0 y 5V) por la interpretación del código proveniente del bloque de Comparadores de Nivel de Señal.
- Proveer de una señal al exterior, proveniente del bloque PWM, que permita tener control sobre un módulo oscilador controlado por tensión para permitir al sistema la realización de barridos en frecuencia, específicamente, el módulo oscilador es el incluido en el generador de funciones marca Hewlett Packard modelo 3311A.
- Por requerimientos en el procesamiento de datos del proyecto realizado, se agregó una expansión de memoria RAM a 64kBytes. El microcontrolador está encargado de llenar y vaciar el contenido de dicha expansión de memoria y de proporcionar todas las señales de control pertinentes, logrando manejar convenientemente dicho bloque de expansión.

- Proveer las señales de control, recepción y transmisión de datos para el manejo de un sensor de temperatura integrado, lo cual permite la obtención de la temperatura ambiente al realizar las caracterizaciones, específicamente las señales de selección de dispositivo (CS*), pulsos de reloj (SCK*) y terminal de entrada/salida de datos (SO/I).

1.7. Bloque comparador de nivel de señal:

La figura 23 muestra el diagrama circuital.

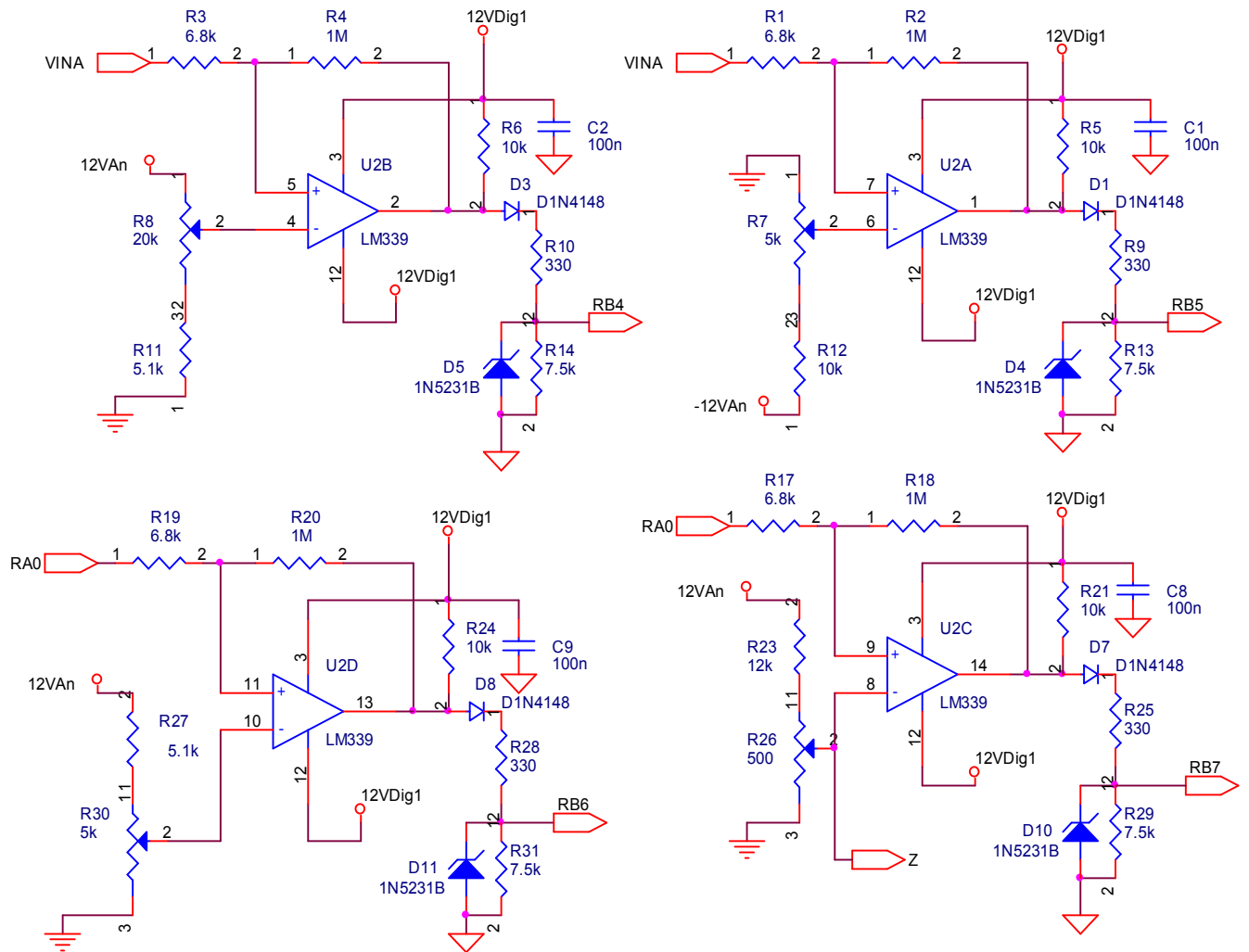


Figura 23. Diagrama del bloque de comparación de la señal.

Este bloque está constituido por 4 comparadores (LM339) a lazo cerrado que detectan los niveles que alcanza la señal, a la salida del bloque adecuador

anteriormente mencionado. Esta información es interpretada por el microcontrolador para ajustar los niveles de ganancia y tensión de pedestal para usar convenientemente el rango del convertidor analógico digital del microcontrolador. Los comparadores poseen ajuste para lograr seleccionar las tensiones de encendido y apagado, pudiendo variar el punto de comparación. En los comparadores, se ajustaron los niveles de tensión de 5V (U2C), 0V (U2D), -2,5V (U2A) y 2,5V (U2B). Los 2 primeros evitan que sean sobrepasados los niveles de tensión permisibles al convertidor analógico digital del microcontrolador. Los 2 últimos evitan que la señal proveniente del amplificador diferencial no sobrepase los niveles 2,5V y -2,5V logrando así que el sistema tenga forma de corregir el nivel D.C. por intermedio del filtro sumador para entrar en el rango del convertidor analógico digital.

1.8. Bloque selector de señal Tensión-Corriente:

La figura 24 muestra el diagrama circuital del bloque.

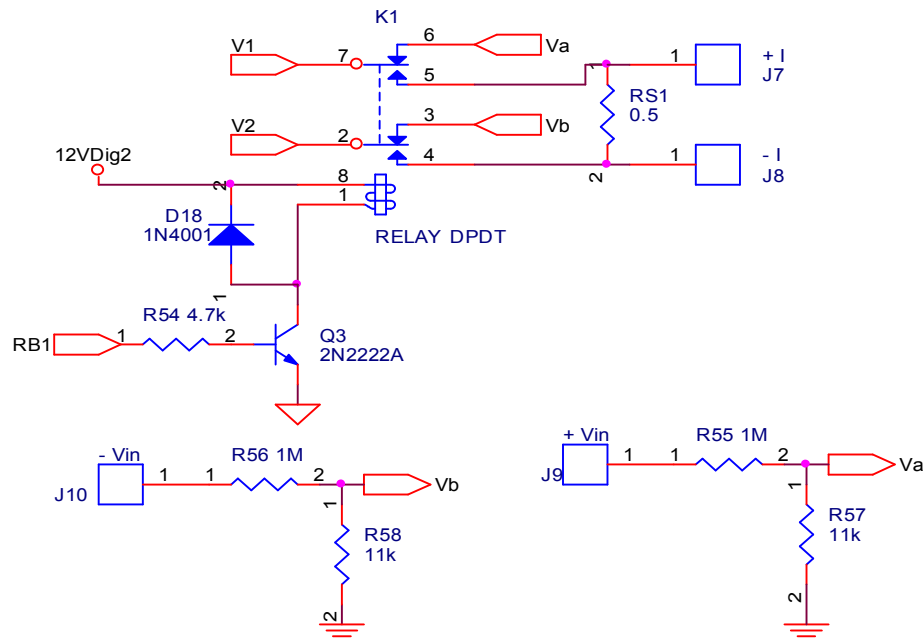


Figura 24. Bloque selector de medición V-I.

El bloque selector de señal de tensión o corriente está implementado con un relé que conmuta las entradas del amplificador instrumental entre los atenuadores de tensión a la entrada (R55, R56, R57, R58) y la resistencia RS1 que es el elemento sensor de corriente implementado. El microcontrolador es el encargado de activar el

relé, logrando activar el modo de medición requerido por parte del usuario. Adicionalmente se observa que en el modo de tensión, el sistema atenúa la señal de entrada en un factor de 0,01 aproximadamente.

1.9. Bloque Referencias de tensión:

Las referencias de tensión del circuito se implementaron con el integrado REF3025 de la casa Texas Instruments® de 2,5V nominales a su salida y con amplificadores operacionales implementados como filtros pasa bajos, -2,5V para el pedestal D.C. del amplificador sumador del bloque adecuador de señal (U20C) y 5V para la referencia de tensión del convertidor analógico digital del microcontrolador (U20B). La tensión de 2,5V se extrae directamente del integrado de referencia antes mencionado. La figura 25 muestra el diagrama circuital del bloque.

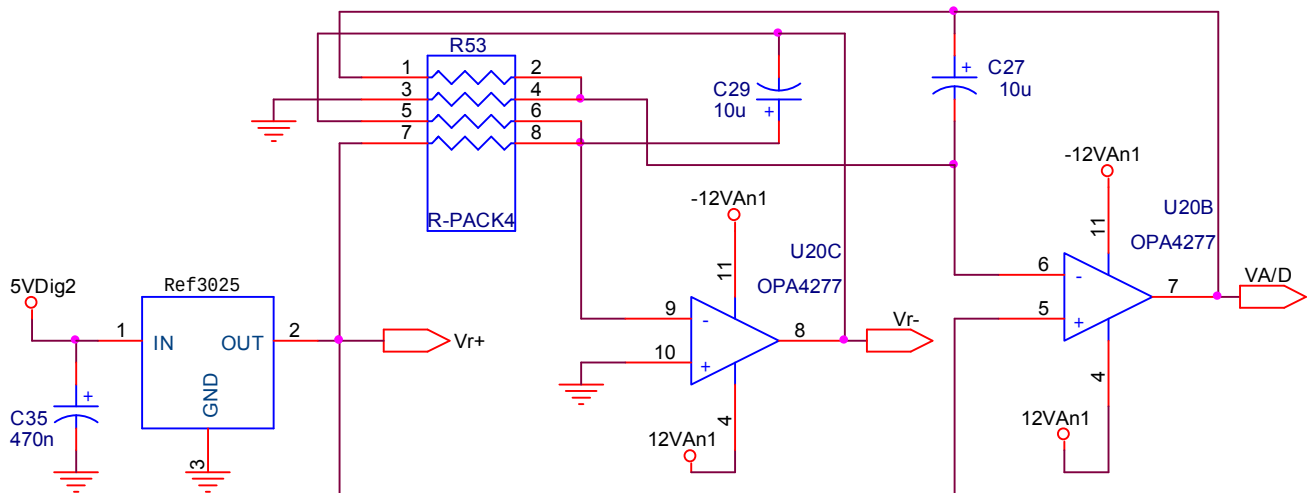


Figura 25. Bloque proporcionador de las referencias de tensión.

1.10. Bloque convertidor PWM-VCO:

Este bloque está constituido por un amplificador inversor y filtro pasa bajo que se encarga de promediar el nivel de señal del módulo PWM a un pedestal D.C. que permitirá manejar el oscilador controlado por tensión del generador de funciones HP3311A. Por requerimiento del generador, la señal de entrada al VCO debe encontrarse en el rango de 0V a -10V. El módulo PWM del microcontrolador se configuró para tener una frecuencia de aproximadamente 1kHz, por tanto, el integrador se diseñó para fijar el polo de baja del mismo a más de 100 veces por

debajo de 1kHz. Este polo se fijó en 8Hz aproximadamente. Por su parte, el microcontrolador recibirá por parte del usuario, la configuración deseada de nivel de señal, a través del módulo de comunicación serial, permitiendo variar el Duty Cycle de salida del bloque PWM y por tanto del nivel D.C. de salida. En la figura 26 se muestra el circuito convertidor antes mencionado.

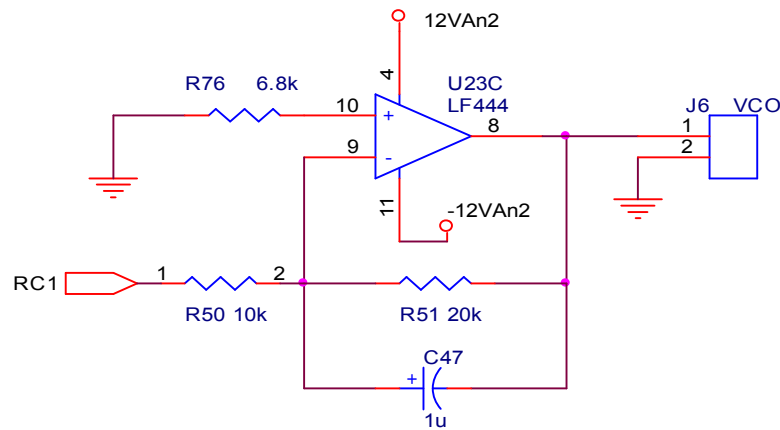


Figura 26. Diagrama circuital del adecuador PWM-VCO.

1.11. Bloque extensión de memoria RAM:

El sistema planteado necesitó de una extensión de la capacidad de memoria con el fin de poder guardar, momentáneamente, los datos de la adquisición, provenientes de la caracterización. Al mismo tiempo, se diseñó el sistema tal que fuera capaz de seguir ejecutando las acciones de índole mecánica logrando disminuir el tiempo de realización de la caracterización de la Función de Red o de Escala. Al culminar la medición en todos los puntos mecánicos de accionamiento, el microcontrolador vaciará la información contenida en el bloque de memoria RAM y la transmitirá a través del módulo USART hacia el computador, en donde se realizarán las gráficas contendientes a representar la Función de Escala que ha sido caracterizada. Para implementar esta extensión de memoria se utilizó el puerto D del microcontrolador, 2 bloques RAM 61256 de 32kBytes cada uno, 2 Latch's 74LS373 y una compuerta NAND 74LS00; los Latch's permiten el direccionamiento de las memorias RAM (dirección de 15 bits). Con la compuerta NAND se diseñó un circuito combinacional que se encarga de seleccionar entre los 2 bloques de memoria

RAM para ejecutar la lectura o escritura de las mismas. La figura 27 muestra el diagrama circuital que contiene al microcontrolador, los bloques RAM y los Latch's.

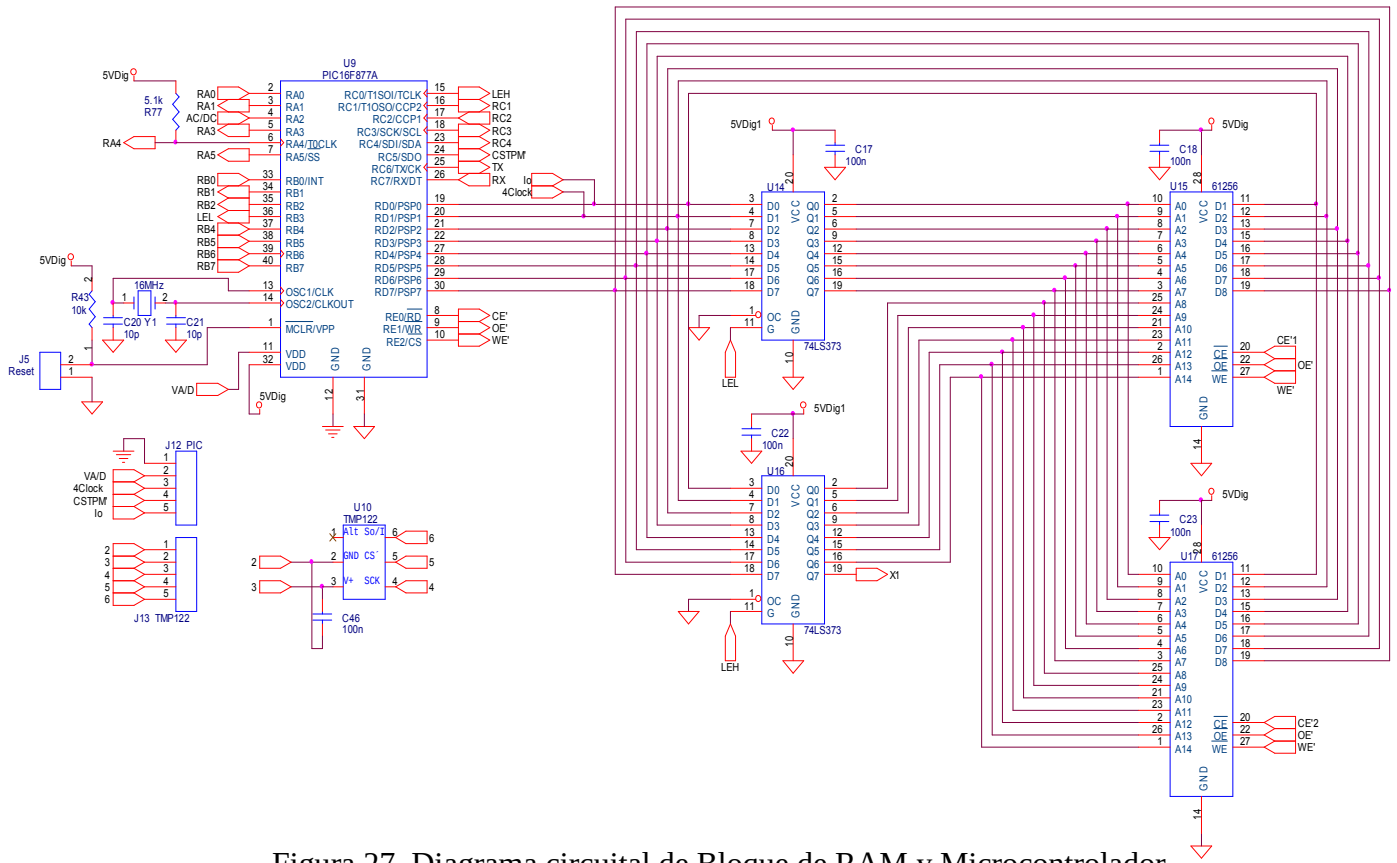


Figura 27. Diagrama circuital de Bloque de RAM y Microcontrolador.

El puerto D del microcontrolador se configuró de tal manera que pudiera servir como bus de direcciones y bus de datos de 8 bits. La figura 28 muestra el circuito selector de memoria RAM.

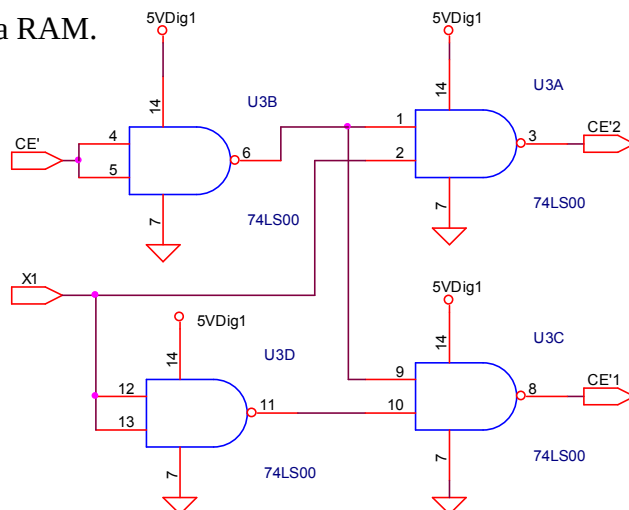


Figura 28. Circuito combinacional para el manejo de la expansión RAM.

1.12. Bloque de alimentación:

Las alimentaciones necesarias del prototipo son específicamente de -12V, 12V, 5V. Sin embargo, para desarrollar adecuadamente al circuito se separaron dichas alimentaciones en digitales y analógicas, con el fin de evitar que el ruido producido por las conmutaciones de los bloques digitales afecten a los bloques analógicos. Se separaron las referencias analógicas de las digitales y se implementaron filtros pasa bajos en las líneas de alimentación. Los integrados utilizados fueron LM7805C(5V), LM7912C(-12V), LM7812C(12V) y SV3122V(12V). La figura 29 muestra los circuitos reguladores lineales de tensión.

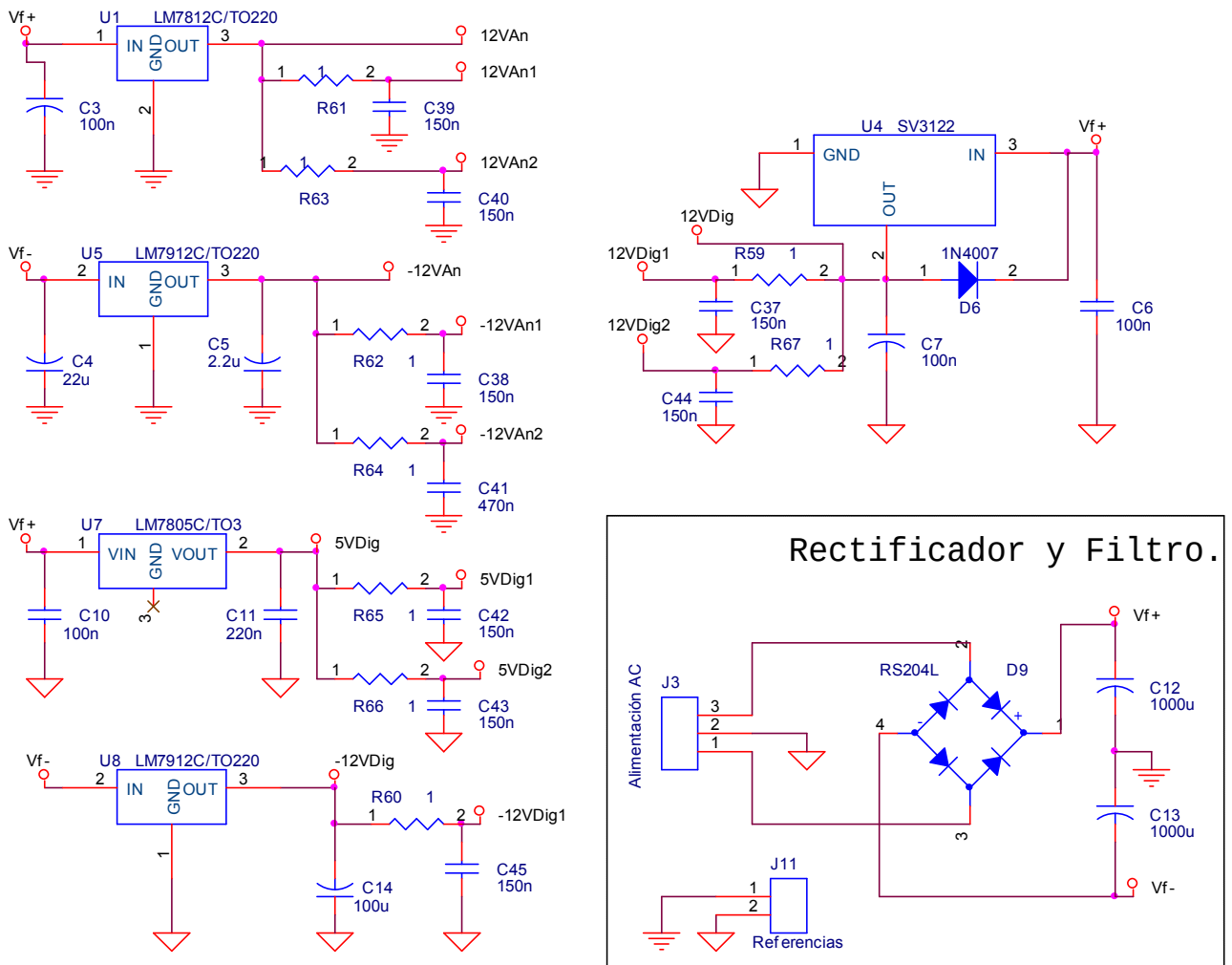


Figura 29. Bloques de alimentación de 5V, 12V y -12V.

1.13. Bloque convertidor de tensión A.C.-D.C.:

Este bloque está constituido por un rectificador de precisión y un integrador que promedia el área bajo la curva de la señal senoidal rectificada, este promedio es contabilizado por el convertidor analógico digital del microcontrolador, logrando así medir la amplitud de la señal de entrada de manera indirecta. Cabe destacar que este método supone que la forma de onda a medir es senoidal ya que el valor digital arrojado por el convertidor será multiplicado por una constante “K” para obtener el valor pico de la señal estudiada, a saber:

$$V_{promedio} = \frac{2 \cdot V_p}{\pi} \Rightarrow K = \frac{\pi}{2} \quad (12)$$

El integrador implementado posee su polo en la frecuencia de 1Hz aproximadamente, por lo tanto, la frecuencia de la señal a caracterizar debe ser mayor a los 10Hz. El tiempo de puesta de la señal promediada en el condensador tiene un valor de 1s aproximadamente, este tiempo deberá ser esperado después de haber accionado mecánicamente el motor de paso para tomar luego la muestra con el convertidor analógico digital. La figura 30 presenta el circuito implementado.

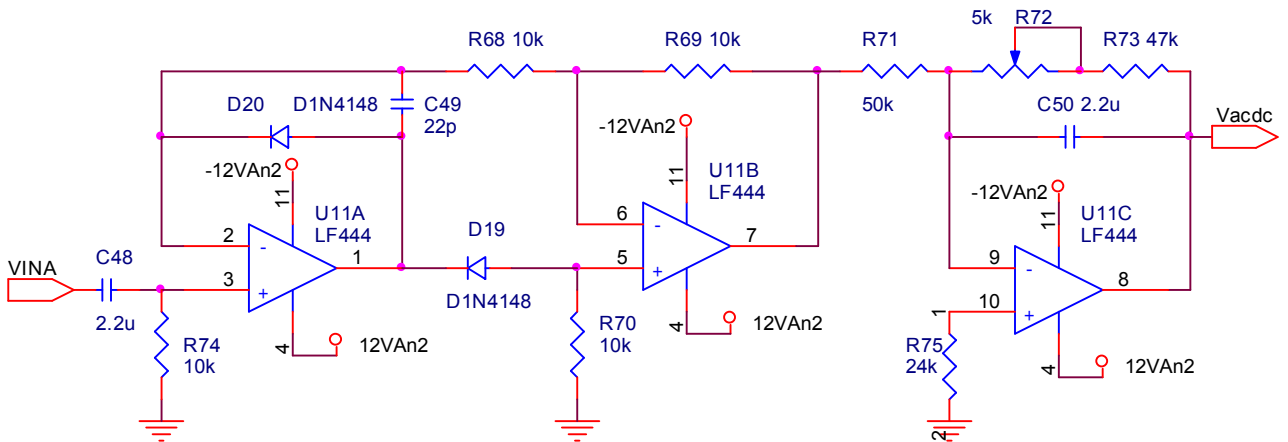


Figura 30. Bloque convertidor A.C.-D.C.

2. Software:

El software fue desarrollado utilizando 2 lenguajes de programación; para la programación del microcontrolador se utilizó el compilador HT-PIC®, de la casa Hi-Tech Software®, el cual permite el desarrollo del programa sobre una plataforma basada en lenguaje ANSI C; la programación del computador personal fue desarrollada con Visual BASIC® 6.0 el cual permite, entre otras cosas, la implementación de un entorno gráfico de ventanas dinámicas y la programación orientada a eventos. Para realizar el trazado de las gráficas que son el resultado de la caracterización del sistema se utilizó la aplicación Microsoft®, Excel®.

2.1. Programación del microcontrolador:

El microcontrolador tiene a su cargo varias operaciones de ejecución en todos los procesos inherentes a la caracterización de las variables eléctricas estudiadas. El microcontrolador ejecuta acciones de configuración a los módulos internos (CCPx⁸, USART, A/D) además de brindarle al hardware implementado todas las condiciones o estados necesarios al iniciar el proceso de caracterización (Potenciómetros digitales, Motor de paso, Integradores, Memoria RAM). A continuación se presenta un resumen de la funcionalidad configurada para cada módulo interno y su utilidad dentro del proyecto.

2.1.1. Módulo USART:

El módulo de comunicación serial del microcontrolador fue configurado para exhibir las siguientes características:

- Velocidad de 9600 Baudios, esta velocidad fue escogida por ser la que menos porcentaje de error presenta en la hoja de datos suministrada por el fabricante del dispositivo.
- 8 bits de datos, sin bit de paridad, tanto para la recepción como para la transmisión.
- Comunicación asíncrona, full duplex.

⁸ Por sus siglas en inglés Capture Compare PWM (Módulo de captura, comparación o modulación en ancho de pulso).

Una de las principales tareas asignadas al módulo USART, es la de recibir los comandos enviados desde el computador personal al microcontrolador, con el fin de que éste último los interprete y logre ejecutar la función asociada a dicha transmisión. Este protocolo de peticiones fue establecido con el fin de eliminar el requerimiento de incluir un periférico de interfaz usuario-prototipo como por ejemplo un teclado. Todos los comandos de petición son enviados por el computador personal en forma de datos binarios, transmitidos a través de la interfaz RS232 e interpretados por el microprocesador para asociarlos a una función preestablecida en el programa, logrando así la ejecución de una acción por parte del sistema prototipo. La interfaz implementada en el sistema se le conoce como interfaz GUI⁹, la misma permite hacer uso de las ventajas en cuanto a la reducción de hardware para el manejo del sistema por parte del usuario, permite que dicha interfaz sea planteada de forma amigable y rápidamente entendible para el operador y, por último, descansa su funcionamiento sobre plataformas de sistemas operativos que trabajen con la filosofía de orientación a eventos. En la tabla 2 se muestran los diferentes comandos enviados por el computador personal y la interpretación que el microcontrolador da a éstos, estableciendo una acción de control sobre el sistema A.F.E. a través del programa principal del computador. Así se consigue eliminar la necesidad de implementar periféricos externos al prototipo A.F.E. como por ejemplo, un teclado.

Otra de las tareas asociadas al módulo USART del microcontrolador es la de enviar los resultados recogidos de las pruebas realizadas así como también el estado general del sistema. Se implementó un protocolo de comunicación, diseñando paquetes de datos arreglados en forma de tramas, así, se aporta información al usuario/operador del tipo de medición al cual pertenecen los datos enviados, los datos de dicha medición y el valor de accionamiento para el momento de la adquisición. Las tablas 3 y 4 muestran la trama de datos y su significado. Estos datos son el resultado de una conversión A/D (para tensión) y una adquisición de período de una señal respectivamente.

⁹ Por su siglas en inglés Graphical Unit Interface. (Unidad de interfaz gráfica).

Tabla 2. Comandos de petición de acción PC $\rightarrow$ μ P.

Número hexadecimal enviado al microcontrolador	Acción solicitada por el usuario al sistema a través de botones virtuales	Función ejecutada por el microcontrolador
64h + 61h	Cancelar acción actual.	<i>cancela()</i>
74h + 69h	Buscar tope mecánico Izquierdo.	<i>topmot()</i>
74h + 64h	Buscar tope mecánico Derecho.	<i>topmot()</i>
70h + xxh	“xx” Pasos a la Izquierda.	<i>pulmot()</i>
80h + xxh	“xx” Pasos a la Derecha.	<i>pulmot()</i>
76h + xxh	“xx” muestras de datos tomados en el mismo accionamiento en Conversión A/D.	<i>libmot()</i>
66h + 02h	Medición de período de señal.	<i>libmot2()</i>
6Fh + xxh	Aumentar o disminuir “xx” pasos del nivel de pedestal D.C.	<i>incpot()</i>
67h + xxh	Aumentar o disminuir la ganancia del sistema.	<i>incpot()</i>
75h + 70h	Potenciómetros digitales. Recuperar la última posición.	<i>ultipots()</i>
75h + 69h	Potenciómetros digitales. Recuperar la posición inicial.	<i>inicpots()</i>
72h + 6Dh	Cambio de ajustes de potenciómetros a modo manual.	Cambio de puerto B inhabilitado.
72h + 61h	Cambio de ajustes de potenciómetros a modo automático.	Cambio de puerto B habilitado.
73h + 61h	Preguntar por el estado del sistema.	<i>configt()</i> y <i>midetmp()</i>
7Ah + 36h	Cancelar inmediatamente la acción que se está ejecutando.	<i>recibiri()</i>
7Ah + 37h	Puesta a cero del accionamiento del eje del motor.	<i>recibiri()</i>
2Ch	Trama recibida sin error, continuar enviando.	<i>recibiri()</i>
2Dh	Trama recibida con error, repetir envío.	<i>recibiri()</i>
77h + xxh	Encendido y variación del módulo PWM.	<i>recibiri()</i>
AAh + 01h	Medición con módulo A/D en un punto fijo.	<i>mpad()</i>
DDh + 01h	Medición de período en un punto fijo.	<i>mfp()</i>
73h + 70h / 73h + 07h	Pausa o reanudar movimiento.	Durante una caracterización.
34h + xxh	Modificación de inicio de caracterización.	# Pasos que el motor dará.
73 h + 67 h / 62h + 67h	Puesta de ganancia al máximo /mínimo posible.	Sólo en modo automático
CC h + 01 h / CAh + 01h	Conecta/ Desconecta relé para medir corriente.	Entrada modo Tensión/Corriente
64h + 63h / 61h + 63h	Selector modo medición A.C./D.C. (Tensión o corriente).	Interruptor analógico.
6Dh + 6Dh	Petición de reporte de modo de medición A.C.. o D.C.	Respuesta del μ Controlador.

Tabla 3. Trama de datos provenientes de la conversión A/D, del μ C al PC.

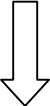
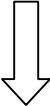
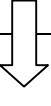
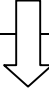
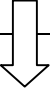
	Cabecera (1er Byte)	Datos adquiridos (2·n Bytes siguientes)	Valor Ganancia y pedestal	Paso actual (Últimos 2 Bytes)	Significado físico de la trama
Datos A/D	FAh	xxxxh...xxxxh	xxhxxh	xxhxxh	
Ejemplo	 FAh	 02FFh02FBh02FCh	 5Dh49h	 11hA3h	3 datos de conversión A/D (3,745V;3,725V;3,730V). Ganancia = 93% Pedestal = 73% Accionamiento = 4515 pasos para llegar a cero mecánico.

Tabla 4. Trama de datos del período de señal y su significado, del μ C al PC.

	Cabecera (1er Byte)	Datos adquiridos (4 Bytes siguientes en orden invertido)	Accionamiento mecánico (Últimos 2 Bytes)	Significado físico de la trama
Datos de período	FDh	xxhxxh...xxh	xxhxxh	
Ejemplo	 FDh	 55h77h01h00h LSB ← MSB	 00hC5h	1 dato de período de señal (96085 cuentas·250ns/cuenta=24,02ms). Accionamiento de 197 pasos para llegar a cero mecánico.

El microcontrolador también fue programado para enviar avisos de acontecimientos, dado que ocurrió un evento imprevisto. Para lograr que el computador personal diferencie dichos avisos, se provee a la trama enviada de una cabecera de identificación y del estado general del sistema. La tabla 5 muestra las tramas pertinentes a los avisos emergentes.

Tabla 5. Trama de datos de avisos emergentes y su significado, del μP al PC.

Trama enviada por el microcontrolador	Significado de la trama de aviso
	Sobrepasamiento del tiempo de espera (más de 100s) con período de señal en un accionamiento.
	Culminación de transmisión de datos de caracterización.
	Modo de ajuste automático de nivel no posible, Debe usar un atenuador para hacer la medición.
	En el modo de ajuste manual de nivel, la señal de entrada está por encima de 5V.
	En el modo de ajuste manual de nivel, la señal de entrada está por debajo de 0V.
	Trama de estado actual. 1: Cabecera; 2: acción potenciómetro pedestal; 3: acción potenciómetro ganancia; 4: Valor del puerto B; 5: Pasos faltantes a la izquierda para llegar a tope mecánico; 6: Pasos dados a la derecha; 7: modo de ajuste (automático o manual); 8: Medición de temperatura ambiente.
	Datos de conversión A/D en punto fijo, para realizar ajustes en dicho punto. 1: Cabecera; 2: 10 Bytes de datos de conversión; 3: Ganancia; 4: Pedestal; 5: paso mecánico actual.
	Datos de período de señal en punto fijo, para realizar ajustes en dicho punto. 1: Cabecera; 2: Período de señal LSB $\rightarrow$ MSB; 3: paso mecánico actual.

2.1.2. Módulo convertidor analógico digital A/D:

El convertidor analógico digital incluido en el microcontrolador fue configurado para realizar las conversiones a 10 bits, con una velocidad máxima de $32 \cdot 12 \cdot (62,5\text{ns}) = 24\mu\text{s}$ entre muestras continuas. El escalón de cuantización del convertidor es de $\frac{V_{ref}}{2^{10}} = \frac{5V}{1024} \approx 4,88\text{mV}$. El canal de conversión utilizado es el canal 0 del microcontrolador, en todo momento. Al adquirir los datos de caracterización de una señal de tensión o corriente, la rutina programada *libmot()*

accionará al motor de paso y procederá a adquirir los datos de conversión pertinentes. La cantidad de datos tomados en un punto del accionamiento mecánico son escogidos por el usuario y a su vez, limitados por el sistema (capacidad de memoria RAM disponible), esto con el fin de poder realizar un promedio de las medidas tomadas por el sistema en el computador personal y obtener un valor más cercano a la realidad al reducir ruido aleatorio de la adquisición. Al haber tomado todas las muestras requeridas por el usuario, el sistema hará avanzar al motor al siguiente paso hasta completar todos los determinados previamente por el tope mecánico.

2.1.3. Módulo CCP1 (Captura, Comparación, PWM):

El módulo de CCP1 fue configurado en modo de captura, específicamente para detectar la ocurrencia de 4 flancos de subida de la señal proveniente del detector de cruce por cero y así cuantificar el período de la señal de entrada. En cada accionamiento mecánico, el sistema esperará la ocurrencia de 2 eventos consecutivos del módulo CCP1 a través de la interrupción de dicho módulo, obteniendo así la información referente a un período completo de la señal observada. El trabajo del módulo CCP1 está directamente asociado con el temporizador TIMER1 del microcontrolador que fungirá de base de tiempo, el mismo está configurado para trabajar con una rata de avance 1:1 con respecto al reloj del microcontrolador (realiza una cuenta cada 250ns) de manera sincrónica. Luego de haber capturado los 2 eventos antes mencionados, el sistema procederá a avanzar un paso mecánico más hasta completar todos los pasos que necesita la caracterización y a su vez, pondrá a cero los registros usados para adquirir el período de la señal (TIMER1 y CCP1). Este módulo pertenece a la rutina de adquisición de datos de período *libmot2()*.

2.1.4. Módulo CCP2 (Captura, Comparación, PWM):

La modulación por ancho de pulso se implementó a través del módulo CCP2, configurado para tal fin. El mismo posee una frecuencia de señal fija de aproximadamente 1kHz (usando el TIMER2 como base de tiempo) con 25 posibles niveles de variación de ancho de pulso (variación mínima de 4%). Como fue mencionado anteriormente, la salida del módulo PWM es procesada por un pseudo

integrador para convertir esta modulación en un nivel D.C. que permita variar la frecuencia de salida de un oscilador controlado por tensión. El control del porcentaje de salida en alto del PWM es manejado a través del computador mediante una pantalla virtual de accionamiento.

2.1.5. Puerto B:

Los Terminales más significativos de puerto B (RB7, RB6, RB5, RB4) fueron conectados a las salidas del bloque comparador de nivel de señal, logrando conocer el nivel de tensión de la señal de entrada al convertidor A/D del μ Controlador. Esta modalidad es conocida como “*Interrupción al cambiar el valor del puerto B*”, la cual forma parte de una de las facilidades existentes en el μ Controlador usado. Según el estado de las salidas de los bloques comparadores detectores del nivel de la señal se programó el microcontrolador para que accione los potenciómetros digitales de ganancia y de nivel de pedestal, buscando la corrección del nivel presente de la señal de entrada al convertidor Analógico Digital. El intento de converger a un valor apropiado de nivel de señal ocurre accionando el potenciómetro de ganancia para disminuir la misma. Si aún después de haber disminuido completamente la ganancia del sistema adecuador se sigue generando el aviso de nivel fuera de rango en el puerto B se procederá a aumentar o a disminuir el valor de pedestal D.C. presente, accionando el potenciómetro de nivel de pedestal. Por último, si después de haber accionado los potenciómetros digitales no se logra adecuar el nivel de la señal en el rango de 0V a 5V, se envía un aviso emergente al computador indicando que la señal se encuentra fuera de rango para el sistema. Adicionalmente, cada vez que el sistema corrija el nivel de la señal de entrada, se enviará un aviso emergente al computador para indicar que hubo un cambio en el tratamiento de la señal por parte del sistema con el nuevo valor de accionamiento de los potenciómetros digitales y el valor de accionamiento mecánico para el cual ocurrió el cambio (paso). Así, dicho cambio es tomado en cuenta por el sistema al trazar la característica de la Función de Red o Función de Escala. El estado del puerto B (lectura de bloque comparador de nivel de señal) se ejecuta haciendo clic en el botón “Edo.Sistema” de la ventana “*Visor del Sistema*” del programa que comanda el computador personal.

2.2. Programación del computador personal:

La programación del computador fue concebida en forma de ventanas dinámicas, dichas ventanas servirán para supervisar y controlar al sistema Analizador de Funciones de Escala. Como fue mencionado anteriormente, las acciones a ejecutar por parte del sistema son transmitidas a través del puerto serial del computador. El programa principal en el computador se encarga de las siguientes acciones:

- Validación de acciones a ejecutar por parte del sistema A.F.E. (configuración y condiciones de las pruebas a realizar).
- Canalización de los avisos emergentes provenientes del sistema A.F.E. con el fin de tomar las acciones operativas pertinentes y ofrecer al usuario un informe del suceso ocurrido.
- Identificar las cabeceras de tramas enviadas por el sistema A.F.E. y realizar los cálculos necesarios que conduzcan a la transformación de los datos digitales en los valores físicos pertinentes.
- Manejar el paquete de cálculo EXCEL® con el fin de vaciar los datos recibidos y convertidos por el Visual Basic® en una hoja de cálculo para luego graficarlos. Esto se realiza bajo el protocolo de VBA¹⁰, el cual permite a la plataforma de Visual Basic® comandar a algunos de los paquetes desarrollados por la misma casa de software. Así mismo, se permite que el usuario tenga opción a graficar los errores de cada caracterización y guardar los datos que resulten de la experiencia realizada para su posterior estudio.
- Permitir cambiar el modo de ajuste de nivel de señal de entrada al convertidor analógico digital entre modo automático o modo manual. En los casos en que se seleccione el modo manual, permite ajustar los niveles de pedestal D.C. y de ganancia del bloque adecuador del sistema A.F.E.
- Realizar la búsqueda del tope mecánico del sistema de manera automática con sólo realizar la petición por pantalla. Se cuenta además con la posibilidad

¹⁰ Por sus siglas en inglés Visual Basic Application. (Interconectividad para aplicaciones con Visual Basic)

de realizar movimientos de ajuste (corrimiento de la posición inicial) sobre el valor de accionamiento encontrado por tope mecánico.

- Manejo del módulo PWM para comandar la entrada del VCO logrando variar la frecuencia de su señal de salida.
- Posibilidad de adquisición de algún parámetro eléctrico (tensión, corriente, frecuencia o período) en un punto de accionamiento cualquiera, logrando así ajustar la función bajo estudio a un valor en un accionamiento mecánico deseado.

CAPÍTULO III

Características y comportamiento del prototipo construido

1. Descripción física y funcionamiento general:

El prototipo construido A.F.E. ha logrado satisfacer los requerimientos previos a su elaboración realizando las adquisiciones de datos deseadas en sincronía con la variación del parámetro θ , el cual rige el comportamiento de la red bajo estudio. La interacción existente entre los bloques que componen el hardware con el software desarrollado, tanto en el microcontrolador como en el manejo del computador personal, cumplen con las características pedidas en la elaboración del equipo.

El sistema está dividido en dos bloques principales, el primero de ellos es el módulo A.F.E. y el segundo es el computador personal, éste será el encargado de enviar los comandos de operación al módulo A.F.E. a su vez que recibirá y procesará el conjunto de datos provenientes de dicho módulo para lograr trazar las gráficas de caracterización. Para usar el sistema, primero se debe establecer comunicación con el computador a través del puerto serial COM1, ésto con el fin de verificar que dicho puerto de comunicaciones se encuentra libre. El usuario ejecutará el programa principal en el computador, el cual chequeará el estado del puerto, de no ser posible la apertura del puerto serial por el programa principal, emergerá un mensaje al usuario indicando que el puerto no puede ser utilizado en este momento por el equipo y la ejecución del programa se detendrá. Si por el contrario, el programa principal logra comunicarse a través del puerto serial, emergerá un mensaje al usuario indicando que encienda o reinicie el módulo A.F.E. para establecer comunicación. En el momento en que el módulo A.F.E. es encendido o reiniciado, se ejecutan las configuraciones iniciales de todo el equipo (valores de puertos, configuración de niveles de ganancia y pedestal D.C. en el bloque de adquisición, estado general del módulo para ser reportado, lectura de temperatura ambiente, etc.). El módulo A.F.E. envía entonces una trama de respuesta para indicar al computador que se encuentra encendido, conectado y listo para ser utilizado por intermedio del programa principal,

este evento es presentado con un mensaje emergente al usuario, en donde se le reporta, visualmente, que el sistema está listo para usarse.

La ventana principal del programa posee un menú de opciones que serán activadas secuencialmente a medida que el usuario interactúe con el sistema, como ejemplo de esto se tiene que el sistema tendrá inhabilitada las opciones de adquisición hasta que el usuario realice la búsqueda automática del tope mecánico del motor de paso, el cual debe ya tener vinculado mecánicamente a su eje el elemento variable a caracterizar. Luego de este paso, se habilitarán las opciones de graficación y adquisición de datos en ventana. El módulo A.F.E. responderá con la cantidad de pasos que resultaron de la detección del tope mecánico en la ventana “*Control de movimiento del motor*” y el estado general del sistema en la ventana “*Visor del sistema*”. En esta última ventana se reportan los siguientes datos:

- Valor actual de temperatura ambiente en grados centígrados.
- Valor en porcentaje del cursor del potenciómetro digital de pedestal que forma parte del sistema de adquisición de datos.
- Valor en porcentaje del cursor del potenciómetro digital de ganancia del sistema de adquisición de datos.
- Modo de ajuste del nivel de la señal que entra al convertidor analógico digital. (Automático o manual).

Se procede ahora a indicarle al sistema, a través del programa principal del computador en la ventana “*con Módulo Graficador*” la prueba que se desea realizar, es decir, una adquisición de datos de tensión (A.C.-D.C.), corriente (A.C.-D.C.), frecuencia o período. Además, en la misma ventana se le indica de que manera se desean adquirir los datos de la caracterización con uno de los dos modos disponibles que son: 1- “*Tomar # muestras por cada paso*” en donde se realiza la prueba tomando “n” muestras en cada paso a dar y se recorrerá el accionamiento con variaciones de paso en paso o 2- “*Tomar muestras cada # pasos*” en el que se realiza la prueba tomando el promedio de 10 muestras entre cada “n” pasos. El programa principal tomará en cuenta si la petición hecha por el usuario puede ser cumplida, dadas las

limitaciones de memoria RAM (módulo A.F.E.) y memoria reservada para la entrada de datos del puerto serial del computador (Programa principal).

Una vez hecha la petición de variable eléctrica a caracterizar y forma de adquisición, el programa principal pide al usuario que introduzca la función teórica que describe el comportamiento de la escala bajo estudio para así obtener los datos numéricos del comportamiento teórico de la misma, envía la configuración y las peticiones del usuario al módulo A.F.E., éste a su vez comienza la prueba. Al terminar de recorrer mecánicamente todo el accionamiento del elemento variable desde el valor máximo ($\theta = 1$) hasta el mínimo ($\theta = 0$), el módulo A.F.E. se encarga de devolver el valor de la variable θ a su posición inicial con el movimiento del eje del motor de paso, vaciar la memoria RAM y ordenar los datos recolectados para luego enviarlos al computador. El programa principal, ejecutándose en el computador, realiza entonces un proceso automatizado de graficación estableciendo un vínculo con una hoja de cálculo en Microsoft® Excel®, donde será vaciado todo el contenido de los datos recopilados, ordenándolos por columnas con las siguientes informaciones: 1- El accionamiento θ normalizado (valor entre 0 y 1 indicando el mínimo y máximo respectivamente) para el cual pertenecen los datos. 2- El valor teórico de la variable eléctrica estudiada. 3- El valor adquirido experimentalmente por el módulo A.F.E. con la información adicional de la temperatura ambiente en la cual fue hecha la caracterización. Para finalizar la ejecución de la automatización, se realiza la gráfica correspondiente al grupo de datos que llenan la hoja de cálculo y se agrega la correspondiente leyenda, nombre de ejes coordenados con unidades y se escala automáticamente el gráfico, adicionalmente, el programa principal pregunta al usuario si desea realizar cálculos del error absoluto entre las caracterizaciones realizadas y el valor teórico previamente introducido. Por último, se grafican también estos resultados, culminando una experiencia completa entre el valor máximo y mínimo de la variable mecánica θ .

El programa principal permite, entre otras cosas, seguir haciendo caracterizaciones; a un conjunto de caracterizaciones se le denomina “Experiencia”

en donde existe un límite en cuanto al número de las mismas, las experiencias pueden tener un máximo de 24 caracterizaciones en una misma hoja de cálculo.

Para lograr proporcionar información al usuario en cuanto al valor de la variable eléctrica estudiada sin querer trazar una gráfica (realizar una caracterización), el sistema está dotado de un monitor de adquisición. El mismo es invocado desde el menú principal con la opción “*Adquisición*”, “*en accionamiento fijo*”, este monitor permite al usuario conocer el valor de θ actual y modificarlo en caso de ser requerido, a la vez que realiza una visualización del valor de la variable a medir (tensión, corriente, frecuencia o período). Esta ventana fue agregada para facilitar el uso del sistema, brindándole al usuario información necesaria al momento de realizar ajustes de la red bajo estudio. Una vez que el usuario termina de ajustar la escala en el punto y valor de interés, se podrá trazar nuevamente la característica de la red bajo estudio con la ventana “*con Módulo Graficador*” en donde se apreciarán los cambios en las respuestas, dada una modificación hecha a la red, debido a que dichas gráficas (anteriores y posteriores al proceso de ajuste) son trazadas en el mismo gráfico, facilitando su comparación.

Los datos recopilados se guardan en un archivo de la aplicación Microsoft® Excel®, esto ocurre cuando culmina una experiencia (grupo de caracterizaciones), de manera automática. El usuario decide cuando termina la recolección de datos indicándoselo al programa con el botón “Abrir-Continuar-Cerrar Experiencia”; la hoja de datos pregunta si se desea guardar los datos, en qué carpeta y/o dispositivo de almacenamiento masivo se desea guardar y qué nombre va a recibir el archivo. Luego de esto se podrá usar nuevamente el sistema A.F.E. para comenzar con una nueva experiencia.

2. Criterios y características del hardware construido:

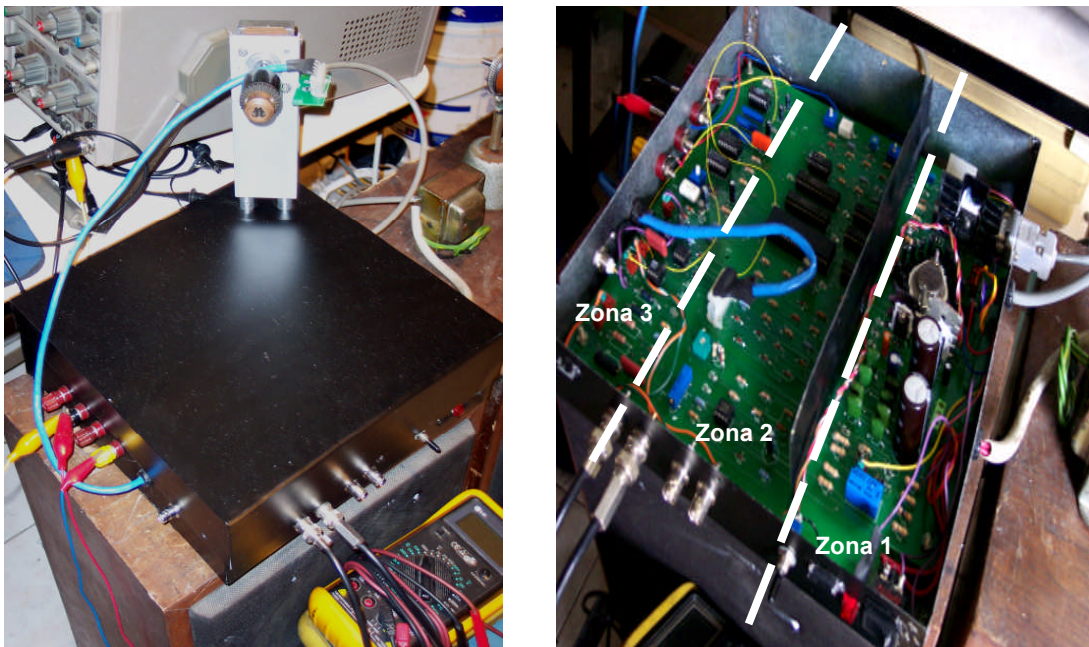
Se construyó una tarjeta de circuito impreso a doble cara para proveer de soporte a la interconexión de dichos componentes, se elaboró una caja metálica que sirviera de recinto para la tarjeta de circuito impreso que a su vez proporcionara blindaje electromagnético al sistema físico antes mencionado, evitando así

interferencias y ruido eléctrico. El equipo ha sido provisto de un interruptor de encendido, un indicador luminoso del estado encendido o apagado, un pulsador para reiniciar el módulo, conectores BNC para entradas de señales de tensión y corriente, salidas de tensión reguladas de valores 5V, 12V, -12V con el fin de ser utilizadas por el usuario con fines de facilitar la alimentación de las redes bajo estudio y por último, salida de tensión variable entre los 0V a -12V DC por medio de un conector BNC con el propósito de ser la entrada del oscilador controlado por tensión de los generadores Hewlett Packard® modelo 3311A permitiendo modular en frecuencia a la señal de salida de dicho generador. Todos los terminales de entrada o salida antes mencionados están debidamente identificados en el aparato. El motor de paso, encargado de realizar las variaciones del parámetro θ del cual depende la red bajo estudio, se encuentra también unido mecánicamente a la caja metálica. Por último, el sensor de temperatura ambiente se encuentra adherido en la superficie exterior de la caja metálica, aislado térmica y eléctricamente con un asiento de material de corcho.

En la distribución de los componentes en el área de la tarjeta de circuito impreso se tuvo especial cuidado en separar a los mismos en tres zonas según la temperatura que dichos componentes alcanzan en su seno al tener encendido el equipo. La primera zona (zona 1) fue constituida por componentes que tienen la función de proveer energía al módulo (Reguladores de tensión, manejador del motor de paso, relé selector de entrada tensión/corriente y convertidor TTL-RS232-TTL), la segunda zona (zona 2) fue compuesta por todos aquellos componentes que tienen funcionamiento digital (Microcontrolador, Memorias RAM, Latch de direccionamiento, Comparadores, Compuertas lógicas). La tercera zona (zona 3) está compuesta de todos aquellos componentes que tienen operación analógica, cuyo comportamiento y respuesta son más sensibles que los anteriores componentes mencionados en cuanto a la temperatura y a los cambios que ésta pueda presentar.

Teniendo la distribución de los componentes planteada por zonas, se agregó una placa deflectora metálica entre la zona 1 y la zona 2 (Fuentes de alimentación - Circuitos digitales) con el fin de lograr una distribución isotérmica en las zonas 2 y 3.

La placa deflectora tiene como propósito desviar el curso del fluido (aire caliente) hacia arriba, buscando que éste no afecte la zona de componentes analógicos al generar fuertes gradientes de temperatura en la superficie de la tarjeta de circuito impreso. En definitiva, puede observarse que la distribución de los componentes en las zonas obedeció a alejar, en lo posible, a los circuitos analógicos de adquisición de datos de las fuentes de calor. La figura 31 muestra una foto del aspecto físico del módulo A.F.E., la figura 32 muestra la tarjeta prototipo con la placa deflectora de temperatura y sus zonas.



Figuras 31 y 32. (Arriba) Aspecto del equipo A.F.E. (Derecha) Detalle de la tarjeta prototipo con la placa deflectora de temperatura indicando las zonas de división.

La alimentación del equipo se lleva a cabo directamente de la red de distribución de energía, luego, por intermedio de un transformador, se disminuye la tensión nominal de $120 \pm 10\% \text{ VA.C.}$ a aproximadamente $30 \pm 10\% \text{ VA.C.}$ Ésta a su vez es rectificada y filtrada para posteriormente ser estabilizada con reguladores monolíticos lineales, los cuales proveen al módulo A.F.E. de tensiones de alimentación de valores 12V, -12V y 5V. El módulo está protegido contra

sobretensiones por un fusible de 500mA en el lado del primario del transformador de potencia. El consumo promedio medido del módulo A.F.E. es de 30W @ $120\pm 10\%$ V.A.C. aproximadamente.

En cuanto al diseño de las fuentes reguladas, se hicieron separaciones para las alimentaciones de los bloques de los circuitos digitales y analógicos, cada uno de estos bloques cuenta con su regulador específico para así evitar problemas de ruido de conmutación digital en la parte analógica del módulo por ser ésta la más sensible, dado que es la que maneja la obtención y conversión de datos de tensión y corriente. Se agregaron además para dichas alimentaciones filtros pasabajos pasivos de primer orden, individuales, los cuales ayudan a eliminar ruido en las fuentes, estabilizando aún más las polarizaciones requeridas por los elementos activos que componen el módulo. Por último, se hizo la separación de las líneas de referencia digitales de las analógicas siendo independientes unas de la otras.

Con referencia a la interconexión entre el Microcontrolador y los bloques de almacenamiento dinámico de datos RAM y Latch's, se tomó la precaución de colocarlos físicamente cercanos para reducir el largo de las pistas de conexión, con el fin de evitar fenómenos de acoplamientos capacitivos y de ruido que pudiesen aparecer y alterar los datos digitales obtenidos, tomando en cuenta además la frecuencia en la que suceden las transacciones de entrada-salida entre ellos, la cual oscila aproximadamente entre los 2 y 4 MHz.

En lo que respecta al motor de paso, luego de realizar varias pruebas, se determinó que el mismo podía funcionar con una corriente máxima de 100mA por bobina. Ésto trajo como ventaja la reducción en la cantidad de energía consumida por el módulo al disminuir la corriente demandada al regulador de 12VD.C.(U4) y por ende, la reducción de la temperatura interna del recinto metálico que contiene el módulo A.F.E.

La detección automática de la traba del motor posee el inconveniente que es altamente dependiente de la temperatura a la cual se encuentra operando el manejador del motor de paso MC3479-D, por esta razón, al encender el equipo, se debe aguardar

alrededor de 1min. para luego hacer la detección de traba del motor con éxito. Basado en las pruebas realizadas, se logró determinar que el error de detección en promedio es de +10 pasos (por exceso). El bloque detector de traba del motor está provisto de dos potenciómetros para hacer los ajustes en cuanto a la sensibilidad en la detección del estado de traba, no obstante, se hace difícil conseguir un punto de ajuste apropiado debido a la naturaleza del circuito concebido y por el corrimiento en temperatura del manejador del motor de paso.

Con relación al funcionamiento del modo automático de adecuación de nivel de señal, se puede comentar que éste basa su operación en el cambio de estado de los comparadores contenidos en el componente U2, estos comparadores son los encargados de indicarle al microcontrolador el nivel de la señal de entrada al convertidor analógico digital. Si el estado de estas 4 salidas (salidas de los 4 comparadores) es inadecuado, el microcontrolador accionará el potenciómetro de ganancia para disminuir la misma, buscando regularizar el estado de salida de los comparadores; ésto sucederá hasta llegar al valor de ganancia mínima (potenciómetro de ganancia al 0% de accionamiento). Si aún luego de disminuir la ganancia hasta su mínimo, no se logra conseguir un valor correcto de las salidas de los comparadores, se procede a accionar convenientemente el potenciómetro digital de pedestal D.C. (aumentando o disminuyendo el valor del pedestal), buscando situar el nivel de entrada del convertidor entre el rango dinámico de valores (0V-5V). Si ocurre que luego de tratar de ajustar el nivel de la señal de pedestal no existe forma de conseguir el valor adecuado en las salidas de los comparadores, se envía al computador una trama que será interpretada por el programa principal como un mensaje emergente al usuario, que le indica que la señal de entrada está fuera del rango de operación y que, de querer medirla, se deberá utilizar un atenuador a la entrada. Es importante destacar que cualquiera sea la medición realizada, el sistema, al comenzar a adquirir los datos, ajustará la ganancia al máximo posible (97% de accionamiento, *criterio de máxima ganancia*). Durante el transcurso de las pruebas y de ser necesario, este nivel de ganancia disminuirá de forma automática, como fue mencionado anteriormente. En los casos que el usuario no desee utilizar el modo de ajuste automático del nivel de la

señal, se tendrá al sistema en modo manual en donde es el usuario el que interactúa con el sistema para fijar la ganancia y el nivel de pedestal D.C. del sistema de adquisición de datos. De igual manera, si el nivel de la señal de entrada al convertidor analógico digital no se encuentra en el rango válido, se enviará al computador una trama que será traducida como un mensaje de advertencia al usuario indicándole la situación y cómo poder subsanarla.

Para realizar las mediciones de frecuencia y período se diseñó un circuito comparador capaz de detectar el cruce por cero de una señal. Dicha señal debe poseer un valor mínimo de 100mV_{pico} para poder ser detectada. El circuito en sí posee una alta impedancia de entrada (12M Ω aproximadamente), la cual no modifica considerablemente la impedancia de entrada del módulo originalmente concebido (1M Ω aproximadamente), adicionalmente tiene la capacidad de soportar amplitudes de la señal de entrada de hasta cientos de Voltios pico.

La conexión entre el amplificador instrumental de entrada y las entradas de tensión o corriente se realizan con ayuda de un relé comandado a voluntad del usuario por intermedio del microcontrolador. Cabe destacar que la entrada de corriente posee una protección de fusible, el cual admite un valor de corriente de hasta 2A.

La forma de calibración de los valores de ganancia y nivel de pedestal del sistema de adquisición, así como también los valores de corrimiento de nivel D.C. del amplificador instrumental, fue implementada con tablas de ajuste. Para elaborarlas, se realizaron mediciones de dichos parámetros a través de todos los posibles puntos de accionamiento, estableciendo los valores correspondientes. Estas tablas son usadas por el programa principal en conjunto con los datos de adquisición para realizar los cálculos de los valores de las caracterizaciones provenientes del módulo A.F.E. El formato real de cada tabla de ajuste es un archivo plano de texto, a su vez, cada adquisición es ajustada según los parámetros de ganancia y nivel de pedestal con la cual fue recogida en el momento de la caracterización.

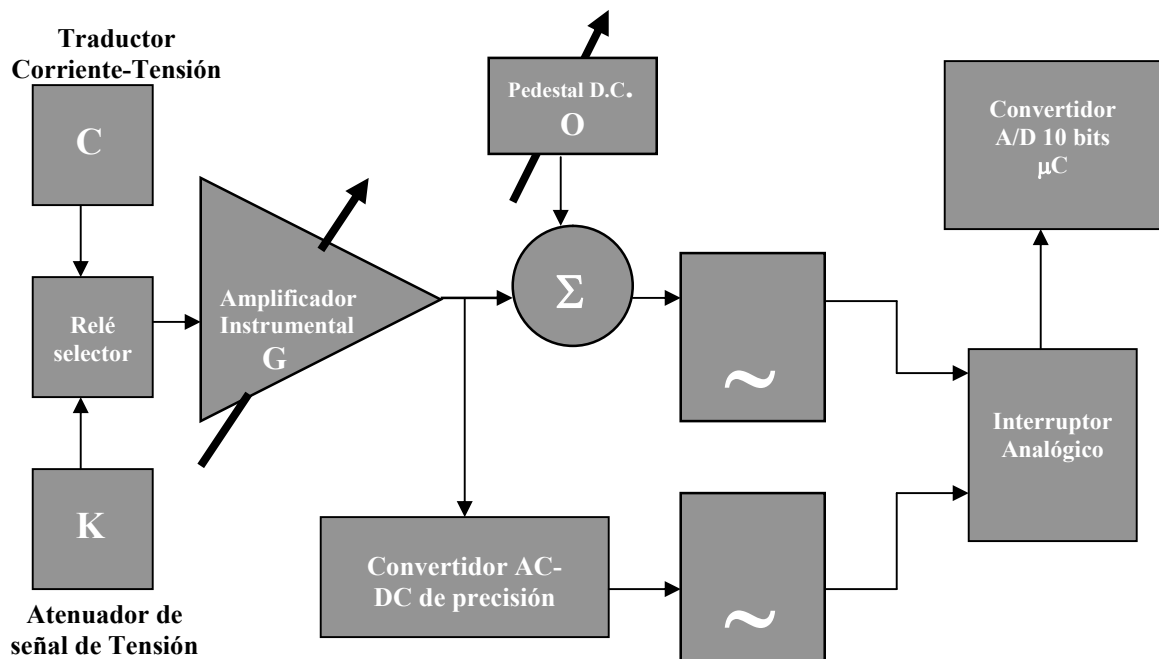


Figura 33. Diagrama de bloques del sistema analógico de adquisición.

Se aprecia en la figura 33 que tanto el bloque “Amplificador Instrumental **G**” como el bloque “Pedestal D.C. **O**” son variables, esto gracias a su diseño, implementado con potenciómetros digitales. También se aprecia que la señal de entrada al convertidor Analógico Digital del microprocesador es filtrada previamente, acotando así el ancho de banda de la información a procesar sin importar la naturaleza de la misma (A.C. o D.C.). La selección de la adquisición entre señales D.C. o A.C. se lleva a cabo con ayuda de un interruptor analógico comandado por el microprocesador.

A continuación se presenta un resumen de la tabla de ajuste de la ganancia del sistema de adquisición en función del valor de accionamiento del potenciómetro digital de ganancia. Como fue mencionado, estos datos pertenecen a un archivo plano de texto usado por el programa principal del computador para calibrar los valores de los parámetros que componen el sistema de adquisición y calcular los valores de las adquisiciones realizadas.

Tabla 6.Valores resumidos de ganancia del sistema de adquisición.

Valor Accionamiento Potenciómetro Digital de Ganancia (x%)	Ganancia teórica G (V/V)	Ganancia experimental (V/V)
	$G = \left(1 + \frac{50000}{\left(5000 \cdot \left(1 - \frac{x}{100} \right) \right)} \right)$	
0%	2	1,18± 0,001
5%	2,05	1,20± 0,001
10%	2,10	1,24± 0,001
15%	2,16	1,26± 0,001
20%	2,23	1,33± 0,001
25%	2,32	1,39± 0,001
30%	2,41	1,42± 0,001
35%	2,54	1,50± 0,001
40%	2,64	1,57± 0,001
45%	2,79	1,67± 0,001
50%	2,96	1,78± 0,001
55%	3,17	1,92± 0,001
60%	3,44	2,06± 0,001
65%	3,78	2,28± 0,001
70%	4,23	2,58± 0,001
75%	5,00	3,00± 0,001
80%	6,00	3,61± 0,001
85%	7,67	4,68± 0,001
86%	8,14	5,01± 0,001
87%	8,69	5,37± 0,001
88%	9,33	5,78± 0,001
89%	10,09	6,29± 0,001
90%	11	6,93± 0,001
91%	12,11	7,69± 0,001
92%	13,50	8,70± 0,001
93%	15,29	10,05± 0,001
94%	17,67	11,93± 0,001
95%	21,00	14,72± 0,001
96%	26,00	19,25± 0,001
97%	34,33	27,97± 0,001
98%	51,00	96,75± 0,001
99%	101,00	698,04± 0,001

Las mediciones presentadas en la tabla 6 fueron realizadas en D.C. utilizando una señal de entrada V_i conocida. La tabla 7 contiene los valores teóricos y prácticos resumidos de los niveles de pedestal del sistema. Esta tabla está contenida en un archivo plano de texto con el fin de que el programa principal del computador realice

el ajuste del nivel de pedestal introducido por el sistema de adquisición en los cálculos resultantes de las adquisiciones.

Tabla 7. Valores resumidos de pedestal D.C. del sistema de adquisición.

Valor Accionamiento Potenciómetro Digital de pedestal (x%)	Pedestal teórico O (V) $O = 2 \cdot \left(-2,5 + 2,5 \cdot \left(\frac{2 \cdot x}{99} \right) \right) v$	Pedestal experimental (V)
0%	-5,000	-4,86±0,01
5%	-4,495	-4,35±0,01
10%	-3,990	-3,84±0,01
15%	-3,485	-3,33±0,01
20%	-2,980	-2,83±0,01
25%	-2,475	-2,33±0,01
30%	-1,970	-1,82±0,01
35%	-1,465	-1,32±0,01
40%	-0,960	-0,82±0,01
45%	-0,455	-0,30±0,01
50%	-0,051	0,19±0,01
55%	0,556	0,69±0,01
60%	1,061	1,19±0,01
65%	1,566	1,69±0,01
70%	2,071	2,07±0,01
75%	2,576	2,57±0,01
80%	3,081	3,08±0,01
85%	3,586	3,69±0,01
90%	4,091	4,20±0,01
95%	4,596	4,70±0,01
99%	5,000	5,01±0,01

Se tiene también un archivo de texto plano para poder realizar el ajuste del nivel de corrimiento D.C. del amplificador instrumental cuando la señal de entrada es de tensión y otro archivo similar para cuando la señal de entrada es corriente, la modificación es debida al cambio de impedancia Thévenin equivalente que experimenta el amplificador instrumental en su entrada, dado el cambio de impedancia que existe al querer adquirir señal de tensión o de corriente (uso del Bloque K o Bloque C en la entrada). Se presenta a continuación una tabla resumen de

los valores para el nivel de desviación D.C. de tensión en la tabla 8. Las mediciones fueron realizadas cortocircuitando las entradas del amplificador instrumental y conectadas a la referencia del módulo A.F.E.

Tabla 8. Cuadro resumen de los niveles de tensión de corrimiento D.C. del amplificador instrumental en función de la ganancia para modo de medición en tensión.

Valor Accionamiento Potenciometro Digital de pedestal (x%)	Pedestal experimental (mV)
0%	27,50±0,1
5%	32,00±0,1
10%	35,05±0,1
15%	38,00±0,1
20%	40,99±0,1
25%	43,85±0,1
30%	46,68±0,1
35%	49,56±0,1
40%	52,60±0,1
45%	55,92±0,1
50%	59,60±0,1
55%	63,45±0,1
60%	68,50±0,1
65%	74,83±0,1
70%	83,60±0,1
75%	95,38±0,1
80%	111,50±0,1
85%	134,90±0,1
90%	161,75±0,1
95%	216,10±0,1
99%	288,20±0,1

La tabla 9 presentada a continuación, contiene los valores resumidos de pedestal D.C. para cuando la señal de entrada es de corriente.

Tabla 9. Cuadro resumen de los niveles de tensión de corrimiento D.C. del amplificador instrumental en función de la ganancia para modo de medición en corriente.

Valor Accionamiento Potenciómetro Digital de pedestal (x%)	Pedestal experimental (mV)
0%	29,44±0,1
5%	34,05±0,1
10%	35,05±0,1
15%	37,10±0,1
20%	40,26±0,1
25%	43,37±0,1
30%	46,36±0,1
35%	49,30±0,1
40%	52,33±0,1
45%	55,63±0,1
50%	59,13±0,1
55%	63,10±0,1
60%	67,11±0,1
65%	72,43±0,1
70%	79,20±0,1
75%	88,45±0,1
80%	100,93±0,1
85%	118,00±0,1
90%	142,85±0,1
95%	180,46±0,1
99%	291,90±0,1

Las mediciones presentadas en las 4 tablas anteriores fueron medidas con un Multímetro digital marca LG, modelo DM-341 con las siguientes especificaciones:

- Exactitud $\pm(0,05\% + 4\text{dgt})$ en las escalas usadas (200mV y 20V), en modo de medición D.C.

El bloque denominado K en la figura 30 es una red atenuadora pasiva, formada por 2 resistencias de valor $1\text{M}\Omega$ y 2 de valor $11\text{k}\Omega$, 1W de disipación de potencia, de película de metal y del 1% de tolerancia. Dichas resistencias se conectaron para formar 2 divisores de tensión que alimentaran las entradas inversora

y no inversora del amplificador instrumental. El valor de atenuación K es de $0,018 \pm 0,0001$ V/V, medido experimentalmente.

El bloque denominado C es una resistencia de valor $1,11\Omega \pm 0,0225\Omega$ que actúa como traductor de la corriente a tensión, dicha señal traducida es llevada a la entrada del sistema de adquisición con ayuda del relé selector de medición en el modo de corriente. La resistencia posee una capacidad máxima de disipación de potencia de 3W. La medición del valor fue hecha con un Multímetro digital marca LG, modelo DM-341 en el rango de 200Ω y exactitud $\pm(2\%+5\text{dgt})$.

El sistema analógico de adquisición de datos posee una respuesta en frecuencia de 20kHz (el doble de lo requerido). En lo que respecta al modo en que adquiere la señal se debe mencionar que puede ser de manera flotante, sin embargo, la degradación en el nivel de rechazo modo común de la entrada puede afectar considerablemente los resultados de la medición por efecto de la frecuencia de la señal de entrada, la figura de rechazo modo común disminuye de 100dB a aproximadamente 40dB en el rango de 0Hz a 20kHz. Dada esta premisa, se escogió trabajar con el *criterio de máxima ganancia* posible en el sistema de adquisición dado que la característica de rechazo modo común mantiene valores altos al usar ganancias altas.

Los rangos de tensión de entrada para los cuales el módulo está acondicionado para trabajar son:

- En A.C.: Hasta 230 VA.C. (Pico) en la entrada.
- En D.C.: Desde -220 hasta 450VD.C. en la entrada.

Para la medición de corriente eléctrica los rangos son:

- En AC y DC: Hasta 2A limitados por la disipación de potencia del elemento traductor de corriente-tensión, con protección de fusible.

La figura de rechazo al modo común de la etapa de adquisición de datos se ve degradada a medida que se exija al amplificador instrumental más ganancia,

adicionalmente, dicha figura también degrada su valor a medida que la señal que se desea amplificar aumenta su frecuencia en Hz. Analizando la hoja de datos que provee el fabricante del dispositivo (Amplificador Instrumental INA 121) se consigue esta característica, la cual se presenta a continuación en la figura 34. Tomando en cuenta este comportamiento, se trabajó con el *criterio de máxima ganancia* en el sistema de adquisición para conseguir minimizar este efecto.

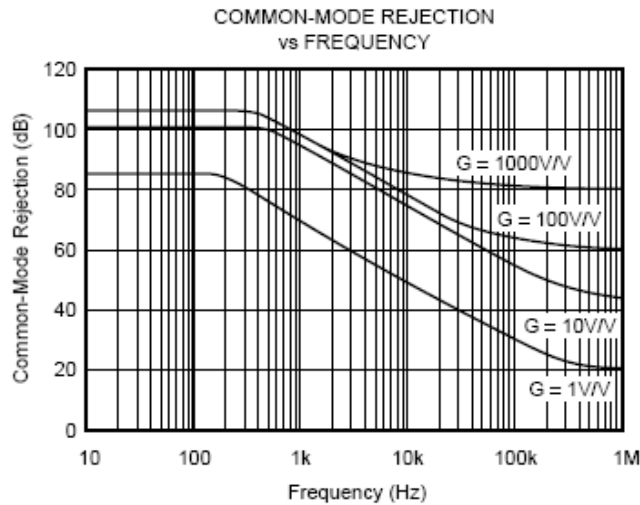


Figura 34. Respuesta en frecuencia del rechazo a modo común del INA121.

En lo que respecta a la salida de tensión variable tenemos como principal característica que su rango está comprendido entre 0V a -10V $\pm 0,4$ V D.C. con un voltaje de rizado de ± 20 mV como peor caso, la capacidad de corriente de salida está limitada a aproximadamente 8mA@-10V D.C. (peor condición) y utiliza la protección de corto circuito existente en el amplificador operacional (LF444) implementado. La protección interna del operacional es la única protección de la corriente de salida. Estas mediciones fueron realizadas con un Osciloscopio analógico marca Hameg, modelo HM203-5.

Como fue mencionado, el módulo A.F.E. provee de salidas de tensión estables, las mismas son tomadas directamente de las fuentes lineales que alimentan el módulo. La capacidad de corriente de salida de las mismas es la siguiente:

- Fuente de tensión D.C. (SI-3122V): +12V: hasta 100mA, sin protección de cortocircuito agregada en dicha salida.
- Fuente de tensión D.C. (LM7912) -12V: hasta 100mA, sin protección de cortocircuito agregada en dicha salida.
- Fuente de tensión D.C. (LM7805) + 5V: hasta 100mA, sin protección de cortocircuito agregada en dicha salida.

Se debe tener especial cuidado en el uso de las fuentes antes mencionadas ya que como se dijo anteriormente, no poseen protección de corriente excepto por la interna de los correspondientes reguladores monolíticos. Al hacer uso indebido de estas fuentes se corre el riesgo de provocar mal funcionamiento en el módulo A.F.E. Las especificaciones de corriente de salida antes mencionadas fueron estimadas tomando en cuenta la máxima corriente que el regulador es capaz de entregar a una carga (información del fabricante) con respecto a la corriente que dicho regulador suministra al módulo A.F.E. en condiciones normales de operación.

3. Características del software implementado:

3.1. Programa del microcontrolador:

Como fue mencionado, el módulo A.F.E. responde a tramas de datos enviadas desde el computador a través del programa principal, dichas tramas son reconocidas e interpretadas por el microcontrolador para ejecutar una acción en específico, de allí deriva el hecho que el módulo A.F.E. carece de un terminal de mandos físico como, por ejemplo, un teclado. Al enviar al módulo A.F.E. una trama de comando, el mismo la atiende por interrupción y se procede a realizar el reconocimiento e interpretación de dicha trama mediante 2 rutinas, la primera de ellas, *recibiri()*, se encuentra vinculada directamente con la vectorización de la interrupción de recepción de datos con el fin de ejecutar una acción con prontitud y sin importar la tarea que esté haciendo el microcontrolador en ese instante; esta rutina contiene todos los comandos que necesitan ser ejecutados para tomar acciones inmediatas sobre el sistema como:

- La detención del motor de paso en medio de una prueba a modo de pausa al igual que la reanulación de las acciones.
- La cancelación de la prueba que se está realizando.
- Devolución a posición original del eje del motor de paso.
- Reporte de la naturaleza de la medición a realizarse, modo A.C. y D.C.
- Cambio de ajuste de nivel de señal (modo automático o manual).
- Modificación del nivel de salida de tensión variable.
- Reportar estado de sistema en general, *configti()*.

La segunda rutina de interpretación de tramas de comandos, *traduce()*, tiene una prioridad secundaria y se ejecuta sólo cuando el microcontrolador no está ocupado en alguna tarea. Está diseñada para realizar las ejecuciones de los comandos que se encargan de realizar las siguientes funciones:

- Rutinas de caracterización al realizar barridos completos de la variable θ , *libmot(valor a, valor b)* y *libmot2(valor x, valor y)*. La rutina *libmot(valor a, valor b)* fue diseñada para trabajar de 2 maneras distintas.

El modo 1 de operación consiste en recorrer mecánicamente todo el accionamiento θ previamente obtenido para caracterizar, paso a paso, #n muestras del valor eléctrico de la señal en tensión o corriente D.C. o A.C. Esta rutina permite realizar peticiones de adquisición al módulo con la posibilidad de reducir estadísticamente el ruido al obtener una cantidad de muestras que luego serán automáticamente promediadas. La cantidad de muestras a solicitar está restringida y validada por el programa principal del computador, el cual se encarga de verificar si la cantidad de datos totales requeridos a manera de muestras son manejables por el módulo A.F.E. El cálculo de la cantidad de muestras depende de la cantidad de pasos a dar, del tipo de caracterización (Tensión o Corriente) y de la cantidad de memoria RAM disponible en el módulo para almacenar los datos mientras la prueba se realiza. La fórmula deducida es la siguiente:

$$\#muestras = \frac{\left(\frac{65535}{pt} - 5 \right)}{2} \quad (12)$$

Donde:

La constante 65535 es la cantidad de Bytes disponible en la memoria RAM instalada en el módulo.

pt = Cantidad de pasos totales obtenida de la detección de la traba del motor.

El modo 2 de operación de la rutina ***libmot(valor a, valor b)*** consiste en realizar la caracterización tomando valores cada #n cantidad de pasos, incluyendo el valor $\theta = 0$ y $\theta = 1$. El valor leído por el módulo se promedia 10 veces con el fin de reducir ruido eléctrico aleatorio. De la misma forma que en el modo de operación 1, el programa principal del

computador verifica que el módulo pueda realizar la caracterización pedida mediante la fórmula siguiente:

$$25 \cdot (n_{pd} + 2) = 65535 \quad (13)$$

Donde:

n_{pd} = Número de pasos en los que se tomarán datos.

La rutina para medición de frecuencia o período **libmot2(valor x, valor y)** tiene 2 modos de funcionamiento, el modo 1 tomará los datos de la caracterización paso a paso durante toda la adquisición. El segundo modo es tomando los datos cada #n de pasos incluyendo los valores de $\theta = 0$ y $\theta=1$. En ambos modos de adquisición se espera por 2 flancos consecutivos de subida de la señal de entrada para avanzar al siguiente paso autorizado.

- Activación del relé selector de señal de entrada, tensión o corriente.
- Rutinas para realizar adquisición de datos de señal en un punto fijo de accionamiento de la variable θ . **mpad()**, **mfp()**. En este modo, para la adquisición de tensión o corriente **mpad()**, se toman 10 muestras para luego promediarlas y reducir el ruido aleatorio de las mediciones. En la rutina **mfp()** se esperan 2 interrupciones consecutivas del módulo CCP1 del microcontrolador, en modo captura, para luego realizar la resta algebraica de ambos valores, obteniendo así el intervalo de tiempo transcurrido ente flancos.
- Cambio de fuente de señal a medir (interruptor analógico A.C.-D.C.).
- Detección de traba del motor de paso **topmot()**.
- Corrimiento del punto inicial de la variable θ luego de haber detectado la traba automáticamente.

- Adquisición de datos de temperatura ambiente. *midetmp()*.
- Accionamiento de potenciómetro digital de ganancia para obtener automáticamente el máximo de ganancia posible del sistema de adquisición (sólo en modo de ajuste de nivel automático).
- Accionamiento de potenciómetro digital de ganancia para obtener el mínimo de ganancia del sistema de adquisición (sólo en modo de ajuste automático).
- Devolución del motor a su θ inicial o a algún punto cualquiera válido de accionamiento.
- Modificación de los valores de ganancia o pedestal D.C. en modo de ajuste manual.

El vector de interrupciones, por su parte, atiende los siguientes eventos:

- Recepción de comandos enviados por el computador.
- Detección de cambio en el puerto B con el fin de ajustar el nivel de la señal adquirida (en modo automático) y de aviso al usuario de la necesidad de corrección de los valores de ganancia y/o tensión de pedestal D.C. (en modo manual). El valor hexadecimal que indica el estado normal del nivel de la señal a medir es A0h el modo D.C. En modo A.C. los valores que indican el estado normal del nivel es A0h o 20h.
- Finalización del tiempo de integración de la señal de corriente consumida por el motor de paso para la detección de traba del eje mecánico del mismo. Dicho tiempo está estipulado experimentalmente en 256 cuentas de 250ns cada una, es decir, 63,75 μ s.
- Finalización de conversión analógica-digital para tomar como válidos los datos que resultan de la misma.
- Aviso de interrupción externa para detectar el estado de la salida del comparador detector de traba del motor de paso y contabilizar los

flancos que dicha señal genere. Para desencadenar las acciones de cambio de giro del motor o detención del mismo se contabilizan 20 interrupciones con el fin de eliminar la respuesta transitoria del mecanismo.

- Detección de flanco válido con el módulo de captura CCP1 para la contabilización de tiempo transcurrido entre 2 consecutivos y adquirir dato de período de la señal. El flanco es contabilizado como válido al haber ocurrido 4 veces en la realidad y ser de subida.
- Sobrepasamiento de la capacidad del temporizador Timer1 para llevar la cuenta del período con 32 bits y así incrementar el rango de medición de esta variable física (Adquisición de Período de señal).
- Al iniciar el sistema, el programa también se encarga de proveer a los componentes del módulo A.F.E. de los valores de configuración de arranque válidos, las acciones son las siguientes:
- Direccionamiento de los puertos A, B, C, D y E (como entrada o salida) del microcontrolador según el requerimiento así como también proveer del valor lógico a cada terminal (*inicpa()*, *incipb()*, *incipc()*, *incipd()*, *incipd()*, *incipd()*).
- Posicionamiento de los cursores de los potenciómetros digitales, el potenciómetro de ganancia estará accionado a 0% (ganancia mínima). El pedestal D.C. se acciona a 73% del accionamiento (aproximadamente la mitad del valor de tensión del rango del convertidor A/D).
- Configuración del sensor de temperatura TMP122 (*configtmp()*).
- Configuración del módulo interno USART para establecer comunicación serial.

Adicionalmente, el microcontrolador posee una rutina, *vaciar()*, encargada de extraer los datos contenidos en RAM del módulo A.F.E. al determinar que éstos

existen. Luego de esto, los datos son transmitidos al computador en su totalidad con la rutina *enviar(#número de datos)*.

3.2. Programa principal del computador:

El software implementado en el computador para realizar el programa principal fue programado en Microsoft® Visual Basic 6®. Dicho ambiente de programación posee la característica de permitir el desarrollo de programas con la filosofía de orientación a eventos y la ventaja de proveer facilidades en el desarrollo de interfaces gráficas facilitando el entendimiento entre el sistema desarrollado y el usuario.

El proyecto posee compatibilidad con la filosofía de interconexión con aplicaciones VBA¹¹ de la casa Microsoft® para que interactúen entre sí. Aprovechando estas facilidades, se logró implementar una interconexión directa entre el programa Visual Basic 6® y la hoja de cálculo Excel®, en donde, a través del Visual Basic® se maneja a conveniencia dicha hoja de cálculo. Primero se realiza el vaciado de los datos y la ejecución de los cálculos pertinentes, provenientes del módulo A.F.E., a la hoja de cálculo. Luego, se procede a graficar dichos datos usando las facilidades del módulo Graph® de Excel®. Por último, se presenta al usuario la gráfica de los datos recolectados y procesados en pantalla con la ventaja de que, tanto los gráficos, como los datos obtenidos forman parte de una hoja de Excel® que podrá ser tratada como un simple archivo de extensión .XLS. Es importante destacar que los gráficos contienen los títulos de ejes, leyenda de datos del gráfico (identificador de curvas) y se realizan a una escala conveniente, permitiendo una completa visualización de las curvas en los valores de accionamiento θ normalizados (entre 0 y 1).

Entre los componentes que fueron agregados al programa principal se encuentra el módulo para manejo de comunicación serial MSComm® de la casa Microsoft®. El módulo posee una cantidad de rutinas destinadas al manejo rápido y

¹¹ Por sus siglas en inglés Visual Basic Applications (Núcleo de la plataforma para automatizaciones con y entre los productos Microsoft Office).

sencillo del puerto serial, incluyendo la atención de eventos por interrupción (al ocurrir actividades específicas en el mismo). La configuración del puerto serial por el programa principal fue la siguiente:

- A través del puerto COM1 del computador.
- Velocidad 9600 Bps¹².
- Trama de 8 bits, sin bit de paridad, 1 bit de parada.
- Comunicación asíncrona.

Entre los eventos utilizados para el desarrollo del programa principal que tienen relación con el uso del control MSComm, se utilizó el evento MSComm.CommEvent = 2, que significa que ha ocurrido la recepción de #i datos en el búfer¹³ de recepción del computador, este evento es desencadenado por la cantidad de datos que llegan al buffer¹³, dicha cantidad de datos (#i) es configurable a través del registro MSComm.SThreshold = (número de datos mínimo que se reciben antes de desencadenar el evento).

La inclusión en el programa principal del control MSScript Control® obedece a la necesidad de manipulación de ecuaciones matemáticas por parte del programa para su posterior evaluación y representación gráfica. Dicha necesidad surgió cuando se planteó el requerimiento de que el usuario le indicase al programa principal cuál es la función teórica que describe el comportamiento de la red a estudiar, ésto con el fin de comparar gráficamente las respuestas teórica y práctica que arroje el estudio (realizando gráficas solapadas en el transcurso de las experiencias), facilitando el estudio. El control MSScript Control® requiere únicamente una entrada de texto, el control se encarga de conseguir la lógica algebraica en la entrada y procede a ejecutar peticiones como por ejemplo la evaluación numérica de la expresión en un punto.

Por último, se incluyó en el programa principal un temporizador de eventos genérico, MSTimer®, con el fin de plantear un intervalo de tiempo que realizará

¹² Por sus siglas en inglés Bauds per second (Baudios por segundo)

¹³ Espacio físico de memoria destinado a almacenar datos para su manipulación.

peticiones al módulo A.F.E. al enviar comandos para ejecutar acciones. Específicamente, esta temporización se utiliza en la ventana “*en accionamiento fijo*”, para hacer peticiones de adquisición en un punto fijo de accionamiento físico θ cualquiera al módulo A.F.E. en donde se realizan adquisiciones cada 1,5s, siendo este el intervalo de tiempo establecido por dicho control.

A continuación se explicarán, una a una, las ventanas que componen al programa principal y su función. La ventana principal está formada por 2 submenús que dan acceso a las funcionalidades del módulo A.F.E. y a su vez sirven para realizar las validaciones pertinentes en cuanto a las peticiones del usuario para interactuar con el módulo. Los submenús antes mencionados son “*Adquisición*” y “*Configuración*”.

La figura 35 muestra el aspecto general del menú principal con los submenús antes mencionados:



Figura 35. Apariencia en ventana del menú principal.

La opción “**Adquisición**” da acceso a 2 ventanas que son “*en accionamiento fijo*” y “*con Módulo Graficador*”. La ventana “*en accionamiento fijo*” sirve para realizar adquisiciones de la variable estudiada (Tensión, Corriente, Frecuencia o Período) en un punto cualquiera del accionamiento θ estático (permitiendo incluso modificarlo a un valor cualquiera de interés para el usuario). La ventana “*con Módulo Graficador*” es usada para cuando se va a realizar un estudio completo de una función de escala, en donde se hace un barrido completo de la variable θ y se trazan las curvas que dicho estudio arroje en un gráfico de Excel®. La ventana “*en accionamiento fijo*” se muestra en la figura 36.



Figura 36. Apariencia de la ventana “*en accionamiento fijo*”.

Como puede observarse en la figura 36, en la ventana existen selectores del parámetro que se desea medir (V, A, f y T), El botón “Mover” sirve para alterar el valor del accionamiento θ actual (valor que se muestra en todo momento actualizado en el recuadro “Theta Actual”). Al pulsar virtualmente el botón se despliega la ventana “*Modificación de accionamiento actual*”, la cual pregunta a qué nuevo valor se desea mover el parámetro θ . El botón “Atrás” conduce al usuario al menú principal nuevamente. El botón “Adquirir” comienza el envío de comandos al módulo A.F.E.

para ejecutar las rutinas *mpad()* o *mfp()*, en los casos en el que se quiera ejecutar una adquisición de tensión o corriente *mpad()*, se despliega previamente una ventana con nombre “*Modo de medición*” en donde el usuario le indica al módulo que la adquisición requerida es A.C. o D.C. Luego de esto, al pulsar el botón “Aceptar” de dicha ventana, el módulo A.F.E. responde con la trama de datos pertinente a la adquisición demandada. Los datos de respuesta son traducidos por el programa principal y presentados en la ventana “*en accionamiento fijo*”, colocando además las unidades físicas correspondientes y el modo en el que fueron adquiridas (A.C. o D.C.). La figura 37 muestra la ventana “*Modificación de accionamiento actual*”. La figura 38 muestra la ventana “*Modo de medición*”.



Figura 37. Ventana para realizar alteración del parámetro θ .



Figura 38. Ventana para seleccionar el modo de adquisición con θ fijo.

Para detener la adquisición de datos con esta ventana, basta con pulsar el botón “Detener” de la ventana “*en accionamiento fijo*” (anterior botón “Adquirir”). Cabe destacar que no se permite realizar alteraciones de la variable θ mientras se están realizando adquisiciones a través de esta ventana, esta condición es evitada por el programa principal inhabilitando el botón “Mover” al comenzar una adquisición.

La ventana “*Modificación de accionamiento actual*” solamente aceptará la introducción por teclado de un número de 2 decimales, separados por punto (Ejemplo: 0.37). De no respetarse esta condición el usuario será avisado con un mensaje emergente, advirtiéndole de esta situación. El botón “Aceptar” permitirá la ejecución de la alteración de la posición actual del eje del motor al valor requerido por el usuario, el módulo selecciona automáticamente el sentido de giro y calcula los pasos requeridos para que el motor altere la variable θ según la petición realizada. El botón “Borrar” limpia el campo de inclusión del nuevo accionamiento. El botón “Cancelar” devuelve a la ventana “*en accionamiento fijo*” sin alterar el valor previo del accionamiento θ .

La siguiente ventana, “*Módulo Graficador*”, se presenta en la figura 39.

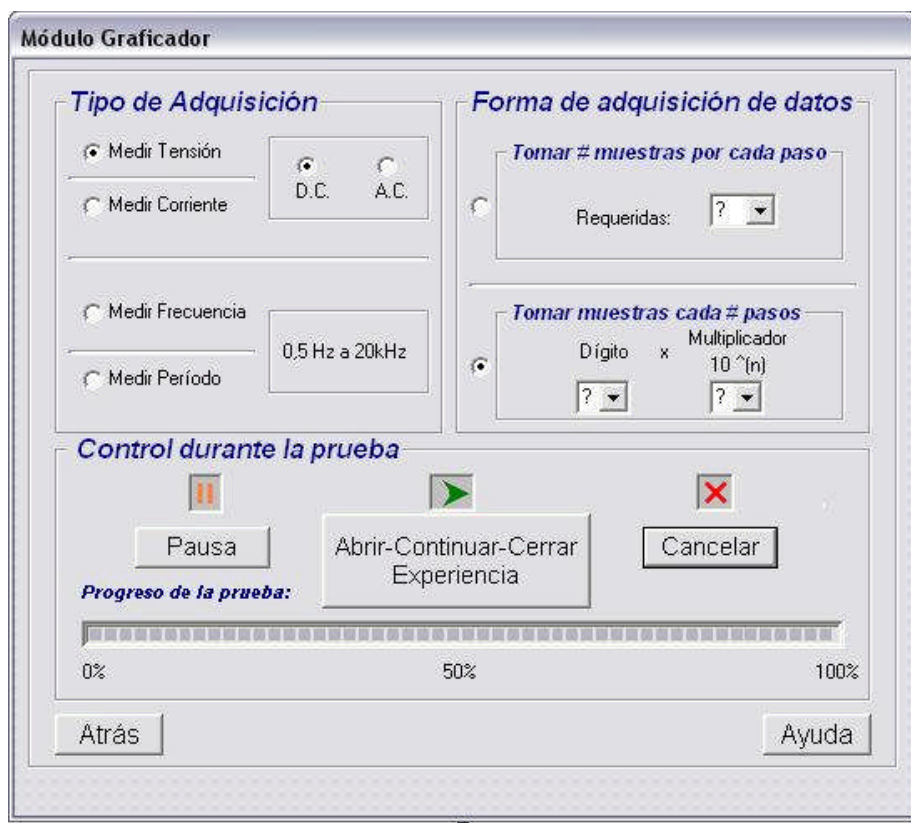


Figura 39. Ventana “*Módulo Graficador*”.

Esta ventana contiene las opciones de adquisición para realizar las caracterizaciones de la función de escala. En ella se reconocen 3 áreas, la primera

llamada “Tipo de Adquisición” encierra las opciones de selección de variable eléctrica a caracterizar. Para el caso de adquisición de tensión o corriente se debe especificar además si la misma será en modo D.C. o A.C. La segunda área, llamada “Forma de Adquisición de Datos” pide al usuario que seleccione entre los 2 modos de adquisición que poseen las rutinas del microcontrolador, a fin de caracterizar la función de red en la forma “Tomar # muestras por cada paso” o “Tomar muestras cada # de pasos”. Para la opción “Tomar # muestras por cada paso” el programa principal verifica el número de mediciones máximas que el módulo A.F.E. puede hacer tomando en cuenta la fórmula (12) y validando contra la selección hecha por el usuario en el recuadro “Requeridas:”. Con la opción “Tomar muestras cada # de pasos”, el programa principal usa la fórmula (13) para realizar la validación según lo seleccionado por el usuario en los recuadros “Dígito” y “Multiplicador” que son la forma de introducir los parámetros de adquisición. El número de pasos entre muestras se consigue con la siguiente fórmula:

$$pem = D \cdot 10^n \quad (14)$$

Donde:

pem = Número *de* pasos entre medición.

D = Dígito (número entre 1 y 9).

n = Exponente para la base 10 (número 0, 1, 2 ó 3).

La última área de la ventana, llamada “Control Durante la Prueba” muestra una barra que indica el progreso de la prueba en porcentaje caracterizado (0 al 100%), que se encuentra sincronizada con la posición de la variable θ actual. El botón “Cancelar” anula inmediatamente el transcurso de la prueba sin importar a que nivel se encuentre; para el módulo A.F.E. y para el programa principal esto es interpretado como que la prueba que ha sido cancelada nunca se llevó a cabo. El botón pausa sirve para detener el avance de la caracterización la cantidad de tiempo que el usuario crea pertinente, esta opción se agregó previendo alguna anomalía en el sistema mecánico o para corregir alguna situación imprevista. El botón “Abrir-Continuar-Cerrar

Experiencia” es el encargado, entre otras cosas, de comenzar realmente el proceso de caracterización de la función de escala de interés, dicho proceso se inicia preguntando al usuario si desea abrir una experiencia nueva, es decir, abrir un archivo en Excel® para anexar todos los datos que sigan a continuación a dicha hoja. Primeramente y de manera obligatoria, el programa principal le pide al usuario que introduzca la expresión matemática que describe el comportamiento teórico de la función de red a través de una ventana llamada “*Función Teórica*”. La ventana “*Función Teórica*” posee un campo para introducir texto por teclado y 2 botones, “Aceptar” y “Borrar”; el botón Borrar limpia el campo de introducción de texto. El botón “Aceptar” se encarga de validar la expresión introducida en el cuadro de texto y verificar errores de sintaxis; de haberlos, se le indicará al usuario que revise la fórmula introducida. La ventana “*Función Teórica*” se muestra en la figura 40 a continuación.

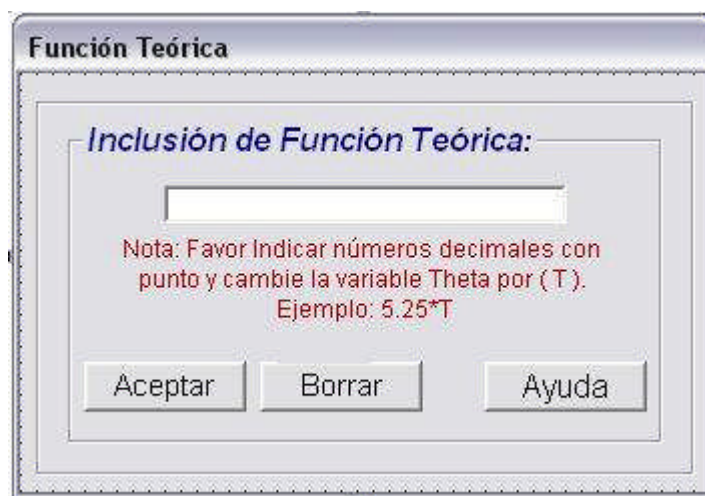


Figura 40. Ventana para la inclusión de la función teórica.

Una vez introducida la fórmula y revisada la sintaxis de la misma, el programa principal indica al usuario que la fórmula a sido introducida con éxito, desaparece la ventana “*Función Teórica*” y se vuelve a la ventana “*Módulo Graficador*” en donde para continuar la experiencia se pide al usuario que vuelva a pulsar el botón “Abrir-Continuar-Cerrar Experiencia”, esta acción deshabilitará todas las opciones de la ventana “*Módulo Graficador*” hasta terminar la experiencia completa. Hecho esto se despliega una hoja de Excel® con los valores de accionamiento mecánico

normalizados que arroja el módulo en una columna de la hoja (columna A) y con los valores de la función teórica que son el resultado de la evaluación de la misma por parte del programa principal en otra de las columnas (columna B) de la hoja de cálculo. Al terminar de llenar con los datos dichas columnas, el programa principal traza la característica de la función teórica en función del accionamiento θ en la gráfica 1 y está listo para comenzar la caracterización de la función de red. A continuación le pregunta al usuario “¿La siguiente adquisición forma parte del conjunto previamente abierto de experiencias?” cuyas opciones son SI o NO. Si el usuario contesta “**NO**”, el programa principal pregunta si se desea calcular los errores absolutos de las mediciones (en este caso, no hay error que calcular debido a que no se ha trazado ninguna caracterización práctica), la hoja de Excel® activa se cerrará preguntándole al usuario si desea guardar los cambios en un archivo. Por otra parte, si el usuario contesta SI a la pregunta, el programa principal envía al módulo A.F.E. los comandos necesarios para ejecutar la caracterización de la función de escala; una vez terminado el barrido de la variable θ , el módulo envía los datos al programa principal y éste a su vez se encarga de llenar la próxima columna disponible de la hoja de Excel® con estos datos, agregando al final de la columna el valor de temperatura ambiente actualizado para el momento en el que se comenzó la caracterización. Al completar esta fase, el programa principal abre una nueva gráfica, y traza la nueva caracterización incluyendo todas las columnas que contienen datos, sin eliminar la(s) anterior (es). Este proceso ocurrirá hasta un máximo de 24 caracterizaciones experimentales de una misma experiencia (en una misma hoja de cálculo). Al llegar a este límite, el programa automáticamente cerrará la experiencia y se dispondrá a preparar sus registros para comenzar una experiencia nueva. No es necesario realizar 24 caracterizaciones para concluir una experiencia en particular, en el momento en el que el usuario desee culminar la experiencia basta con indicarle al programa principal que la próxima adquisición “**NO** forma parte del conjunto de experiencias previamente abiertas” con lo que, seguidamente, se le pregunta al usuario si quiere realizar cálculos de los errores absolutos de las curvas trazadas por intermedio de la ventana “*Selección de columnas para cálculo de error*”. Esta ventana presenta un

cuadro de texto preparado para que el usuario le indique al programa principal a cuales columnas quiere calcular el error, las columnas se separan con coma y no poseen restricciones en cuanto al orden en que se introducen en el cuadro. (Ver figura 41) El sistema verifica la validez de la entrada y procede a realizar el cálculo del error para todo el rango de valores de la variable θ mediante la siguiente fórmula:

$$ErrorAbsoluto = V_t - V_p \quad (15)$$

Donde:

V_t = Valor de la función, producto de la fórmula teórica.

V_p = Valor de la adquisición de la variable.

Luego de llenar la hoja de datos en Excel® con estos resultados se realiza una nueva gráfica de todos los errores calculados en función de la variable de accionamiento θ . De no querer realizar cálculos del error, el procedimiento antes descrito se obvia pasando directamente a cerrar la aplicación Excel®, se le pregunta al usuario si desea guardar los datos de la hoja de cálculo, bajo qué nombre y en qué lugar (Elemento de almacenamiento masivo) desea guardarlos. De no querer guardar la hoja de Excel®, toda la experiencia será desechada de manera definitiva e irreversible.

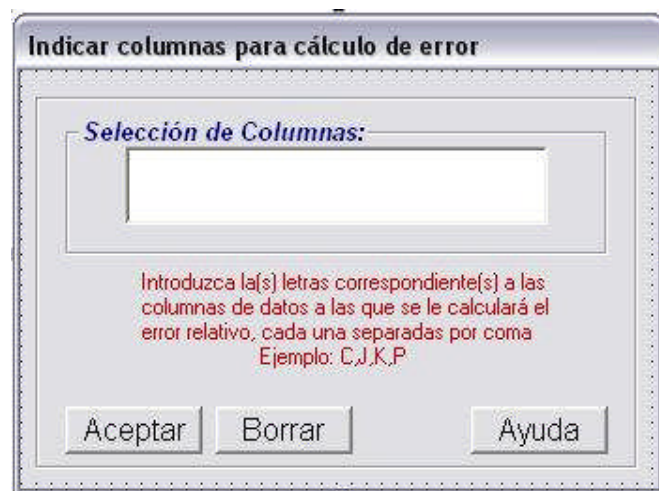


Figura 41. Aspecto de la ventana selectora de columnas para el cálculo de error.

A continuación se da una descripción de la opción “**Configuración**” del menú principal. Ésta da acceso a 3 nuevas ventanas que son: “*Al motor de paso*”, “*A potenciómetros*” y “*A excitación externa*”. La ventana “*Al motor de paso*” contiene todos los comandos que hacen falta para ejecutar la búsqueda de tope mecánico de manera automática con el botón “Tope Derecho”. También cuenta con el botón “Volver a Cero”, que se usa para cuando se desea volver al eje de motor a su posición θ inicial (por defecto $\theta=1$). La ventana también reporta al usuario los valores siguientes:

- Número de pasos a la izquierda del punto de partida inicial ($\theta=1$).
- Número de pasos a la derecha del punto inicial.
- Ángulo mecánico expresado en grados, equivalente al recorrido total del accionamiento ($0 \leq \theta \leq 1$) y calculado con la siguiente relación matemática:

$$Ar = \left(\frac{360 \frac{^\circ}{\text{vuelta}}}{200 \text{ pasos / vuelta}} \right) \cdot np = 1,8 \cdot np \left(\frac{^\circ}{\text{paso}} \right) \quad (16)$$

Donde:

np = número de pasos dados, contados desde el origen ($\theta = 1$).

Ar = Ángulo mecánico recorrido.

La ventana “*Al motor de paso*” posee también la facilidad de ajustar el comienzo del punto inicial del valor $\theta = 1$. Al haber hecho la detección automática de la traba del motor, se puede disminuir la cantidad de pasos contabilizados por el módulo llenando la entrada de texto disponible con un número entero de valor comprendido entre 1 y 255 que equivale directamente a la cantidad de pasos a los que desea ajustarse el valor máximo de θ . Si se desea modificar aún más el punto de accionamiento máximo, se puede repetir el comando antes mencionado hasta llegar al valor deseado. El programa principal se encarga de realizar las validaciones

correspondientes en cuanto a los caracteres introducidos por el usuario así como también de la posibilidad de ejecutar el comando con los requerimientos pedidos.

Es necesario mencionar que el módulo A.F.E. realiza los barridos de la variable θ desde su valor máximo ($\theta = 1$) hasta su valor mínimo ($\theta = 0$). Luego de realizar la detección de la traba del motor el módulo lleva, en todo momento, la cantidad de pasos dados en sentido antihorario (pasos a la izquierda) como los pasos dados en sentido horario (pasos a la derecha), por tanto, siempre se tiene información de la posición angular relativa del eje del motor. La figura 42 muestra el aspecto gráfico de la ventana “Al motor de paso”.

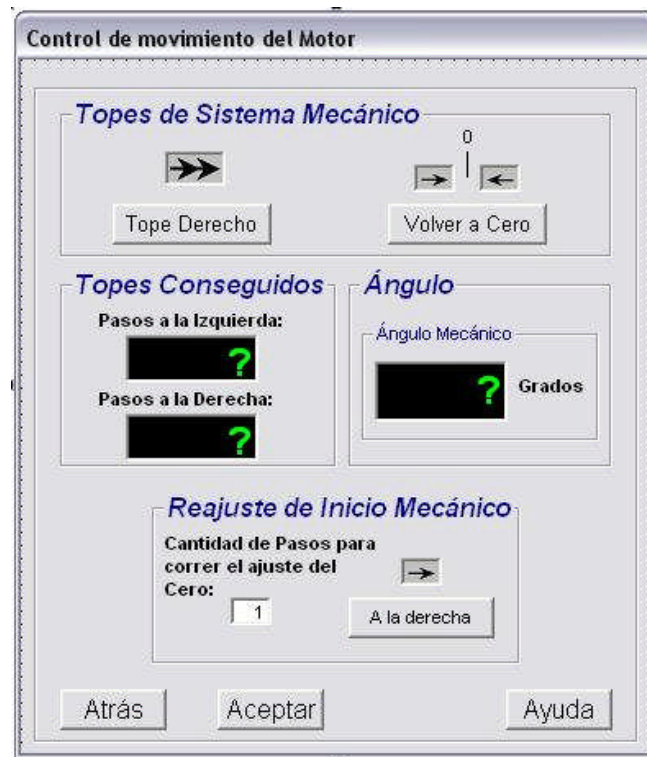


Figura 42. Ventana para el manejo del motor de paso.

La ventana “A potenciómetros” es la interfaz con el usuario que permite realizar la interacción con los parámetros variables del sistema de adquisición de datos (ganancia y pedestal D.C.). Existen 2 grupos principales de controles que la componen. El grupo denominado “Ajuste de Nivel de Señal” posee las opciones para poder cambiar el modo en el que el módulo A.F.E. realiza el ajuste del nivel de la

señal a la entrada del convertidor Analógico/Digital con las opciones “Automático” (modo por defecto) y “Manual”. En el modo “Automático” el módulo modificará, de ser necesario los niveles de ganancia y pedestal D.C. del sistema de adquisición para hacer que la señal adquirida se encuentre dentro del nivel de tensión permitido por el convertidor del microcontrolador (Nivel entre 0 y 5V). En el modo manual, es el usuario quien interactúa con los controles de ganancia y pedestal D.C. para realizar la adquisición. Se muestra a continuación el aspecto de dicha ventana en la figura 43.



Figura 43. Ventana de configuración para potenciómetros digitales.

Los cursores de los potenciómetros digitales sólo serán susceptibles de modificar cuando el módulo se encuentre en modo de ajuste de nivel de señal “Manual”, en donde el programa principal da acceso a mover virtualmente los controles deslizantes con los que se comanda cada potenciómetro. Tanto los valores de ganancia total del sistema de adquisición como el nivel de pedestal D.C. se muestran en sus correspondientes recuadros a manera de información al usuario, dichos valores cambiarán automáticamente al realizar cambios a través de los controles deslizables, manteniéndose actualizados en todo momento. Los valores por defecto al iniciar el módulo A.F.E son:

- Ganancia: (0% de accionamiento), equivalente a 0,02 V/V.
- Pedestal D.C.:(73% de accionamiento), equivalente a 2,515 V.

Los valores límites de los accionamientos de los potenciómetros digitales son:

- Accionamiento de potenciómetro de ganancia: Máx.: 98%, Mín.: 0%.
- Accionamiento de potenciómetro de pedestal: Máx.: 97%, Mín.: 0%.

La ventana “*A excitación externa*” es la encargada de controlar el nivel D.C. de tensión de salida, diseñada para ser usada junto con el generador de funciones HP3311A. Así se logra realizar barridos en frecuencia mediante el oscilador controlado por tensión del generador de funciones antes mencionado. La salida es comandada por un control deslizable cuyo recorrido corresponde al nivel de tensión D.C. en la salida. Se agregó un indicador virtual de Encendido/Apagado (simulando un LED¹⁴) y un botón para Encender/Apagar el módulo. Adicionalmente existe un recuadro indicador del valor de tensión de salida actual. La escala de salida viene dada en escalones de tensión (escala discreta), dado que el núcleo principal del módulo está basado en el periférico PWM¹⁵ del microcontrolador empleado. Los pasos de la escala que representa la salida de este módulo son en total 26. La tabla 10 muestra los valores de accionamiento del control deslizable y la tensión experimental de salida del módulo.

Tabla 10. Valores resumidos de la tensión D.C. de salida para controlar el V.C.O¹⁶ del generador de funciones HP3311A.

Valor del control deslizante (en pasos)	Tensión D.C. experimental (V)
0	-0,016±0,001
4	-1,581±0,001
6	-2,363±0,001
8	-3,145±0,001
10	-3,928±0,001
12	-4,709±0,001
14	-5,490±0,001
16	-6,272±0,001
20	-7,835±0,001
22	-8,616±0,001
25	-9,784±0,001

¹⁴ Por sus siglas en inglés Light Emitting Diode. Diodo emisor de luz.

¹⁵ Por sus siglas en inglés Pulse Width Modulation. Modulación por ancho de pulso.

¹⁶ Por sus siglas en inglés Voltage Controlled Oscillator. Oscilador controlado por tensión.

Los valores mostrados en la tabla 10 fueron medidos con un Multímetro digital marca LG, modelo DM-341 con: Exactitud 0,05% + 4dgt, en la escala de 20V, modo de medición D.C.

La figura 44 muestra la ventana controladora de la tensión D.C. variable.



Figura 44. Aspecto de la Ventana “*A excitación externa*”.

La manera de visualizar el estado general del módulo es por medio de ventana “*Visor del sistema*” que se presenta a continuación en la figura 42.

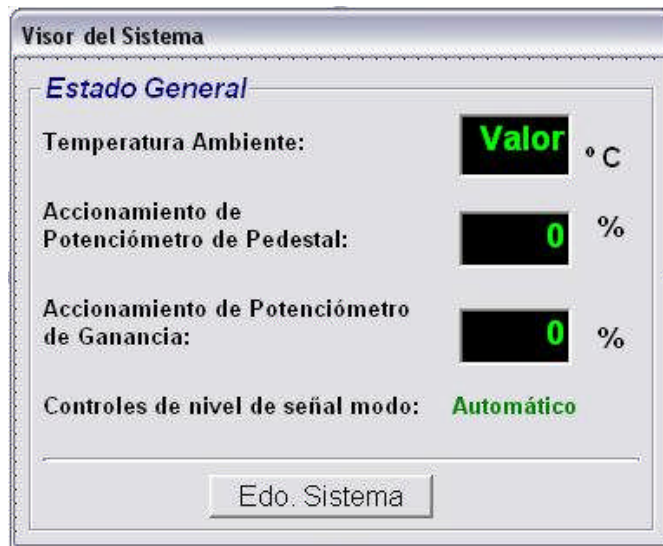


Figura 45. Aspecto de la ventana “*Visor del Sistema*”.

La ventana “*Visor del sistema*” le presenta al usuario información del valor de algunos de los parámetros relevantes en el sistema como:

- Valor de temperatura ambiente en grados Centígrados.
- Valor de accionamiento en porcentaje del potenciómetro de Pedestal D.C. del sistema de adquisición.
- Valor de accionamiento en porcentaje del potenciómetro de ganancia del sistema de adquisición.
- Estado en el modo de ajuste de nivel de señal; modo “Automático” o “Manual”.

La ventana posee además un botón “Edo. Sistema” para realizar una petición de actualización de los datos del módulo A.F.E. pidiendo la ejecución de la rutina *configti()*.

Es relevante mencionar que en todas las ventanas diseñadas se estableció una diferencia entre los cuadros de texto que sirven para que el usuario visualice valores del módulo y los que sirven como entrada de información para modificación de parámetros o inclusión de datos. Los cuadros de texto que sirven para visualizar reportes del módulo están diseñados con fondo de color negro y caracteres en color verde, por su parte, los cuadros de texto para incluir datos por parte del usuario están diseñados en fondo color blanco con caracteres en color negro. El programa principal se encarga de proteger los cuadros de texto informativos para que no puedan ser modificados por el usuario. Adicionalmente, las ventanas que así lo ameritan están dotadas de archivos de ayuda, activables en el transcurso de las experiencias. Las ayudas diseñadas dan explicación de los componentes y modos de uso de los mismos, facilitando así la interacción entre el operador y el sistema A.F.E.

4. Resultados obtenidos:

Con el fin de caracterizar el comportamiento eléctrico del sistema A.F.E., se realizaron pruebas orientadas a obtener el valor de precisión que brinda el aparato en cada una de las variables que puede caracterizar.

4.1. Precisión alcanzada:

Para lograr obtener medidas de la precisión del sistema se realizó una experiencia de adquisición compuesta por 24 curvas experimentales, cada una de ellas repitiendo la adquisición del mismo valor. Luego se estableció el análisis de los resultados con el estudio de la desviación estándar, promedio y mediana del conjunto para, al final, asumir un modelo lineal, de primer orden, y peor caso, estableciendo la tendencia del comportamiento del sistema A.F.E.

Para construir el modelo lineal de primer orden y de peor caso que represente la dispersión en la característica de precisión se usan las siguientes fórmulas:

$$V_n = \frac{(V_{\max} + V_{\min})}{2} \quad (17)$$

$$\Delta V_n = \frac{(V_{\max} - V_{\min})}{2} \quad (18)$$

Donde:

V_n = Valor nominal promedio entre los extremos.

ΔV_n = Error simétrico del valor nominal indicando la magnitud del entorno de valores.

4.1.1. Precisión en la adquisición de Tensión A.C. y D.C.:

Las pruebas realizadas para el caso de adquisición de tensión D.C. consistieron en 2 experiencias. La primera de ellas consistió en aplicar a la entrada del sistema una señal de 5V D.C. constante (independiente de la variable θ), buscando caracterizar el comportamiento estático de adquisición del sistema. La segunda consistió en aplicar una señal variable con respecto al accionamiento θ a un potenciómetro multivuelta lineal, marca SAKAE TSUSHINKOGYO®, modelo “HELICALOHM®” S46HP-3, de valor $5k\Omega \pm 1\%$, con una tolerancia en su linealidad $\leq 0,25\%$ en toda su escala, esto con el fin de poner a prueba el comportamiento dinámico de adquisición. Los resultados se presentan en la tabla 11 a continuación. Cabe destacar que la obtención experimental de los datos fue lograda mientras la variable de accionamiento θ se barre desde su máximo hasta su mínimo,

lo que implica el aumento del número de mediciones efectuadas. Por otra parte, se utilizó el método de adquisición del sistema denominado “# pasos entre muestras” de la ventana “*Módulo Graficador*”.

Tabla 11. Resultados estadísticos de la adquisición de una señal de tensión D.C.

Modo de prueba	Desviación Estándar (V)	Mediana (V)	Promedio aritmético (V)	Modelo (V) de dispersión (para peor caso)
Entrada estática (5V const.)	Máximo = 0,043V Mínimo = 0,022V	Máximo = 5,016V Mínimo = 4,992V	Máximo = 5,010V Mínimo = 4,998V	0,033±0,011
Entrada dinámica (0 a 5V)	Máximo = 0,055V Mínimo = 0,021V	4,998V	Máximo = 5,010V Mínimo = 4,998V <i>(para el caso Tensión entrada = 5V)</i>	0,038±0,017

Para el caso de adquisición de tensión A.C., tanto en modo estático como en modo dinámico, se utilizó una señal de entrada senoidal de 1kHz de frecuencia y amplitud 10,4Vpico extraída de un generador de funciones marca HEATHKIT, modelo IG-1275. La prueba de entrada dinámica fue realizada con el mismo potenciómetro usado en la prueba D.C. de precisión. La experiencia en general consta de 24 gráficos que arrojan el resultado estadístico y fue efectuada en las mismas condiciones que la correspondiente para determinar la precisión en las medidas tensión D.C. La tabla 12 contiene los resultados de la caracterización.

Tabla 12. Resultados estadísticos de la adquisición de una señal de tensión A.C.

Modo de prueba	Desviación Estándar (Vpico)	Mediana (Vpico)	Promedio aritmético (Vpico)	Modelo (Vpico) de dispersión (para peor caso)
Entrada estática (10,4Vp const.)	Máximo = 0,069V Mínimo = 0,020V	Máximo = 10,523V Mínimo = 10,342V	Máximo = 10,519V Mínimo = 10,361V	0,045±0,025
Entrada dinámica (0 a 10,4Vp)	Máximo = 0,036V Mínimo = 0,015V	10,421Vp <i>(para el caso Tensión entrada = 10,4Vp)</i>	Máximo = 10,569V Mínimo = 10,361V <i>(para el caso Tensión entrada = 10,4Vp)</i>	0,036±0,015

4.1.2. Precisión en la adquisición de Corriente A.C. y D.C.:

Las pruebas para determinar la dispersión de resultados en cuanto a la adquisición de corriente D.C. y A.C. fueron hechas tomando como herramientas las mismas usadas para la medición de precisión en el modo de tensión. Para medir en corriente D.C. se utilizó un regulador de tensión monolítico externo LM340 de 5V conectado en configuración de fuente de corriente constante. Para realizar la medición de corriente en modo A.C. se utilizó un generador de funciones marca HEATHKIT, modelo IG-1275 alimentando a un circuito resistivo. Se realizó una experiencia de medición (24 gráficas) para cada modo de corriente de entrada (D.C. y A.C.). La tabla 13 resume los valores estadísticos para la adquisición de corriente D.C. y la tabla 14 resume los resultados estadísticos para corriente A.C.

Tabla 13. Resultados estadísticos de comportamiento de la adquisición de una señal de Corriente D.C.

Modo de prueba	Desviación Estándar (mA)	Mediana (mA)	Promedio aritmético (mA)	Modelo (mA) de dispersión (para peor caso)
Entrada 46mA (const.)	Máximo = 0,322mA Mínimo = 0,065mA	Máximo = 45,530mA Mínimo = 45,340mA	Máximo = 45,435mA Mínimo = 45,278mA	0,243±0,078

Tabla 14. Resultados estadísticos de comportamiento de la adquisición de una señal de Corriente A.C.

Modo de prueba	Desviación Estándar (mApico)	Mediana (mApico)	Promedio aritmético (mApico)	Modelo (mApico) de dispersión (para peor caso)
Entrada 106,5mA (pico.)	Máximo = 1,022mA Mínimo = 0,410mA	Máximo=102,920mA Mínimo=102,200mA	Máximo =102,948mA Mínimo =102,441mA	0,716±0,306

4.1.3. Precisión en la adquisición de Frecuencia:

La adquisición de la variable frecuencia fue efectuada mediante un circuito temporizador LM555 en configuración astable, se modifica la frecuencia de su señal de salida al alterar el parámetro θ . Se plantearon 2 experiencias para la adquisición de frecuencia, en modo dinámico y estático. Las experiencias está conformadas a su vez

por 24 gráficos experimentales de donde se extraen los parámetros estadísticos presentados en la tabla 15 para la caracterización. El método adquisición usado fue el de “# pasos entre muestras”.

Tabla 15. Caracterización de precisión al adquirir datos de frecuencia.

Modo de prueba	Desviación Estándar	Mediana (Hz)	Promedio aritmético (Hz)	Modelo de dispersión (para peor caso)
Frecuencia (1100 Hz const.)	Máximo=0,389 Hz Mínimo=0,106 Hz	Máximo=1100,678Hz Mínimo=1100,448Hz	Máximo=1100,725Hz Mínimo=1100,403Hz	0,248±0,142 Hz
Frecuencia Entrada dinámica (540 a 9,5kHz)	Máximo=4,43% de la lectura. Mínimo=0,14% de la lectura.	8154,137Hz <i>(para el caso de frecuencia entrada = 8200 Hz)</i>	8159,758Hz <i>(para el caso de frecuencia entrada = 8200 Hz)</i>	2,28±2,14 % del valor nominal adquirido.

4.1.4. Precisión en la adquisición de Período:

La adquisición de la variable período fue realizada con el mismo circuito electrónico para caracterizar la adquisición de frecuencia, con 2 experiencias planteadas en modo dinámico y estático. Las experiencias está conformadas por 24 gráficos experimentales. El método adquisición fue el de “# pasos entre muestras”.

Tabla 16. Caracterización de precisión al adquirir datos de Período.

Modo de prueba	Desviación Estándar	Mediana (ms)	Promedio aritmético (ms)	Modelo de dispersión (para peor caso)
Período Entrada estática (0,907ms const.)	Máximo=0,0003 ms Mínimo=0,00008ms	Máximo = 0,909ms Mínimo = 0,909ms	Máximo = 0,909ms Mínimo =0,909ms	0,00022 ±0,00014ms
Período Entrada dinámica (0,109ms a 1,863ms)	Máximo = 0,008ms. Mínimo = 0,002ms.	1,224ms <i>(para el caso de periodo de señal =1,224ms)</i>	1,224ms <i>(para el caso de periodo de señal =1,224ms)</i>	0,005 ±0,003ms.

Cabe destacar que utilizando el modo de adquisición estadístico “# de muestras por cada paso” se consigue mejorar sustancialmente las figuras de precisión

del sistema en la adquisición de parámetros de tensión y corriente. Muestra de ello son los resultados de las pruebas siguientes en donde se tomaron 23 muestras por paso para luego promediarlas. Las Tablas 17 y 18 muestran los resultados estadísticos del uso de este método de adquisición.

Tabla 17. Precisión alcanzada con el modo de medición “# muestras en cada paso” para tensión D.C.

Modo de prueba	Desviación Estándar (V)	Mediana (V)	Promedio aritmético (V)	Modelo (V) de dispersión (para peor caso)
Entrada estática (5V const.)	Máximo = 0,043V Mínimo = 0,022V	Máximo = 5,016V Mínimo = 4,992V	Máximo = 5,010V Mínimo = 4,998V	0,022±0,009
Entrada dinámica (0 a 5V)	Máximo = 0,055V Mínimo = 0,021V	4,998V	Máximo = 5,010V Mínimo = 4,998V (para el caso Tensión entrada = 5V)	0,029±0,014

Tabla 18. Precisión alcanzada con el modo de medición “# muestras en cada paso” para tensión A.C.

Modo de prueba	Desviación Estándar (Vpico)	Mediana (Vpico)	Promedio aritmético (Vpico)	Modelo (Vpico) de dispersión (para peor caso)
Entrada estática (10,4Vp const.)	Máximo = 0,069V Mínimo = 0,020V	Máximo = 10,523V Mínimo = 10,342V	Máximo = 10,519V Mínimo = 10,361V	0,021±0,010
Entrada dinámica (0 a 10,4Vp)	Máximo = 0,067V Mínimo = 0,023V	10,411Vp (para el caso Tensión entrada = 10,4Vp)	Máximo = 10,469V Mínimo = 10,337V (para el caso Tensión entrada = 10,4Vp)	0,045±0,022

4.2. Exactitud alcanzada:

El parámetro exactitud del sistema A.F.E. en la adquisición de las variables eléctricas no pudo ser ponderado por no contar con el equipo de medición adecuado y las herramientas necesarias para su realización. Sin embargo, cabe destacar que el sistema A.F.E. no fue concebido para ser un instrumento de medición de dichas

variables, sus alcances fueron especificados y delimitados desde el comienzo del proyecto para concebirlo como un sistema capaz de trazar gráficamente la característica de una función de red o de escala al realizar variaciones del parámetro θ sin que ésto implique la obtención de valores de forma exacta.

CONCLUSIONES

El sistema prototipo A.F.E. cumplió satisfactoriamente los requerimientos exigidos en el Anteproyecto de Tesis ya que logra disminuir significativamente el tiempo empleado en trazar gráficamente una característica descriptiva de una función de escala en forma de tensión, corriente, frecuencia o período.

Los resultados arrojados por las pruebas realizadas a las adquisiciones revelan que el sistema A.F.E. posee un nivel bajo en la dispersión de los datos resultantes de las caracterizaciones. Al mismo tiempo, el sistema exhibe un comportamiento estable al permitir hacer repeticiones de una caracterización bajo las mismas condiciones.

La automatización implementada en el manejo de los datos de adquisición resultó muy conveniente y cómoda para el usuario, ésto debido a que los datos son homologados con respecto a una de las hojas de cálculo más populares en el mundo de la informática, las hojas de cálculo de Excel®, las cuales poseen opciones de interconectividad con otros métodos de almacenamiento, graficación, análisis numérico, análisis gráficos y ordenamiento disponibles en el mercado del software actual.

En cuanto a la automatización de la variación del parámetro θ hecha con el motor de paso, se debe resaltar que este método fue ventajoso ya que la estabilidad mecánica conseguida en el parámetro °mecánicos/paso fue importante en el desenvolvimiento del sistema completo. Cabe destacar que la concepción del diseño de los motores de paso obedece a la necesidad de poder realizar variaciones mecánicas precisas de posición sin tener necesidad de usar lazos de control que la corrijan. Así se logra implementar sistemas electromecánicos a lazo abierto con poco error en la posición angular al realizar movimientos, menos costosos y con menos componentes para su implementación.

El lenguaje utilizado para programar el computador personal facilitó enormemente el trabajo inherente que esto representa, la programación orientada a

eventos posee una cantidad de módulos preestablecidos que han sido diseñados para trabajar con interfaces gráficas de fácil aplicación y de rápido diseño. Las ventajas antes mencionadas no repercuten en la calidad, funcionalidad y presentación del programa realizado, por el contrario, inducen al programador de las mismas a enriquecer sus programas con todas las ventajas que el lenguaje ofrece. Se debe advertir de igual manera que el lenguaje de programación Visual Basic® no es el más idóneo para utilizarse cuando se requieran implementaciones de corte matemático (como evaluaciones de funciones, operaciones matemáticas no básicas, manejo directo de bloques de memoria, etc.), debido, entre otras cosas a que el mismo no está concebido para tal fin, sin embargo, ésto no significa que dichas tareas no puedan ser realizadas por dicho lenguaje.

En lo que respecta al lenguaje de programación utilizado para implementar el código en el microcontrolador se hace mención a que el mismo (ANSI C) resultó muy práctico y útil debido entre otras cosas a que posee funciones y rutinas ya programadas, las cuales optimizaron el tiempo de desarrollo del programa implementado.

El método empleado en la detección de traba del motor de paso no resultó el más idóneo, el circuito analógico implementado es muy sensible a los cambios de temperatura que sufre el manejador del motor de paso (MC3479D) por la corriente promedio suministrada al motor. Aunado a esto, el ajuste de la sensibilidad del detector es difícil de realizar y el mismo es vulnerable al ruido eléctrico del ambiente (por su concepción analógica). Por estas razones, y para obtener un mejor desempeño del sistema, el circuito antes mencionado debe ser sustituido. Una de las opciones para realizar dicha sustitución es presentada posteriormente en el aparte “RECOMENDACIONES”.

RECOMENDACIONES

Aún cuando se lograron cumplir con las premisas planteadas en el anteproyecto propuesto y el funcionamiento del prototipo construido es satisfactorio, se listan a continuación algunas posibles mejoras al sistema A.F.E., obedeciendo a criterios de diseño electrónico, simplificación del sistema en general, costos y avances tecnológicos.

El bloque adecuador puede ser sustituido prácticamente en su totalidad por el componente AD7731 de la casa Analog Devices®. La figura 46 muestra un diagrama de implementación, como bloque de adquisición de datos.

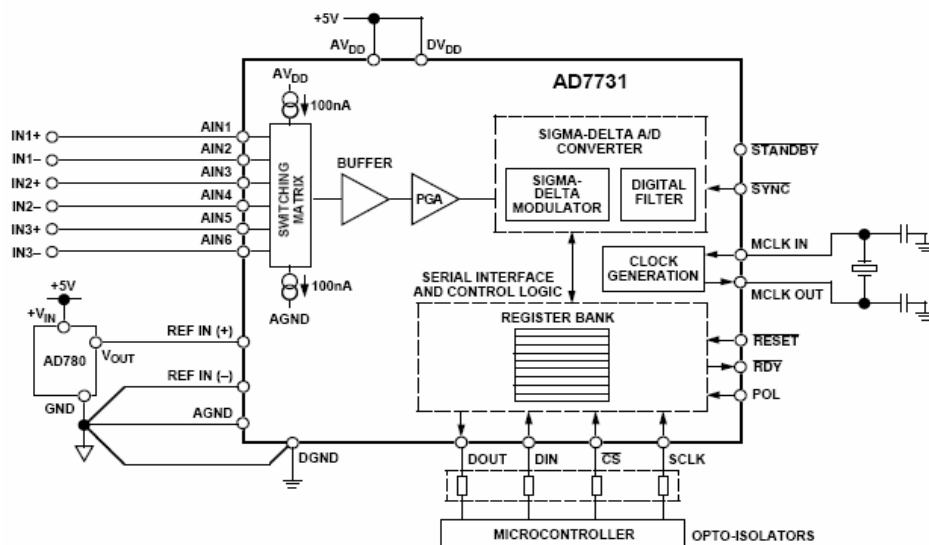


Figura 46. Integrado AD7731 como bloque de adquisición.

Este componente es un sistema integrado, formado internamente por un convertidor A/D, tipo Σ - Δ de 24 bits (bipolar o unipolar), amplificador de ganancia variable, filtro digital programable, interfaz de comunicación serial y microcontrolador, el cual se encarga de realizar las transacciones internas del dispositivo. El uso de este componente trae ventajas como la reducción considerable de las dimensiones de la tarjeta del prototipo, reducción de la cantidad de componentes a utilizar, posee mejores características de conversión A/D en

comparación a las del convertidor A/D del microcontrolador utilizado (PIC16F877A), reducción de los efectos de segundo orden en el comportamiento de los elementos usados de manera discreta (como amplificadores, resistencias, condensadores). Adicionalmente, disminuye considerablemente el costo del prototipo (alrededor de \$17¹⁷ por pieza) comparado con el costo del bloque adecuador (alrededor de \$23¹⁷ por pieza).

Atendiendo a los cambios tecnológicos y a la realidad de periféricos de los computadores personales actuales, cuya tendencia es la eliminación paulatina, pero definitiva de puertos de comunicación serial para ser sustituidos por puertos USB¹⁸, se hace la recomendación de implementar las próximas versiones del sistema A.F.E. suplantando el microcontrolador PIC16F877A por el microcontrolador PIC18F2450 el cual posee un módulo de comunicación serial de tecnología USB incorporado. Además de esto, el microcontrolador tiene un módulo de comunicación USART que podría servir para manejar al bloque de adquisición AD7703 antes mencionado ya que éste posee protocolo serial para el intercambio de datos de conversión.

En lo que respecta al bloque de detección de traba, el mismo puede ser mejorado considerablemente al implementarse con otra filosofía de diseño, de tal manera hacerlo independiente de circuitos con concepción analógica. La modificación propuesta consta de un circuito digital óptico, (implementado con 2 opto transmisores y 2 opto receptores), los cuales se ubicarán en una rueda agujereada con cierto criterio (la distancia entre agujeros debe permitir resolver la misma cantidad de grados por paso que el motor utilizado). Esta rueda a su vez estará vinculada mecánicamente (fija) al eje del motor de paso. Al realizar la búsqueda del tope mecánico del sistema, el microcontrolador autorizará un pulso para permitir el movimiento angular del eje del motor, seguidamente, el microcontrolador verificará el estado (Encendido-Apagado) de los opto receptores y lo comparará con el valor de estado previo al pulso de paso. De no ocurrir cambios en los opto receptores, se

¹⁷ Precio unitario extraído de la página www.Digikey.com a la fecha

¹⁸ Por sus siglas en inglés Universal Serial Bus. Bus serial universal.

decidirá entonces que el sistema ha alcanzado el tope mecánico. Como puede apreciarse, el detector de traba se hace independiente de señales analógicas, es insensible a ruido del ambiente y prácticamente inmune a corrimientos por temperatura de los componentes que lo conforman, mejorando sustancialmente (en teoría) su comportamiento y capacidad de decisión. A continuación una vista frontal y de perfil del sistema óptico para la detección de traba, incluyendo el disco codificado y los sensores. Ver figura 47.

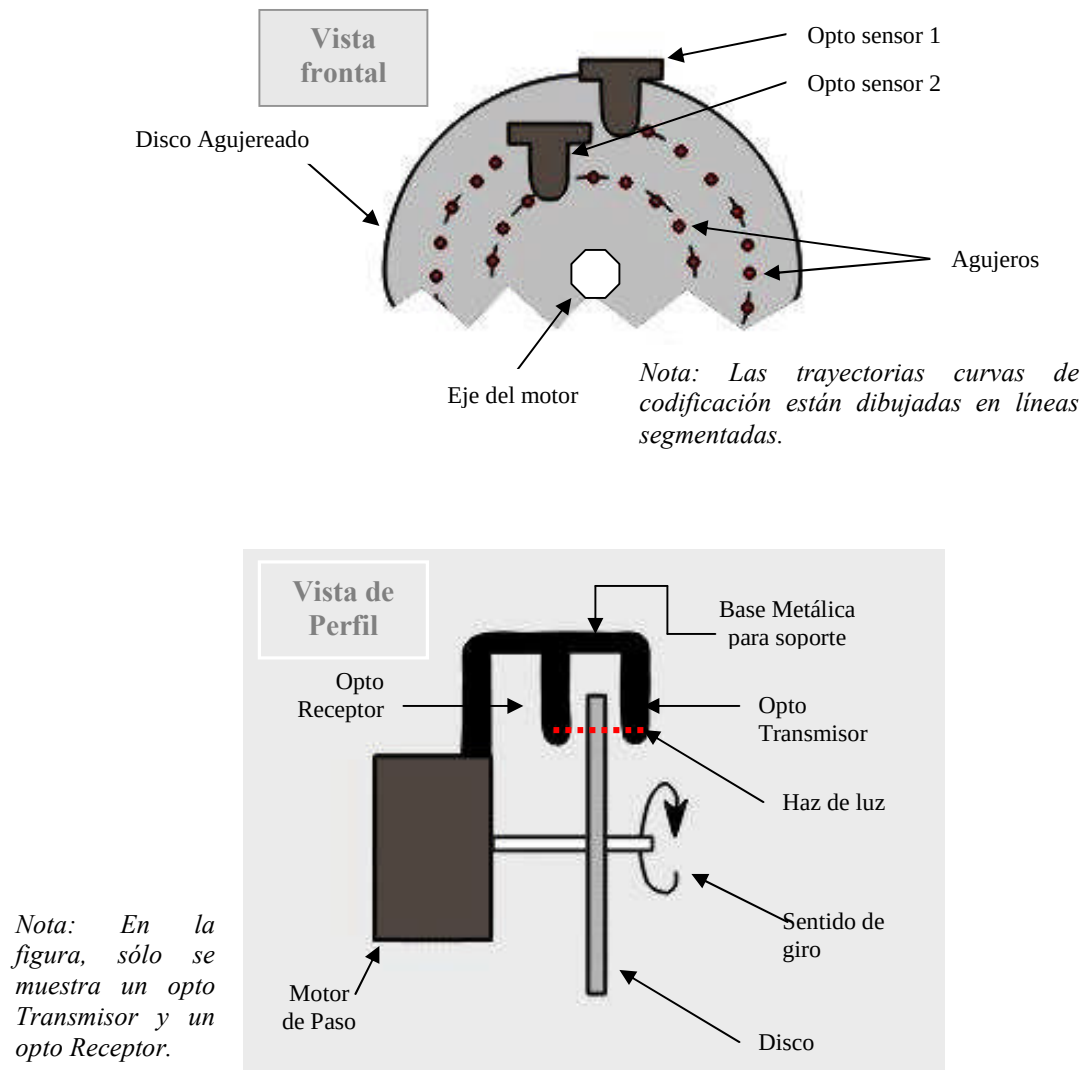


Figura 47. Vista frontal y de perfil del disco codificado para detección de traba del motor de paso y su implementación física en el sistema de accionamiento mecánico del prototipo A.F.E.

BIBLIOGRAFÍA

Libros.

Kenjo, Takashi; Sugawara, Akira. "STEPPING MOTORS and their microprocessor controls", Clarendon Press, Oxford 1994.

Kuo, Benjamin C. "Step Motors", West Publishing CO, St Paul. New York. Boston. Los Angeles. San Francisco.

Milá de la Roca, José M. "Diseño de Equipo Electrónico", Editorial Ed-mil, Caracas Venezuela 2002.

Navarro D, Héctor A. "Instrumentación Electrónica Moderna", Editorial Innovación Tecnológica, Facultad de Ingeniería, Universidad Central de Venezuela, Caracas Venezuela 1995.

Kernighan W, Brian.; Ritchie Dennis M. "El Lenguaje de Programación C" , Editorial Prentice Hall Hispanoamericana, México 1991.

Walkenbach, John, "Excel 2000 Programming for Dummies," IDG Books Worldwide, Inc., Foster City, CA., 1999.

Internet.

Microchip Technology, < <http://www.microchip.com> >

TI Semiconductors, < <http://www.ti.com/> >

Analog Devices Inc, < <http://www.analog.com/> >

Johanson Dielectrics, < <http://www.johansondielectrics.com/> >

National Semiconductors, < <http://www.national.com/> >

Maxim IC, < <http://www.maxim-ic.com/> >

Intersil, < <http://www.intersil.com/cda/home/> >

Semiconductor Equipment, < <http://www.onsemi.com/PowerSolutions/home.do> >

Buscador de hojas de Datos, < <http://www.alldatasheet.com/> >

Sitio WEB Sakae , http://www.sakae-tsushin.co.jp/eng_page/top.shtml

Digikey Corporation, < <http://www.digikey.com/> >

Hi TECH software, < <http://www.htsoft.com/> >

Microsoft Corporation, < <http://www.microsoft.com/> >

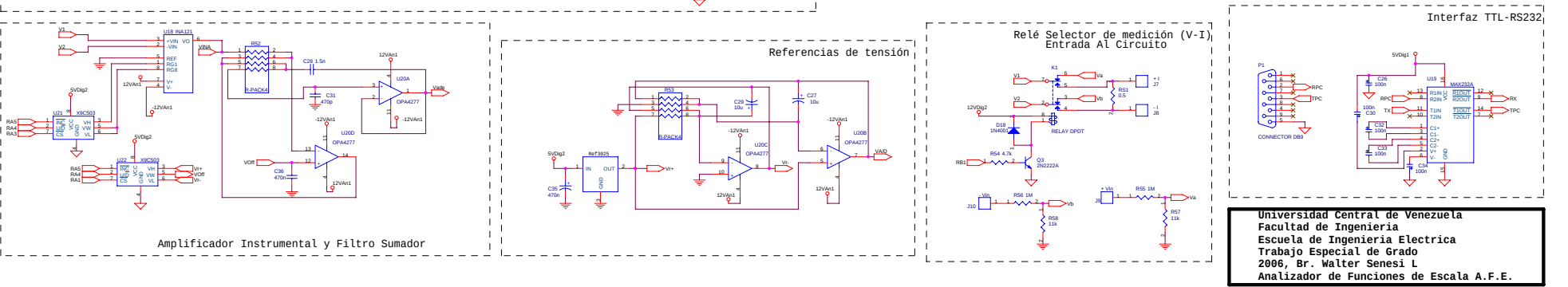
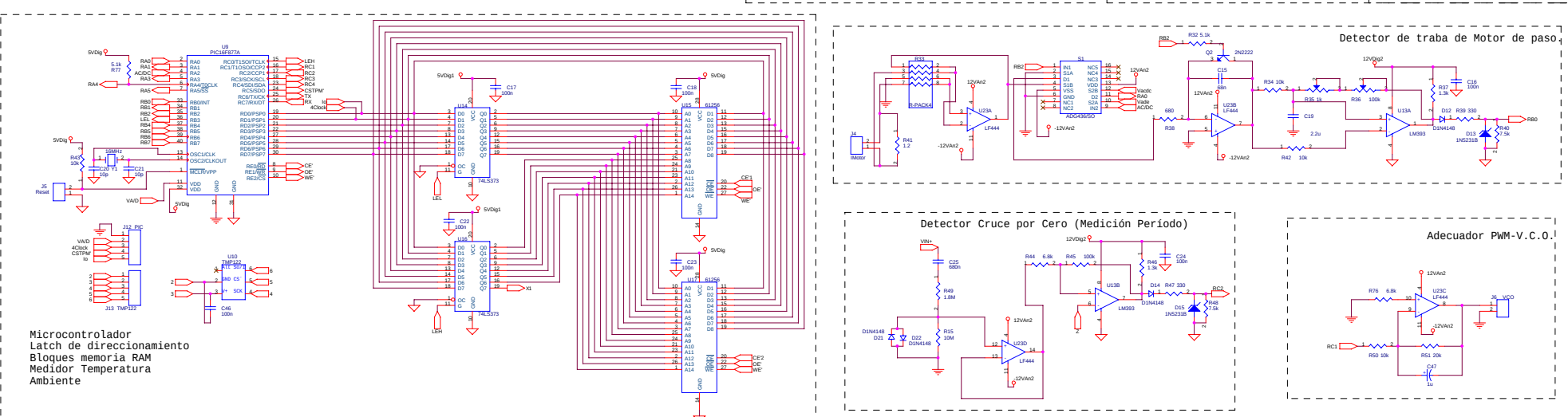
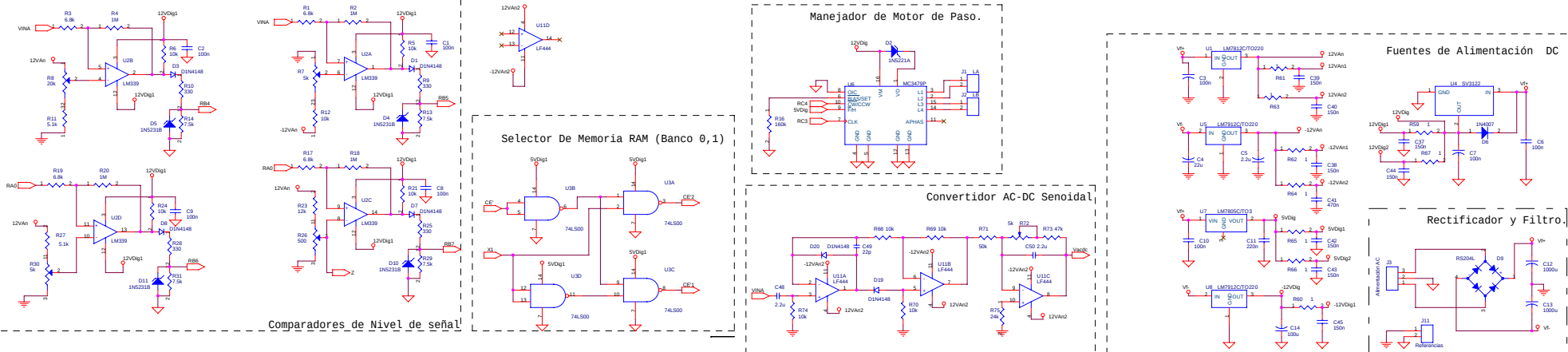
Página de recursos para Programación, < <http://www.elguruprogramador.com.ar/> >

Página de recursos para Programación, <http://deadline.3x.ro/sitemap.html>

Página de recursos para Programación, < <http://www.developerfusion.co.uk/vbnet/> >

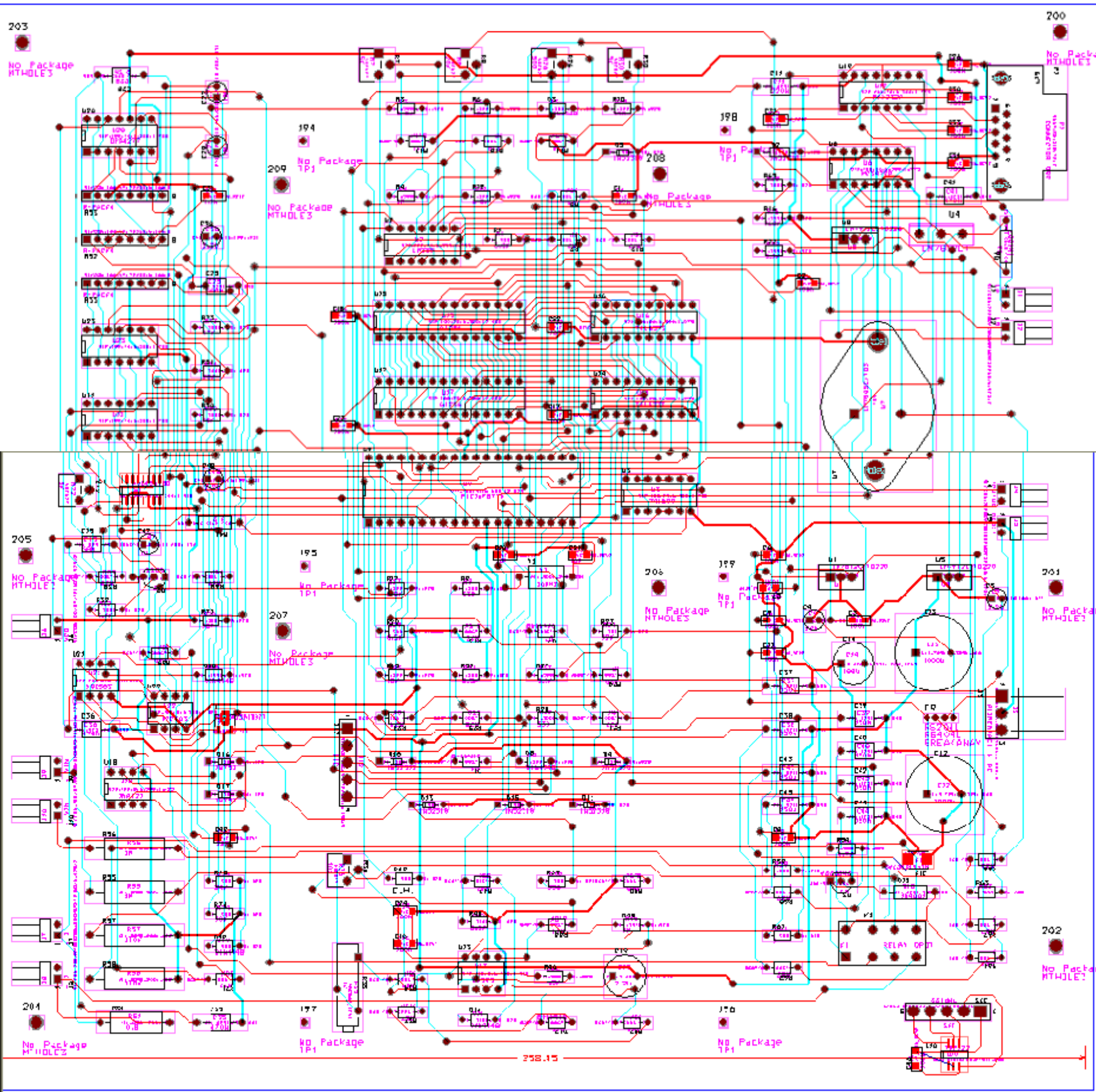
Página de recursos para Programación, < <http://www.lawebdelprogramador.com/> >

Página de recursos para Programación, < <http://www.idg.es/pcworld/> >



Universidad Central de Venezuela
 Facultad de Ingeniería
 Escuela de Ingeniería Eléctrica
 Trabajo Especial de Grado
 2006, Br. Walter Senesi L.
 Analizador de Funciones de Escala A.F.E.

ANEXO #2. Distribución de la interconexión de la tarjeta prototipo A.F.E.



Universidad Central de Venezuela
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
Trabajo Especial de Grado
2006, Br. Walter E. Senesi L.

Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Adquisición de módulo A/D en un punto de accionamiento mecánico (ajuste en un punto)

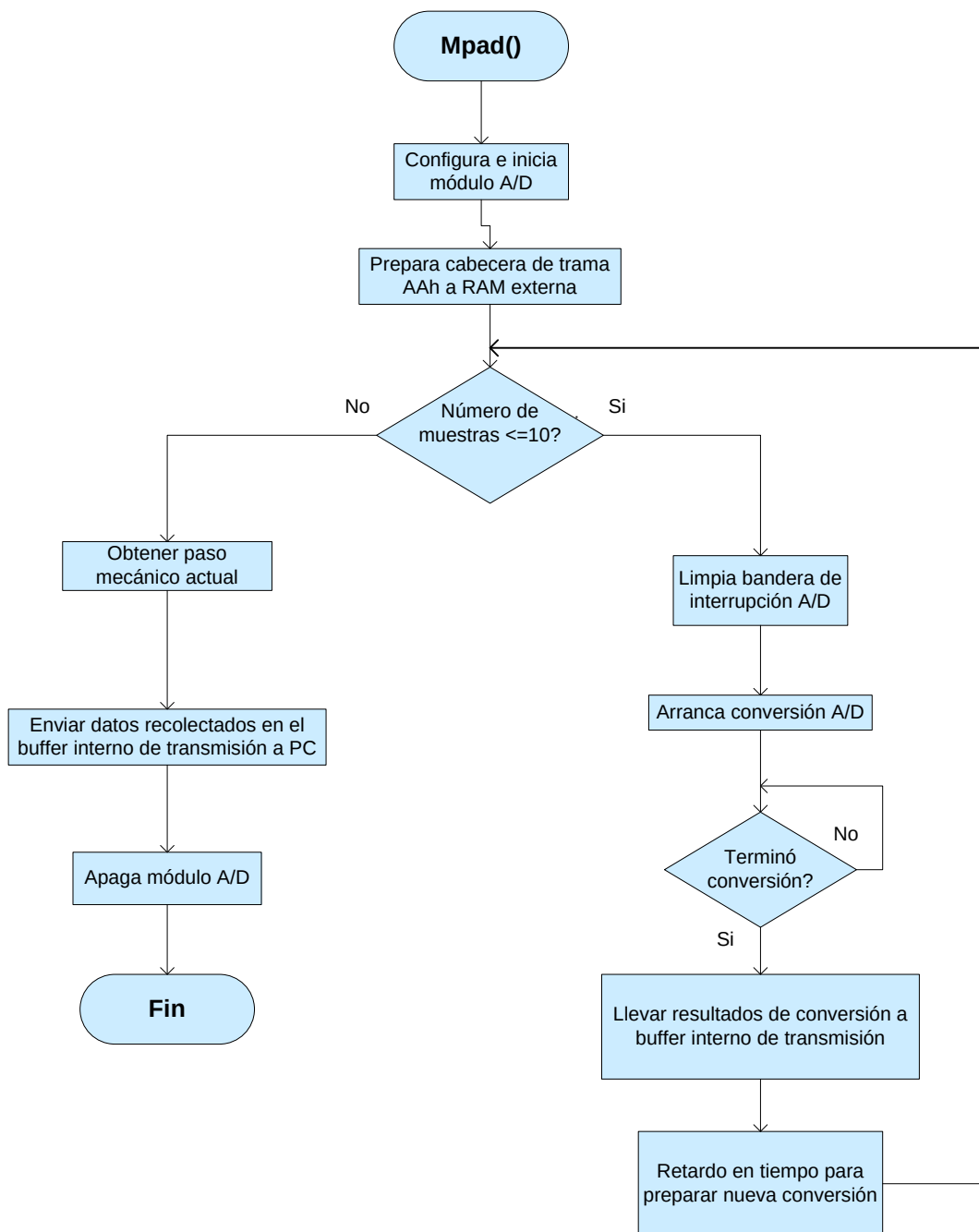


Diagrama de flujo 1

Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Adquisición de período de señal en un punto de accionamiento mecánico
(ajuste en un punto)

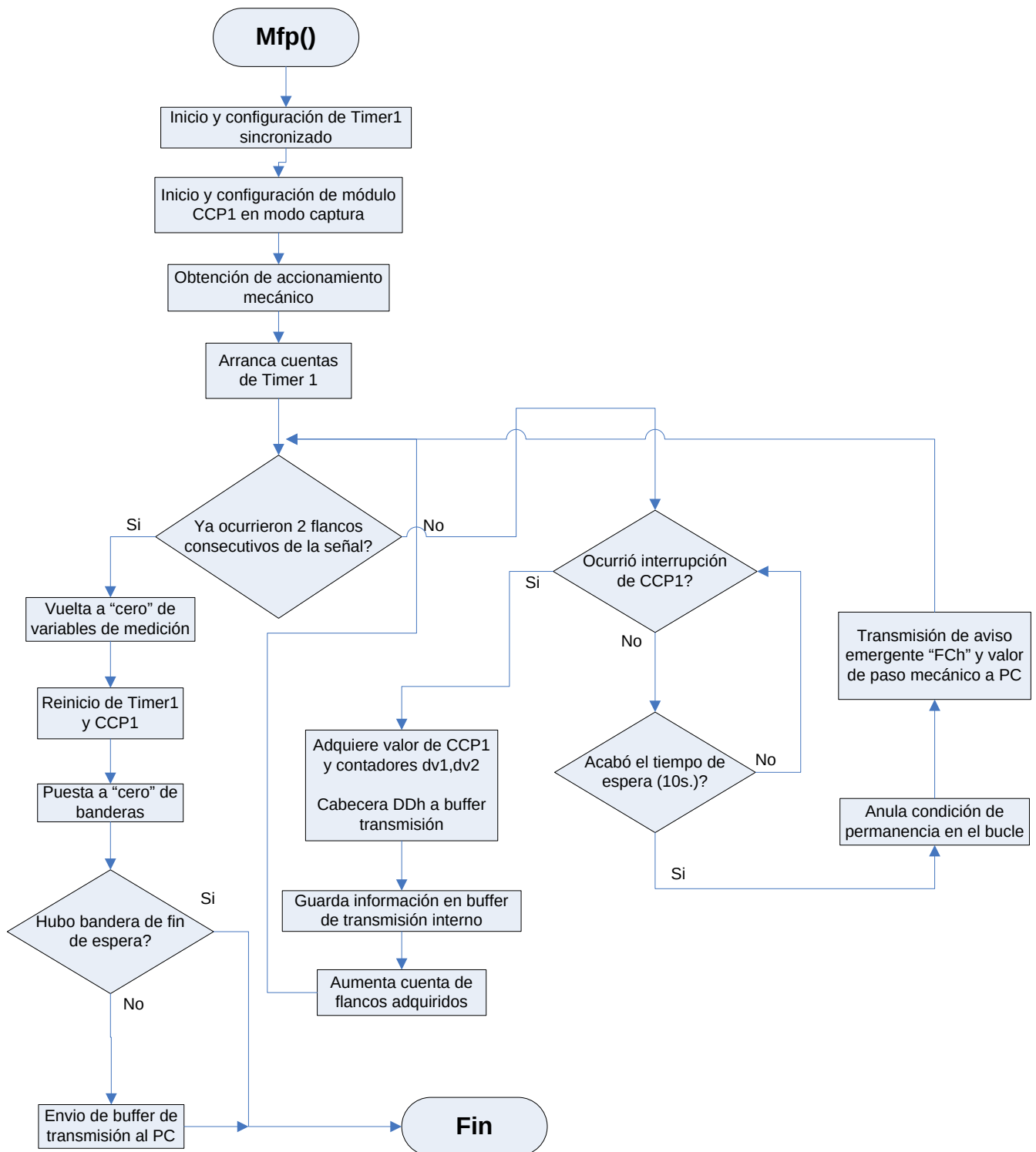


Diagrama de flujo 2

**Anexo #3 Diagramas de Flujo del programa del
microcontrolador PIC del prototipo A.F.E.**
Accionamiento del motor de paso para conseguir tope mecánico

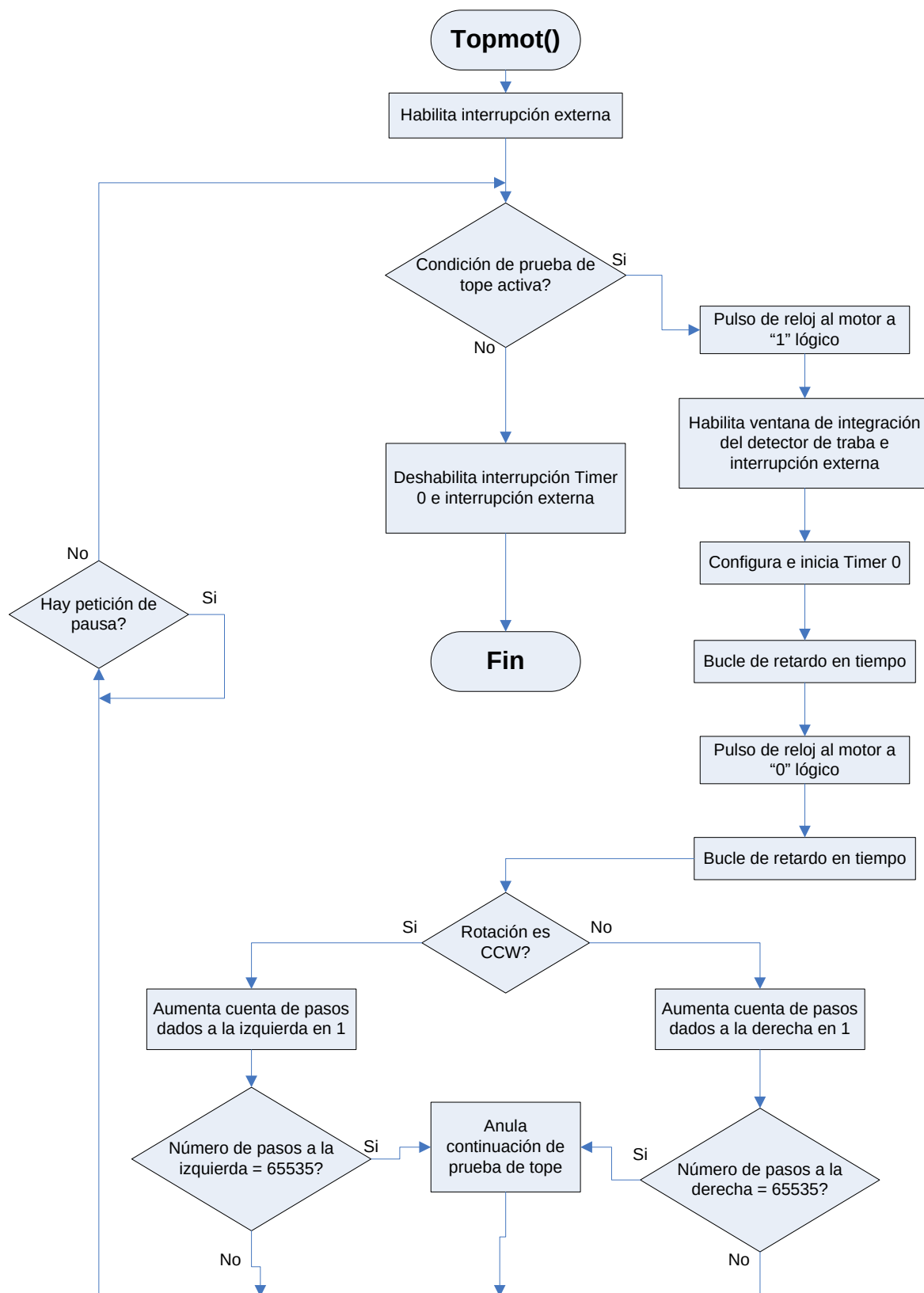


Diagrama de flujo 3

Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Adquisición de datos A/D con sincronización mecánica para caracterizar una función de escala

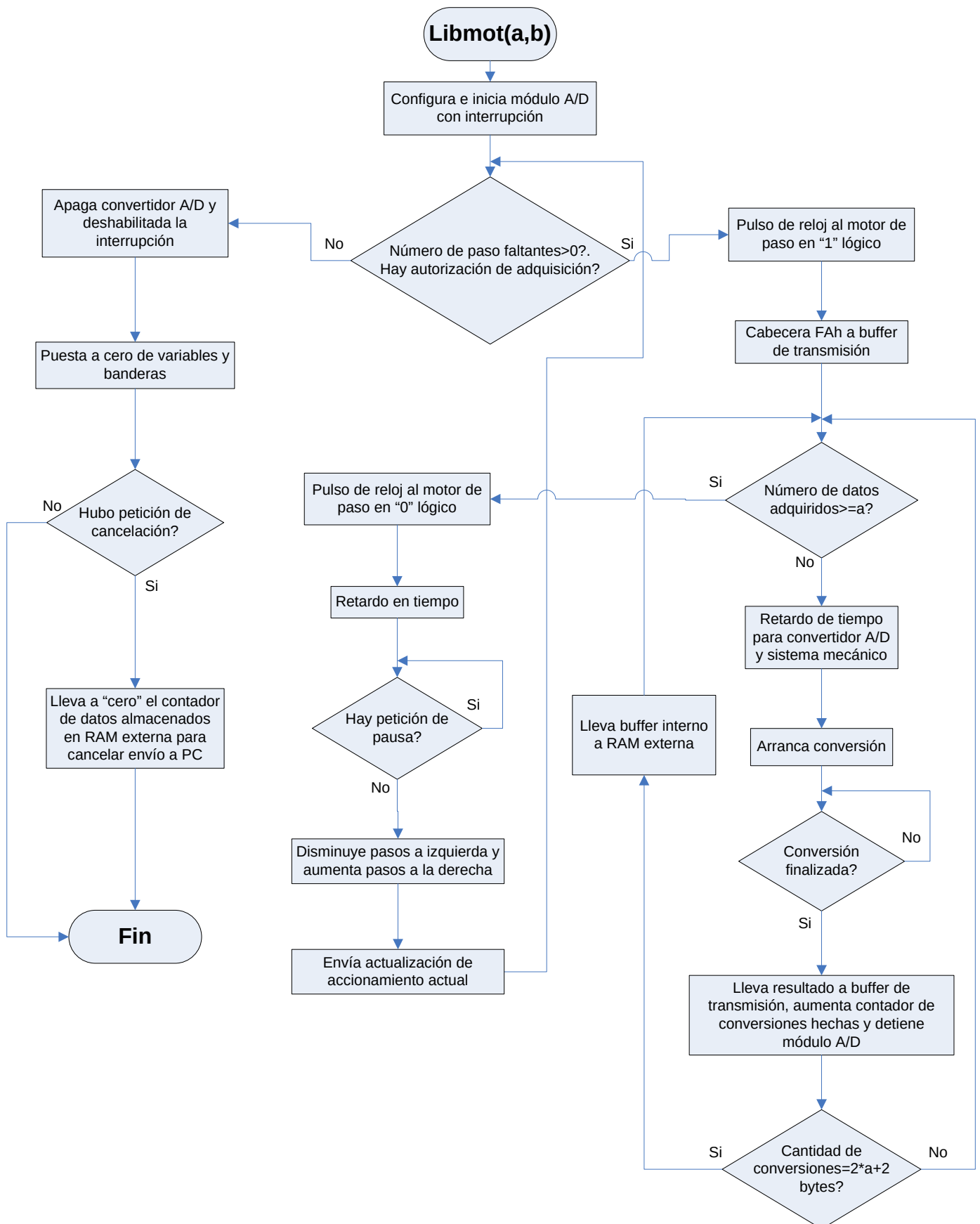


Diagrama de flujo 4

Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Medición de período de señal con sincronización mecánica para caracterizar funciones de escala

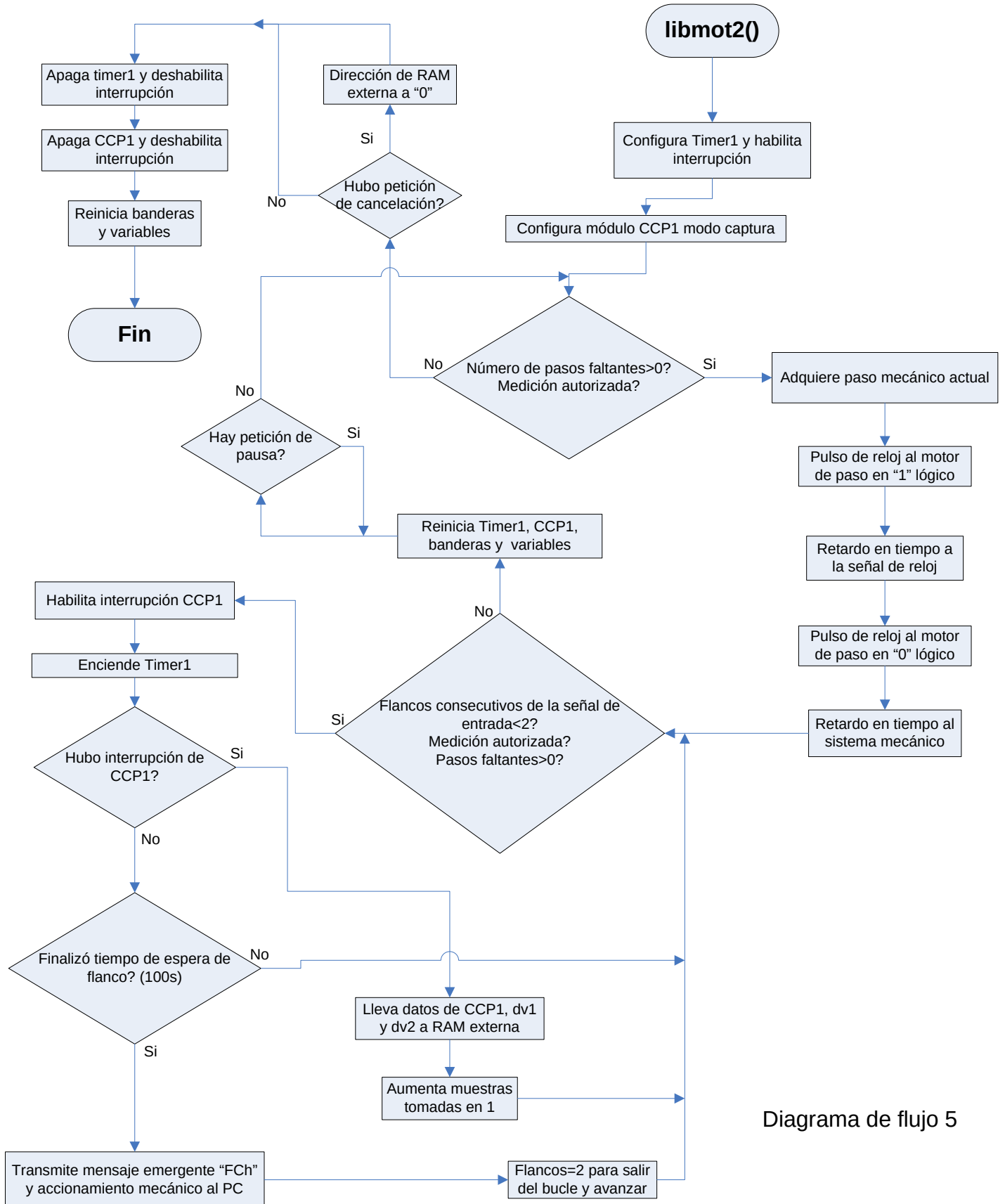


Diagrama de flujo 5

Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Vector de Interrupciones del sistema

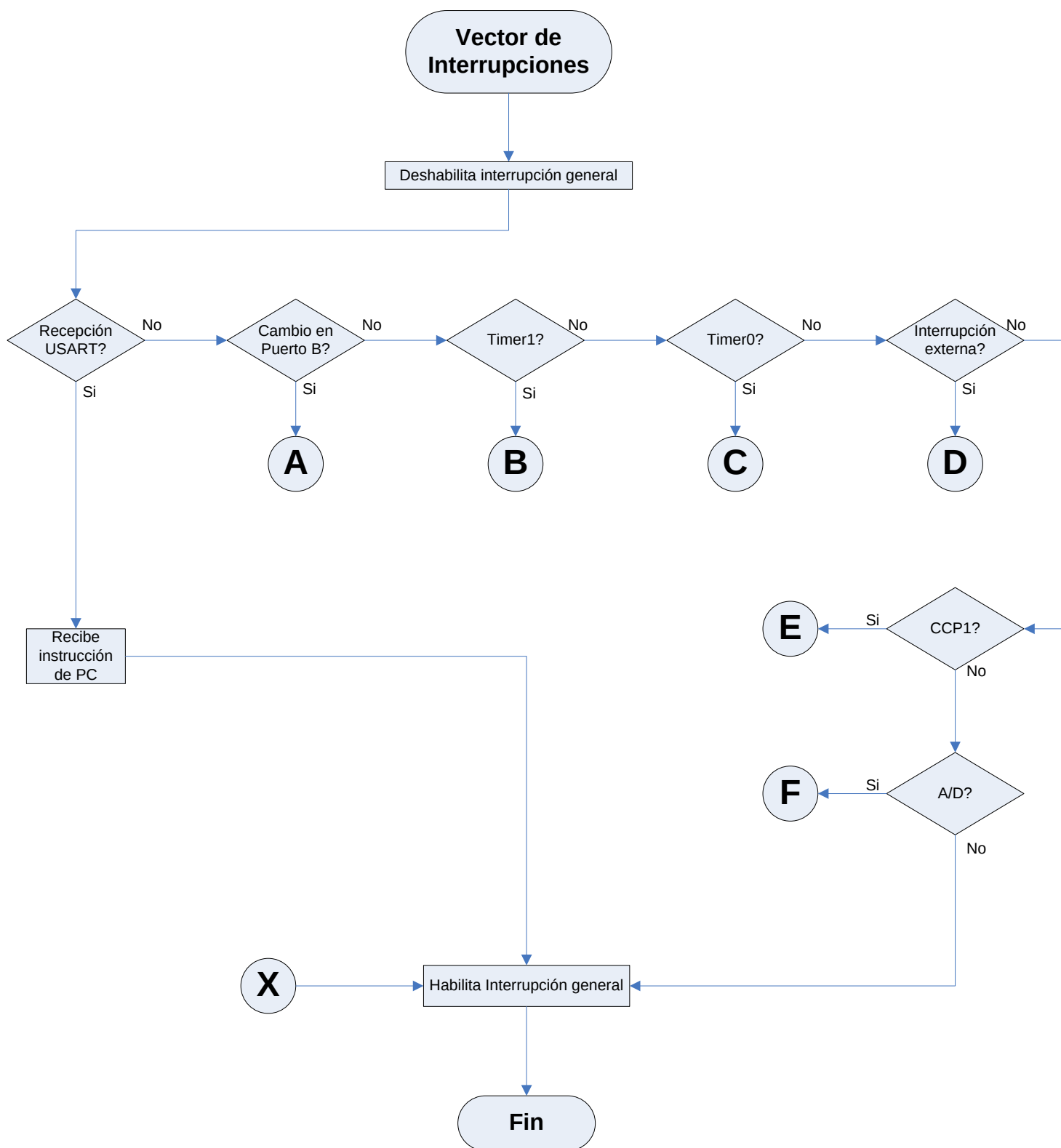
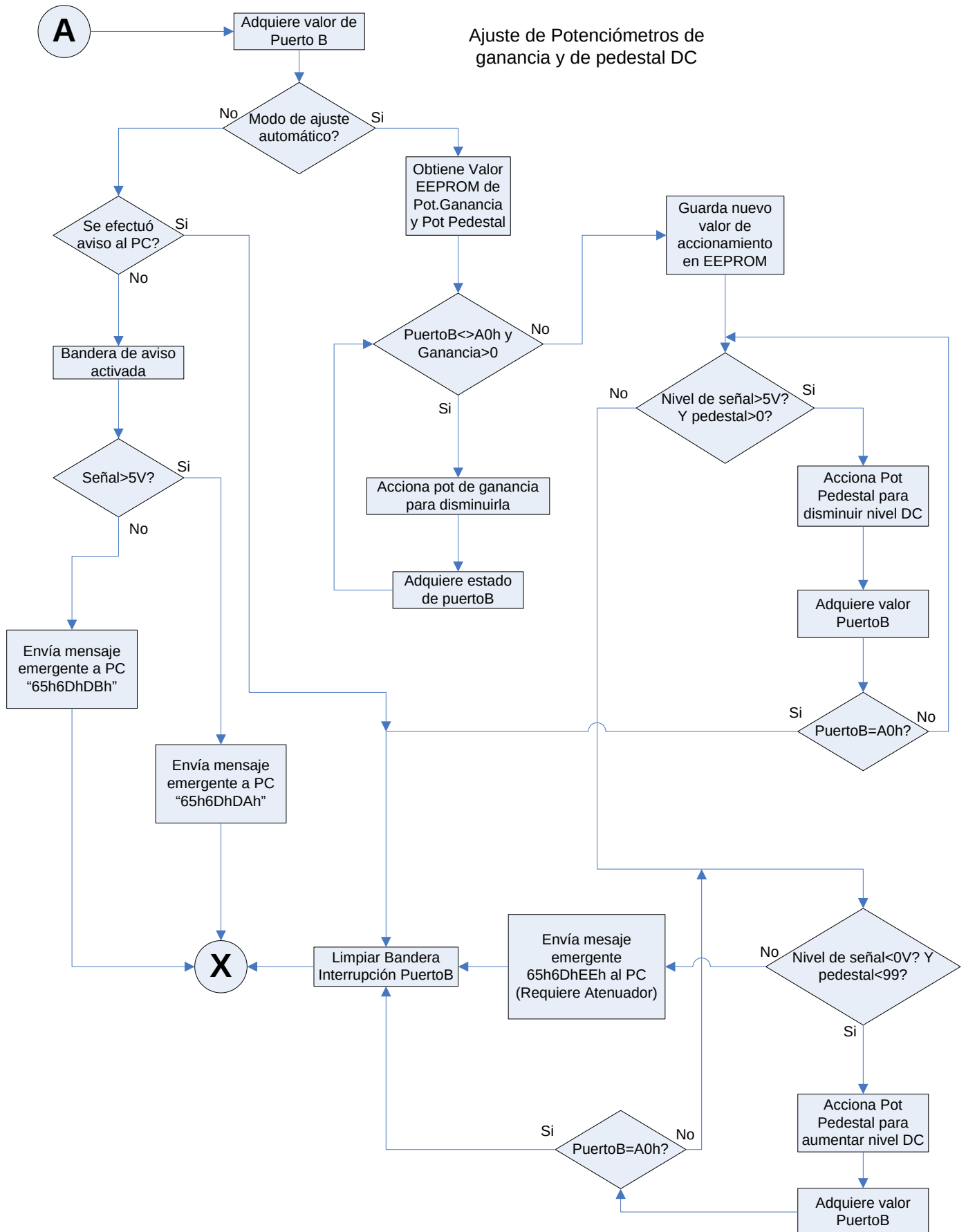


Diagrama de flujo 6

Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

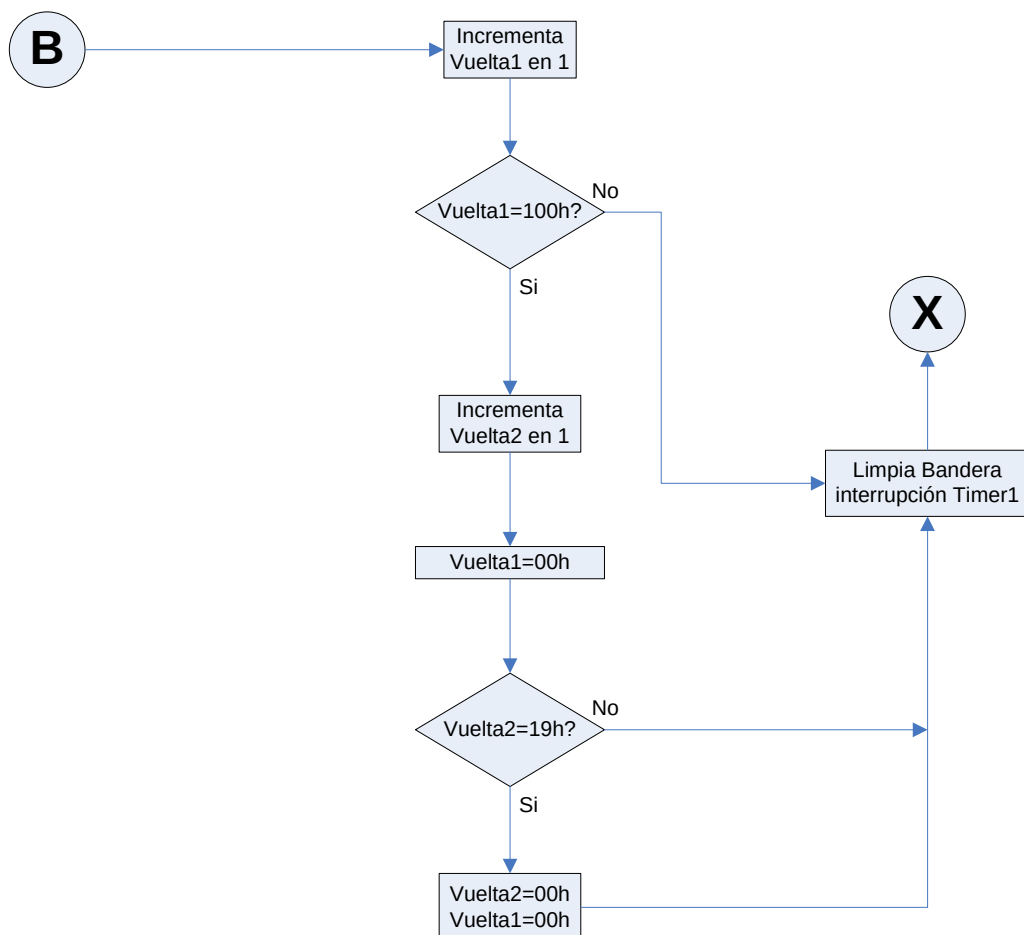
Continuación de vector de Interrupciones del sistema



Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Continuación de vector de Interrupciones del sistema

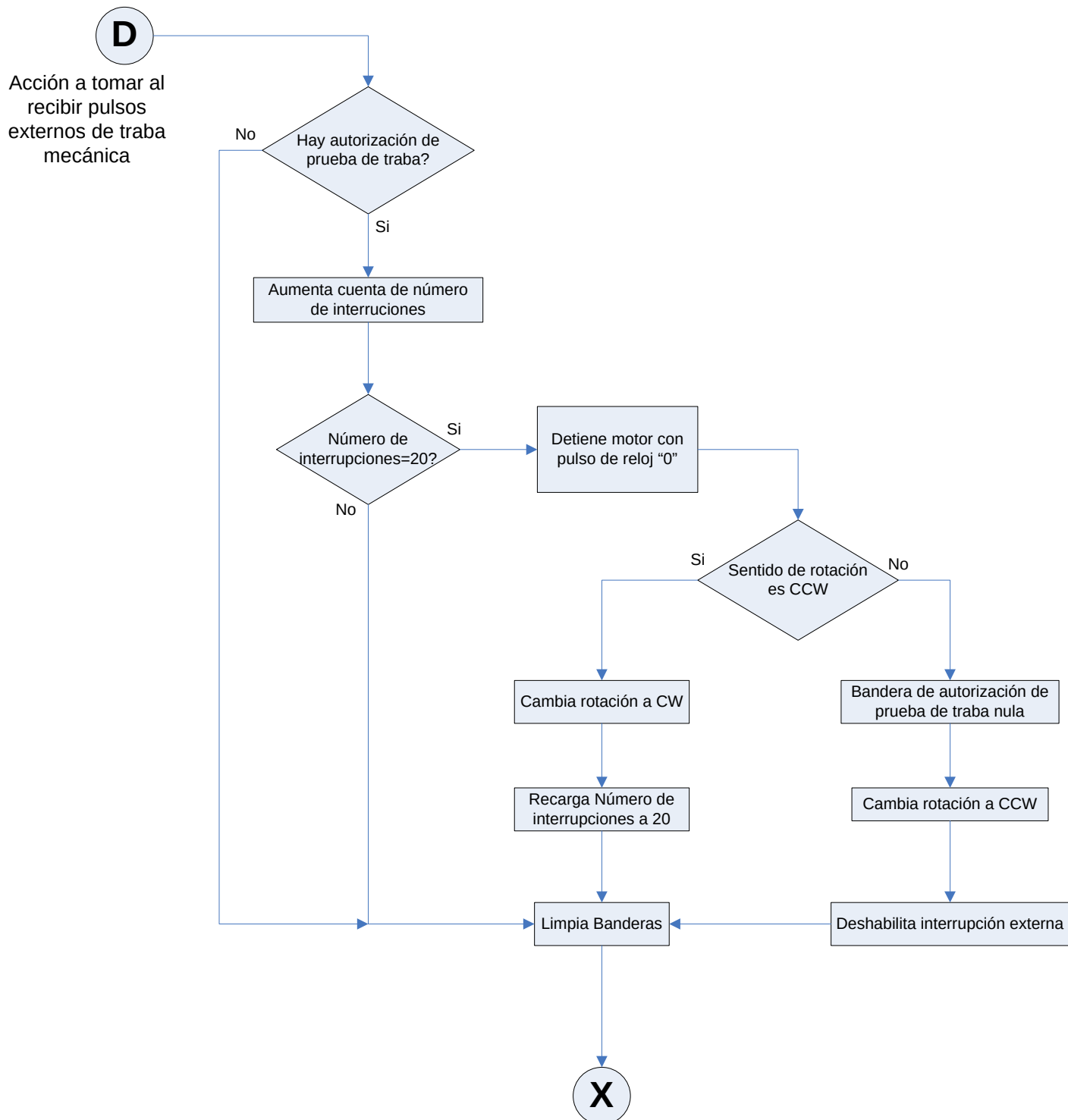
Acción a tomar al desbordar Timer1 (16.383ms) en cada cuenta



Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Continuación de vector de Interrupciones del sistema

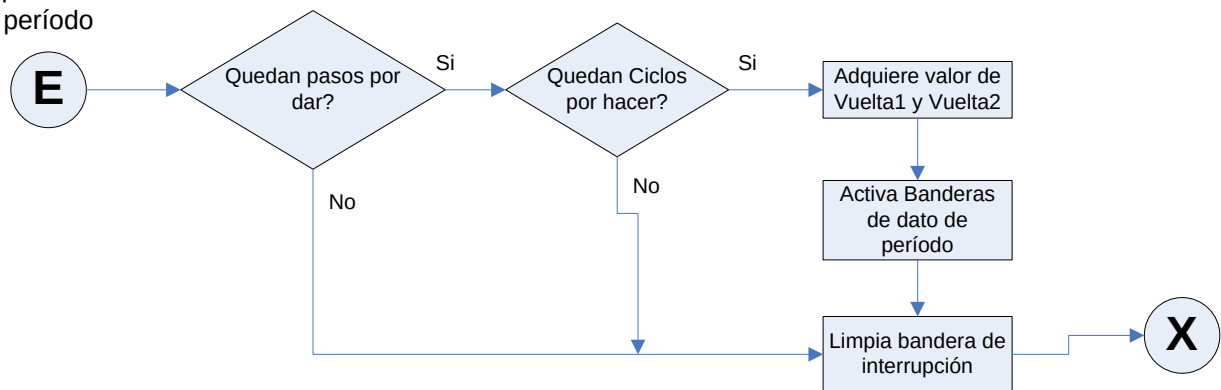
Acción a tomar al desbordar Timer0 (fin de tiempo integración)



Anexo #3 Diagramas de Flujo del programa del microcontrolador PIC del prototipo A.F.E.

Continuación de vector de Interrupciones del sistema

Acción al recibir flanco de señal en adquisición de período



Acción al culminar conversión A/D



Anexo # 4

Programa del microcontrolador Sistema A.F.E. (ANSI C para PIC)

Elaborado por el Br. Walter E. Senesi L para optar por el título de Ingeniero Electricista, mención Electrónica,
de la Universidad Central de Venezuela.

```
#include<pic168xa.h>
#include<pic.h>

//=====
//                Prototipos de Funciones.
//=====

void inicpa();           //inicia puerto A.
void inicpb();           //inicia puerto B.
void inicpc();           //inicia puerto C.
void inicpd();           //inicia puerto D.
void inicpe();           //inicia puerto E.
void inicpots();         //inicializa potenciómetros Digitales a punto original.
void ultipots();         //Mueve potenciómetros a su último set (Grabado en EEPROM).
void inrxtx();           //inicia Configuración TX y RX.
void inicad();           //inicia Convertidor A/D.
void iniccp1();          //inicia Módulo Captura 1.
void inicint();          //inicia vector interrupciones.
void inict1();           //inicia Timer 1.
void inict0();           //inicia Timer 0.
void enviar(unsigned int);//procedimiento en condición de Transmisión.
void niveli();           //procedimiento en condición de niveles de señal F.D.R.
void pulmot(int);        //procedimiento para dar n pulso a motor.
void cancela();          //procedimiento para cancelar petición de movimiento+#pasos dados.
void traduce();          //procedimiento de traducción de data del PC (Comandos).
void libmot(int,int);     //procedimiento Tensión ó corriente de movimiento en condición de prueba
(100hZ).
void libmot2(int,int);    //procedimiento Período ó Frecuencia de movimiento en condición de prueba
(100hZ).
void delay(int);         //Retardo de tiempo programable.
void incpot(int);        //procedimiento para mover cursor del potenciómetro.
void topmot();           //procedimiento para búsqueda de tope mecánico.
void vaciar();           //procedimineto para vaciar data de RAM.
void banderas();         //Iniciación de banderas del programa.
void CRC( unsigned char *,unsigned int );
void ajni();             //ajuste de nivel A/D en modo automático (rauto=1)
                        //Calcula CRC de Datos para enviar sin error.

void iniccp2();          //inicia Módulo PWM. Nivel DC para control de VCO externo.
void inict2();           //Inicia Temporizador 2 para uso del PWM.
void midetmp();          //procedimiento para obtener temperatura del sensor.
void conftmp();          //procedimiento para configurar sensor de temperatura.
void configt();          //procedimiento para enviar configuración total del medidor.
void mpad();             //medición con A/D en 1 punto de la característica.
void mfp();              //medición con CCP1 en 1 punto de la característica.
//NOTA: Los módulos trabai(),incpoti(),topmoti() y niveli() son exactos a los originales //(aquellos que no
poseen la "i" al final del nombre), pero atienden a la rutina de //servicio de interrupciones en vez de al
programa principal. Debieron ser duplicados por //limitación del compilador usado, ya que éste no admite
códigos reentrantes en operación.
```

```
//=====
//          Variables del Programa.
//=====

//Banderas un bit(al banco 0 de RAM):
bit pulso,trad,dper,stop,error,adl,tout,med,rauto,aviso,hd,ma,cn;
//pulso:Estado de Búsqueda de tope del motor.
//trad:Petición de traducción de conjunto de comandos RS232.
//dper:Hubo Atención a medición de período de señal.
//stop:Bandera para detener el sistema en espera de continuación por parte del PC.
//error:Bandera indicadora de error en trama de datos transmitida.
//adl:Bandera indicadora de conversión A/D fue realizada.
//tout:Bandera indicadora que hubo sobrepasamiento de 100s. en espera de algún evento.
//hd:Bandera indicadora de que hubo datos de medición de período.
//med:Bandera autorizadora de mediciones A/D y período.
//rauto:Bandera para Acondicionamiento automático de nivel AC y DC de señal de entrada.
//aviso:Bandera indicadora de haber hecho el aviso una vez (Uso de atenuador).
//ma:autorización de realizar medición en un punto determinado(pasos por muestra).

//Variables de 8 bits(al banco 0 de RAM):
unsigned char d1=0,d2=0,pde=1,vm1=0,vm2=0,vp=0,x=1,y=1,ndato=1,pds=1,
valx=0,pasol=0,pasoh=0,CCRCHI=0xFF,CCRCLo=0xFF,tmpl=0x00,tmph=0x00,vuelta2=0,
psck=1,dv1=0,dv2=0;
//d1:primer dato de la trama de comandos del PC a PIC.
//d2:segundo dato de la trama de comandos del PC a PIC.
//pde:Puntero de dato de entrada (organizador).
//vm1:Variable que contiene valor actual del Potenciómetro de Offset.
//vm2:Variable que contiene valor actual del Potenciómetro de Ganancia.
//vp:Valor del puerto B.
//ndato:Cantidad de datos que de tomarán por paso dado por el motor.
//pds:Puntero de datos de salida.
//valx:Variable para manipulación del cursor de los potenciómetros digitales.
//vuelta2:Acumulador parte alta de tiempo transcurrido al activar Timer1.
//vuelta1:Acumulador parte baja de tiempo transcurrido al activar Timer1.
//pasol:Variable que contiene parte baja del paso actual del sistema.
//pasoh:Variable que contiene parte alta del paso actual del sistema.
//CCRCHI:Valor parte alta de CRC calculado para una trama.
//CCRCLo:Valor parte baja de CRC calculado para una trama.
//tmpl:Valor parte baja de lectura de temperatura del sistema.
//tmph:Valor parte alta de lectura de temperatura del sistema.
//psck:Contador de pulsos de clock al medidor de temperatura.

unsigned int pdb=1,ni=0,miya=0;
//pdb:Puntero de datos del buffer de intercambio (txbuffer).
//ni:Contador del número de interrupciones por traba del eje del motor, a partir de
//un # de 50(CW) y 25(CCW)interrupciones permite acción del sistema sobre el motor.
//Variables de 16 bits (al banco 1 de RAM):
//miya condición para tomar medición en ese punto de accionamiento.
bank2 unsigned long adr=0,ciclo=0,nupasd=0,vuelta1=0,nupasi=0,vcr=0,bc=0,tmpreg=0,pwm=0,sadr=0;
//adr:Puntero de dirección de almacenamiento en RAM externa desde el PIC.
//ciclo:Cantidad de veces que se repetirá una medición+movimiento del motor.
//nupasd:Número de pasos dados a la derecha al buscar tope mecánico.
//nupasi:Número de pasos dados a la izquierda al buscar tope mecánico.
//vcr:Valor de configuración del medidor de temperatura.
//bc:Bit de configuración al medidor de temperatura.
```



```

inicpc();
inicpd();
inicpe();
banderas();
inicpots();
inrxtx();
conftmp();
inicint();
TXEN=1;
TXREG=0x55;
while(!TRMT);
TXREG=0x43;
while(!TRMT);
TXREG=0x56;
while(!TRMT);
TXREG=0x57;
while(!TRMT);
TXREG=0x53;
while(!TRMT);
TXREG=0x4C;
while(!TRMT);
TXEN=0;
while(1)
{
    traduce();           //intérprete de comandos recibidos por PC.
    vaciar();           //si hay dato en RAM, se saca a PIC.
}
}

//=====
//          Puerto A Iniciación.
//=====
void inicpa()
{
    PCFG3=1;           //RA5..RA1 Digitales, RA0 Analógica.
    PCFG2=1;
    PCFG1=1;
    PCFG0=0;
    TRISA=0x01;        //RA5..RA1 salidas, RA0 Entrada.
    PORTA=0xFF;        //Todas las salidas a "1".
}

//=====
//          Puerto B Iniciación.
//=====
void inicpb()
{
    TRISB=0xF1;        //RB7..RB4,RB0 entradas comparadores e int.; RB3..RB1
    Salidas.           //Encendidas la Pull Up internas del micro.
    RBPU=1;
    TRISB2=0;
    RBPU=1;
    RB2=1;
    RB3=0;             //RB3..RB1->"0", los demás en "1".
    RB2=1;
    RB1=0;
}
}

```

```

//=====
//          Puerto C Iniciación.
//=====
void inicpc()
{
    TRISC=0x84;          //RC7 y RC2 entradas, RC6,RC5,RC4,RC3,RC1,RC0 Salidas.
    PORTC=0b00100000;    //Salidas en "0" menos RC5="1".
}
//=====
//          Puerto D Iniciación.
//=====
void inicpd()
{
    TRISD=0x00;          //Todos los pines de salida.
    PORTD=0x00;          //iniciados en "0".
}
//=====
//          Puerto E Iniciación.
//=====
void inicpe()
{
    TRISE=0x00;          //todos los pines de salida.
    PORTE=0xFF;          //iniciados en "0".
}
//=====
//Iniciación Módulo Capture/Compare/PWM 2 Modo PWM.
//=====
void iniccp2()
{
    CCPR2L=0x00;         //Duty Cycle variable por usuario.
    CCP2CON=0b00001100; //modo "PWM" periodo de 976.56Hz.
}
//=====
//          Iniciación Timer 2.
//=====
void inict2()
{
    TOUTPS3=0;           //Frecuencia 976.56Hz.
    TOUTPS2=0;           //Sin post escalador.
    TOUTPS1=0;
    TOUTPS0=0;
    T2CKPS1=1;
    T2CKPS0=1;
    TMR2ON=0;            //TIMER2 apagado.
    PR2=0xFF;            //Valor de comparación 255h.
    TMR2IF=0;
    TMR2IE=0;
}
void banderas()
{
    trad=0;
    dper=0;
    pulso=0;
    stop=0;
    error=1;
    adl=0;
}

```

```

    tout=0;
    aviso=0;
    med=0;
    rauto=1;
    hd=0;
    cn=1;
}
//=====
//      función Lectura Temperatura.
//=====
void midetmp()
{
    TRISD1=0;                //Pin RD1 de salida. pulsos clock.
    TRISD0=1;                //Pin RD0 de entrada.
    RD1=0;                   //Prepara clock en 0.
    RC5=0;                   //Selecciona dispositivo.
    delay(10);
    psck=1;                  //Inicia contador extractor.
    while(psck<=16)          //Extrae 16 bits serial...
    {
        delay(10);           //Tiempo de puesta.
        RD1=1;               //Da pulso de clock.
        tmpreg=tmpreg<<1;    //Rota 1 bit a izq al registro de temp.
        delay(50);           //Retardo para puesta estable.
        tmpreg=(tmpreg|(PORTD&0x01)); //Extrae el bit de respuesta.
        psck++;              //Aumenta contador de bits.
        RD1=0;
        delay(40);           //Retardo para puesta estable.
    }
    RC5=1;                   //Deselecciona dispositivo.
    tmp1=tmpreg&0xFF;        //Envia la información recolectada
    tmp2=(tmpreg>>8)&0xFF;   //del sensor a USART. PIC->PC.
    TRISD0=0;
}
//=====
//      Función configuración sensor temperatura.
//TMed:cada 8seg,Res:0.25°C,6 fallas->alarma,Pol:0=activo
//Modo:Comparador con histéresis.Tlow=75°C,THigh=80°C.
//=====
void conftmp()
{
    psck=1;
    TRISD0=1;                //Pin RD0 de entrada.
    TRISD1=0;
    RD1=0;                   //Pulso de clock en 0.
    RC5=0;                   //Selección de dispositivo.
    delay(10);
    vcr=0x0DC2;              //Valor de configuración
    while(psck<=32)          //contará 32 bits, los 16 primeros se
    {                          //desecharán (valor lectura
temperatura).
        if(psck<=16)         //verifica los 16 primeros bits.
        {
            RD1=1;           //Pulsos de clock...
            delay(50);
            RD1=0;

```



```

        delay(50);
    }
    if(psck>=17)                //al llegar los siguientes 16 bits.
    {
        TRISD0=0;                //Pin RD0 de Salida.
        bc=vcr&0x8000;           //se filtra el MSB de la rotación.
        if(bc==0x8000)           //verifica si es 1 ó 0.
        {
            RD0=1;
        }
        if(bc==0x0000)
        {
            RD0=0;
        }
        vcr=vcr<<1;              //Rota 1 bit a izq. a la palabra de
                                //configuración.
        delay(10);
        RD1=1;
        delay(50);                //Pulsos y retardo de clock..
        RD1=0;
        delay(50);
    }
    psck++;                      //aumenta contador.
}
TRISD0=0;                      //Configura RD0 de salida.
RC5=1;                          //Deselecciona dispositivo.
}
//=====
//          Iniciación USART asíncrono (RX y TX).
//=====
void inrxtx()
{
    SPBRG=0x67;                  //9600 baud @ 16MHz.
    TXSTA=0b00000110;           //Configuración de registros Tx y RX.
    RCSTA=0x90;                  //habilitación de interrupciones TX y RX.
    SPEN=1;
    CREN=1;
}
//=====
//          Iniciación Potenciómetros digitales PDC y PG.
//=====
void inicpots()
{
    PEIE=0;
    TOIE=0;
    RBIE=0;
    INTE=0;
//-----
//Pot.dig Ganancia.
//-----
    RA4=0;                      //decremento al mínimo de ganancia.
    RA1=1;
    RA3=0;                      //hasta encontrar tope inferior.
    delay(10);
    incpot(101);
    RA5=0;

```

```

    RA3=1;
    EEPROM_WRITE(0x02,0);//salva la nueva posición en EEPROM.
    EEIF=0;
//-----
//Pot.dig.Offset.
//-----
    RA4=0;                //decremento al mínimo de offset posible.
    RA1=0;                //hasta encontrar tope inferior.
    delay(10);
    incpot(101);
    RA4=1;                //incremento de offset.
    incpot(73);           //A aprox. 2,5V de offset.
    EEPROM_WRITE(0x03,73);//salva la nueva posición en EEPROM.
    EEIF=0;
    RA5=0;
    RA1=1;
    PEIE=1;
    T0IE=1;
    RBIF=0;
    if (rauto)
    {
        cn=1;
        RBIE=1;
    }
    INTE=1;
}
//=====
//Potenciómetros digitales PDC y PG a último Set Point.
//=====
void ultipots()
{
    PEIE=0;
    T0IE=0;
    RBIE=0;
    INTE=0;
    inicpots();
//-----
//Pot.dig.Ganancia.
//-----
    vm2=EEPROM_READ(0x02);
    if (vm2<10)
    {
        RA4=0;                //decremento al mínimo de ganancia.
        RA3=0;                //hasta encontrar tope inferior.
        delay(10);
        vm2--;
        incpot((10-vm2));
        vm2++;
        RA5=0;
        RA3=1;
    }
    if (vm2>10)
    {
        RA4=1;                //incremento al mínimo de ganancia.
        RA3=0;                //hasta encontrar tope inferior.
        delay(10);
    }
}

```

```

        vm2--;
        incpot((vm2-10));
        vm2++;
        RA5=0;
        RA3=1;
    }
//-----
//Pot.dig.Offset.
//-----
    vm1=EEPROM_READ(0x03);
    if (vm1<49)
    {
        RA4=0;                //decremento al mínimo de ganancia.
        RA1=0;                //hasta encontrar tope inferior.
        delay(10);
        vm1--;
        incpot((49-vm1));
        vm1++;
        RA5=0;
        RA1=1;
    }
    if (vm1>49)
    {
        RA4=1;                //incremento al mínimo de ganancia.
        RA1=0;                //hasta encontrar tope inferior.
        delay(10);
        vm1--;
        incpot((vm1-49));
        vm1++;
        RA5=0;
        RA1=1;
    }
    PEIE=1;
    TOIE=1;
    RBIF=0;
    cn=1;
    RBIE=1;
    INTE=1;
}
//=====
//          Iniciación Módulo Convertidor A/D.
//=====
void inicad()
{
    ADCON0=0b10000000; //A/D,CH0,Fosc/32,Apagado.
    ADCON1=0b10001110; //RA0 entrada Analógica,Justif.der.
                        //Conversión de señal AC.
}
//=====
//          Iniciación Módulo Capture/Compare/PWM 1
//=====
void iniccp1()
{
    CCP1CON=0b00000110; //modo "Capture" Cada 4 flancos de subida,
    TRISC2=1;           //Pin RC2 de entrada a CCP1.
}

```

```

//=====
//          Iniciación Timer 0
//=====
void inict0()
{
    TMR0IF=0;           //Borra bandera de interrupción Timer0.
    TMR0=0x00;          //Reescribe valor de cuenta de Timer0.
    T0CS=0;              //Sincronizado con ciclo interno del micro.
    PSA=0;              //Prescalador asignado a Timer0.
    PS2=0;              //Evolución cada ciclo del micro (250ns).
    PS1=0;              //En sobrepasamiento "FFh" habrá contado 63.75us.
    PS0=0;
    TMR0IE=1;
}
//=====
//          Iniciación Timer 1
//=====
void inict1()
{
    T1CON=0b00000000;   //Timer1 como timer,prescaler 1:1,sin osc, sincroniz.interna.
    TMR1L=0x00;
    TMR1H=0x00;
}
//=====
//función inicializar interrupciones deseadas.
//=====
void inicint()
{
    int i;
    TXEN=0;
    PEIE=1;              //Activa interrupción de periféricos.
    RBIF=0;
    if (rauto)
    {
        cn=1;
        RBIE=1;          //Activa interrupción cambio en PORTB.
    }
    else
    {
        RBIE=0;
    }
    T0IE=0;              //Desactiva interrupción sobrepasamiento en Timer0.
    INTE=0;              //Desactiva interrupción externa detecta estado de traba.
    ADIE=0;              //Desactiva interrupción Dato convertido A/D.
    I=RCREG;
    RCIE=1;              //Activa interrupción Recepción de comando del PC.
    CCP1IE=0;            //Activa interrupción Módulo Capture (medición de período).
    TMR1IE=0;            //Desactiva interrupción sobrepasamiento de timer 1.
    INTEDG=1;            //Activa interrupción externa con flanco de bajada(Pin RB0).
    GIE=1;              //Activa interrupción General.
}

```

```

//=====
//      función cálculo de CRC.
//=====
void CRC(unsigned char *ptxbuffer,unsigned int Longi)
{
//      *ptxbuffer      puntero del arreglo Txbuffer.
//      Longi           Cantidad de Datos a Procesar.
    unsigned pcc;      // Puntero extractor de datos del arreglo.
    CCRCHi=0xFF;       // Iniciación Calculo CRC alto.
    CCRCLo=0xFF;       // Iniciación Calculo CRC bajo.
    while(Longi--)      // hasta haber procesado todos los bytes...
    {
        pcc=CCRCLo^*ptxbuffer++; //Usa Tabla para cálculo de CRC.
        CCRCLo=CCRCHi^TCRCHi[pcc];
        CCRCHi=TCRCLo[pcc];      //Retorna resultado de CRC.
    }
}
//=====
//función para transmitir conversión hecha(PIC->PC)
//=====
void enviar(unsigned int largo)
{
    while(error)
    {
        unsigned char j=1;
        TXEN=1;
        while(j<=largo)      //Envía los datos modo F.I.F.O.hasta vaciar txbuffer.
        {
            while(!TRMT);
            TXREG=txbuffer[j];    //mueve resultado de conversión a puerto USART de
salida.
            j++;
        }
        while(!TRMT);
        TXREG=CCRCLo;            //mueve resultado de CRC LOW a USART de
salida.
        while(!TRMT);
        TXREG=CCRCHi;            //mueve resultado de CRC HI a puerto USART
de salida.
        while(!TRMT);
        stop=0;
        error=0;
        while(stop);            //Espera llegada y respuesta del PC.
    }
    TXEN=0;
}
//=====
//      función para pulsos incremento pot.dig. (inc')
//=====
void incpot(int n)
{
    int i=0;
    while (n>0)                //duración ciclo definido por "n"
    {
        RA5=0;                //pulso "0".
        for (i=0;i<150;i++);    //delay.
    }
}

```

```

        RA5=1;                                //pulso "1".
        for (i=0;i<150;i++);                  //delay.
        n--;                                  //decrementa ciclo.
    }
}
//=====
//función para pulsos de motor a 900hZ (busqueda de tope mecánico).
//=====
void topmot()
{
    int lee;
    lee=PORTB;
    RBIF=0;
    RBIE=0;
    INTF=0;                                //Limpia bandera de traba.
    INTE=1;                                //Habilita aviso de traba.
    while(pulso)                            //Mientras se quiera movimiento...
    {
        int i=0;
        RC3=1;                                //Pulso en "1" al driver motor.
        RB2=0;                                //Habilita inicio de la ventana.(170)
        inict0();                             //Habilita tiempo de ventana de integración.
        for(i=0;i<500;i++);                  //Retardo en tiempo.
        RC3=0;                                //Pulso en "0" al driver motor.
        RB2=0;
        inict0();                             //Habilita tiempo de ventana de integración.
        for(i=0;i<500;i++);                  //Delay.
        if (RC4)                             //Verifica sentido de rotación (CW=0 ó CCW=1).
        {
            izquierda.
            nupasi++;                          //Incremento del Número de pasos Mecánicos dados a la
            if (nupasi==0xFFFF)//Verifica no sobrepasar rango de variable.
            {
                pulso=0;//Anula continuación del ciclo.
            }
        }
        else
        {
            derecha.
            nupasd++;                          //Incremento del Número de pasos Mecánicos dados a la
            if (nupasd==0xFFFF)//Verifica no sobrepasar rango de variable.
            {
                pulso=0;//Anula continuación del ciclo.
            }
        }
    }
    while(stop);
}
TOIE=0;
TOIF=0;
INTE=0;
INTF=0;
if (rauto)
{
    RBIF=0;
    cn=1;
}

```

```

        RBIE=1;
    }
}

//=====
//función interprete de comandos RX de RS232 (PC->PIC)
//=====
void ajni()
{
    while(!EEIF);
    if (rauto)
    {
        int vpx=0;
        vpx=PORTB&0x40;
        RC3=0; //Detiene motor.
        vm2=EEPROM_READ(0x02); //obtiene el valor anterior de posición.
        vm1=EEPROM_READ(0x03); //obtiene el valor anterior de posición.
        if ((vp!=0xA0)&(vm2>0))
        {
            RA1=1;
            RA4=0;
            RA3=0; //sentido de decremento de ganancia (u/d'=0).
            RA4=0;
            while(((vp!=0xA0)&(vm2>0))&((vpx==0x40)&(RA2==0)&(vm2>0)))
            {
                int i=0;
                int x=0;
                if (vm2>=1)
                {
                    x=(vm2-1);
                }
                else
                {
                    x=0;
                }
                while(vm2>=x)
                {
                    RA5=0; //pulso "0".
                    for (i=0;i<150;i++); //delay.
                    RA5=1; //pulso "1".
                    for(i=0;i<150;i++); //delay.
                    vm2--;
                }
                vp=PORTB&0xF0; //verifica si hubo cambio...
                vpx=PORTB&0x40;
            }
            EEPROM_WRITE(0x02,vm2);//Guarda última posición ganancia.
            RBIF=0;
            EEIF=0;
            RA4=1;
            RA3=1;
        }
        vp=PORTB&0x40; //filtra valor de Puerto B.(4 msb).
        RBIF=0;
        if ((vp==0x40)&(vm1>0)&(vm2==0)&(RA2==1))
            //salida del sumador por encima de 5V.
    }
}

```

```

{
    RA3=1;
    RA4=0;                //sentido de decremento (u/d'=0).
    RA1=0;                //selección del dispositivo Voffset.
    RA4=0;
    while((vp!=0xA0)&(vm1>0))
    {
        int i=0;
        RA5=0;                //pulso "0".
        for (i=0;i<150;i++);    //delay.
        RA5=1;                //pulso "1".
        for (i=0;i<150;i++);    //delay.
        vm1--;
        vp=PORTB&0xF0;//verifica si hubo cambio...
    }
    EEPROM_WRITE(0x03,vm1);    //Guarda última posición pedestal.
    RBIF=0;
    EEIF=0;
    RA4=1;
    RA1=1;
}
vp=PORTB&0x80;                //filtra valor de Puerto B.(4 msb).
RBIF=0;
if ((vp==0x00)&(vm1<=97)&(vm2==0)&(RA2==1))
    //salida del sumador por debajo de 0V.
{
    RA3=1;
    RA4=1;                //sentido de incremento (u/d'=1).
    RA1=0;                //selección del dispositivo.
    RA4=1;
    while((vp!=0xA0)&(vm1<97))
    {
        int i=0;
        RA5=0;                //pulso "0".
        for (i=0;i<150;i++);    //delay.
        RA5=1;                //pulso "1".
        for (i=0;i<150;i++);    //delay.
        vm1++;
        vp=PORTB&0xF0;        //verifica si hubo cambio...
    }
    EEPROM_WRITE(0x03,vm1);    //Guarda última posición pedestal.
    RBIF=0;
    EEIF=0;
    RA1=1;
}
vp=PORTB&0xF0;                //filtra valor de Puerto B.(4 msb).
vpx=PORTB&0x40;
RBIF=0;
if (((vp!=0xA0)&(vm2==0)&(RA2==1))&((vpx==0x40)&(vm2==0)&(RA2==0)))
    //Entrada al A/D fuera de rango modo AC+DC.
{
    int v=1;
    TXEN=1;
    while (v<=4)
    {
        TXREG=0x65;
    }
}

```



```

        while(!TRMT);
        TXREG=0x6D;
        while(!TRMT);
        TXREG=0xEE;

        while(!TRMT);
        v++;
    }
    TXEN=0;
    RBIE=0;
    vp=PORTB&0xF0;
    RBIF=0;
    rauto=0;
    aviso=1;
}
if ((!rauto)&(!aviso))
{
    int aux=0;
    vp=PORTB&0xF0;
    aviso=1;
    aux=vp&0x40;
    RBIF=0;
    if (aux==0x40)

        {
            int v=1;
            TXEN=1;
            while (v<=4)
            {
                TXREG=0x65;
                while(!TRMT);
                TXREG=0x6D;
                while(!TRMT);
                TXREG=0xDA;
                while(!TRMT);
                v++;
            }
            TXEN=0;
        }
    aux=vp&0x80;
    RBIF=0;
    if (aux==0x00)

        {
            int v=1;
            TXEN=1;
            while (v<=4)
            {
                TXREG=0x65;
                while(!TRMT);
                TXREG=0x6D;
                while(!TRMT);
                TXREG=0xDB;
                while(!TRMT);
                v++;
            }
        }
}

```

//anuncia error por amplitud.
//Debe usar atenuador
//a la entrada del sistema.

//filtra valor de Puerto B.(4 msb).

//filtra valor de Puerto B.(4 msb).

//filtra valor de Puerto B.(4 msb).

//salida del sumador por encima de 5V.

//filtra valor de Puerto B.(4 msb).

//salida del sumador por debajo de 0V.

```

    }
    TXEN=0;
}
}
}
//=====
//función interprete de comandos RX de RS232 (PC->PIC)
//=====
void traduce()
{
    if (trad&pde)
    {
        if (d1==0x64) //Menú Cuadro 1. Pantalla de Prueba.
        {
            if (d2==0x61) //Botón "CANCELAR".
            {
                cancela(); //atiende comando de detención de
                motor.
            }
            if (d2==0x63)
            {
                RA2=1; //selección de señal directo del
                filtro(AC+DC).(Switch Analógico)
            }
        }
        if (d1==0xCC)
        {
            if (d2==0x01)
            {
                RB1=1; //Conectar Relé de corriente.
            }
        }
        if (d1==0xCA)
        {
            if (d2==0x01)
            {
                RB1=0; //Desconectar Relé de corriente.
            }
        }
        if (d1==0x61)
        {
            if (d2==0x63)
            {
                RA2=0; //selección de señal directo del Rectif
                (AC).(Switch Analógico)
            }
        }
        if(d1==0xAA) //medir tensión ó corriente a paso fijo.
        {
            if (d2==0x01)
            {
                pde=1;
                mpad(); //medición A/D,paso fijo.
            }
        }
    }
}

```

```

if(d1==0xDD)
{
    if (d2==0x01)
    {
        pde=1;
        mfp();
        //medición de frecuencia,paso fijo.
    }
}
if (d1==0x74)
{
    if (d2==0x69)
    {
        RC4=1;
        pulso=1;
        nupasi=0;
        topmot();
        //movimiento hasta encontrar tope
        configt();
    }
    if (d2==0x64)
    {
        RC4=0;
        pulso=1;
        nupasd=0;
        topmot();
        //movimiento hasta encontrar tope
        configt();
    }
}
if (d1==0x70)
{
    int c;
    if (nupasd==0)
    {
        c=1;
    }
    if (nupasd>0&nupasd>=d2)
    {
        c=2;
    }
    switch(c)
    {
        case 1:
        {
            RC4=1;
            pulmot(d2);
            nupasi=nupasi-d2;
            break;
        }
        case 2:
        {
            RC4=1;
            pulmot(d2);
            nupasd=nupasd+d2;
        }
    }
}

```

mecánico.

mecánico.

CCW.(Izq.)
izquierda.

//Menú Cuadro 2. Manejo del Motor.

//Botón "Tope mecánico izquierdo".

//sentido de rotación CCW.(Izq.)
//autoriza movimiento del motor.

//Botón "Tope mecánico derecho".

//sentido de rotación CW.(Der.)
//autoriza movimiento del motor.

//Sentido de rotación
//Da d2 movimientos a la
//actualiza posición CCW.

//Sentido de rotación CCW.(Izq.)
//Da d2 movimientos a la izquierda.
//actualiza posición CCW.

```

        nupasi=nupasi-d2;//actualiza posición CCW.
        break;
    }
}
configt();
}
if (d1==0x80) //Botón "n pasos a la derecha".
{
    int c;
    if (nupasi>0&nupasi>=d2)
    {
        c=1;
    }
    if (nupasi==0)
    {
        c=2;
    }
    switch(c)
    {
        case 1:
        {
            RC4=0; //sentido de rotación CW.(Der.)
            pulmot(d2); //Da d2 movimientos a la derecha.
            RC4=1; //Sentido de rotación CCW.(Izq.)
            nupasi=nupasi+d2;//actualiza posición CCW.
            nupasd=nupasd-d2;//actualiza posición CW.
            break;
        }
        case 2:
        {
            RC4=0; //sentido de rotación CW.(Der.)
            pulmot(d2); //Da d2 movimientos a la derecha.
            RC4=1; //Sentido de rotación CCW.(Izq.)
            nupasd=nupasd-d2;//actualiza posición CW.
            break;
        }
    }
    configt();
}
if (d1==0x76&nupasi>0) //Menú Cuadro 4. Botón "Tensión".
{
    if (d2<=0x28)
    {
        libmot(d2,1); //Medición AC, d2 muestras(40 máx.).
        //Envío de pulsos al driver de motor.(100hZ).
    }
    if (d2>=0x29)
    {
        //Medición DC, d2-0x29 muestras (40 máx.).
        int mpp=0;
        int mul=0;
        int dig=0;
        mpp=d2-0x29;
        dig=mpp>>4&0x0F;
        mul=mpp&0x0F;
        mpp=1;
        while (mul>0)
        {

```

```

        mpp=mpp*10;
        mul--;
    }
    libmot((0x0A),(dig*mpp));
    //Envío de pulsos al driver de motor.(100hZ
}
}
if (d1==0x63&nupasi>0) //Botón "Corriente".
{
    if (d2<=0x28)
    {
        //Medición AC, d2 muestras(40 máx).
        libmot(d2,1); //Envío de pulsos al driver de motor.(100hZ).
    }
    if (d2>=0x29)
    {
        //Medición DC, d2-0x29 muestras.
        int mpp=0;
        int mul=0;
        int dig=0;
        mpp=d2-0x29;
        dig=mpp>>4&0x0F;
        mul=mpp&0x0F;
        mpp=1;
        while (mul>0)
        {
            mpp=mpp*10;
            mul--;
        }
        libmot((0x0A),(dig*mpp));
        //Envío de pulsos al driver de motor.(100hZ).
    }
}
}
if (d1==0x66&nupasi>0) //Botón "Frecuencia".
{
    if (d2<=0x28)
    {
        //Divisor por 1.
        libmot2(d2,1); //Envío de pulsos al driver de motor.(100hZ).
    }
    if (d2>=0x29)
    {
        int mpp=0;
        int mul=0;
        int dig=0;
        mpp=d2-0x29;
        dig=mpp>>4&0x0F;
        mul=mpp&0x0F;
        mpp=1;
        while (mul>0)
        {
            mpp=mpp*10;
            mul--;
        }
        libmot2((d2-0x29),(dig*mpp));
        //Envío de pulsos al driver de motor.(100hZ).
    }
}
}

```

```

if (d1==0x7D&nupasi>0) //Botón "Período".
{
    if (d2<=0x28)
    {
        libmot2(d2,1);
        //Envío de pulsos al driver de motor.(100hZ).
    }
    if (d2>=0x29)
    {
        int mpp=0;
        int mul=0;
        int dig=0;
        mpp=d2-0x29;
        dig=mpp>>4&0x0F;
        mul=mpp&0x0F;
        mpp=1;
        while (mul>0)
        {
            mpp=mpp*10;
            mul--;
        }
        libmot2((d2-0x29),(dig*mpp));
        //Envío de pulsos al driver de motor.(100hZ).
    }
}
if (d1==0x6F) //Botón "DC offset".
{
    vm1=EEPROM_READ(0x03);//obtiene el valor anterior de posición.
    if (d2>vm1) //Si el signo de incremento es (+).
    {
        RA4=1; //Señal de incremento al pot.
        RA3=1;
        RA1=0; //Señal Selección de dispositivo.
        valx=d2-vm1;
        if (valx>99)
        {
            valx=99;
        }
        RA4=1;
        incpot(valx); //(#)Pulsos para incrementar.
        RA1=1; //deselecciona el dispositivo.
        valx=vm1+valx;
        if (valx>99)
        {
            valx=99;
        }
        EEPROM_WRITE(0x03,valx);
        //salva la nueva posición en EEPROM.
        EEIF=0;
    }
    if (d2<vm1) //Si el signo de incremento es (-).
    {
        RA4=0; //Señal de decremento al pot.
        RA3=1;
        RA1=0; //Señal Selección de dispositivo.
        valx=vm1-d2;
    }
}

```

```

        if (valx>99)
        {
            valx=99;
        }
        incpot(valx);           //(#)Pulsos para decrementar.
        RA1=1;                 //deselecciona el dispositivo.
        valx=vm1-valx;
        if (valx>99)
        {
            valx=99;
        }
        EEPROM_WRITE(0x03,valx);
                                //salva la nueva posición en EEPROM.
        EEIF=0;
    }
}
if (d1==0x67)                 //Botón "Ganancia".
{
    pde=1;                     //habilita recepción rxbuffer.
    vm2=EEPROM_READ(0x02);     //obtiene el valor anterior de posición.
    if (d2>vm2)                //Si el signo de incremento es (+).
    {
        RA4=1;                 //Señal de incremento al pot.
        RA1=1;
        RA3=0;                 //Señal Selección de dispositivo.
        valx=(d2-vm2);
        if (valx>99)
        {
            valx=99;
        }
        RA4=1;
        incpot(valx);           //(#)Pulsos para incrementar.
        RA3=1;                 //deselecciona el dispositivo.
        valx=vm2+valx;
        if (valx>99)
        {
            valx=99;
        }
        EEPROM_WRITE(0x02,valx);
                                //salva la nueva posición en EEPROM.
        EEIF=0;
    }
}
if (d2<vm2)                   //Si el signo de incremento es (-).
{
    RA4=0;                     //Señal de decremento al pot.
    RA1=1;
    RA3=0;                     //Señal Selección de dispositivo.
    valx=vm2-d2;
    incpot(valx);              //(#)Pulsos para decrementar.
    RA3=1;                     //deselecciona el dispositivo.
    valx=vm2-valx;
    if (valx>99)
    {
        valx=99;
    }
}

```

```

        EEPROM_WRITE(0x02, valx);
                                //salva la nueva posición en EEPROM.
        EEIF=0;
    }
}
if (d1==0x75)                    //Botón "Puesta a Último Set Point".
{
    if (d2==0x70)
    {
        ultipots();              //Restaura último set point de los pot's.
    }
}
if (d1==0x69)
{
    if (d2==0x7D)
    {
        inicpots();              //Restaura set point original: G=10; Voff=0V.
    }
}
if (d1==0x6D)
{
    if (d2==0x74)
    {
        midetmp();              //Adquiere dato de temperatura ambiente.
    }
}
if (d1==0x73&d2==0x67&rauto)
{
    pde=1;
    while(!EEIF);
    vm2=EEPROM_READ(0x02);        //obtiene el valor anterior de posición.
    if (vm2==0)
    {
        int tt;
        RBIE=0;
        RA4=1;
        RA1=1;
        RA3=0;                    //Selección Pot. Ganacia
        incpot(97);
                                //incremento de pot ganancia a casi max. posible.
        RA3=1;
        EEPROM_WRITE(0x02, 97); //Guarda última posición.
        EEIF=0;
        for(tt=0; tt<15000; tt++);
        if (rauto)
        {
            cn=1;
            RBIF=0;
            RBIE=1;
        }
    }
}
if (d1==0x62&d2==0x67&rauto)    //Baja la ganancia al mínimo
{
    int pb=0;
    pde=1;

```



```

        RBIE=0;
        RA4=0;
        RA1=1;
        RA3=0;          //Selección Pot.Ganacia
        incpot(102);     //decremento de pot ganancia al min. posible.
        EEPROM_WRITE(0x02,0);//Guarda última posición.
        EEIF=0;
        RA3=1;
        RA4=1;
        pb=PORTB;
        if (rauto)
        {
            RBIF=0;
            cn=1;
            RBIE=1;
        }
    }
    if (d1==0x43)
    {
        RC4=1;          //Sentido de rotación CCW.(Izq.)
        pulmot(d2);      //Da d2 movimientos a la izquierda.
        nupasi=nupasi-d2;//actualiza posición CCW.
        nupasd=nupasd+d2;//actualiza posición CCW.
    }
    if (d1==0x34)
    {
        RC4=0;          //sentido de rotación CW.(Der.)
        pulmot(d2);      //Da d2 movimientos a la derecha.
        RC4=1;          //Sentido de rotación CCW.(Izq.)
        nupasi=nupasi+d2;//actualiza posición CCW.
        nupasd=nupasd-d2;//actualiza posición CW.
    }
    d1=0x00;            //borra petición solicitada anteriormente.
    d2=0x00;            //reset de la variable de opción recibida.
    trad=0;
    pde=1;
}

}

//=====
//Función para enviar datos de configuración de sistema.(desde vector int.)
//=====
void configti()
{
    int p;
    while(!EEIF);
    p=PORTB&0xF0;        //Estado de Comparadores (MSB puertoB):
    vm1=EEPROM_READ(0x03); //valor pot Offset.
    vm2=EEPROM_READ(0x02); //valor pot Ganancia.
    TXEN=1;
    TXREG=0x73;
    while(!TRMT);
    TXREG=0x61;
    while(!TRMT);
    TXREG=vm1;
    while(!TRMT);
    TXREG=vm2;
}

```

```

while(!TRMT);
TXREG=p;
while(!TRMT);
p=nupasi&0xFF;
TXREG=p;
while(!TRMT);
p=(nupasi>>8)&0xFF;
TXREG=p;
while(!TRMT);
p=nupasd&0xFF;
TXREG=p;
while(!TRMT);
p=(nupasd>>8)&0xFF;
TXREG=p;
while(!TRMT);
if (rauto)
{
    p=0x61;
}
else
{
    p=0x6D;
}
TXREG=p;
while(!TRMT);
TXREG=tmpl;           //Envia la información recolectada
while(!TRMT);
TXREG=tmph;           //del sensor a USART. PIC->PC.
while(!TRMT);
TXEN=0;
}

```

```

//=====
//función para recibir comandos del PC (PC->PIC)
//=====
void recibiri()
{
    pde++;
    if (pde==2)
    {
        d1=RCREG;
        pde++;
    }
    if (pde==4)
    {
        d2=RCREG;
        pde=1;
        trad=1;
    }
    if (d1==0x72&pde==1)
    {
        if (d2==0x6D)
        {
            rauto=0;
            aviso=0;
        }
    }
}

```

```

        if (d2==0x61)
        {
            int pa=0;
            cn=1;
            rauto=1;
            pa=PORTB;
            RBIF=0;
            RBIE=1;
        }
        trad=0;
        d1=0;
    }
    if (d1==0x73&pde==1)
    {
        if (d2==0x61)
        {
            trad=0;
            configti();           //envia dato de configuración de sistema.
                                   //del PIC al PC.
        }
        if (d2==0x70)
        {
            trad=0;
            stop=1;               //pausa al evento realizandose en el sistema.
        }
        if (d2==0x07)
        {
            trad=0;
            stop=0;               //reanuda acciones del evento en el sistema.
        }
    }
    if(d1==0x7A&pde==1)
    {
        if (d2==0x36)
        {
            pulso=0;             //atiende comando de detención de motor.
            med=0;
            RBIE=0;
            TMR0IE=0;
            TMR1IE=0;
            ADIE=0;
            CCP1IF=0;
            TMR0IF=0;
            TMR1IF=0;
            RBIF=0;
            ADIF=0;
            RC3=0;
            trad=0;
            if (rauto)
            {
                cn=1;
                RBIE=1;
            }
        }
        if (d2==0x37&nupasd>0)   //Botón "Puesta a Cero"
        {
            trad=0;

```

```

        pde=1;
        if (RC4)
        {
            RC4=0;
            while(nupasd>0)
            {
                int i;
                RC3=1;
                for (i=0;i<600;i++);
                RC3=0;
                for (i=0;i<600;i++);
                nupasd--;
                nupasi++;
            }
            RC4=1;
        }
        med=0;
        pulso=0;
    }
    if(d1==0x2C&pde==3)
    {
        pde=1;
        stop=0;
        error=0;
        d1=0;
    }
    if (d1==0x2D&pde==3)
    {
        pde=1;
        stop=0;
        error=1;
        d1=0;
    }
    if (d1==0x6D&pde==1&d2==0x6D)
    {
        int val=0;
        val=PORTA&0x04;
        TXEN=1;
        TXREG=0x6D;
        while(!TRMT);
        if (val==0)
        {
            TXREG=0xAC;
        }
        else
        {
            TXREG=0xDC;
        }
        while(!TRMT);
        TXEN=0;
    }
    if (d1==0x77&pde==1)
    {
        if(d2<=0x19)
        {

```

//habilita recepción rxbuffer.
 //si esta girando a la izq...

//cambio de sentido de giro.
 //autoriza movimiento.

//pulso en "1" al driver motor.
 //delay.
 //pulso en "0" al driver motor.
 //delay.
 //decrementa variable de control.

//cambio de sentido de giro.

//anula prueba.
 //anula búsqueda de tope.

//recibió txbuffer...

//no repite último envío.

//manda a repetir último envío.

```

    trad=0;
    iniccp2();
    inict2();
    pwm=d2*0x0A;           //constante para adecuar a formato Hex.
    CCP2L=pwm&0xFF;
    pwm=pwm>>8;
    pwm=pwm&0xFF;
    switch(x)
    {
        case 1:if (pwm==0x0000)
        {
            CCP2X=0;
            CCP2Y=0;
            break;
        }
        case 2:if (pwm==0x0001)
        {
            CCP2X=0;
            CCP2Y=1;
            break;
        }
        case 3:if (pwm==0x0002)
        {
            CCP2X=1;
            CCP2Y=0;
            break;
        }
        case 4:if (pwm==0x0003)
        {
            CCP2X=1;
            CCP2Y=1;
            break;
        }
    }
    TMR2ON=1;
    d1=0;
    d2=0;
}
if(d2>=0x1A)
{
    pde=1;
    iniccp2();
    inict2();
    trad=0;
    d1=0;
    d2=0;
}
}

//=====
//Función para enviar datos de configuración de sistema.
//=====
void configt()
{
    int p;
    while(!EEIF);

```

```

p=PORTB&0xF0;
vm1=EEPROM_READ(0x03);
vm2=EEPROM_READ(0x02);
TXEN=1;
TXREG=0x73;
while(!TRMT);
TXREG=0x61;
while(!TRMT);
TXREG=vm1;
while(!TRMT);
TXREG=vm2;
while(!TRMT);
TXREG=p;
while(!TRMT);
p=nupasi&0xFF;
TXREG=p;
while(!TRMT);
p=(nupasi>>8)&0xFF;
TXREG=p;
while(!TRMT);
p=nupasd&0xFF;
TXREG=p;
while(!TRMT);
p=(nupasd>>8)&0xFF;
TXREG=p;
while(!TRMT);
if (rauto)
{
    p=0x61;
}
else
{
    p=0x6D;
}
TXREG=p;
while(!TRMT);
TXREG=tmpl;
while(!TRMT);
TXREG=tmph;
while(!TRMT);
TXEN=0;
}

//=====
//Función medición A/D, motor estático.
//=====
void mpad()
{
    int t=0;
    inicad();
    miya=0;
    ADON=1;
    ADIF=0;
    ADIE=1;
    pdb=1;
    txbuffer[pdb]=0xAA;

```

//Estado de Comparadores (MSB puertoB):
//valor pot Offset.
//valor pot Ganancia.

//Envía la información recolectada
//del sensor a USART. PIC->PC.

```

    pdb++;
    ndato=1;
    med=1;
    while(ndato<=10)
    {
        for(t=0;t<5000;t++);           //retardo 16ms.
        adl=0;
        ADIF=0;
        ADGO=1;
        while(!adl);
        txbuffer[pdb]=ADRESL;           //Mueve parte baja de conv. a buffer.
        pdb++;                          //aumenta puntero posición de llenado.
        txbuffer[pdb]=ADRESH;          //Mueve parte alta de conv. a buffer.
        pdb++;                          //aumenta puntero posición de llenado.
        ndato++;
    }
    ADGO=0;                             //Detiene módulo A/D.
    pasol=(nupasi)&0xFF;
    pasoh=((nupasi)>>8)&0xFF;
    txbuffer[pdb]=pasol;                //Envía al buffer RAM del pic.
    pdb++;
    txbuffer[pdb]=pasoh;
    while(!EEIF);
    pdb++;
    txbuffer[pdb]=EEPROM_READ(0x02);    //Manda valor de ganancia actual
    pdb++;
    txbuffer[pdb]=EEPROM_READ(0x03);    //Manda valor de offset actual
    errorr=1;
    enviar(25);
    ADON=0;
    ndato=1;
    med=0;
    pdb=1;
}
//=====
//Función medición frecuencia, motor estático.
//=====
void mfp()
{
    int w=2;
    inict1();
    iniccp1(1);                        //inicia Módulo Captura 1.(divisor x 4)
    CCP1IF=0;
    CCP1IE=1;
    TMR1IF=0;
    TMR1IE=1;
    pasol=(nupasi)&0xFF;                //Extractor de paso mecánico actual.
    pasoh=((nupasi)>>8)&0xFF;
    pdb=1;
    med=1;
    TMR1ON=1;
    while(w>0)
    {
        if(dper)
        {

```

```

        txbuffer[pdb]=0xDD;           //mueve resultado de conversión a
                                        //RAM.
        pdb++;
        txbuffer[pdb]=CCPR1L;         //mueve resultado de conversión a
                                        //RAM.
        pdb++;
        txbuffer[pdb]=CCPR1H;         //mueve resultado de conversión a
                                        //RAM.
        pdb++;
        txbuffer[pdb]=dv1;            //mueve resultado de conversión a
                                        //RAM.
        pdb++;
        txbuffer[pdb]=dv2;            //mueve resultado de conversión a
                                        //RAM.
        pdb++;
        txbuffer[pdb]=pasol;          //mueve resultado de conversión a
                                        //RAM.
        pdb++;
        txbuffer[pdb]=pasoh;          //mueve resultado de conversión a
                                        //RAM.
        pdb++;
        w--;
        dper=0;
        CCP1IF=0;
        CCP1IE=1;
        GIE=1;
    }
    if(tout)                           //Si pasaron 100s...
    {
        tout=0;                       //Limpia indicador.
        w=0;                          //avanza al próximo paso de motor.
        if (!hd)
        {
            int v=10;
            TXEN=1;
            while (v>0)
            {
                TXREG=0xFC;            //Sobrepasamiento de 100s. al PC
                while(!TRMT);          //indicador que no hubo dato en este paso dado.
                v--;
            }
            TXREG=pasol;
            while(!TRMT);              //Sobrepasamiento de 100s. al PC
            TXREG=pasoh;               //indicador que no hubo dato en este paso dado.
            while(!TRMT);              //Sobrepasamiento de 100s. al PC
            TXEN=0;                   //indicador que no hubo dato en este paso dado.
        }
    }
}
TMR1ON=0;
vuelta1=0;
vuelta2=0;
TMR1L=0x00;

```



```

    TMR1H=0x00;
    TMR1IE=0;
    CCP1IE=0;
    TMR1IF=0;
    CCP1IF=0;
    tout=0;
    dper=0;
    hd=0;
    pdb=1;
    CCPR1L=0x00;
    CCPR1H=0x00;
    CCP1CON=0x00;
    errort=1;
    enviar(14);
    med=0;
}
//=====
//Función para pulsos de motor a 100hZ (Tensión y Corriente)
//=====
void libmot(int w,int a)
{
    int ix=0;
    int lee=0;
    int inf=15;
    int trans=0;
    while(trans<32000)
    {
        trans++;
    }
    trans=0;
    while(trans<32000)
    {
        trans++;
    }
    miya=0;
    pdb=1;
    med=1;
    ma=0;
    cn=1;
    ciclo=nupasi+1;
    ndato=1;
    inicad();
    ADON=1;
    ADIE=1;
    configt();
    while (ciclo>0&med)
    {
        adl=0;
        if (ciclo==1)
        {
            miya=0;
        }
        if (miya==0)
        {
            ma=1;
            txbuffer[pdb]=0xFA;
        }
    }
}

```

//carga de #pasos cw detectados a contador.

//Configura A/D para medición AC-DC.

//enciende convertidor A/D.

//#pasos faltantes para llegar al tope der->izq.

//se asegura de tomar datos de último paso.

//medición autorizada en este accionamiento.

//marca próximo dato a PC es Conv.A/D.

```

        pdb++;
        miya=a;
    }
    while(ndato<=w&med&ma)
    {
        ADIF=0;
        ADGO=1;                                //arranca conversión.
        while(!adl);
        adl=0;
        txbuffer[pdb]=ADRESL;                    //Mueve parte baja de conv. a buffer.
        pdb++;                                    //aumenta puntero posición de llenado.
        txbuffer[pdb]=ADRESH;                    //Mueve parte alta de conv. a buffer.
        pdb++;                                    //aumenta puntero posición de llenado.
        if (pdb==(2*w+2))                        //verifica no sobrepasar long.de buffer.
        {
            int valor=0;
            ADGO=0;                                //Detiene módulo A/D.
            while(!EEIF);
            valor=EEPROM_READ(0x02);//valor actual de ganancia.
            pasol=nupasi&0xFF;
            pasoh=(nupasi>>8)&0xFF;
            txbuffer[pdb]=pasol;//envia al buffer RAM del pic.
            pdb++;
            txbuffer[pdb]=pasoh;
            pdb++;
            txbuffer[pdb]=valor;
            pdb++;
            while(!EEIF);
            valor=EEPROM_READ(0x03);
            txbuffer[pdb]=valor;
            PORTE=0xFF;
            TRISD=0x00;
            GIE=0;
            RC5=1;
            while(pdb>=1)                        //Rutina para pasar datos RAM pic
            {                                    //a RAM externa.
                RC0=0;
                RB3=0;
                PORTE=0xFF;
                PORTD=(adr>>8)&0xFF;//filtra parte alta de dirección.
                RC0=1;                            //Lleva dato a Latch adress High.
                RC0=0;
                PORTD=(adr&0xFF);                //filtra parte baja de dirección.
                RB3=1;                            //Lleva dato a Latch adress Low.
                RB3=0;
                TRISD=0xFF;
                RE2=0;
                RE0=0;
                TRISD=0x00;
                for(ix=0;ix<1;ix++);
                PORTD=txbuffer[((2*w+6)-pdb)];
                //mueve resultado de conversión a RAM externa.
                for(ix=0;ix<1;ix++);
                RE0=1; //Protocolo de escritura en RAM externa.
                RE2=1;
                adr++;                            //incrementa dirección RAM.
            }
        }
    }

```

```

                                pdb--;
                                }
                                pdb=1;           //reset de puntero posición.
                                GIE=1;
                                }
                                ndato++;
                                }

while(stop);
RC3=1;           //pulso al driver motor "1".
ciclo--;        //decrementa cantidad de pasos.
if (ciclo>0)
{
    nupasi--;
    nupasd++;
    miya--;
    inf--;
}
if (inf==1)
{
    pasol=nupasi&0xFF;
    pasoh=(nupasi>>8)&0xFF;
    TXEN=1;
    inf=15;
    TXREG=0x69;
    while(!TRMT);
    TXREG=pasol;
    while(!TRMT);
    TXREG=pasoh;
    while(!TRMT);
    TXEN=0;
}
ix=0;
while (ix<1200)
{
    ix++;
}
ndato=1;
ma=0;
RC3=0;
ix=0;           //pulso al driver motor "0".
while (ix<1200)
{
    ix++;
}
if (rauto&(miya==0|ciclo==1))
{
    trans=0;
    while (trans<32000)
    {
        trans++;
    }
    trans=0;
    while (trans<32000)
    {
        trans++;
    }
}

```

```

    }
    vp=PORTB&0xF0;
    RBIE=0;
    RBIF=0;
    if (((vp!=0xA0)&(RA2==1))||((vp==0xE0)&(RA2==0)))
    {
        ajni();
    }
}
}
ADGO=0;
ADGO=0;
ADON=0;
ADIE=0;
ADIF=0;
ma=0;
if (rauto)
{
    RBIE=0;
    RA4=0;
    RA1=1;
    RA3=0;
    incpot(100);
    EEPROM_WRITE(0x02,0);
    EEIF=0;
    RA3=1;
    RA1=0;
    delay(10);
    incpot(101);
    RA4=1;
    incpot(73);
    EEPROM_WRITE(0x03,73);
    EEIF=0;
    RA5=0;
    RA1=1;
    cn=1;
    RBIE=1;
}
if (!med)
{
    adr=0;
    if (lee!=0xA0)
    {
        rauto=0;
    }
}
if (RC4)
{
    RC4=0;
    while(nupasd>0)
    {
        int iy=0;
        RC3=1;
        for (iy=0;iy<600;iy++);
        RC3=0;
    }
}

```

//Selección Pot.Ganacia
//decremento de pot ganancia al min. posible.
//Guarda última posición.
//hasta encontrar tope inferior.
//incremento de offset.
//A aprox. 2,5V de offset.
//salva la nueva posición en EEPROM.

//si esta girando a la izq...
//cambio de sentido de giro.
//autoriza movimiento.
//pulso en "1" al driver motor.
//delay.
//pulso en "0" al driver motor.

```

        for (iy=0;iy<600;iy++);           //delay.
        nupasd--;                         //decrementa variable de control.
        nupasi++;
    }
    RC4=1;                               //cambio de sentido de giro.
}

//=====
//Función para pulsos de motor a 100hZ (Frecuencia y Período.)
//=====
void libmot2(int w,int d)
{
    int inf=15;
    int it=0;
    int pb;
    miya=0;
    pb=PORTB;
    RBIE=0;
    RBIF=0;
    w=2;
    med=1;
    ciclo=nupasi+1;                      //carga de #pasos cw detectados a contador.
    inict1();
    TMR1IE=1;
    while (ciclo>0&med)                  // #pasos faltantes para llegar al tope der->izq.
    {
        pasol=nupasi&0xFF;               //Extractor de paso mecánico
        pasoh=(nupasi>>8)&0xFF;
        iniccp1(1);                      //inicia Módulo Captura 1.(divisor x 4)
        CCP1IF=0;
        if (ciclo==1)
        {
            miya=0;                      //se asegura de tomar datos de último paso.
        }
        if (miya==0)
        {
            ma=1;
            miya=d;
        }
        while(w>0&med&ciclo>0&ma)        //141.5us para grabar una trama(20kHz máximo).
        {
            TMR1ON=1;
            CCP1IE=1;
            if(dper)
            {
                int j=0;
                CCP1IE=0;
                TRISD=0x00;
                RC0=0;
                RB3=0;
                PORTE=0xFF;
                PORTD=(adr>>8)&0xFF;
                //filtra parte alta de dirección.
                RC0=1;                    //Lleva dato a Latch adress High.
                RC0=0;
            }
        }
    }
}

```

PORTD=(adr&0xFF);	//filtra parte baja de dirección.
RB3=1;	//Lleva dato a Latch adress Low.
RB3=0;	
TRISD=0xFF;	
RE2=0;	
RE0=0;	
TRISD=0x00;	
for(j=0;j<1;j++);	
PORTD=0xFD;	//mueve resultado de conversión a
	//RAM.
j=0;	
RE0=1;	//Protocolo de escritura en RAM.
RE2=1;	
PORTE=0xFF;	
adr++;	//incrementa dirección RAM.
PORTD=(adr>>8)&0xFF;	
	//filtra parte alta de dirección.
RC0=1;	//Lleva dato a Latch adress High.
RC0=0;	
PORTD=(adr&0xFF);	//filtra parte baja de dirección.
RB3=1;	//Lleva dato a Latch adress Low.
RB3=0;	
TRISD=0xFF;	
RE2=0;	
RE0=0;	
TRISD=0x00;	
for(j=0;j<1;j++);	
PORTD=CCPR1L;	//mueve resultado de conversión a
	//RAM.
j=0;	
RE0=1;	//Protocolo de escritura en RAM.
RE2=1;	
PORTE=0xFF;	
adr++;	//incrementa dirección RAM.
PORTD=(adr>>8)&0xFF;	//filtra parte alta de dirección.
RC0=1;	//Lleva dato a Latch adress High.
RC0=0;	
PORTD=(adr&0xFF);	//filtra parte baja de dirección.
RB3=1;	//Lleva dato a Latch adress Low.
RB3=0;	
TRISD=0xFF;	
RE2=0;	
RE0=0;	
TRISD=0x00;	
for(j=0;j<1;j++);	
PORTD=CCPR1H;	//mueve resultado de conversión a
	//RAM.
j=0;	
RE0=1;	//Protocolo de escritura en RAM.
RE2=1;	
PORTE=0xFF;	
adr++;	//incrementa dirección RAM.
PORTD=(adr>>8)&0xFF;	//filtra parte alta de dirección.

RC0=1;	//Lleva dato a Latch adress High.
RC0=0;	
PORTD=(adr&0xFF);	//filtra parte baja de dirección.
RB3=1;	//Lleva dato a Latch adress Low.
RB3=0;	
TRISD=0xFF;	
RE2=0;	
RE0=0;	
TRISD=0x00;	
for(j=0;j<1;j++);	
PORTD=dv1;	//mueve resultado de conversión a //RAM.
j=0;	
RE0=1;	//Protocolo de escritura en RAM.
RE2=1;	
PORTE=0xFF;	
adr++;	//incrementa dirección RAM.
PORTD=(adr>>8)&0xFF;	//filtra parte alta de dirección.
RC0=1;	//Lleva dato a Latch adress High.
RC0=0;	
PORTD=(adr&0xFF);	//filtra parte baja de dirección.
RB3=1;	//Lleva dato a Latch adress Low.
RB3=0;	
TRISD=0xFF;	
RE2=0;	
RE0=0;	
TRISD=0x00;	
for(j=0;j<1;j++);	
PORTD=dv2;	//mueve resultado de conversión a //RAM.
j=0;	
RE0=1;	//Protocolo de escritura en RAM.
RE2=1;	
PORTE=0xFF;	
adr++;	//incrementa dirección RAM.
PORTD=(adr>>8)&0xFF;	//filtra parte alta de dirección.
RC0=1;	//Lleva dato a Latch adress High.
RC0=0;	
PORTD=(adr&0xFF);	//filtra parte baja de dirección.
RB3=1;	//Lleva dato a Latch adress Low.
RB3=0;	
TRISD=0xFF;	
RE2=0;	
RE0=0;	
TRISD=0x00;	
for(j=0;j<1;j++);	
PORTD=pasol;	//mueve resultado de conversión a //RAM.
j=0;	
RE0=1;	//Protocolo de escritura en RAM.
RE2=1;	
PORTE=0xFF;	
adr++;	//incrementa dirección RAM.

```

PORTD=(adr>>8)&0xFF; //filtra parte alta de dirección.
RC0=1; //Lleva dato a Latch adress High.
RC0=0;
PORTD=(adr&0xFF); //filtra parte baja de dirección.
RB3=1; //Lleva dato a Latch adress Low.
RB3=0;
TRISD=0xFF;
RE2=0;
RE0=0;
TRISD=0x00;
for(j=0;j<1;j++);
PORTD=pasoh; //mueve resultado de conversión a
//RAM.

j=0;
RE0=1; //Protocolo de escritura en RAM.
RE2=1;
PORTE=0xFF;
adr++;
dper=0;
w--;
CCP1IF=0;
CCP1IE=1;
}
if(tout) //Si pasaron 12.58s...
{
    if(!hd)
    {
        dper=1;
        tout=1;
        CCPR1L=0;
        CCPR1H=0;
        dv1=0;
        dv2=0;
        w=2;
        hd=1;
    }
    if (hd&w==1)
    {
        CCPR1L=0xFF;
        CCPR1H=0xFF;
        dv1=0xFF;
        dv2=0xFF;
        dper=1;
    }
}
}
w=2;
ma=0;
hd=0;
tout=0;
CCP1IE=0;
CCP1CON=0x00;
CCP1IF=0;
TMR1ON=0;
vuelta1=0;
vuelta2=0;

```



```

TMR1L=0x00;
TMR1H=0x00;
if (stop)
{
    CCP1IE=0;
    CCP1IF=0;
}
while(stop);
RC3=1;
for (it=0;it<1200;it++);
RC3=0;
for (it=0;it<1200;it++);
ciclo--;
if (ciclo>0)
{
    nupasi--;
    nupasd++;
    miya--;
    inf--;
}
if (inf==1)
{
    TXEN=1;
    inf=15;
    TXREG=0x69;
    while(!TRMT);
    TXREG=pasol;
    while(!TRMT);
    TXREG=pasoh;
    while(!TRMT);
    TXEN=0;
}
}
if (!med)
{
    adr=0;
}
TMR1ON=0;
TMR1IE=0;
vuelta1=0;
vuelta2=0;
CCP1IE=0;
TMR1IF=0;
CCP1IF=0;
tout=0;
dper=0;
hd=0;
ma=0;
CCPR1L=0x00;
CCPR1H=0x00;
if (RC4)
{
    RC4=0;
    while(nupasd>0)
    {
        int iz;
    }
}
//pulso al driver motor "1".
//delay.
//pulso al driver motor "0".
//delay.
//decrementa cantidad de pasos.
//si esta girando a la izq...
//cambio de sentido de giro.
//autoriza movimiento.

```

```

        RC3=1;                                //pulso en "1" al driver motor.
        for (iz=0;iz<600;iz++);                //delay.
        RC3=0;                                //pulso en "0" al driver motor.
        for (iz=0;iz<600;iz++);                //delay.
        nupasd--;                              //decrementa variable de control.
        nupasi++;
    }
    RC4=1;                                    //cambio de sentido de giro.
}
if (rauto)
{
    pb=PORTB;
    RBIF=0;
    cn=1;
    RBIE=1;
}
}
//=====
//      Función Recuperar data de RAM y Enviar Data a PC.
//=====
void vaciar()
{
    if ((ciclo==0)&(adr>0)&(sadr<adr))
    {
        int i;
        int v=1;
        inicint();
        sadr=0;
        pdb=1;
        RC5=1;
        while (pdb<=12&sadr<adr)
        {
            RC0=0;
            RB3=0;
            PORTE=0xFF;
            TRISD=0x00;                        //PuertoD de salida.
            PORTD=(sadr>>8)&0xFF;              //filtra parte alta de dirección.
            RC0=1;                             //Lleva dato a Latch adress High.
            RC0=0;
            PORTD=(sadr&0xFF);                 //filtra parte baja de dirección.
            RB3=1;                             //Lleva dato a Latch adress Low.
            RB3=0;
            TRISD=0xFF;                        //PuertoD como entrada de datos.
            RE0=0;                             //Protocolo de Lectura en RAM.
            RE1=0;
            RE0=1;
            for(i=0;i<y;i++);
            txbuffer[pdb]=PORTD;               //mueve dato en RAM al PIC.
            for(i=0;i<y;i++);
            RE1=1;
            sadr++;                            //incrementa dirección RAM de salida.
            pdb++;                             //incrementa puntero buffer TX.
            if (sadr==adr|pdb==13)
            {
                CRC((txbuffer+1),pdb-1); //calcula CRC de trama extraída.
                error=1;                  //Supone que habrá error en comunicación.
            }
        }
    }
}

```

```

        stop=1;                //Bandera de parada.
        enviar(pdb-1);         //manda trama de data con CRC calculado.
        pdb=1;
    }
}
sadr=0;                        //Reset de direcciones de RAM para
adr=0;                          //ser reutilizada luego de envío.
TXEN=1;
while (v<=12)
{
    TXREG=0xBB;                //aviso de culminación de vaciado de datos.
    while(!TRMT);
    v++;
}
TXEN=0;
}
}
//=====
//      función retardo en el tiempo.
//=====
void delay (int x)
{
    int u;
    for (u=0;u<x;u++);
}
//=====
//función para generar 1 pulso de motor a 100hZ.
//=====
void pulmot(int p)
{
    int i;
    while(p>0)
    {
        RC3=1;                //Pulso de clock al driver del motor.
        for (i=0;i<600;i++);   //Delay.
        RC3=0;
        for (i=0;i<600;i++);
        p--;
    }
}
//=====
//      función para detener motor (instantaneamente).
//=====
void cancela()
{
    RC3=0;                    //pin de señal de clock al driver en "0".TMR0IE=0;
    RBIE=0;
    TMR1IE=0;
    CCP1IE=0;
    ADIE=0;
    CCP1IF=0;
    TMR0IF=0;
    TMR1IF=0;
    ADIF=0;
    CCP1IF=0;
    pulso=0;                 //inhibe movimiento del motor.
}

```

```

        ciclo=0;
        vuelta1=0;
        vuelta2=0;
        aviso=0;
        stop=0;
        med=0;
        TXEN=0;
    }
    //=====
    //      función retardo en el tiempo.(Desde Vector INT)
    //=====
    void delayi (int x)
    {
        int u;
        for (u=0;u<x;u++);
    }
    //=====
    //      Rutinas de Interrupción.
    //=====
    #pragma interrupt_level 0;
    void interrupt rutina_servicio()
    {
        GIE=0;                                //desactiva interrupción global.
        if (RCIF&CREN)                          //Si hubo dato de Entrada (RX)...
        {
            recibiri();                          //ejecuta rutina de recepción de data.
            RCIF=0;
        }
        if (RBIF&cn)                            //Si hubo Cambio en el nivel de la señal de entrada...
        {
            RBIE=0;
            vp=PORTB&0xF0;                      //filtra valor de Puerto B.(4 msb).
            RBIF=0;
            cn=0;
            while(!EEIF);
            if (rauto)
            {
                int vpx=0;
                vpx=PORTB&0x40;
                RC3=0;                            //Detiene motor.
                vm2=EEPROM_READ(0x02);            //obtiene el valor anterior de posición.
                vm1=EEPROM_READ(0x03);            //obtiene el valor anterior de posición.
                if
                (((vp!=0xA0)&(vm2>0)&(RA2==1))|((vpx==0x40)&(RA2==0)&(vm2>0)))
                {
                    RA1=1;
                    RA4=0;
                    RA3=0;                        //sentido de decremento de ganancia (u/d'=0).
                    RA4=0;

                    while((vp!=0xA0)&(vm2>0)&(RA2==1)|((vpx==0x40)&(RA2==0)&(vm2==0)))
                    {
                        int i=0;
                        int x=0;
                        if (vm2>=1)
                        {

```

```

        x=(vm2-1);
    }
    else
    {
        x=0;
    }
    while(vm2>=x)
    {
        RA5=0;           //pulso "0".
        for (i=0;i<150;i++); //delay.
        RA5=1;           //pulso "1".
        for(i=0;i<150;i++); //delay.
        vm2--;
    }
    vp=PORTB&0xF0;
    //verifica si hubo cambio...
    vpx=PORTB&0x40;
}
EEPROM_WRITE(0x02,vm2);
//Guarda última posición ganancia.

RBIF=0;
EEIF=0;
RA4=1;
RA3=1;
}
vp=PORTB&0x40;           //filtra valor de Puerto B.(4 msb).
RBIF=0;
if ((vp==0x40)&(vm1>0)&(vm2==0)&(RA2==1))
    //salida del sumador por encima de 5V.
{
    RA3=1;
    RA4=0;           //sentido de decremento (u/d'=0).
    RA1=0;           //selección del dispositivo Voffset.
    RA4=0;
    while((vp!=0xA0)&(vm1>0))
    {
        int i;
        RA5=0;           //pulso "0".
        for (i=0;i<150;i++); //delay.
        RA5=1;           //pulso "1".
        for (i=0;i<150;i++); //delay.
        vm1--;
        vp=PORTB&0xF0;
        //verifica si hubo cambio...
    }
    EEPROM_WRITE(0x03,vm1);
    //Guarda última posición pedestal.

    RBIF=0;
    EEIF=0;
    RA4=1;
    RA1=1;
}
vp=PORTB&0x80;           //filtra valor de Puerto B.(4 msb).
RBIF=0;
if ((vp==0x00)&(vm1<=97)&(vm2==0)&(RA2==1))
    //salida del sumador por debajo de 0V.

```

```

{
    RA3=1;
    RA4=1;           //sentido de incremento (u/d'=1).
    RA1=0;           //selección del dispositivo.
    RA4=1;
    while((vp!=0xA0)&(vm1<97))
    {
        int i;
        RA5=0;       //pulso "0".
        for (i=0;i<150;i++); //delay.
        RA5=1;       //pulso "1".
        for (i=0;i<150;i++); //delay.
        vm1++;
        vp=PORTB&0xF0; //verifica si hubo cambio...
    }
    EEPROM_WRITE(0x03,vm1);
                                //Guarda última posición pedestal.
    RBIF=0;
    EEIF=0;
    RA1=1;
}
vp=PORTB&0xF0;           //filtra valor de Puerto B.(4 msb).
vpx=PORTB&0x40;
RBIF=0;
if
(((vp!=0xA0)&(vm2==0)&(RA2==1))|((vpx==0x40)&(vm2==0)&(RA2==0)))
                                //Entrada al A/D fuera de rango modo AC+DC.
{
    int v=1;
    TXEN=1;
    while (v<=4)
    {
        TXREG=0x65;
        while(!TRMT);
        TXREG=0x6D;
        while(!TRMT);
        TXREG=0xEE;           //anuncia error por amplitud.

                                //Debe usar atenuador
        while(!TRMT);         //a la entrada del sistema.
        v++;
    }
    TXEN=0;
    RBIF=0;
    vp=PORTB&0xF0;           //filtra valor de Puerto B.(4 msb).
    RBIF=0;
    rauto=0;
    aviso=1;
}
}
if ((!rauto)&(!aviso))
{
    int aux=0;
    vp=PORTB&0xF0;           //filtra valor de Puerto B.(4 msb).
    aviso=1;
    aux=vp&0x40;             //filtra valor de Puerto B.(4 msb).
}

```

```

RBIF=0;
if (aux==0x40)
    //salida del sumador por encima de 5V.
    {
        int v=1;
        TXEN=1;
        while (v<=4)
        {
            TXREG=0x65;
            while(!TRMT);
            TXREG=0x6D;
            while(!TRMT);
            TXREG=0xDA;
            while(!TRMT);
            v++;
        }
        TXEN=0;
    }
aux=vp&0x80; //filtra valor de Puerto B.(4 msb).
RBIF=0;
if (aux==0x00)
    //salida del sumador por debajo de 0V.
    {
        int v=1;
        TXEN=1;
        while (v<=4)
        {
            TXREG=0x65;
            while(!TRMT);
            TXREG=0x6D;
            while(!TRMT);
            TXREG=0xDB;
            while(!TRMT);
            v++;
        }
        TXEN=0;
    }
}
if (TMR1IF)
{
    if(med)
    {
        vuelta1++; //incrementa contador de sobrepasamiento.
                    //cada vuelta 250ns*65535=16.383ms.
        if (vuelta1==0x100) //verifica el transcurrir de 4.17s..
        {
            vuelta1=0; //reset de #veces de sobrepasamiento.
            vuelta2++;
            if (vuelta2==0x03) //verifica el transcurrir de 12.58s..
            {
                tout=1; //bandera indicadora de 12.58s.
                vuelta2=0;
                vuelta1=0;
            }
        }
    }
}

```

```

    }
    TMR1IF=0;
}
if(TOIF&pulso)                //Si acabó el tiempo de integración...
{
    RB2=1;                    //reset de Integrador, Apertura SW analógico.
    TOIF=0;                  //reset de bandera.
    TOIE=0;                  //inhibe int.T0 para detener ventana integración.
}
if (INTF)                      //Si la señal cruzó 0v (Flanco de bajada de Traba)...
{
    if(pulso)
    {
        ni++;                //contador de pulsos INT para filtrar ruido y transitoria.
        INTF=0;
        if(ni==20)           //cuenta 10 interrupciones de transitoria (Experimental).
        {
            RC3=0;           //detiene motor. (Pulso a "0")
            if (!RC4)
            {
                pulso=0;      //Anula condición para seguir enviando pulsos.
                RC4=1;        //Sentido rotación a Izquierda (CCW).
                INTF=0;       //reset de bandera interrupción.
                INTE=0;
                ni=0;
                nupasi=nupasd-45;
                nupasd=0;
            }
            else
            {
                ni=0;         //Recarga Contador de interrupciones.
                RC4=0;
                //cambia rotación motor a CW. (CCW por defecto).
                INTF=0;       //reset de bandera interrupción.
            }
        }
    }
    else
    {
        INTF=0;
        INTE=0;
    }
}
if(CCP1IF)                    //Si la señal cruzó 0v (captura margen de tiempo).
{
    if(med)
    {
        dv1=vuelta1&0xFF;
        dv2=vuelta2;
        dper=1;              //Bandera de activación de grabación de medición.
        hd=1;                //hubo al menos 1 dato tomado.
    }
    CCP1IF=0;                //Reset de bandera.
}
if(ADIF)
{

```



```
        if(med)
        {
            adl=1;           //aviso de conversión lista.
            ADIF=0;
        }
        ADIF=0;
    }
    GIE=1;                  //reactiva interrupción global.
}
```

Anexo # 5

Programa Principal del computador para el Sistema A.F.E.

(Visual Basic 6.0)

Elaborado por el Br. Walter E. Senesi L para optar por el título de Ingeniero Electricista, mención
Electrónica, de la Universidad Central de Venezuela.

- ***Módulo Principal de declaraciones (Module1):***

Option Explicit

Public funesc As Boolean

Public ep As Boolean

Public pet As Boolean

Public v As Integer

Public o As Integer

Public c6 As Integer

Public nupasi

Public nupasd

Public temperatura

Public puertob

Public offset

Public ganancia

Public auto As Boolean

Public tm

Public cantmue

Public pb

Public I

Public tester As Integer

Public totalpaso

Public ajg

Public ajo

Public ajp

Public pausa

Public C As Integer

Public conjunto As Integer

Public mini

Public maxi

Public xlApp As Excel.Application

Public xlBook As Workbook

Public xlSheet As Worksheet

Public grafica As Excel.Chart

Public preg3

Public exlap3 As Excel.Application

```
Public xlBook3 As Workbook
Public xlSheet3 As Worksheet
Public grafica3 As Excel.Chart
```

```
Public Function retardo(ByVal x)
    Dim tx
    Do While tx < x
        tx = tx + 1
    Loop
End Function
```

- ***Código para Formulario1 (Menú Principal):***

```
Private Sub Form_Load()
On Local Error GoTo Error_Com
    Form1.MSComm1.Settings = "9600,N,8,1"
    Form1.MSComm1.SThreshold = 2
    Form1.MSComm1.RThreshold = 1
    Form1.MSComm1.InputLen = 1
    Form1.MSComm1.CommPort = 1
    Form1.MSComm1.PortOpen = True
    If Form1.MSComm1.PortOpen = True Then
        Dim m4
        Form1.Configuración.Enabled = False
        Form1.Medición.Enabled = False
        Form1.frecexciext.Enabled = False
        Form1.Motor.Enabled = False
        Form1.Potdo.Enabled = False
        Form1.mapmpf.Enabled = True
        Form1.modgraf.Enabled = True
        m4 = MsgBox("Encienda el equipo ó bien reinícielo para establecer comunicación", vbInformation)
    End If
Exit Sub
Error_Com:
    Dim x
    x = MsgBox("Se produjo un error al tratar de abrir el puerto COM1, Revise su disponibilidad y luego arranque nuevamente el programa principal", vbCritical)
    Unload Me
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Unload frmAbout
    Unload Form12
    Unload Form11
    Unload Form10
    Unload form9
```

```

Unload Form8
Unload Form7
Unload Form6
Unload Form5
Unload Form4
Unload Form3
Unload Form2
End Sub

Private Sub frecexciext_Click()
    Form5.Show
End Sub

Private Sub he_Click()
    Form12.RichTextBox1.FileName = "C:\AFE\ayugen.rtf"
    Form12.Show
End Sub

Private Sub mapmpf_Click()
    Form2.Show
End Sub

Private Sub modgraf_Click()
    Form8.Show
End Sub

Private Sub Motor_Click()
    Form3.Show
End Sub

Private Sub Potdo_Click()
    Form4.Show
End Sub

Private Sub MSComm1_OnComm()

    If MSComm1.CommEvent = 2 Then
        Dim data As Variant
        Dim pb2
        Dim lWord

        data = MSComm1.Input
        pb = Mid(data, 1, 1)
        If pb = "ü" Then
            MSComm1.InputLen = 5
            data = MSComm1.Input
            MSComm1.InputLen = 1
            Form2.Label2.Caption = "F.R."
        End If
        If pb = "i" Then
            Dim pl

```

```

Dim ph
MSComm1.InputLen = 2
data = MSComm1.Input
MSComm1.InputLen = 1
pl = Asc(Mid(data, 1, 1))
ph = Asc(Mid(data, 2, 1))
nupasi = (ph * &H100) Or pl
nupasd = totalpaso - nupasi
Form3.Text4.Text = CStr(nupasi)
Form3.Text5.Text = CStr(nupasd)
Form8.ProgressBar1.Value = 100 * (1 - (nupasi / totalpaso))
End If
If pb = "m" Then
    MSComm1.InputLen = 1
    data = MSComm1.Input

    If Asc(Mid(data, 1, 1)) = "Ü" Then 'conectado AC
        Form10.Option2 = True
    End If
    If Asc(Mid(data, 1, 1)) = "¬" Then 'conectado DC
        Form10.Option1 = True
    End If
End If
If pb = "U" Then
    MSComm1.InputLen = 6
    data = MSComm1.Input
    MSComm1.InputLen = 1
    MSComm1.Output = Chr(&H6D)
    MSComm1.Output = Chr(&H74)
    MsgBox "Sistema Conectado y en espera", vbExclamation
    Unload Form2
    Unload Form3
    Unload Form4
    Unload Form5
    Unload Form8
    Unload form9
    Unload Form10
    Unload Form11
    retardo (100000)
    Form1.MSComm1.Output = Chr(&H73)
    Form1.MSComm1.Output = Chr(&H61)
    Form1.Motor.Enabled = True
    Form1.frecexciext.Enabled = True
    Form1.Configuración.Enabled = True
    Form7.Show

```

```

End If

If pb = "e" Then
    Dim dat1
    Dim dat2

    MSComm1.InputLen = 11

    Do While MSComm1.InBufferCount < 11
        Loop
        data = MSComm1.Input
        dat1 = Mid(data, 1, 1)
        dat2 = Mid(data, 2, 1)
        If dat1 = "m" Then
            If dat2 = "Ú" Then
                MsgBox "Aviso: Ajuste manual, señal medida fuera de rango superior. Decrementa Pedestal DC", vbInformation
            End If
            If dat2 = "Ů" Then
                MsgBox "Aviso: Ajuste manual, señal medida fuera de rango Inferior. Incrementa Pedestal DC", vbInformation
            End If
            If dat2 = "ĩ" Then
                MsgBox "Aviso: Ajuste automático, señal medida fuera de rango. Use atenuador a la entrada", vbInformation
            End If
        End If
        MSComm1.InputLen = 1
    End If

    If pb = "a" Then

        Dim punt
        Dim adl1
        Dim adh1
        Dim prom
        Dim paso1
        Dim co

        MSComm1.InputLen = 24

        Do While MSComm1.InBufferCount <= 23
            Loop
            data = MSComm1.Input

            punt = 1
            Do While punt <= (20)

```

```

    adl1 = Asc(Mid$(data, punt, 1))
    adh1 = Asc(Mid$(data, punt + 1, 1))
    adh1 = (adh1 * &H100) Or adl1
    prom = (adh1 + prom)
    punt = punt + 2

Loop
adl1 = Asc(Mid$(data, punt, 1))
punt = punt + 1
adh1 = Asc(Mid$(data, punt, 1))
paso1 = (adh1 * &H100) Or adl1
punt = punt + 1
ganancia = Asc(Mid$(data, punt, 1))
punt = punt + 1
offset = Asc(Mid$(data, punt, 1))
If totalpaso <> 0 Then
    Form2.Text1.Text = Round(paso1 / totalpaso, 3)
End If
Open "C:\AFE\tablaPedestal.txt" For Input As #1
Do Until co = offset + 1 Or EOF(1)
    Line Input #1, ajp
    co = co + 1
Loop
co = 0
ajp = 2 * Val(ajp)
Close #1
If Form2.Option1 = True Then
    Open "C:\AFE\tablaOffset.txt" For Input As #1
    Do Until co = ganancia + 1 Or EOF(1)
        Line Input #1, ajo
        co = co + 1
    Loop
    Close #1
End If
If Form2.Option2 = True Then
    Open "C:\AFE\tablaOffsetCorr.txt" For Input As #1
    Do Until co = ganancia + 1 Or EOF(1)
        Line Input #1, ajo
        co = co + 1
    Loop
    Close #1
End If
co = 0
ajo = Val(ajo) * 0.001

```

```

Open "C:\AFE\tablaGanancia.txt" For Input As #1
Do Until co = ganancia + 1 Or EOF(1)
    Line Input #1, ajg
    co = co + 1
Loop
Close #1
ajg = 0.018 * Val(ajg)
Form7.Text6.Text = CStr(offset)
Form7.Text7.Text = CStr(ganancia)
Form4.Slider1.SelStart = ganancia
Form4.Slider2.SelStart = offset
If Form2.Option1 = True Then
    If Form10.Option1 = True Then
        prom = Round(5 * ((prom) / (10)) / 1024, 4)
        prom = prom - ajp + ajo + 0.015
        prom = (prom / (ajg))
    End If
    If Form10.Option2 = True Then
        prom = Round(5 * ((prom) / (10)) / 1024, 4)
        prom = (prom / (ajg))
    End If
End If
If Form2.Option2 = True Then
    If Form10.Option1 = True Then
        prom = Round(5 * ((prom) / (10)) / 1024, 4)
        prom = Round(prom - ajp + ajo - 0.003, 2)
        prom = prom / (ajg / 0.018)
        prom = Round(1000 * prom / 0.95, 2)
    End If
    If Form10.Option2 = True Then
        prom = Round(5 * ((prom) / (10)) / 1024, 4)
        prom = Round(prom, 2)
        prom = prom / (ajg / 0.018)
        prom = Round(1000 * prom / 1.111, 2)
    End If
End If
MSComm1.InputLen = 1
MSComm1.RThreshold = 1
Form2.Label2.FontSize = 34
If Form2.Option1.Value = True Then
    If Form10.Option1.Value = True Then
        Form2.Label2.Caption = Round(prom, 2)
    End If
    If Form10.Option2.Value = True Then
        Form2.Label2.Caption = Round((prom * (3.1415 / 2)), 2)
    End If
End If

```



```

    End If
End If
If Form2.Option2.Value = True Then
    If Form10.Option1.Value = True Then
        Form2.Label2.Caption = Round(prom, 2)
    End If
    If Form10.Option2.Value = True Then
        Form2.Label2.Caption = Round(prom * (3.1415 / 2), 2)
    End If
End If
End If

If pb = "Y" Then

    Dim punt2 As Integer
    Dim adl2 As Long
    Dim adh2 As Long
    Dim f1 As Long
    Dim f2 As Long
    Dim resulta As Long
    Dim paso2

    MSComm1.InputLen = 13

    Do While MSComm1.InBufferCount < 13
        Loop
        data = MSComm1.Input
        punt2 = 1
        adl2 = Asc(Mid$(data, punt2 + 1, 1))
        adl2 = (adl2 * &H100) Or Asc(Mid$(data, punt2, 1))
        adh2 = Asc(Mid$(data, punt2 + 3, 1))
        adh2 = (adh2 * &H100) Or Asc(Mid$(data, punt2 + 2, 1))
        f1 = (adh2 * &H10000)
        f1 = f1 Or adl2
        paso2 = Asc(Mid$(data, punt2 + 5, 1)) * &H100 Or Asc(Mid$(data, punt2 + 4, 1))
        If totalpaso <> 0 Then
            Form2.Text1.Text = Round(paso2 / totalpaso, 3)
        End If
        adl2 = Asc(Mid$(data, punt2 + 8, 1))
        adl2 = (adl2 * &H100)
        adl2 = adl2 Or Asc(Mid$(data, punt2 + 7, 1))
        adh2 = Asc(Mid$(data, punt2 + 10, 1))
        adh2 = (adh2 * &H100)
        adh2 = adh2 Or Asc(Mid$(data, punt2 + 9, 1))
        f2 = (adh2 * &H10000)
    
```

```

f2 = f2 Or adl2
MSComm1.InputLen = 1
MSComm1.RThreshold = 1
Form2.Label2.FontSize = 34
resulta = Round(4 / (0.00000025 * (f2 - f1)), 2)
If resulta <= 1 Then
    Form2.Timer1.Interval = 2100
Else
    Form2.Timer1.Interval = 1000
End If
If Form2.Option3 = True Then
    If resulta < 50500 Then
        Form2.Label2.Caption = Round(4 / (0.00000025 * (f2 - f1)), 2)
    Else
        Form2.Label2.Caption = "F.R."
    End If
End If
If Form2.Option4 = True Then
    If resulta < 50500 Then
        Form2.Label2.Caption = Round(1000 / resulta, 3)
    Else
        Form2.Label2.Caption = "F.R."
    End If
End If
End If
If pb = "s" Then
    Dim cont1
    MSComm1.InputLen = 11
    Do While MSComm1.InBufferCount < 11
        Loop
        data = MSComm1.Input
        pb = Asc(Mid$(data, 5, 1))
        pb2 = Asc(Mid$(data, 6, 1))
        lWord = (pb2 * &H100) Or pb
        nupasi = lWord
        Form3.Text4.Text = CStr(nupasi)
        Form3.Text2.Text = (nupasi * 360) / 200
        pb = Asc(Mid$(data, 7, 1))
        pb2 = Asc(Mid$(data, 8, 1))
        lWord = (pb2 * &H100) Or pb
        nupasd = lWord
        totalpaso = nupasi + nupasd
        Form3.Text5.Text = CStr(nupasd)
        pb = Mid$(data, 9, 1)
        If pb = "a" Then

```

```

    auto = True
Else
    auto = False

End If
If auto = False Then
    Form7.Label7.Caption = "Manual"
    Form7.Label7.ForeColor = &H80FF&
    Form4.Option2 = True
Else
    Form7.Label7.Caption = "Automático"
    Form7.Label7.ForeColor = &H8000&
    Form4.Option1 = True
End If
offset = Asc(Mid$(data, 2, 1))
ganancia = Asc(Mid$(data, 3, 1))
Open "C:\AFE\tablaPedestal.txt" For Input As #1
Do Until cont1 = offset + 1 Or EOF(1)
    Line Input #1, ajp
    cont1 = cont1 + 1
Loop
cont1 = 0
ajp = 2 * Val(ajp)
Close #1
Open "C:\AFE\tablaOffset.txt" For Input As #1
Do Until cont1 = ganancia + 1 Or EOF(1)
    Line Input #1, ajo
    cont1 = cont1 + 1
Loop
Close #1
cont1 = 0
ajo = Val(ajo) * 0.001
Open "C:\AFE\tablaGanancia.txt" For Input As #1
Do Until cont1 = ganancia + 1 Or EOF(1)
    Line Input #1, ajg
    cont1 = cont1 + 1
Loop
Close #1
cont1 = 0
ajg = 0.018 * Val(ajg)
Form4.Text2.Text = CStr(Round(ajp, 2))
Form4.Text1.Text = CStr(Round(ajg, 3))
Form7.Text6.Text = CStr(offset)
Form7.Text7.Text = CStr(ganancia)

```

```

Form4.Slider1.SelStart = ganancia
Form4.Slider2.SelStart = offset
puertob = Asc(Mid$(data, 4, 1))
pb = Asc(Mid$(data, 10, 1))
pb2 = Asc(Mid$(data, 11, 1))
temperatura = Round(((pb2 * &H100) Or pb) / 128 + 0.117, 1)
If totalpaso <> 0 Then
    Form2.Text1.Text = Round(nupasi / totalpaso, 3)
    Form1.Medición.Enabled = True
    Form1.mapmpf.Enabled = True
    Form1.modgraf.Enabled = True
    Form1.frecexciext.Enabled = True
    Form1.Motor.Enabled = True
    Form1.Potdo.Enabled = True
Else
    Form1.Medición.Enabled = False
    Form1.mapmpf.Enabled = False
    Form1.modgraf.Enabled = False
    Form1.frecexciext.Enabled = True
    Form1.Motor.Enabled = True
    Form1.Potdo.Enabled = False
    Form2.Text1.Text = "?"
End If
Form7.Text8.Text = CStr(temperatura)
MSComm1.InputLen = 1
MSComm1.RThreshold = 1
pb = "s"
End If

If pb = "ú" Then

    Dim puntero
    Dim promedio
    Dim indice
    Dim casilla
    Dim n
    Dim adl
    Dim adh
    Dim paso
    Dim objfilas As Range
    Dim filas
    Dim colum
    Dim sel As Object
    Dim myRange As Range
    Dim seleccion

```

```

Dim d
Dim pot
Dim base
Dim x
Dim nd1
Dim pun
Dim v
Dim nc1
Dim cont2

If (conjunto >= 1) Then
    Set xlApp = GetObject(, "Excel.Application")
    Set xlBook = xlApp.Workbooks("Libro1")
    Set xlSheet = xlBook.Worksheets("Hoja1")

    C = C + 1
End If

Form8.Command2.Enabled = True
Form8.Command7.Enabled = False
Form8.ProgressBar1.Value = 0

If Form8.Option6 = True Then
    cantmue = 10
End If

d = Val(Form8.Combo2.Text)
pot = Val(Form8.Combo3.Text)
base = 1

Do While pot > 0

    pot = pot - 1
    base = 10 * base

Loop

MSComm1.RThreshold = 0
nd1 = totalpaso
x = 0
If Form8.Option6 = True Then
    Do While (nd1 > 0)
        nd1 = nd1 - (d * base)
        x = x + 1
    Loop

```

```

End If
If Form8.Option5 = True Then
    x = totalpaso
End If
Do While MSComm1.InBufferCount < (2 * cantmue + 5) * (x + 1)
Loop

MSComm1.InputLen = 0
data = MSComm1.Input

If Form8.Option1 = True Or Form8.Option2 = True Then

    n = 0
    indice = (2 * cantmue + 5)
    puntero = 1
    I = 2
    Do While puntero <= (2 * cantmue + 5) * (x + 1)

        Do While puntero < (indice * (n + 1)) - 4

            adl = Asc(Mid$(data, puntero, 1))
            adh = Asc(Mid$(data, puntero + 1, 1))
            adh = (adh * &H100) Or adl
            promedio = (adh + promedio)
            puntero = puntero + 2

        Loop
        promedio = 5 / 1024 * ((promedio) / (cantmue))
        adl = Asc(Mid$(data, puntero, 1))
        adh = Asc(Mid$(data, puntero + 1, 1))
        paso = (adh * &H100) Or adl
        ganancia = Asc(Mid$(data, puntero + 2, 1))
        offset = Asc(Mid$(data, puntero + 3, 1))

        Open "C:\AFE\tablaPedestal.txt" For Input As #1
        Do Until cont2 = offset + 1 Or EOF(1)
            Line Input #1, ajp
            cont2 = cont2 + 1
        Loop
        cont2 = 0
        ajp = 2 * Val(ajp)
        Close #1
        If Form2.Option1 = True Then
            Open "C:\AFE\tablaOffset.txt" For Input As #1
            Do Until cont2 = ganancia + 1 Or EOF(1)

```

```

        Line Input #1, ajo
        cont2 = cont2 + 1
    Loop
    Close #1
End If
If Form2.Option2 = True Then
    Open "C:\AFE\tablaOffsetCorr.txt" For Input As #1
    Do Until cont2 = ganancia + 1 Or EOF(1)
        Line Input #1, ajo
        cont2 = cont2 + 1
    Loop
    Close #1
End If
cont2 = 0
ajo = Val(ajo) * 0.001
Open "C:\AFE\tablaGanancia.txt" For Input As #1
Do Until cont2 = ganancia + 1 Or EOF(1)
    Line Input #1, ajg
    cont2 = cont2 + 1
Loop
Close #1
cont2 = 0
ajg = 0.018 * Val(ajg)
If Form8.Option7 = True Then    'si la medición es DC+AC
    If Form8.Option1 = True Then
        promedio = promedio + 0.015 - ajp + ajo
        promedio = promedio / ajg
    End If
    If Form8.Option2 = True Then
        promedio = Round(promedio - ajp + ajo, 2)
        promedio = promedio / (ajg / 0.018)
        promedio = Round(1000 * promedio / 0.95, 2)
    End If
End If
If Form8.Option8 = True Then    'si la medición es AC
    If Form8.Option1 = True Then
        promedio = ((promedio) / (ajg)) * (3.1415 / 2)
    End If
    If Form8.Option2 = True Then
        promedio = ((promedio) / (ajg / 0.018)) * (3.1415 / 2)
        promedio = Round(1000 * (promedio / 1.111), 3)
    End If
End If
If C > 1 Then
    xlSheet.Cells(I, C).Value = promedio

```

```

End If
I = I + 1
n = n + 1
puntero = puntero + 5
promedio = 0
Loop
If Form8.Option1 = True Then
If C = 1 Then
xlSheet.Cells(1, C + 1).Value = "Tensión(V) " & CStr(C)
Else
xlSheet.Cells(1, C).Value = "Tensión(V) " & CStr(C - 1)
End If
End If

If Form8.Option2 = True Then
If C = 1 Then
xlSheet.Cells(1, C + 1).Value = "Corriente(mA)" & CStr(C)
Else
xlSheet.Cells(1, C).Value = "Corriente(mA)" & CStr(C - 1)
End If
End If
xlSheet.Cells(I, C).Value = "Temp.°C"
xlSheet.Cells(I + 1, C).Value = temperatura
xlSheet.Columns(C).AutoFit
MsgBox "Los datos han sido procesados, Graficando..."

Set objfilas = xlSheet.Range("A2").CurrentRegion
filas = objfilas.Rows.Count
column = objfilas.Columns.Count

Open "C:\AFE\TablaAlfabeto.txt" For Input As #1
Do Until pun = column Or EOF(1)
Line Input #1, v
pun = pun + 1
Loop
Close #1

If (C > 1) Then
filas = filas - 2
End If
seleccion = "A2:" & v & filas
xlBook.Charts.Add
Set grafica = xlBook.ActiveChart
xlApp.Visible = True

```



```

With grafica
    .ChartType = xlXYScatterSmooth
    .SetSourceData Source:=xlSheet.Range(seleccion), PlotBy _
:=xlColumns
    .Location Where:=xlLocationAsNewSheet
End With

With grafica.Axes(xlCategory)
    .HasMajorGridlines = True
    .HasMinorGridlines = False
End With

With grafica
    .HasLegend = True
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Acc.Theta"
    .Axes(xlValue, xlPrimary).HasTitle = True
    If Form8.Option1 = True Then
        If Form8.Option7 = True Then
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Tensión D.C.(V)"
        End If
        If Form8.Option8 = True Then
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Tensión A.C.(V)"
        End If
    End If
    If Form8.Option2 = True Then
        If Form8.Option7 = True Then
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Corriente D.C.(mA)"
        End If
        If Form8.Option8 = True Then
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Corriente A.C.(mA)"
        End If
    End If
End With
nc1 = C
Do While (nc1 >= 2)
    grafica.SeriesCollection(nc1 - 1).Name = "=Hoja1!R1C" & nc1
    nc1 = nc1 - 1
Loop
With grafica.Axes(xlCategory)
    .MinimumScaleIsAuto = True
    .MaximumScale = 1
    .MinorUnitIsAuto = True
    .MinimumScale = 0
    .MajorUnitIsAuto = True

```

```

.Crosses = xlAutomatic
.ReversePlotOrder = False
.ScaleType = xlLinear
.DisplayUnit = xlNone
End With

With grafica.Axes(xlValue, xlPrimary)
.HasMajorGridlines = True
.HasMinorGridlines = False
.MinimumScale = (0.95) * mini
.MaximumScale = (1.05) * maxi
.MinorUnitIsAuto = True
.MajorUnitIsAuto = True
.Crosses = xlAutomatic
.ReversePlotOrder = False
.ScaleType = xlLinear
.DisplayUnit = xlNone
End With

With grafica.ChartArea.Border
.ColorIndex = 16
.Weight = xlThin
.LineStyle = xlContinuous
End With

With grafica.Axes(xlValue).TickLabels
.NumberFormat = "0.000"
.Font.Name = "Arial"
.Font.Bold = True
.Font.Size = 11
.Font.Strikethrough = False
.Font.Underline = xlUnderlineStyleNone
End With

xlApp.Visible = True
MsgBox ("Graficado")
I = 0
MSComm1.InBufferCount = 0
MSComm1.RThreshold = 1
pb = "0"
MSComm1.InputLen = 1
If preg3 = 7 And conjunto = 0 Then
    xlBook.Close
    xlApp.Quit
End If
Form1.MSComm1.Output = Chr(&H73)

```

```

Form1.MSComm1.Output = Chr(&H61)
Set filas = Nothing
Set colum = Nothing
Set sel = Nothing
Set myRange = Nothing
Set objfilas = Nothing
Set seleccion = Nothing
Set grafica = Nothing
Set xlSheet = Nothing
Set xlBook = Nothing
Set xlApp = Nothing
End If

End If

If pb = "Y" Then
    Dim punt3
    Dim per3 As Single
    Dim v0 As Long
    Dim v1 As Long
    Dim t0 As Long
    Dim t1 As Long
    Dim com3 As Long
    Dim paso3
    Dim objfilas3 As Range
    Dim filas3
    Dim seleccion3
    Dim sel3 As Object
    Dim d3
    Dim pot3
    Dim base3
    Dim nd
    Dim cd
    Dim pun3
    Dim colum3
    Dim v3
    Dim nc

    If (conjunto >= 1) Then
        Set exlApp3 = GetObject(, "Excel.Application")
        Set xlBook3 = exlApp3.Workbooks("Libro1")
        Set xlSheet3 = xlBook3.Worksheets("Hoja1")
        C = C + 1
    End If

```

```

Form8.ProgressBar1.Value = 0
Form8.Command2.Enabled = True
Form8.Command7.Enabled = False

d3 = Val(Form8.Combo2.Text)
pot3 = Val(Form8.Combo3.Text)
base3 = 1

Do While pot3 > 0

    pot3 = pot3 - 1
    base3 = 10 * base3

Loop

MSComm1.RThreshold = 0
cd = 0
nd = totalpaso
Do While (nd > 0)
    nd = (nd - (base3 * d3))
    cd = cd + 1
Loop
Do While MSComm1.InBufferCount < (14 * (cd + 1))
Loop

MSComm1.InputLen = 0
data = MSComm1.Input

If Form8.Option3 = True Or Form8.Option4 = True Then
    punt3 = 1
    I = 2
    Do While punt3 <= (14) * (cd + 1)

        t0 = Asc(Mid$(data, punt3 + 1, 1))
        com3 = Asc(Mid$(data, punt3, 1))
        t0 = (t0 * &H100) Or com3
        com3 = Asc(Mid$(data, punt3 + 2, 1))
        v0 = Asc(Mid$(data, punt3 + 3, 1))
        v0 = (v0 * &H100) Or com3
        t0 = (v0 * &H10000) Or t0
        punt3 = punt3 + 7
        t1 = Asc(Mid$(data, punt3 + 1, 1))
        com3 = Asc(Mid$(data, punt3, 1))
        t1 = (t1 * &H100) Or com3
        com3 = Asc(Mid$(data, punt3 + 2, 1))

```

```

v1 = Asc(Mid$(data, punt3 + 3, 1))
v1 = (v1 * &H100) Or com3
If v1 >= 65535 Then
    per3 = 32000000          'límite de período mínimo 1Hz
Else
    t1 = (v1 * &H10000) Or t1
    per3 = t1 - t0
End If
com3 = Asc(Mid$(data, punt3 + 4, 1))
paso3 = Asc(Mid$(data, punt3 + 5, 1))
paso3 = paso3 * &H100 Or com3
punt3 = punt3 + 7

If Form8.Option3 = True Then
    per3 = (4) / (per3 * 0.00000025)
End If
If Form8.Option4 = True Then
    per3 = ((1000 * per3) / 4) * 0.00000025
End If
If (C = 1) Then
    xlSheet3.Cells(I, C).Value = paso3 / totalpaso
    xlSheet3.Cells(I, C + 1).Value = per3
End If
If (C > 1) Then
    xlSheet3.Cells(I, C).Value = per3
End If
I = I + 1
per3 = 0

Loop

If Form8.Option3 = True Then
    xlSheet3.Cells(1, C).Value = "Frec.(Hz)" & CStr(C - 1)
End If
If Form8.Option4 = True Then
    xlSheet3.Cells(1, C).Value = "Per.(ms)" & CStr(C - 1)
End If
xlSheet3.Cells(I, C).Value = "Temp." & CStr(C)
xlSheet3.Cells(I + 1, C).Value = temperatura
xlSheet3.Columns(1).AutoFit
MsgBox "Los datos han sido procesados, Graficando..."
Set objfilas3 = xlSheet3.Range("A2").CurrentRegion
filas3 = objfilas3.Rows.Count
colum3 = objfilas3.Columns.Count
Open "C:\AFE\TablaAlfabeto.txt" For Input As #1
Do Until pun3 = colum3 Or EOF(1)

```

```

    Line Input #1, v3
    pun3 = pun3 + 1
Loop
Close #1

If (C > 1) Then
    filas3 = filas3 - 2
End If
seleccion3 = "A2:" & v3 & filas3
xlBook3.Charts.Add
Set grafica3 = xlBook3.ActiveChart
exlApp3.Visible = True
With grafica3
    .ChartType = xlXYScatterSmooth
    .SetSourceData Source:=xlSheet3.Range(seleccion3), PlotBy _
:=xlColumns
    .Location Where:=xlLocationAsNewSheet
End With

nc = C
Do While (nc >= 2)
    grafica3.SeriesCollection(nc - 1).Name = "=Hoja1!R1C" & nc
    nc = nc - 1
Loop

With grafica3.Axes(xlCategory)
    .HasMajorGridlines = True
    .HasMinorGridlines = False
End With

With grafica3
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Acc.Theta"
    .Axes(xlValue, xlPrimary).HasTitle = True
    If Form8.Option3 = True Then
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Frec.(Hz)"
    End If
    If Form8.Option4 = True Then
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Tiempo(ms)"
    End If
End With

With grafica3.Axes(xlCategory)
    .MinimumScaleIsAuto = True
    .MaximumScale = 1

```

```

.MinorUnitIsAuto = True
.MajorUnitIsAuto = True
.Crosses = xlAutomatic
.ReversePlotOrder = False
.ScaleType = xlLinear
.DisplayUnit = xlNone
End With

With grafica3.Axes(xlValue, xlPrimary)
.HasMajorGridlines = True
.HasMinorGridlines = False
.MinimumScale = (0.95) * mini
.MaximumScale = (1.05) * maxi
.MinorUnitIsAuto = True
.MajorUnitIsAuto = True
.Crosses = xlAutomatic
.ReversePlotOrder = False
.ScaleType = xlLinear
.DisplayUnit = xlNone
End With

With grafica3.ChartArea.Border
.ColorIndex = 16
.Weight = xlThin
.LineStyle = xlContinuous
End With

With grafica3.Axes(xlValue).TickLabels
.NumberFormat = "0.000"
.Font.Name = "Arial"
.Font.Bold = True
.Font.Size = 11
.Font.Strikethrough = False
.Font.Underline = xlUnderlineStyleNone
End With

exlApp3.Visible = True
MsgBox ("Graficado")
I = 0
MSComm1.InBufferCount = 0
MSComm1.RThreshold = 1
pb = "0"
MSComm1.InputLen = 1
If preg3 = 7 And conjunto = 0 Then
xlBook3.Close
exlApp3.Quit

```

```

End If
Form1.MSComm1.Output = Chr(&H73)
Form1.MSComm1.Output = Chr(&H61)
exlApp3.EnableEvents = False
Set filas3 = Nothing
Set colum3 = Nothing
Set objfilas3 = Nothing
Set grafica3 = Nothing
Set xlSheet3 = Nothing
Set xlBook3 = Nothing
Set exlApp3 = Nothing
Set sel3 = Nothing
End If
End If
End If
End Sub

```

- ***Código para Formulario2 (Monitor para ajuste en un punto):***

```

Private Sub Command1_Click()
    Form1.MSComm1.Output = Chr(&H73)
    Form1.MSComm1.Output = Chr(&H61)
    Unload Form2
    Form1.Show
End Sub

Private Sub Command2_Click()

    tester = tester + 1

    If tester = 1 Then
        Dim d
        Form2.Command2.Caption = "Detener"
        Form2.Command1.Enabled = False
        Form2.Command4.Enabled = False
        Form2.Option1.Enabled = False
        Form2.Option2.Enabled = False
        Form2.Option3.Enabled = False
        Form2.Option4.Enabled = False
        If Option1.Value = True Then
            retardo (1500000)
            Form1.MSComm1.Output = Chr(&HCA)    'entrada de modo tensión
            Form1.MSComm1.Output = Chr(&H1)
        End If
        If Option2.Value = True Then

```



```

        retardo (1500000)
        Form1.MSComm1.Output = Chr(&HCC)      'entrada modo corriente
        Form1.MSComm1.Output = Chr(&H1)
    End If
    If Form4.Option1 = True Then                'si está en modo automático
        If Option1.Value = True Or Option2.Value = True Then
            retardo (1500000)
            Form1.MSComm1.Output = Chr(&H73)    'ganancia a maximo
            Form1.MSComm1.Output = Chr(&H67)
        End If
    End If
    If Option1.Value = True Or Option2.Value = True Then
        Form10.Visible = True
        Form10.Show
    End If
    If Option3.Value = True Or Option4.Value = True Then
        Form2.Timer1.Interval = 1000
        Form2.Timer1 = True
    End If
End If

If tester = 2 Then

    tester = 0
    Form2.Timer1 = False
    Form2.Command2.Caption = "Adquirir"
    Form2.Label9.Caption = ""
    Form2.Label1.Caption = ""
    Form2.Label2.Caption = ""
    Form4.Slider1.Enabled = False
    Form2.Command1.Enabled = True
    Form2.Command4.Enabled = True
    Form2.Option1.Enabled = True
    Form2.Option2.Enabled = True
    Form2.Option3.Enabled = True
    Form2.Option4.Enabled = True
    Form1.MSComm1.Output = Chr(&H62)      'Baja ganancia al mínimo
    Form1.MSComm1.Output = Chr(&H67)
    retardo (2000000)
    Form1.MSComm1.Output = Chr(&H73)      'edo sistema
    Form1.MSComm1.Output = Chr(&H61)
End If

End Sub

```

```

Private Sub Command3_Click()
    Form12.RichTextBox1.FileName = "C:\AFE\multi.rtf"
    Form12.Show
End Sub

Private Sub Command4_Click()
    Form1.MSComm1.Output = Chr(&H73)    'edo sistema
    Form1.MSComm1.Output = Chr(&H61)
    form9.Visible = True
End Sub

Private Sub Timer1_Timer()

    If Form2.Option1.Value = True Or Form2.Option2.Value = True Then

        If Form2.Option1.Value = True Then 'medición de tensión
            If Form10.Option1 = True Then 'Para medicion DC
                Label9.Caption = "DC"
                Form1.MSComm1.Output = Chr(&HAA)
                Form1.MSComm1.Output = Chr(&H1)
            End If
            If Form10.Option2 = True Then          'Para medicion AC
                Label9.Caption = "AC"
                Form1.MSComm1.Output = Chr(&HAA)
                Form1.MSComm1.Output = Chr(&H1)
            End If
        End If
        If Form2.Option2.Value = True Then 'medición de corriente
            If Form10.Option1 = True Then 'Para medicion DC
                Label9.Caption = "DC"
                Form1.MSComm1.Output = Chr(&HAA)
                Form1.MSComm1.Output = Chr(&H1)
            End If
            If Form10.Option2 = True Then 'Para medicion AC
                Label9.Caption = "AC"
                Form1.MSComm1.Output = Chr(&HAA)
                Form1.MSComm1.Output = Chr(&H1)
            End If
        End If

        If Form2.Option1.Value = True Then
            Label1.Caption = "V"
        End If
        If Form2.Option2.Value = True Then
            Label1.Caption = "mA"
        End If
    End If

```

```

End If

End If

If Form2.Option3.Value = True Or Form2.Option4.Value = True Then 'medición de frecuencia o período
    Form2.Label9.Caption = ""
    Form1.MSComm1.Output = Chr(&HDD)
    Form1.MSComm1.Output = Chr(&H1)
    If Form2.Option3.Value = True Then
        Label1.Caption = "Hz"
    End If
    If Form2.Option4.Value = True Then
        Label1.Caption = "ms"
    End If

End If

End Sub

```

- ***Código para Formulario3 (Control de movimiento del Motor):***

```

Private Sub Command1_Click()
    If Form1.MSComm1.PortOpen = True Then
        Dim datsal As Variant
        Dim C As Variant
        datsal = Abs(Val(Trim(Text3.Text)))
        C = Val(Trim(Text4.Text))
        If datsal = 0 Then
            MsgBox ("Introduzca caracteres válidos"), vbCritical
            Text3.Text = ""
            Text3.SetFocus
        End If
        If datsal > &HFF Then
            MsgBox ("Excesiva cantidad de pasos, máximo 255")
        End If
        If datsal > C Then
            MsgBox ("Excesiva cantidad de pasos, máximo " & C)
        End If
        If datsal <= &HFF And C >= datsal Then
            Dim s
            s = Chr(&H70)
            Form1.MSComm1.Output = s
            datsal = Chr(datsal)
            Form1.MSComm1.Output = datsal
        End If
    End If
End Sub

```

```

    End If
End Sub

Private Sub Command10_Click()
    Form12.RichTextBox1.FileName = "C:\AFE\mot.rtf"
    Form12.Show
End Sub

Private Sub Command2_Click()
    If Form1.MSComm1.PortOpen = True Then
        Dim datsal As Variant
        datsal = Chr(&H7A)
        Form1.MSComm1.Output = datsal
        datsal = Chr(&H37)
        Form1.MSComm1.Output = datsal
    End If
End Sub

Private Sub Command3_Click()

    If Form1.MSComm1.PortOpen = True Then
        Dim datsal As Variant
        Dim C As Variant
        datsal = Val(Trim(Text3.Text))
        C = Val(Trim(Text5.Text))
        If datsal > &HFF Or C < datsal Then
            MsgBox ("Excesiva cantidad de pasos")
        End If
        If datsal <= &HFF And C >= datsal Then
            Dim s
            s = Chr(&H80)
            Form1.MSComm1.Output = s
            datsal = Chr(datsal)
            Form1.MSComm1.Output = datsal
        End If
    End If
End Sub

Public Sub Command4_Click()
    If Form1.MSComm1.PortOpen = True Then
        Dim datsal As Variant
        datsal = Chr(&H74)
        Form1.MSComm1.Output = datsal
        datsal = Chr(&H69)
        Form1.MSComm1.Output = datsal
    End If
End Sub

```

```

        Command4.Enabled = True
    End If
End Sub

Private Sub Command5_Click()
    If Form1.MSComm1.PortOpen = True Then
        Dim datsal As Variant
        datsal = Chr(&H74)
        Form1.MSComm1.Output = datsal
        datsal = Chr(&H64)
        Form1.MSComm1.Output = datsal
    End If
End Sub

Private Sub Command6_Click()
    c6 = c6 + 1
    If Form1.MSComm1.PortOpen = True And c6 = 1 Then
        Dim datsal As Variant
        Dim r As Integer
        datsal = Chr$(p)
        Form1.MSComm1.Output = datsal
        datsal = Chr$(m)
        Form1.MSComm1.Output = datsal
        Command6.Caption = "Proseguir..."
        Picture7.Picture = LoadPicture("C:\AFE\comenzar.BMP")
        r = MsgBox("Acción en pausa.", 0, "Información")
    End If

    If Form1.MSComm1.PortOpen = True And c6 = 2 Then
        c6 = 0
        Command6.Caption = "Pausa"
        datsal = Chr$(p)
        Form1.MSComm1.Output = datsal
        datsal = Chr$(p)
        Form1.MSComm1.Output = datsal
        Picture7.Picture = LoadPicture("C:\AFE\pausa.BMP")
    End If

End Sub

Private Sub Command7_Click()
    If Form1.MSComm1.PortOpen = True Then
        Dim datsal As Variant
        datsal = Chr(&H64)
        Form1.MSComm1.Output = datsal
    End If
End Sub

```

```

        datsal = Chr(&H61)
        Form1.MSComm1.Output = datsal
    End If
End Sub

Private Sub Command8_Click()
    Unload Form3
    Form1.Show
End Sub

Private Sub Command9_Click()
    Form1.Show
End Sub

Private Sub Form_Load()

    If Form1.MSComm1.PortOpen = False Then
        MsgBox ("Puerto Comm Inhabilitado")
        Form6.Show
    End If

End Sub

```

- ***Código para Formulario4 (Menú Potenciómetros):***

```

Private Sub Form_Load()

    If auto = False Then
        Option2.Value = True
        Slider1.Enabled = False
        Slider2.Enabled = False
    Else
        Option1.Value = True
        Slider1.Enabled = True
        Slider2.Enabled = True
    End If

End Sub

Private Sub slider1_scroll()
    Dim acc
    Dim cont1
    Dim vg
    acc = Slider1.Value
    Open "C:\AFE\tablaGanancia.txt" For Input As #1
    Do Until cont1 = acc + 1 Or EOF(1)

```

```

        Line Input #1, vg
        cont1 = cont1 + 1
    Loop
    Close #1
    Text1.Text = Round(0.018 * Val(vg), 3)
    cont1 = 0
End Sub
Private Sub slider2_scroll()
    Dim off
    Dim cont2
    Dim ped
    off = Slider2.Value
    Open "C:\AFE\tablaPedestal.txt" For Input As #1
    Do Until cont2 = off + 1 Or EOF(1)
        Line Input #1, ped
        cont2 = cont2 + 1
    Loop
    Close #1
    Text2.Text = CStr(Round(2 * Val(ped), 2))
End Sub
Private Sub slider1_change()
    If Option2.Value = True Then
        Dim valor
        valor = Chr(Slider1.Value)
        Form1.MSComm1.Output = Chr(&H67)
        Form1.MSComm1.Output = valor
        retardo (1000000)
        Form1.MSComm1.Output = Chr(&H73)
        Form1.MSComm1.Output = Chr(&H61)
    End If
End Sub
Private Sub slider2_change()
    If Option2.Value = True Then
        Dim valor
        valor = Slider2.Value
        Form1.MSComm1.Output = Chr(&H6F)
        Form1.MSComm1.Output = Chr(valor)
        retardo (1000000)
        Form1.MSComm1.Output = Chr(&H73)
        Form1.MSComm1.Output = Chr(&H61)
    End If
End Sub

```

- **Código para Formulario5 (A excitación Externa):**

```

Private Sub Command1_Click()
    Unload Form5
    Form1.Show
End Sub

Private Sub Command3_Click()
    Form12.RichTextBox1.FileName = "C:\AFE\vco.rtf"
    Form12.Show
End Sub

Private Sub Command4_Click()
    Static t As Integer
    t = t + 1
    If (t = 1) Then
        Shape1.FillColor = RGB(0, 230, 0)
        Command4.Caption = "Apagar"
        Slider1.Enabled = True
    End If
    If (t = 2) Then
        Shape1.FillColor = RGB(230, 0, 0)
        Command4.Caption = "Encender"
        Slider1.Enabled = False
        Slider1.Value = 0
        Text1.Text = "0"
        t = 0
        Form1.MSComm1.Output = Chr(&H77)
        Form1.MSComm1.Output = Chr(&H0)
    End If
End Sub

Private Sub Form_Load()
    Slider1.Enabled = False
    Shape1.FillColor = RGB(230, 0, 0)
End Sub

Private Sub slider1_scroll()
    Text1.Text = Round(-(3.95 / 10) * Slider1.Value, 1)
End Sub

Private Sub slider1_change()
    Dim valor
    valor = Chr(Slider1.Value)
    Form1.MSComm1.Output = Chr(&H77)
    Form1.MSComm1.Output = valor
End Sub

```


- ***Código para Formulario6 (Indicar columnas para cálculo de error):***

```
Private Sub Command1_Click()
    Form12.RichTextBox1.FileName = "C:\AFE\col.rtf"
    Form12.Show
End Sub
```

```
Private Sub Command2_Click()
    Dim texto
    Dim cc
    Dim p
    Dim col
    Dim ha As Excel.Application
    Dim libro As Workbook
    Dim hoja As Worksheet
    Dim area As Range
    Dim grafer As Excel.Chart
    Dim nele
    Dim at
    Dim a
    Dim b
    Dim d
    Dim ms
    Dim unid
    Dim m
    Dim ncol
    Dim prueba
    Dim xp
    Dim ind
```

```
Set ha = GetObject("Libro1").Application
Set libro = ha.Workbooks(1)
Set hoja = libro.Worksheets(1)
Set area = hoja.Range("A2").CurrentRegion
```

```
If Form8.Option1 = True Then
    unid = "(V)"
    If Form8.Option7 = True Then
        unid = unid & "DC"
    Else
        unid = unid & "AC"
    End If
End If
If Form8.Option2 = True Then
```

```

    unid = "(mA)"
    If Form8.Option7 = True Then
        unid = unid & "DC"
    Else
        unid = unid & "AC"
    End If
End If
If Form8.Option3 = True Then
    unid = "(Hz)"
End If
If Form8.Option4 = True Then
    unid = "(ms)"
End If
nele = area.Rows.Count
at = -1
a = "A" & CStr(nele + at)
b = "Z" & CStr(nele + at)

Set seco = hoja.Range(a & ":" & b)

Form6.Visible = False

With seco
    .Font.Bold = True
End With

at = 2
a = "A" & CStr(nele + at)
b = "Z" & CStr(nele + at)

Set seco = hoja.Range(a & ":" & b)

With seco
    .Font.Bold = True
End With

texto = Trim(Form6.Text1.Text)
cc = Len(texto)
p = 1
Do While p <= cc
    col = UCase(Mid(texto, p, 1))
    p = p + 1
    xp = col & "2"
    prueba = hoja.Evaluate(xp)
    If Asc(col) >= 66 And Asc(col) <= 90 Then

```

```

If prueba <> "" Then
    a = col & CStr(nele + at)
    hoja.Evaluate(a).Value = "Error Absoluto " & col & " " & unid
    hoja.Columns.AutoFit
    Do While at <= nele - 2
        at = at + 1
        a = col & CStr(nele + at)
        b = "B" & CStr(at - 1)
        d = col & CStr(at - 1)
        hoja.Evaluate(a).Value = Round((hoja.Evaluate(b) - hoja.Evaluate(d)), 3)
    Loop
    at = 2
End If
End If
Loop
a = "A1: " & "A" & CStr(nele - 2)
Set seco = hoja.Range(a)
seco.Copy (hoja.Cells(nele + 2, 1))
a = "A" & CStr(nele + 2)
ncol = area.Columns.Count
Open "C:\AFE\TablaAlfabeto.txt" For Input As #1
    Do Until m = ncol Or EOF(1)
        Line Input #1, col
        m = m + 1
    Loop
Close #1
m = 0
ind = ncol
a = col & CStr(nele + 2)
Do While ind >= 1 And ncol = ind
    prueba = hoja.Evaluate(a)
    If prueba = "" Then
        ncol = ncol - 1
        Open "C:\AFE\TablaAlfabeto.txt" For Input As #1
            Do Until m = ncol Or EOF(1)
                Line Input #1, col
                m = m + 1
            Loop
        Close #1
        m = 0
        a = col & CStr(nele + 2)
    End If
    ind = ind - 1
Loop
b = col & CStr(2 * (nele + 1) - 3)

```

```

a = "A" & CStr(nele + 2)
a = a & ":" & b
Set grafer = libro.Charts.Add
grafer.Name = "Errores Absolutos"

With grafer
.ChartType = xlXYScatterSmooth
.SetSourceData Source:=hoja.Range(a), PlotBy _
:=xlColumns
.Location Where:=xlLocationAsNewSheet
End With

libro.Worksheets(3).Visible = False
libro.Worksheets(2).Visible = False

With grafer.Axes(xlCategory)
.HasMajorGridlines = True
.HasMinorGridlines = False
End With

With grafer
.HasLegend = True
.Axes(xlCategory, xlPrimary).HasTitle = True
.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Acc.Theta"
.Axes(xlValue, xlPrimary).HasTitle = True
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Error Absoluto"
End With

With grafer.Axes(xlCategory)
.MinimumScaleIsAuto = True
.MaximumScale = 1
.MinorUnitIsAuto = True
.MinimumScale = 0
.MajorUnitIsAuto = True
.Crosses = xlAutomatic
.ReversePlotOrder = False
.ScaleType = xlLinear
.DisplayUnit = xlNone
End With

With grafer.Axes(xlValue, xlPrimary)
.HasMajorGridlines = True
.HasMinorGridlines = False
.MinorUnitIsAuto = True
.MajorUnitIsAuto = True

```

```

.Crosses = xlAutomatic
.ReversePlotOrder = False
.ScaleType = xlLinear
.DisplayUnit = xlNone
End With

With grafer.ChartArea.Border
.ColorIndex = 16
.Weight = xlThin
.LineStyle = xlContinuous
End With

With grafer.Axes(xlValue).TickLabels
.NumberFormat = "0.000"
.Font.Name = "Arial"
.Font.Bold = True
.Font.Size = 11
.Font.Strikethrough = False
.Font.Underline = xlUnderlineStyleNone
End With
ha.Visible = True
ms = MsgBox("Error(es) Graficado(s)", vbOKOnly)
Form6.Text1.Text = ""
libro.Close
ha.Quit
Form8.Option1.Enabled = True
Form8.Option2.Enabled = True
Form8.Option3.Enabled = True
Form8.Option4.Enabled = True
Form8.Option5.Enabled = True
Form8.Option6.Enabled = True
Form8.Option7.Enabled = True
Form8.Option8.Enabled = True
Form8.Combo1.Enabled = True
Form8.Combo2.Enabled = True
Form8.Combo3.Enabled = True
Form8.Command1.Enabled = True
Form8.Command7.Enabled = False
Form8.Command2.Enabled = True
Form3.Enabled = True
Form4.Enabled = True
conjunto = 0
funesc = False
Form3.Enabled = True
C = 0

```

```

Set area = Nothing
Set grafer = Nothing
Set seco = Nothing
Set hoja = Nothing
Set libro = Nothing
Set ha = Nothing

```

```
End Sub
```

```

Private Sub Command3_Click()
    Form6.Text1 = ""
    Text1.SetFocus
End Sub

```

- ***Código para Formulario7 (Visor del Sistema):***

```

Private Sub Command11_Click()

    Dim salida

    If Form1.MSComm1.PortOpen = True Then

        Dim x
        x = 0
        salida = Chr(&H6D)
        Form1.MSComm1.Output = salida
        salida = Chr(&H74)
        Form1.MSComm1.Output = salida
        Do While x < 10000
            x = x + 1
        Loop
        salida = Chr(&H73)
        Form1.MSComm1.Output = salida
        salida = Chr(&H61)
        Form1.MSComm1.Output = salida

    End If

    If Form1.MSComm1.PortOpen = False Then

        MsgBox ("Puerto Comm Inhabilitado")
        Form6.Show

    End If

```

```

End Sub
Private Sub Form_Load()

    If Form1.MSComm1.PortOpen = False Then

        MsgBox ("Puerto Comm Inhabilitado")
        Form6.Show

    End If

End Sub

```

- ***Código para Formulario8 (Módulo Graficador):***

```

Private Sub Command1_Click()
    Unload Form8
    Form1.Show
End Sub

Private Sub Command2_Click()
    Form1.MSComm1.Output = Chr(&H73)
    Form1.MSComm1.Output = Chr(&H61)
    If conjunto = 1 Then
        conjunto = conjunto + 1
        C = C + 1
    End If
    If conjunto = 0 Then
        If (nupasd <> 0) Then
            Form1.MSComm1.Output = Chr(&H7A)
            Form1.MSComm1.Output = Chr(&H37)
        End If
        preg3 = MsgBox("¿La siguiente adquisición formará parte de un conjunto de experiencias? ", vbYesNo)
        If preg3 = 6 Then
            conjunto = conjunto + 1
            pet = True
        End If
        If preg3 = 7 Then
            conjunto = 0
            pet = False
        End If
        If Form11.Text1.Text <> "" Then

            Dim vez
            Dim w
            Dim valor

```

```

Dim sc
Dim dv
Dim f
Dim k
Dim sel As Object
Dim ar As Range
Dim gt As Chart
Dim canf
Dim area

Set xlApp = CreateObject("Excel.Application")
Set xlBook = xlApp.Workbooks.Add
Set xlSheet = xlBook.Worksheets(1)
C = 1

Form4.Enabled = False
xlBook.Worksheets(3).Visible = False
xlBook.Worksheets(2).Visible = False

dv = "DIM " & "t"
t = totalpaso
sv = "t" & " = " & CStr(totalpaso)
ScriptControl1.AddCode dv
ScriptControl1.AddCode sv

dv = "DIM " & "x"
Text2.Text = totalpaso
sv = Trim(Text3.Text) & " = " & Trim(Text2.Text)
ScriptControl1.AddCode dv
ScriptControl1.AddCode sv

dv1 = "DIM " & "T"
sv1 = "T" & " = " & "x/t"
Form11.ScriptControl1.AddCode dv1
Form11.ScriptControl1.AddCode sv1

Form8.Option1.Enabled = False
Form8.Option2.Enabled = False
Form8.Option3.Enabled = False
Form8.Option4.Enabled = False
Form8.Option5.Enabled = False
Form8.Option6.Enabled = False
Form8.Option7.Enabled = False
Form8.Option8.Enabled = False
Form8.Combo1.Enabled = False

```



```

Form8.Combo2.Enabled = False
Form8.Combo3.Enabled = False
If Form8.Option5.Value = True Then
    w = 1
End If
If Form8.Option6.Value = True Then
    w = Val(Trim(Form8.Combo2.Text)) * 10 ^ (Val(Trim(Form8.Combo3.Text)))
End If
vez = totalpaso

Set sel = xlApp.Range("A1:Z1")
xlSheet.Columns("A:Z").NumberFormat = "0.000"
With sel
    .Font.Bold = True
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .ShrinkToFit = False
    .MergeCells = False
End With
If Form8.Option1 = True Then
    xlSheet.Cells(1, C + 1).Value = "Fnc.Teórica(V) "
End If

If Form8.Option2 = True Then
    xlSheet.Cells(1, C + 1).Value = "Fnc.Teórica(mA)"
End If

If Form8.Option3 = True Then
    xlSheet.Cells(1, C + 1).Value = "Fnc.Teórica(Hz) "
End If
If Form8.Option4 = True Then
    xlSheet.Cells(1, C + 1).Value = "Fnc.Teórica(ms) "
End If
xlSheet.Cells(1, 1) = "Acc.Theta"
xlSheet.Columns(1).AutoFit
xlSheet.Columns(2).AutoFit
f = 2
k = 1
dv = "DIM " & "x"
sv = Trim(Text3.Text) & " = " & Trim(Text2.Text)
ScriptControl1.AddCode dv
ScriptControl1.AddCode sv

```

```

dv = "DIM " & "t"
sv = "t" & " = " & CStr(totalpaso)
ScriptControl1.AddCode dv
ScriptControl1.AddCode sv

dv = "DIM " & "T"
sv = "T" & " = " & "x/t"
ScriptControl1.AddCode dv
ScriptControl1.AddCode sv
xlApp.Visible = True
Dim final
Do While (vez > 0)
    xlSheet.Cells(f, k) = Round(vez / totalpaso, 3)
    xlSheet.Cells(f, k + 1) = Round(ScriptControl1.Eval(Form11.Text1.Text), 3)
    f = f + 1
    Text2.Text = vez - w
    vez = Val(Text2.Text)

    dv = "DIM " & "x"
    sv = Trim(Text3.Text) & " = " & Trim(Text2.Text)
    ScriptControl1.AddCode dv
    ScriptControl1.AddCode sv

    dv = "DIM " & "t"
    sv = "t" & " = " & CStr(totalpaso)
    ScriptControl1.AddCode dv
    ScriptControl1.AddCode sv

    dv = "DIM " & "T"
    sv = "T" & " = " & "x/t"
    ScriptControl1.AddCode dv
    ScriptControl1.AddCode sv

If (vez <= 0) Then

    dv = "DIM " & "x"
    Text2.Text = "0"
    sv = Trim(Text3.Text) & " = " & Trim(Text2.Text)
    ScriptControl1.AddCode dv
    ScriptControl1.AddCode sv

    dv = "DIM " & "t"
    sv = "t" & " = " & CStr(totalpaso)

```

```

ScriptControl1.AddCode dv
ScriptControl1.AddCode sv

dv = "DIM " & "T"
Text2.Text = "0"
sv = "T" & " " & "x/t"
ScriptControl1.AddCode dv
ScriptControl1.AddCode sv

xlSheet.Cells(f, k) = 0
End If
If Form8.Option2 = True Then
    xlSheet.Cells(f, k + 1) = Round(ScriptControl1.Eval(Form11.Text1.Text), 3)
Else
    xlSheet.Cells(f, k + 1) = Round(ScriptControl1.Eval(Form11.Text1.Text), 3)
End If
Loop
Set ar = xlSheet.Range("A2").CurrentRegion
Set gt = xlBook.Charts.Add
mini = ar.Application.WorksheetFunction.Min(ar)
maxi = ar.Application.WorksheetFunction.Max(ar)
canf = ar.Rows.Count
area = "A2:" & "B" & CStr(canf)
With gt
    .ChartType = xlXYScatterSmooth
    .SetSourceData Source:=xlSheet.Range(area), PlotBy _
:=xlColumns
    .Location Where:=xlLocationAsNewSheet
End With

With gt.Axes(xlCategory)
    .HasMajorGridlines = True
    .HasMinorGridlines = False
End With

With gt
    .HasLegend = True
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Acc.Theta"
    .Axes(xlValue, xlPrimary).HasTitle = True
    If Form8.Option1 = True Then
        If Form8.Option7 = True Then
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Tensión D.C.(V)"
        End If
        If Form8.Option8 = True Then

```

```

        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Tensión A.C.(V)"
    End If
End If
If Form8.Option2 = True Then
    If Form8.Option7 = True Then
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Corriente D.C.(mA)"
    End If
    If Form8.Option8 = True Then
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Corriente A.C.(mA)"
    End If
End If
End With
gt.SeriesCollection(1).Name = "=Hoja1!R1C2"
With gt.Axes(xlCategory)
    .MinimumScaleIsAuto = True
    .MaximumScale = 1
    .MinorUnitIsAuto = True
    .MinimumScale = 0
    .MajorUnitIsAuto = True
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With

With gt.Axes(xlValue, xlPrimary)
    .HasMajorGridlines = True
    .HasMinorGridlines = False
    .MinimumScale = (0.95) * mini
    .MaximumScale = (1.05) * maxi
    .MinorUnitIsAuto = True
    .MajorUnitIsAuto = True
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With

With gt.ChartArea.Border
    .ColorIndex = 16
    .Weight = xlThin
    .LineStyle = xlContinuous
End With

With gt.Axes(xlValue).TickLabels

```

```

.NumberFormat = "0.000"
.Font.Name = "Arial"
.Font.Bold = True
.Font.Size = 11
.Font.Strikethrough = False
.Font.Underline = xlUnderlineStyleNone
End With
xlApp.Visible = True
Form1.MSComm1.Output = Chr(&H73)
Form1.MSComm1.Output = Chr(&H61)
Set ar = Nothing
Set sel = Nothing
Set gt = Nothing
Set xlSheet = Nothing
Set xlBook = Nothing
Set xlApp = Nothing
funesc = True
conjunto = conjunto + 1
Form11.Text1.Text = ""
C = 2
Else
    MsgBox ("Primero debe Introducir la función teórica que describe el comportamiento, luego podrá realizar el estudio")
    If Form8.Option6 = True And (Form8.Combo2.Text = "?" Or Form8.Combo3.Text = "?") Then
        MsgBox ("Seleccione el numero de pasos mecánicos a dar entre cada muestra")
    End If
    If Form8.Option5 = True And Form8.Combo1.Text = "?" Then
        MsgBox ("Seleccione el numero de muestras a tomar en cada paso mecánico")
    End If
    Form11.Visible = True
    Form11.Show
    Form11.Text1.SetFocus
    funesc = False
    conjunto = 0
End If
End If
If conjunto >= 2 Then
    Form3.Enabled = False
    If pet = True Then
        preg3 = MsgBox("¿La siguiente adquisición formará parte del conjunto previamente abierto de experiencias? ",
vbYesNo)
    Else
        preg3 = 7
    End If
    If preg3 = 7 Then

```

```

Dim perr
perr = MsgBox("¿Desea realizar cálculos del error absoluto para alguna(s) de las(s) adquisiciones?", vbYesNo)
If perr = 6 Then
    Form6.Show
    Form6.Text1.SetFocus
    funesc = False
End If

If perr = 7 Then
    Dim ha As Excel.Application
    Dim libro As Workbook
    Dim hoja As Worksheet

    Set ha = GetObject("Libro1").Application
    Set libro = ha.Workbooks(1)
    Set hoja = libro.Worksheets(1)

    Form8.Option1.Enabled = True
    Form8.Option2.Enabled = True
    Form8.Option3.Enabled = True
    Form8.Option4.Enabled = True
    Form8.Option5.Enabled = True
    Form8.Option6.Enabled = True
    Form8.Option7.Enabled = True
    Form8.Option8.Enabled = True
    Form8.Combo1.Enabled = True
    Form8.Combo2.Enabled = True
    Form8.Combo3.Enabled = True
    Form8.Command1.Enabled = True
    Form8.Command7.Enabled = False
    Form8.Command2.Enabled = True
    Form4.Enabled = False
    conjunto = 0
    funesc = False
    Form3.Enabled = True
    Form4.Enabled = True
    libro.Close
    ha.Quit
    C = 0
    Set ha = Nothing
    Set libro = Nothing
End If
End If
If preg3 = 6 Then
    conjunto = conjunto + 1

```

```

        funesc = True
    End If
End If
If funesc = True And (Option1.Value = True Or Option2.Value) = True Then
    retardo (1500000)
    If Option8.Value = True Then
        Form1.MSComm1.Output = Chr(&H61)
        Form1.MSComm1.Output = Chr(&H63)
    End If
    If Option7.Value = True Then
        Form1.MSComm1.Output = Chr(&H64)
        Form1.MSComm1.Output = Chr(&H63)
    End If
End If
If funesc = True And Option5.Value = True And (Option1.Value = True Or Option2.Value = True) Then
    If nupasi > 0 And Combo1.Text <> "?" And Val(Combo1.Text) <= cantmue Then

        If Val(Combo1.Text) <= cantmue And Option1.Value = True Or Option2.Value = True Then

            If Form4.Option1 = True Then
                Form1.MSComm1.Output = Chr(&H73)
                Form1.MSComm1.Output = Chr(&H67)
            End If
            retardo (1500000)
            Form8.Command1.Enabled = False
            Form8.Command7.Enabled = True
            Form8.Command2.Enabled = False
            Form1.MSComm1.Output = Chr(&H76)
            cantmue = (Val(Combo1.Text))
            Form1.MSComm1.Output = Chr(cantmue)

        End If
    Else

        If nupasi = 0 Then

            MsgBox "Por favor, Realice búsqueda de tope mecánico", vbInformation
            Form3.Show

        End If

        If Combo1.Text = "?" Then

            MsgBox "Por favor, Seleccione la cantidad de muestras por paso", vbInformation

```

```

End If

If Val(Combo1.Text) > cantmue Then

    MsgBox "Disminuya la cantidad de muestras requeridas", vbCritical
    Form8.Combo1.Enabled = True

End If

End If

End If

End If

If funesc = True And Option6.Value = True And (Option1.Value = True Or Option2.Value = True) Then
    Dim digito          'mide tensión ó corriente
    Dim mul
    Dim preg
    Form8.Command1.Enabled = False
    Form8.Command2.Enabled = False
    Form8.Command7.Enabled = True
    retardo (1500000)
    If Form4.Option1 = True Then
        Form1.MSComm1.Output = Chr(&H73)
        Form1.MSComm1.Output = Chr(&H67)
    End If
    digito = Val(Combo2.Text)
    digito = Hex(digito)
    mul = Val(Combo3.Text)
    mul = Hex(mul)
    preg = MsgBox("La cantidad de pasos entre muestras es: " & " " & Val(Combo2.Text) * 10 ^ (Val(Combo3.Text)),
vbOKCancel)
    If preg = vbOK Then
        If Option1.Value = True Then
            Form1.MSComm1.Output = Chr(&H76)
            cantmue = "&" & "H" & (digito & mul)
            cantmue = cantmue + &H29
            Form1.MSComm1.Output = Chr(cantmue)
        End If
        If Option2.Value = True Then
            Form1.MSComm1.Output = Chr(&H63)
            cantmue = "&" & "H" & (digito & mul)
            cantmue = cantmue + &H29
            Form1.MSComm1.Output = Chr(cantmue)
        End If
    End If
End If

```



```

End If

If funesc = True And Option5.Value = True Then
    If Option3.Value = True Or Option4.Value = True Then
        Form8.Command1.Enabled = False
        Form8.Command2.Enabled = False
        Form8.Command7.Enabled = True
        Form1.MSComm1.Output = Chr(&H66) 'mide frecuencia ó período
        Form1.MSComm1.Output = Chr(&H2)
    End If

End If

End If

If funesc = True And Option6.Value = True And (Option3.Value = True Or Option4.Value = True) Then

    Dim dig2
    Dim mul2
    Dim preg2
    Form8.Command1.Enabled = False
    Form8.Command2.Enabled = False
    Form8.Command7.Enabled = True
    dig2 = Val(Combo2.Text)
    dig2 = Hex(dig2)
    mul2 = Val(Combo3.Text)
    mul2 = Hex(mul2)
    preg2 = MsgBox("La cantidad de pasos entre muestras es: " & " " & Val(Combo2.Text) * 10 ^ Val(Combo3.Text),
vbOKCancel)
    If preg2 = vbOK Then
        Form1.MSComm1.Output = Chr(&H66)
        cantmue = "&" & "H" & (dig2 & mul2)
        cantmue = cantmue + &H29
        Form1.MSComm1.Output = Chr(cantmue)
    End If

End If

End Sub

Private Sub Command3_Click()
    Form12.RichTextBox1.FileName = "C:\AFE\graf.rtf"
    Form12.Show
End Sub

Private Sub Command6_Click()
    pausa = pausa + 1

```

```

If pausa = 1 Then
    Form1.MSComm1.Output = Chr(&H73)
    Form1.MSComm1.Output = Chr(&H70)
    Command6.Caption = "Reanudar"
End If

If pausa = 2 Then
    Form1.MSComm1.Output = Chr(&H73)
    Form1.MSComm1.Output = Chr(&H7)
    Command6.Caption = "Pausa"
    pausa = 0
End If

End Sub

Private Sub Command7_Click()
    Form1.MSComm1.Output = Chr(&H7A)
    Form1.MSComm1.Output = Chr(&H36)
    Form8.Command2.Enabled = True
    Form8.Command7.Enabled = False
    Form8.ProgressBar1.Value = 0
End Sub

Private Sub Form_Load()
    If nupasi = 0 Then
        MsgBox "Por favor, Realice búsqueda de tope mecánico ó devuelva el motor a su posición original", vbInformation
        Form3.Show
        Command6.Enabled = True
        Command7.Enabled = True
    End If

    If Form8.Option6 = True Then
        Form8.Combo1.Enabled = False
        Form8.Combo2.Enabled = True
        Form8.Combo3.Enabled = True
    End If

    If Form8.Option5 = True Then
        Form8.Combo1.Enabled = True
        Form8.Combo2.Enabled = False
        Form8.Combo3.Enabled = False
    End If

End Sub

Private Sub Option1_Click()
    Form1.MSComm1.Output = Chr(&HCA)
    Form1.MSComm1.Output = Chr(&H1)
End Sub

Private Sub Option2_Click()

```

```

Form1.MSComm1.Output = Chr(&HCC)
Form1.MSComm1.Output = Chr(&H1)
End Sub

Private Sub Option5_Click()
Form8.Combo1.Enabled = True
Form8.Combo2.Enabled = False
Form8.Combo3.Enabled = False
If totalpaso > 0 Then
    If Option1.Value = True Or Option2.Value = True Then

        Dim peticion
        Dim numdat
        cantmue = Round(((65535 / totalpaso) - 5) / 2, 0)
        numdat = (cantmue * 2 + 5) * totalpaso
        If numdat > 32000 Then
            cantmue = Round(((32000 / totalpaso) - 5) / 2, 0)
        End If
        peticion = MsgBox("Según la medición requerida, la cantidad máxima de muestras por paso es: " & " " & cantmue,
vbInformation)
    End If
    If Option1.Value = False And Option2.Value = False And Option3.Value = False And Option4.Value = False Then

        MsgBox ("Por favor, Seleccione el tipo de prueba a ejecutar")

    End If

Else

    MsgBox "Por favor, Realice búsqueda de tope mecánico", vbInformation
    Form3.Show

End If
End Sub

Private Sub Option6_Click()
Form8.Combo1.Enabled = False
Form8.Combo2.Enabled = True
Form8.Combo3.Enabled = True
End Sub

```

- ***Código para Formulario9 (Modificación de accionamiento actual):***

```

Private Sub Command1_Click()
Dim n1

```

```

Dim n
If Text1.Text = "" Then
    MsgBox "Por favor, ingrese valor de nuevo theta", vbOKOnly
    Text1.SetFocus
Else
    n = CCur(form9.Text1.Text)
    n1 = Val(form9.Text1.Text)
    If (n1 > 0) Then
        If (n >= 1) And (n < 10) Then
            n = 0.1 * n
        End If
        If (n >= 10) Then
            n = 0.01 * n
        End If
    End If
    If form9.Text1.Text <> "0" And form9.Text1.Text <> "1" Then
        If CCur(form9.Text1.Text) > 1 Or (n > 0) And (n1 <> 0) Then
            form9.Text1.Text = ""
            MsgBox "Por favor, ingrese nuevo theta válido e indique la posición decimal con coma.", vbCritical
            Text1.SetFocus
        End If

        Dim res
        Dim m
        If (n1 > 0) Then
            If (n >= 1) And (n < 10) Then
                n = 0.1 * n
            End If
            If (n >= 10) Then
                n = 0.01 * n
            End If
        End If
        res = MsgBox(("¿Quiere desplazar a ") & n & " el valor Theta ", vbYesNo)
        If res = 6 Then
            Dim aa
            Dim c1
            n = totalpaso * n
            aa = nupasi
            If (n > aa) Then
                c1 = Chr(&H34)
            Else
                c1 = Chr(&H43)
            End If
            If Abs(n - aa) > 255 Then
                Dim vez

```

```

Dim cont
Dim datsal1 As Variant
Dim t

cont = Abs(n - aa)
Do While (cont >= 255)
    cont = cont - 255
    vez = vez + 1
Loop
Do While (vez > 0)
    Form1.MSComm1.Output = c1
    datsal1 = Chr(&HFF)
    Form1.MSComm1.Output = datsal1
    vez = vez - 1
    t = 0
    Do While (t < 10000000)
        t = t + 1
    Loop
Loop
Form1.MSComm1.Output = c1
cont = Round(cont)
cont = Chr(cont)
Form1.MSComm1.Output = cont
form9.Visible = False
Form2.Visible = True

Else
    Dim datsal As Variant
    datsal = nupasi
    If (n > datsal) Then
        c2 = Chr(&H34)
    Else
        c2 = Chr(&H43)
    End If
    datsal = Abs(Round(nupasi - n))
    If datsal > 0 Then
        Form1.MSComm1.Output = c2
        datsal = Chr(datsal)
        Form1.MSComm1.Output = datsal
        form9.Visible = False
        Form2.Visible = True
        form9.Text1.Text = ""
    End If
    form9.Visible = False
    Form2.Visible = True

```

```

        End If
    End If
End If
Else

Dim res1
Dim m1
If (n1 > 0) Then
    If (n >= 1) And (n < 10) Then
        n = 0.1 * n
    End If
    If (n >= 10) Then
        n = 0.01 * n
    End If
End If
If form9.Text1.Text = "0" Then
    n = 0
End If
If form9.Text1.Text = "1" Then
    n = 1
End If
res1 = MsgBox("¿Quiere desplazar a ") & n & " el valor Theta ", vbYesNo)
If res1 = 6 Then
    Dim aa1
    Dim c11
    n = totalpaso * n
    aa1 = nupasi
    If (n > aa1) Then
        c11 = Chr(&H34)
    Else
        c11 = Chr(&H43)
    End If
    If Abs(n - aa1) > 255 Then
        Dim vez1
        Dim cont1
        Dim datsal11 As Variant
        Dim t1

        cont1 = Abs(n - aa1)
        Do While (cont1 >= 255)
            cont1 = cont1 - 255
            vez1 = vez1 + 1
        Loop
        Do While (vez1 > 0)
            Form1.MSComm1.Output = c11

```

```

        datsal11 = Chr(&HFF)
        Form1.MSComm1.Output = datsal11
        vez1 = vez1 - 1
        t1 = 0
        Do While (t1 < 10000000)
            t1 = t1 + 1
        Loop
    Loop
    Form1.MSComm1.Output = c11
    cont1 = Round(cont1)
    cont1 = Chr(cont1)
    Form1.MSComm1.Output = cont1
    form9.Visible = False
    Form2.Visible = True

Else
    Dim datsal111 As Variant
    datsal111 = nupasi
    If (n > datsal111) Then
        c21 = Chr(&H34)
    Else
        c21 = Chr(&H43)
    End If
    datsal111 = Abs(Round(nupasi - n))
    If datsal111 > 0 Then
        Form1.MSComm1.Output = c21
        datsal111 = Chr(datsal111)
        Form1.MSComm1.Output = datsal111
        form9.Visible = False
        Form2.Visible = True
        form9.Text1.Text = ""
    End If
    form9.Visible = False
    Form2.Visible = True
End If
End If
End If
Dim yv As Double
yv = 0
Do While (yv < 10000000)
    yv = yv + 1
Loop
Form1.MSComm1.Output = Chr(&H73)
Form1.MSComm1.Output = Chr(&H61)
End If

```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    form9.Text1.Text = ""
```

```
    Text1.SetFocus
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    form9.Visible = False
```

```
    Form2.Visible = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    form9.Text1.Text = ""
```

```
End Sub
```

- ***Código para Formulario10 (Modo de medición):***

```
Private Sub Command1_Click()
```

```
    If Form10.Option1 = True Then
```

```
        Form1.MSComm1.Output = Chr(&H64)    'modo DC+AC
```

```
        Form1.MSComm1.Output = Chr(&H63)
```

```
    End If
```

```
    If Form10.Option2 = True Then
```

```
        Form1.MSComm1.Output = Chr(&H61)    'modo AC
```

```
        Form1.MSComm1.Output = Chr(&H63)
```

```
    End If
```

```
    Form2.Timer1.Interval = 1500
```

```
    retardo (3000000)
```

```
    Form2.Timer1 = True
```

```
    Form2.Show
```

```
    Form10.Visible = False
```

```
End Sub
```

- ***Código para Formulario11 (Función Teórica):***

```
Private Sub Command1_Click()
```

```
    Dim error As Boolean
```

```
    Dim valor
```

```
    Dim dv1
```

```
    Dim sv1
```

```
    On Local Error GoTo Error_Ejecutar
```

```
    error = False
```

```
    If Form11.Text1.Text = "" Then
```

```
        MsgBox ("Introduzca la expresión de la función teórica")
```



```

Form11.Text1.SetFocus
Else

dv1 = "DIM " & "t"
sv1 = "t" & " " & CStr(totalpaso)
Form11.ScriptControl1.AddCode dv1
Form11.ScriptControl1.AddCode sv1

dv1 = "DIM " & "x"
Form8.Text2.Text = totalpaso
sv1 = Trim(Form8.Text3.Text) & " " & Trim(Form8.Text2.Text)
Form11.ScriptControl1.AddCode dv1
Form11.ScriptControl1.AddCode sv1

dv1 = "DIM " & "T"
sv1 = "T" & " " & "x/t"
Form11.ScriptControl1.AddCode dv1
Form11.ScriptControl1.AddCode sv1

valor = Form11.ScriptControl1.Eval(Form11.Text1.Text)
error = False
If error = False Then
    Dim m
    m = MsgBox("Ecuación teórica válida, pulse el botón 'Abrir-Cerrar-Continuar experiencia' para continuar", vbInformation)
    Form11.Visible = False
    Form8.Show
End If
Exit Sub
End If

Error_Ejecutar:
Dim m1
m1 = MsgBox("Se produjo un error de sintaxis, corrija la fórmula usada", vbCritical)
MsgBox "Se ha producido el siguiente error:" & vbCrLf & _
    Form11.ScriptControl1.error.Number & ", " & _
    Form11.ScriptControl1.error.Description
Err = 0
error = True
Text1.Text = ""
Text1.SetFocus
End Sub

Private Sub Command2_Click()
Text1.Text = ""
Text1.SetFocus

```

End Sub

Private Sub Command3_Click()

Form12.RichTextBox1.FileName = "C:\AFE\Funteo.rtf"

Form12.Show

End Sub

- ***Código para Formulario12 (Pantalla de Ayuda A.F.E.):***

Private Sub Command1_Click()

Form12.RichTextBox1.Text = ""

Form12.Visible = False

End Sub

ANEXO # 6

Lista de Partes Electrónicas del prototipo A.F.E.

Universidad Central de Venezuela. Facultad de Ingeniería, Escuela de Eléctrica

Br. Walter E. Senesi L., Prototipo A.F.E. 2006.

Componente	Cantidad	Referencia en el plano	Valor o # de parte
1	20	C1,C2,C3,C6,C7,C8,C9,C10, C16,C17,C18,C22,C23,C24, C26,C30,C32,C33,C34,C46	100nf
2	1	C4	22uf
3	4	C5,C19,C48,C50	2.2uf
4	1	C11	220nf
5	2	C12,C13	1000uf
6	1	C14	100uf
7	1	C15	68nf
8	2	C21,C20	10pf
9	1	C25	680nf
10	2	C27,C29	10uf
11	1	C28	1.5nf
12	1	C31	470pf
13	3	C35,C36,C41	470nf
14	8	C37,C38,C39,C40,C42,C43, C44,C45	150nf
15	1	C47	1uf
16	1	C49	22pf
17	10	D1,D3,D7,D8,D12,D14,D19, D20,D21,D22	D1N4148
18	1	D2	1N5221A
19	6	D4,D5,D10,D11,D13,D15	1N5231B
20	1	D6	1N4007
21	1	D9	RS204L
22	1	D18	1N4001
23	1	J1	Conector LA
24	1	J2	Conector LB
25	1	J3	Conector AC
26	1	J4	Conector IMotor
27	1	J5	Pulsador
28	1	J6	BNC para VCO
29	1	J7	Conector + I
30	1	J8	Conector - I
31	1	J9	Conector + Vin
32	1	J10	Conector - Vin
33	1	J11	Referencias
34	1	J12	40 Pines PIC
35	1	U10	TMP122
36	1	K1	RELAY DPDT
37	1	P1	Conector DB9
38	1	Q2	2N2222A
39	1	Q3	2N2222A
40	1	RS1	1Ω
41	6	R1,R3,R17,R19,R44,R76	6.8kΩ
42	6	R2,R4,R18,R20,R55,R56	1MΩ
43	13	R5,R6,R12,R21,R24,R34, R42,R43,R50,R68,R69,R70, R74	10kΩ

44	3	R7,R30,R72	5k Ω
45	2	R51,R8	20k Ω
46	6	R9,R10,R25,R28,R39,R47	330 Ω
47	4	R11,R27,R32,R77	5.1k Ω
48	6	R13,R14,R29,R31,R40,R48	7.5k Ω
49	1	R15	10M Ω
50	1	R16	160k Ω
51	1	R23	12k Ω
52	1	R26	500 Ω
53	3	R33,R53(12k Ω) R52(5.6k Ω)	R-PACK4
54	1	R35	1k Ω
55	2	R36,R45	100k Ω
56	2	R37,R46	1.3k Ω
57	1	R38	680 Ω
58	1	R41	1.2 Ω
59	1	R49	1.8M Ω
60	1	R54	4.7k Ω
61	2	R57,R58	11k Ω
62	9	R59,R60,R61,R62,R63,R64, R65,R66,R67	1 Ω
63	1	R71	50k Ω
64	1	R73	47k Ω
65	1	R75	24k Ω
66	1	U1	LM7812C/TO220
67	1	U2	LM339
68	1	U3	74LS00
69	1	U4	LM7812CT
70	2	U5,U8	LM7912C/TO220
71	1	U6	MC3479P
72	1	U7	LM7805C/TO3
73	1	U9	PIC16F877A
74	2	U23,U11	LF444
75	1	U13	LM393
76	2	U16,U14	74LS373
77	2	U15,U17	61256
78	1	U18	INA121
79	1	U19	MAX232A
80	1	U20	OPA4277
81	2	U22,U21	X9C503
82	1	Y1	Cristal 16MHz
83	1	1A	REF3025AIDBZT

LF444

Quad Low Power JFET Input Operational Amplifier

General Description

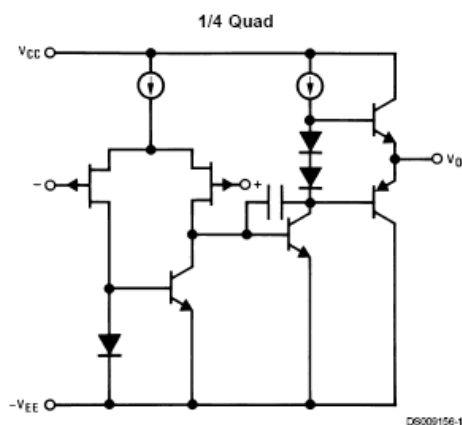
The LF444 quad low power operational amplifier provides many of the same AC characteristics as the industry standard LM148 while greatly improving the DC characteristics of the LM148. The amplifier has the same bandwidth, slew rate, and gain (10 k Ω load) as the LM148 and only draws one fourth the supply current of the LM148. In addition the well matched high voltage JFET input devices of the LF444 reduce the input bias and offset currents by a factor of 10,000 over the LM148. The LF444 also has a very low equivalent input noise voltage for a low power amplifier.

The LF444 is pin compatible with the LM148 allowing an immediate 4 times reduction in power drain in many applications. The LF444 should be used wherever low power dissipation and good electrical characteristics are the major considerations.

Features

- 1/4 supply current of a LM148: 200 μ A/Amplifier (max)
- Low input bias current: 50 pA (max)
- High gain bandwidth: 1 MHz
- High slew rate: 1 V/ μ s
- Low noise voltage for low power: 35 nV/ $\sqrt{\text{Hz}}$
- Low input noise current: 0.01 pA/ $\sqrt{\text{Hz}}$
- High input impedance: $10^{12}\Omega$
- High gain: 50k (min)

Simplified Schematic



Ordering Information

LF444XYZ

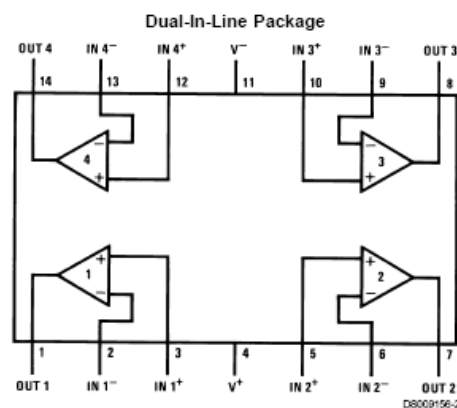
X indicates electrical grade

Y indicates temperature range

"M" for military, "C" for commercial

Z indicates package type "D", "M" or "N"

Connection Diagram



Top View

Order Number LF444CM, LF444CMX,
LF444ACN, LF 444CN or LF444MD/883
See NS Package Number D14E, M14A or N14A

Bi-FET™ and Bi-FET II™ are trademarks of National Semiconductor Corporation.



INA121

FET-Input, Low Power INSTRUMENTATION AMPLIFIER

FEATURES

- LOW BIAS CURRENT: $\pm 4\text{pA}$
- LOW QUIESCENT CURRENT: $\pm 450\mu\text{A}$
- LOW INPUT OFFSET VOLTAGE: $\pm 200\mu\text{V}$
- LOW INPUT OFFSET DRIFT: $\pm 2\mu\text{V}/^\circ\text{C}$
- LOW INPUT NOISE:
 $20\text{nV}/\sqrt{\text{Hz}}$ at $f = 1\text{kHz}$ ($G = 100$)
- HIGH CMR: 106dB
- WIDE SUPPLY RANGE: $\pm 2.25\text{V}$ to $\pm 18\text{V}$
- LOW NONLINEARITY ERROR: 0.001% max
- INPUT PROTECTION TO $\pm 40\text{V}$
- 8-PIN DIP AND SO-8 SURFACE MOUNT

APPLICATIONS

- LOW-LEVEL TRANSDUCER AMPLIFIERS
Bridge, RTD, Thermocouple
- PHYSIOLOGICAL AMPLIFIERS
ECG, EEG, EMG, Respiratory
- HIGH IMPEDANCE TRANSDUCERS
- CAPACITIVE SENSORS
- MULTI-CHANNEL DATA ACQUISITION
- PORTABLE, BATTERY OPERATED SYSTEMS
- GENERAL PURPOSE INSTRUMENTATION

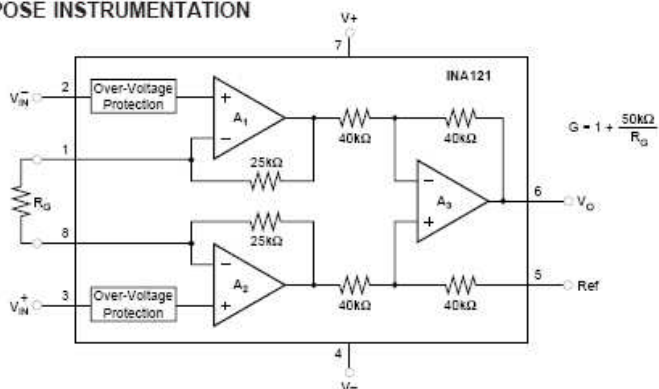
DESCRIPTION

The INA121 is a FET-input, low power instrumentation amplifier offering excellent accuracy. Its versatile three-op amp design and very small size make it ideal for a variety of general purpose applications. Low bias current ($\pm 4\text{pA}$) allows use with high impedance sources.

Gain can be set from 1V to 10,000V/V with a single external resistor. Internal input protection can withstand up to $\pm 40\text{V}$ without damage.

The INA121 is laser-trimmed for very low offset voltage ($\pm 200\mu\text{V}$), low offset drift ($\pm 2\mu\text{V}/^\circ\text{C}$), and high common-mode rejection (106dB at $G = 100$). It operates on power supplies as low as $\pm 2.25\text{V}$ ($+4.5\text{V}$), allowing use in battery operated and single 5V systems. Quiescent current is only $450\mu\text{A}$.

Package options include 8-pin plastic DIP and SO-8 surface mount. All are specified for the -40°C to $+85^\circ\text{C}$ industrial temperature range.



International Airport Industrial Park • Mailing Address: PO Box 11400, Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd., Tucson, AZ 85706 • Tel: (520) 746-1111 • Twx: 910-952-1111
Internet: <http://www.burr-brown.com/> • FAXLine: (800) 546-6133 (US/Canada Only) • Cable: BBRCORP • Telex: 066-6431 • FAX: (520) 889-1510 • Immediate Product Info: (800) 546-6132

50ppm/°C Max, 50μA in SOT23-3 CMOS VOLTAGE REFERENCE

FEATURES

- **MicroSIZE PACKAGE:** SOT23-3
- **LOW DROPOUT:** 1mV
- **HIGH OUTPUT CURRENT:** 25mA
- **HIGH ACCURACY:** 0.2%
- **LOW I_O :** 50μA max
- **EXCELLENT SPECIFIED DRIFT PERFORMANCE:**
 - 50ppm/°C (max) from 0°C to +70°C
 - 75ppm/°C (max) from -40°C to +125°C

APPLICATIONS

- PORTABLE, BATTERY-POWERED EQUIPMENT
- DATA ACQUISITION SYSTEMS
- MEDICAL EQUIPMENT
- HAND-HELD TEST EQUIPMENT

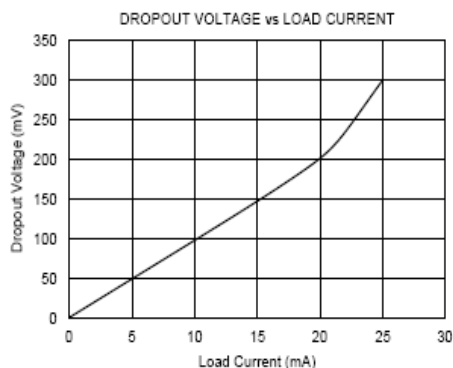
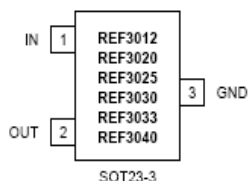
DESCRIPTION

The REF30xx is a precision, low power, low voltage dropout voltage reference family available in a tiny SOT23-3.

The REF30xx small size and low power consumption (50μA max) make it ideal for portable and battery-powered applications. The REF30xx does not require a load capacitor, but is stable with any capacitive load.

Unloaded, the REF30xx can be operated with supplies within 1mV of output voltage. All models are specified for the wide temperature range, -40°C to +125°C.

PRODUCT	VOLTAGE (V)
REF3012	1.25
REF3020	2.048
REF3025	2.5
REF3030	3.0
REF3033	3.3
REF3040	4.096



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

All trademarks are the property of their respective owners.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS
www.ti.com

Copyright © 2002-2004, Texas Instruments Incorporated

1.5°C Accurate Programmable Digital Temperature Sensors with SPI™ Interface

FEATURES

- DIGITAL OUTPUT: SPI-Compatible Interface
- PROGRAMMABLE RESOLUTION:
9- to 12-Bits + Sign
- ACCURACY:
±1.5°C from –25°C to +85°C (max)
±2.0°C from –40°C to +125°C (max)
- LOW QUIESCENT CURRENT: 50µA
- WIDE SUPPLY RANGE: 2.7V to 5.5V
- TINY SOT23-6 AND SO-8 PACKAGES
- OPERATION TO 150°C
- PROGRAMMABLE HIGH/LOW SETPOINTS

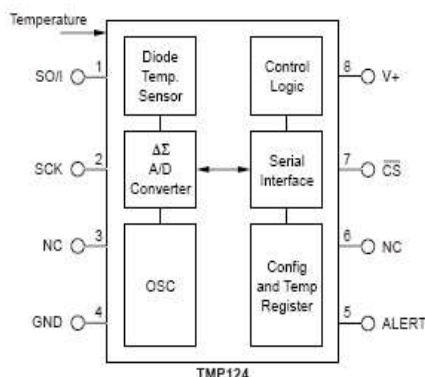
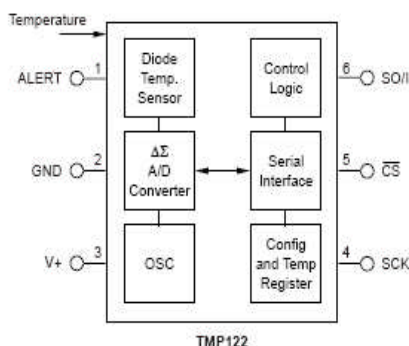
APPLICATIONS

- POWER-SUPPLY TEMPERATURE MONITORING
- COMPUTER PERIPHERAL THERMAL PROTECTION
- NOTEBOOK COMPUTERS
- CELL PHONES
- BATTERY MANAGEMENT
- OFFICE MACHINES
- THERMOSTAT CONTROLS
- ENVIRONMENTAL MONITORING and HVAC
- ELECTROMECHANICAL DEVICE TEMPERATURE

DESCRIPTION

The TMP122 and TMP124 are SPI-compatible temperature sensors available in SOT23-6 and SO-8 packages. Requiring only a pull-up resistor for complete function, the TMP122 and TMP124 temperature sensors are capable of measuring temperatures within 2°C of accuracy over a temperature range of –40°C to +125°C, with operation up to 150°C. Programmable resolution, programmable set points and shut down function provide versatility for any application. Low supply current and a supply range from 2.7V to 5.5V make the TMP122 and TMP124 excellent candidates for low-power applications.

The TMP122 and TMP124 are ideal for extended thermal measurement in a variety of communication, computer, consumer, environmental, industrial, and instrumentation applications.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

SPI is a registered trademark of Motorola. All other trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS
www.ti.com

Copyright © 2003, Texas Instruments Incorporated



Dual SPDT Switch

ADG436

FEATURES

44 V supply maximum ratings

V_{SS} to V_{DD} analog signal range

Low on resistance ($12\ \Omega$ typ)

Low ΔR_{ON} ($3\ \Omega$ max)

Low R_{ON} match ($2.5\ \Omega$ max)

Low power dissipation

Fast switching times

$t_{ON} < 175\ \text{ns}$

$t_{OFF} < 145\ \text{ns}$

Low leakage currents (5 nA max)

Low charge injection (10 pC)

Break-before-make switching action

APPLICATIONS

Audio and video switching

Battery-powered systems

Test equipment

Communications systems

GENERAL DESCRIPTION

The ADG436 is a monolithic CMOS device comprising two independently selectable SPDT switches. It is designed on an LC^2 MOS process, which provides low power dissipation yet gives high switching speed and low on resistance.

The on resistance profile is very flat over the full analog input range, ensuring good linearity and low distortion when switching audio signals. High switching speed also makes the part suitable for video signal switching. CMOS construction ensures ultralow power dissipation, making the part ideally suited for portable and battery-powered instruments.

Each switch conducts equally well in both directions when on and has an input signal range which extends to the power supplies. In the off condition, signal levels up to the supplies are blocked. All switches exhibit break-before-make switching action for use in multiplexer applications. Inherent in the design is low charge injection for minimum transients when switching the digital inputs.

FUNCTIONAL BLOCK DIAGRAM

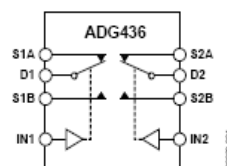


Figure 1.

PRODUCT HIGHLIGHTS

1. Extended signal range.
The ADG436 is fabricated on an enhanced LC^2 MOS process, giving an increased signal range that extends to the supply rails.
2. Low power dissipation.
3. Low R_{ON} .
4. Single-supply operation.
For applications where the analog signal is unipolar, the ADG436 can be operated from a single rail power supply.

Rev. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700
Fax: 781.461.3113

www.analog.com

© 2005 Analog Devices, Inc. All rights reserved.

MC3479

Stepper Motor Driver

The MC3479 is designed to drive a two-phase stepper motor in the bipolar mode. The circuit consists of four input sections, a logic decoding/sequencing section, two driver-stages for the motor coils, and an output to indicate the Phase A drive state.

- Single Supply Operation: 7.2 to 16.5 V
- 350 mA/Coil Drive Capability
- Clamp Diodes Provided for Back-EMF Suppression
- Selectable CW/CCW and Full/Half Step Operation
- Selectable High/Low Output Impedance (Half Step Mode)
- TTL/CMOS Compatible Inputs
- Input Hysteresis: 400 mV Minimum
- Phase Logic Can Be Initialized to Phase A
- Phase A Output Drive State Indication (Open-Collector)

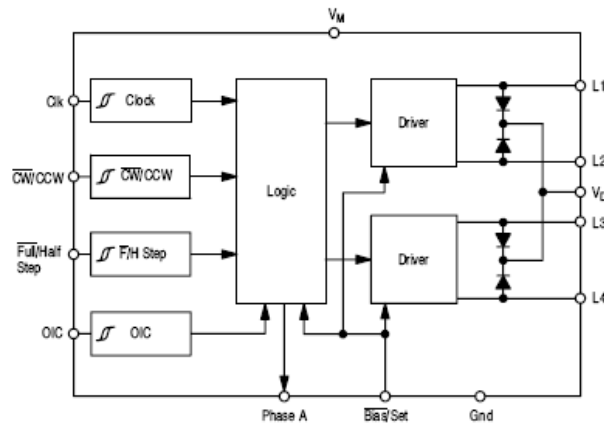


Figure 1. Representative Block Diagram

ORDERING INFORMATION

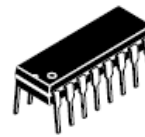
Device	Operating Temperature Range	Package	Shipping
MC3479P	T _A = 0° to +70°C	PDIP	25 Units / Rail
MC3479P	T _A = 0° to +70°C	PDIP	2000 Units / Box



ON Semiconductor®

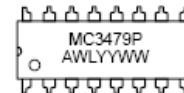
<http://onsemi.com>

STEPPER MOTOR DRIVER



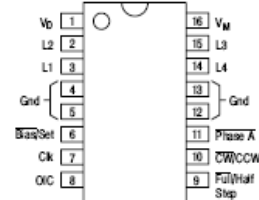
PDIP
P SUFFIX
PLASTIC PACKAGE
CASE 648C

MARKING DIAGRAM



MC3479P = Specific Device Code
A = Assembly Location
WL = Wafer Lot
YY = Year
WW = Work Week

PIN CONNECTIONS



(Top View)

INPUT TRUTH TABLE

	Input Low	Input High
CW/CCW	CW	CCW
Full/Half Step	Full Step	Half Step
OIC	Hi Z	Low Z
Clk	Positive Edge Triggered	



+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

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

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than $5\mu W$. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

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

Features

Superior to Bipolar

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

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

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



Maxim Integrated Products 1

For free samples and the latest literature, visit www.maxim-ic.com or phone 1-800-998-8800.
For small orders, phone 1-800-835-8769.

MAX220-MAX249

Digitally Controlled Potentiometer (XDCP™)

FEATURES

- Solid-state potentiometer
- 3-wire serial interface
- 100 wiper tap points
 - Wiper position stored in nonvolatile memory and recalled on power-up
- 99 resistive elements
 - Temperature compensated
 - End to end resistance, $\pm 20\%$
 - Terminal voltages, $\pm 5V$
- Low power CMOS
 - $V_{CC} = 5V$
 - Active current, 3mA max.
 - Standby current, 750 μ A max.
- High reliability
 - Endurance, 100,000 data changes per bit
 - Register data retention, 100 years
- X9C102 = 1k Ω
- X9C103 = 10k Ω
- X9C503 = 50k Ω
- X9C104 = 100k Ω
- Packages
 - 8-lead SOIC and DIP

DESCRIPTION

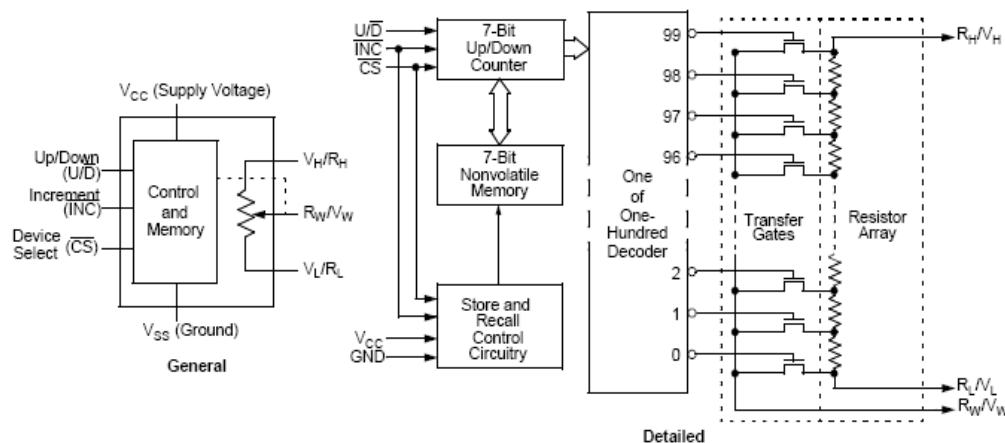
The X9Cxxx are Intersil digitally controlled (XDCP) potentiometers. The device consists of a resistor array, wiper switches, a control section, and nonvolatile memory. The wiper position is controlled by a three-wire interface.

The potentiometer is implemented by a resistor array composed of 99 resistive elements and a wiper switching network. Between each element and at either end are tap points accessible to the wiper terminal. The position of the wiper element is controlled by the $\overline{CS}$, U/D , and INC inputs. The position of the wiper can be stored in nonvolatile memory and then be recalled upon a subsequent power-up operation.

The device can be used as a three-terminal potentiometer or as a two-terminal variable resistor in a wide variety of applications including:

- control
- parameter adjustments
- signal processing

BLOCK DIAGRAM



LM339, LM239, LM2901, LM2901V, NCV2901, MC3302

Single Supply Quad Comparators

These comparators are designed for use in level detection, low-level sensing and memory applications in consumer, automotive, and industrial electronic applications.

Features

- Single or Split Supply Operation
- Low Input Bias Current: 25 nA (Typ)
- Low Input Offset Current: ± 5.0 nA (Typ)
- Low Input Offset Voltage
- Input Common Mode Voltage Range to GND
- Low Output Saturation Voltage: 130 mV (Typ) @ 4.0 mA
- TTL and CMOS Compatible
- ESD Clamps on the Inputs Increase Reliability without Affecting Device Operation
- NCV Prefix for Automotive and Other Applications Requiring Site and Control Changes
- Pb-Free Packages are Available*

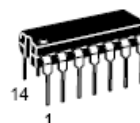


ON Semiconductor®

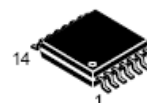
<http://onsemi.com>



SOIC-14
D SUFFIX
CASE 751A

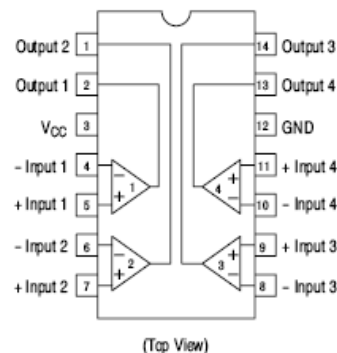


PDIP-14
N, P SUFFIX
CASE 646



TSSOP-14
DTB SUFFIX
CASE 948G

PIN CONNECTIONS



ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 6 of this data sheet.

DEVICE MARKING INFORMATION

See general marking information in the device marking section on page 8 of this data sheet.

*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

LM193/LM293/LM393/LM2903

Low Power Low Offset Voltage Dual Comparators

General Description

The LM193 series consists of two independent precision voltage comparators with an offset voltage specification as low as 2.0 mV max for two comparators which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though operated from a single power supply voltage.

Application areas include limit comparators, simple analog to digital converters; pulse, squarewave and time delay generators; wide range VCO; MOS clock timers; multivibrators and high voltage digital logic gates. The LM193 series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, the LM193 series will directly interface with MOS logic where their low power drain is a distinct advantage over standard comparators.

The LM393 and LM2903 parts are available in National's innovative thin micro SMD package with 8 (12 mil) large bumps.

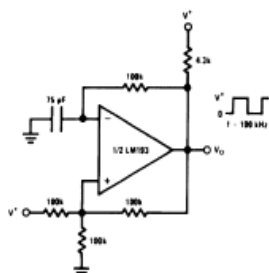
Advantages

- High precision comparators
- Reduced V_{OS} drift over temperature
- Eliminates need for dual supplies
- Allows sensing near ground
- Compatible with all forms of logic
- Power drain suitable for battery operation

Features

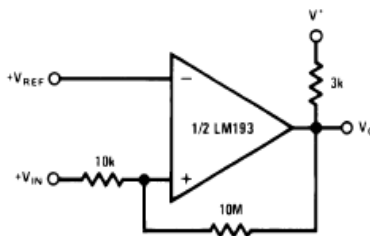
- Wide supply
 - Voltage range: 2.0V to 36V
 - Single or dual supplies: $\pm 1.0V$ to $\pm 18V$
- Very low supply current drain (0.4 mA) — independent of supply voltage
- Low input biasing current: 25 nA
- Low input offset current: ± 5 nA
- Maximum offset voltage: ± 3 mV
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Low output saturation voltage,; 250 mV at 4 mA
- Output voltage compatible with TTL, DTL, ECL, MOS and CMOS logic systems
- Available in the 8-Bump (12 mil) micro SMD package
- See AN-1112 for micro SMD considerations

Squarewave Oscillator



00670938

Non-Inverting Comparator with Hysteresis



00670909



PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM),
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin
PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during Sleep via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™
(Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) – 8 bits wide with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital
Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference
(VREF) module
 - Programmable input multiplexing from device
inputs and internal voltage reference
 - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash
program memory typical
- 1,000,000 erase/write cycle Data EEPROM
memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™)
via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low-power, high-speed Flash/EEPROM
technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I²C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2



UTRON

REV 1.0

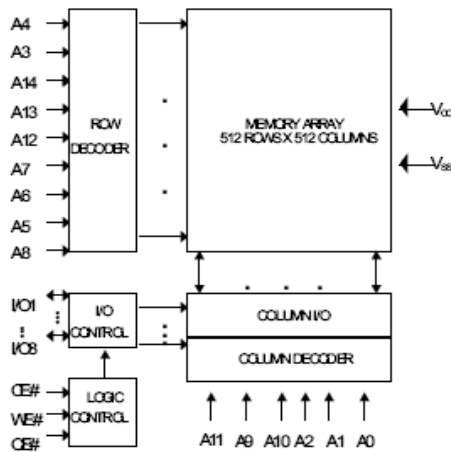
32K X 8 BIT HIGH SPEED CMOS SRAM

UT61256

FEATURES

- Fast access time : 8/10/12/15 ns (max.)
- Low operating power consumption : 80 mA (typical)
- Single 5V power supply
- All inputs and outputs TTL compatible
- Fully static operation
- Three state outputs
- Package : 28-pin 300 mil skinny PDIP
28-pin 300 mil SOJ
28-pin 330 mil SOP
28-pin 8x13.4mm TSOP-I

FUNCTIONAL BLOCK DIAGRAM



PIN DESCRIPTION

SYMBOL	DESCRIPTION
A0 - A14	Address Inputs
I/O1 - I/O8	Data Inputs/Outputs
CE#	Chip Enable Inputs
WE#	Write Enable Inputs
OE#	Output Enable Inputs
V _{CC}	Power Supply
V _{SS}	Ground

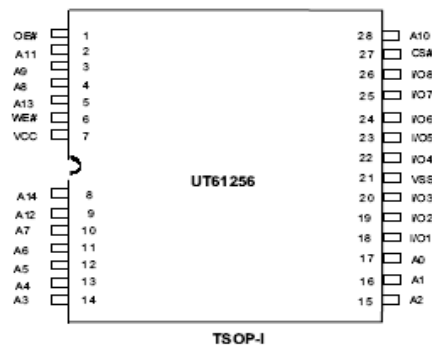
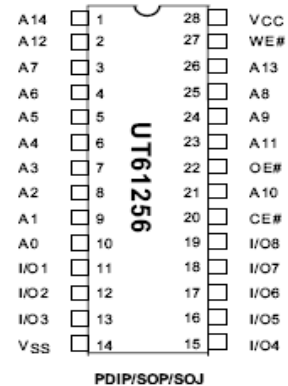
GENERAL DESCRIPTION

The UT61256 is a 262,144-bit high-speed CMOS static random access memory organized as 32,768 words by 8 bits. It is fabricated using high performance, high reliability CMOS technology.

The UT61256 is designed for high-speed system applications. It is particularly suited for use in high-density high-speed system applications.

The UT61256 operates from a single 5V power supply and all inputs and outputs are fully TTL compatible.

PIN CONFIGURATION





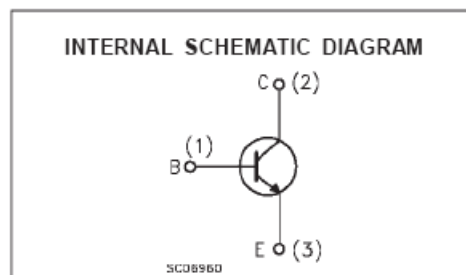
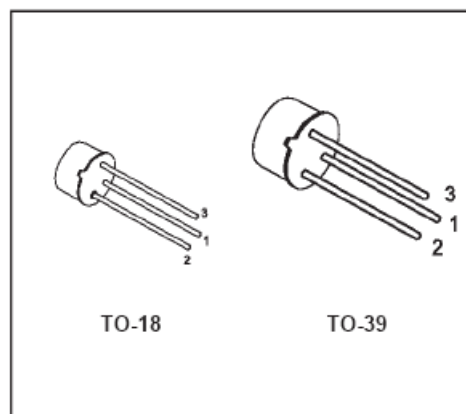
2N2219A
2N2222A

HIGH SPEED SWITCHES

DESCRIPTION

The 2N2219A and 2N2222A are silicon planar epitaxial NPN transistors in Jedec TO-39 (for 2N2219A) and in Jedec TO-18 (for 2N2222A) metal case. They are designed for high speed switching application at collector current up to 500mA, and feature useful current gain over a wide range of collector current, low leakage currents and low saturation voltage.

● 2N2219A approved to CECC 50002-100,
2N2222A approved to CECC 50002-101
available on request.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CB0}	Collector-Base Voltage ($I_E = 0$)	75	V
V_{CE0}	Collector-Emitter Voltage ($I_S = 0$)	40	V
V_{EB0}	Emitter-Base Voltage ($I_C = 0$)	6	V
I_C	Collector Current	0.8	A
P_{tot}	Total Dissipation at $T_{amb} \leq 25^\circ\text{C}$ for 2N2219A	0.8	W
	for 2N2222A	0.5	W
	at $T_{case} \leq 25^\circ\text{C}$ for 2N2219A	3	W
	for 2N2222A	1.8	W
T_{stg}	Storage Temperature	-65 to 200	$^\circ\text{C}$
T_J	Max. Operating Junction Temperature	175	$^\circ\text{C}$

June 1999

1/8

SN54LS373, SN54LS374, SN54S373, SN54S374,
SN74LS373, SN74LS374, SN74S373, SN74S374

OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

SDLS165B – OCTOBER 1975 – REVISED AUGUST 2002

- Choice of Eight Latches or Eight D-Type Flip-Flops in a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel Access for Loading
- Buffered Control Inputs
- Clock-Enable Input Has Hysteresis to Improve Noise Rejection ('S373 and 'S374)
- P-N-P Inputs Reduce DC Loading on Data Lines ('S373 and 'S374)

description

These 8-bit registers feature 3-state outputs designed specifically for driving highly capacitive or relatively low-impedance loads. The high-impedance 3-state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pullup components. These devices are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches, meaning that while the enable (C or CLK) input is high, the Q outputs follow the data (D) inputs. When C or CLK is taken low, the output is latched at the level of the data that was set up.

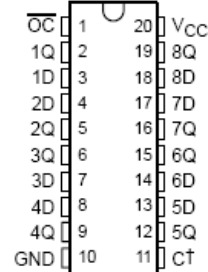
The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs are set to the logic states that were set up at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines of the 'S373 and 'S374 devices simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output-control ($\overline{OC}$) input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or the high-impedance state. In the high-impedance state, the outputs neither load nor drive the bus lines significantly.

$\overline{OC}$ does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered, even while the outputs are off.

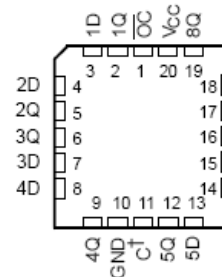
SN54LS373, SN54LS374, SN54S373,
SN54S374 ... J OR W PACKAGE
SN74LS373, SN74LS374 ... DW, N, OR NS PACKAGE
SN74LS374 ... DB, DW, N, OR NS PACKAGE
SN74S373 ... DW OR N PACKAGE

(TOP VIEW)



† C for 'LS373 and 'S373; CLK for 'LS374 and 'S374.

SN54LS373, SN54LS374, SN54S373,
SN54S374 ... FK PACKAGE
(TOP VIEW)



† C for 'LS373 and 'S373; CLK for 'LS374 and 'S374.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA: Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2002, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

1

DM74LS00

Quad 2-Input NAND Gate

General Description

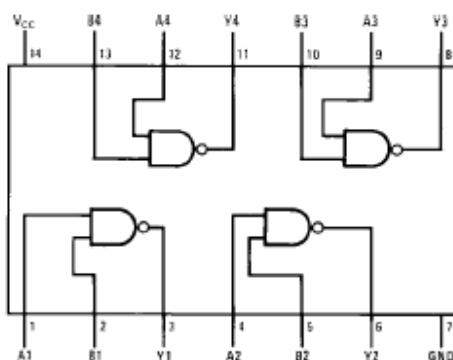
This device contains four independent gates each of which performs the logic NAND function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS00M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS00SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS00N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = \overline{AB}$$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

NTE309K Integrated Circuit Voltage Regulator

Description:

The NTE309K is a complete 5V regulator fabricated on a single silicon chip. It is designed for local regulation on digital logic cards, eliminating the distribution problems associated with single-point regulation. With the TO3 power package, the available output current is greater than 1A.

The regulator is essentially blowout proof. Current limiting is included to limit the peak output current to a safe value. In addition, the thermal shutdown is provided to keep the IC from overheating. If internal dissipation becomes too great, the regulator will shut down to prevent excessive heating.

Considerable effort was expended to make these devices easy to use and to minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response somewhat. Input bypassing is needed, however, if the regulator is located very far from the filter capacitor of the power supply. Stability is also achieved by methods that provide very good rejection of load or line transients as are usually seen with TTL logic.

Although designed primarily as a fixed-voltage regulator, the output can be set to voltages above 5V, as shown below. It is also possible to use the circuits as the control element in precision regulators, taking advantage of the good current-handling capability and the thermal overload protection.

Features:

- Specified to be compatible, worst case, with TTL and DTL
- Output current in excess of 1A
- Internal thermal overload protection
- No external components required

Absolute Maximum Ratings:

Input Voltage, V_{IN}	35V
Power Dissipation, P_D	Internally Limited
Operating Junction Temperature Range, T_{opr}	0°C to +125°C
Storage Temperature Range, T_{stg}	-65°C to +150°C
Typical Thermal Resistance, Junction-to-Case (Note 1), R_{thJC}	2.5°C/W
Lead Temperature (Soldering, 10 seconds), T_L	+300°C

Note 1. Without a heat sink, the thermal resistance of the TO-3 package is approximately 35°C/W.
With a heat sink,

LM79XX Series 3-Terminal Negative Regulators

General Description

The LM79XX series of 3-terminal regulators is available with fixed output voltages of $-5V$, $-8V$, $-12V$, and $-15V$. These devices need only one external component—a compensation capacitor at the output. The LM79XX series is packaged in the TO-220 power package and is capable of supplying 1.5A of output current.

These regulators employ internal current limiting, safe area protection and thermal shutdown for protection against virtually all overload conditions.

Low ground pin current of the LM79XX series allows output voltage to be easily boosted above the preset value with a resistor divider. The low quiescent current drain of

these devices with a specified maximum change with line and load ensures good regulation in the voltage boosted mode.

For applications requiring other voltages, see LM137 data sheet.

Features

- Thermal, short circuit and safe area protection
- High ripple rejection
- 1.5A output current
- 4% tolerance on preset output voltage

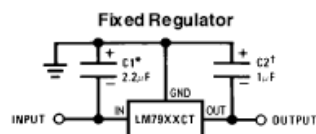
Connection Diagrams



TL/H/7340-14

Order Number LM7905CT, LM7912CT or LM7915CT
See NS Package Number TO3B

Typical Applications



TL/H/7340-3

*Required if regulator is separated from filter capacitor by more than 3". For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted.

†Required for stability. For value given, capacitor must be solid tantalum. 25 μF aluminum electrolytic may be substituted. Values given may be increased without limit.

For output capacitance in excess of 100 μF , a high current diode from input to output (1N4001, etc.) will protect the regulator from momentary input shorts.

SI-3000V Series

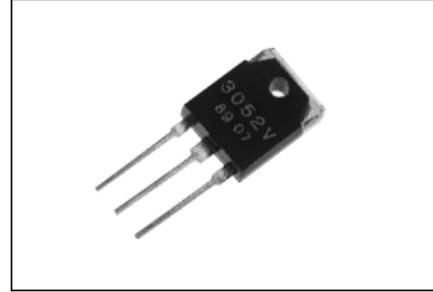
3-Terminal, Low Dropout Voltage Dropper Type

■ Features

- TO-3P package 3-terminal regulator
- Output current: 2.0A
- Low dropout voltage: $V_{DIF} \leq 1V$ (at $I_o = 2.0A$)
- Built-in foldback overcurrent protection circuit

■ Applications

- For stabilization of the secondary stage of switching power supplies
- Electronic equipment



■ Absolute Maximum Ratings

($T_a = 25^\circ C$)

Parameter	Symbol	Ratings		Unit
		SI-3052V	SI-3122V/3152V	
DC Input Voltage	V_{IN}	25	30	V
DC Output Current	I_o	2.0		A
Power Dissipation	P_{D1}	50($T_c = 25^\circ C$)		W
	P_{D2}	1.6(Without heatsink, stand-alone operation)		W
Junction Temperature	T_j	-30 to +125		$^\circ C$
Ambient Operating Temperature	T_{op}	-20 to +100		$^\circ C$
Storage Temperature	T_{stg}	-30 to +125		$^\circ C$
Thermal Resistance (junction to case)	$R_{th(j-c)}$	2.0		$^\circ C/W$

■ Electrical Characteristics

($T_a = 25^\circ C$)

Parameter	Symbol	Ratings									unit
		SI-3052V			SI-3122V			SI-3152V			
		min.	typ.	max.	min.	typ.	max.	min.	typ.	max.	
Input Voltage	V _{IN}	6		15	13		25	16		25	V
Output Voltage	V _O	4.9	5.0	5.1	11.8	12.0	12.2	14.8	15.0	15.2	V
	Conditions	V _{IN} =8V, I _o =1.0A			V _{IN} =16V, I _o =1.0A			V _{IN} =20V, I _o =1.0A			
Dropout Voltage	V _{DIF}			0.5			0.5			0.5	V
	Conditions	I _o =1.0A									
				1.0			1.0			1.0	
	Conditions	I _o =2.0A									
Line Regulation	ΔV _{OLINE}		10	30		20	60		20	60	mV
	Conditions	V _{IN} =6 to 15V, I _o =1.0A			V _{IN} =13 to 25V, I _o =1.0A			V _{IN} =16 to 25V, I _o =1.0A			
Load Regulation	ΔV _{OLOAD}		40	100		80	200		80	200	mV
	Conditions	V _{IN} =8V, I _o =0 to 2.0A			V _{IN} =16V, I _o =0 to 2.0A			V _{IN} =20V, I _o =0 to 2.0A			
Temperature Coefficient of Output Voltage	ΔV _O /ΔT _s		±0.5			+1.5			±1.5		mV/°C
Ripple Rejection	R _{REJ}		54			54			54		dB
	Conditions	f=100 to 120Hz									
Overcurrent Protection Starting Current	I _{O1}	2.4			2.4			2.4			A
	Conditions	V _{IN} =8V			V _{IN} =16V			V _{IN} =20V			

The following are also available: SI-3522V(5.2V), SI-3062V(6V), SI-3082V(8V), SI-3922V(9.2V), SI-3102V(10V), SI-3132V(13.1V), SI-3182V(18V), SI-3202V(20V).

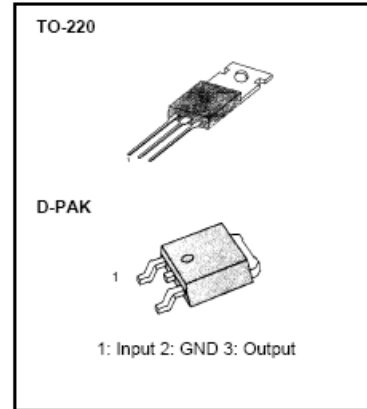
LM78XX (KA78XX, MC78XX) FIXED VOLTAGE REGULATOR (POSITIVE)

3-TERMINAL 1A POSITIVE VOLTAGE REGULATORS

The LM78XX series of three-terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

FEATURES

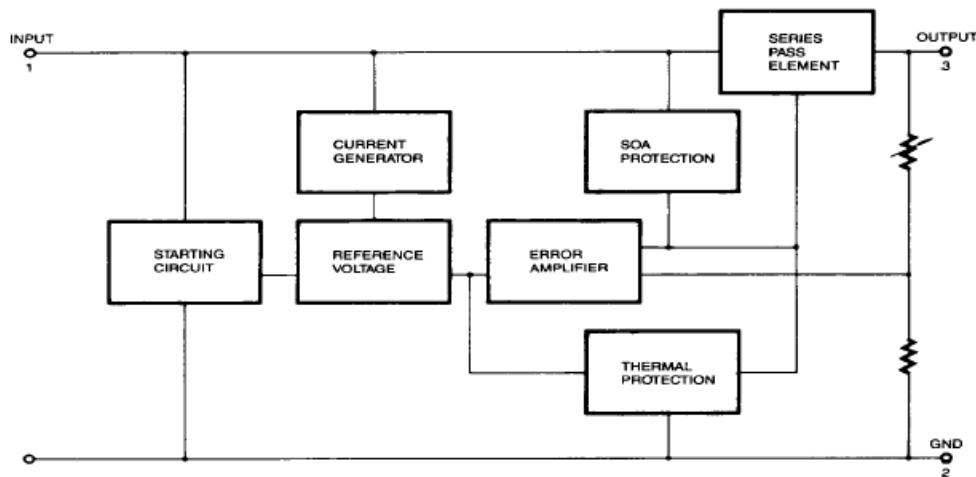
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 11, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor SOA Protection



ORDERING INFORMATION

Device	Output Voltage Tolerance	Packag	Operating Temperature
KA78XXCT	$\pm 4\%$	TO-220	0 ~ +125 °C
KA78XXAT	$\pm 2\%$		-40 ~ +125 °C
KA78XXIT	$\pm 4\%$		0 ~ +125 °C
KA78XXR	$\pm 2\%$	D-PAK	0 ~ +125 °C
KA78XXAR	$\pm 2\%$		-40 ~ +125 °C
KA78XXIR	$\pm 4\%$		-40 ~ +125 °C

BLOCK DIAGRAM



FAIRCHILD
SEMICONDUCTOR™

©1999 Fairchild Semiconductor Corporation

Rev. B