

89
2.41



**UNIVERSIDAD
NACIONAL
AUTONOMA DE
MEXICO
FACULTAD DE INGENIERIA**

**SISTEMA DE ADQUISICION Y DESPLIEGUE DE INFORMACION
PARA LA LAMINADORA HILLE 25**

E INGENIERO MECANICO ELECTRICISTA

PRESENTA:

ALVARO CUAUHEMOC GUZMAN SANCHEZ

DIRECTOR DE TESIS:

M. I. JORGE RODRIGUEZ CUEVAS

MEXICO D. F.

1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A MIS PADRES:

Por el apoyo brindado a lo largo de todos estos años de formación profesional.

AL M. I. JORGE RODRIGUEZ CUEVAS:

Por su magnífica orientación y excelente apoyo, pero sobre todo por su gran calidad humana.

AL ING. DANIEL MONTAÑO FRIAS:

Porque gracias a su apoyo profesional y amistad incondicional éste trabajo pudo salir adelante.

RICARDO BECERRIL HERRERA

AGRADECIMIENTOS Y DEDICATORIAS

A MIS PADRES, ALVARO Y MARGARITA:

Por darme todo en la vida, por sus enseñanzas, apoyo, cariño y comprensión incondicional. A ellos principalmente.

A MIS HERMANAS ROSALIA V. Y CLAUDIA M.:

Como un aliciente para seguir adelante.

A MIS ABUELOS ROSALIO, ELVA, MANUEL Y MERCEDES.

AL ING. JORGE RODRIGUEZ CUEVAS:

Por dirigir el trabajo, y en general a todos mis maestros por compartirme sus conocimientos.

A RICARDO, VICTOR HUGO, DANIEL Y ALFREDO, ENTRE OTROS:

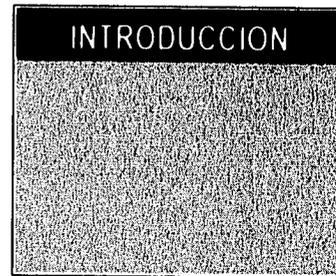
Por su amistad y compañerismo.

ALVARO CUAUHEMOC GUZMAN SANCHEZ

INDICE

	PAGINA
INTRODUCCION	1
CAPITULO 1 PLANTEAMIENTO DEL PROBLEMA Y CONSIDERACIONES DEL DISEÑO	4
CAPITULO 2 CONCEPTOS GENERALES	7
CAPITULO 3 CARACTERIZACION DE LOS TRANSDUCTORES A UTILIZAR	21
CAPITULO 4 DISEÑO DE LOS ACONDICIONADORES DE SEÑALES	34
CAPITULO 5 DISEÑO DE LA ARQUITECTURA DEL SISTEMA	54
CAPITULO 6 DISEÑO DE LA PROGRAMACION DEL SISTEMA	60
CAPITULO 7 PRUEBAS Y MEDICIONES	95
CAPITULO 8 CONCLUSIONES	109
APENDICE A: PROGRAMA FUENTE DEL SISTEMA PARA COMPUTADORA PC.	
APENDICE B: PROGRAMA FUENTE PARA EL MICROCONTROLADOR.	
BIBLIOGRAFIA	

SISTEMA DE ADQUISICION Y DESPLIEGUE DE INFORMACION PARA LA LAMINADORA HILLE 25



En la actualidad son muchos los tipos de industrias y las áreas de las mismas, que se han visto beneficiadas por los avances alcanzados en la rama de la ingeniería que se especializa en el estudio del control automático de procesos; haciendo posible que en la actualidad el control automático sea aplicable a cualquier tipo de procesos, sin importar la complejidad del mismo.

Dentro del campo del control automático se desarrollan una gran variedad de tareas, las cuales pueden ir desde la simple adquisición de datos del proceso, los cuales nos ayuden a realizar un análisis del mismo; hasta la automatización total del proceso basándonos en los datos adquiridos, en equipos digitales de procesamiento y en actuadores que intervienen directamente en el proceso, y que ayudan a la aplicación de técnicas de control más avanzadas como son la compensación por "tiempo muerto", desacoplamiento de procesos multivariables, algoritmos de control digital y control adaptable.

La adquisición de datos por sí misma puede tener varias aplicaciones fundamentales dentro de los procesos como pueden ser, adquirir datos para efectuar balances de materia y energía, calcular eficiencias y rendimientos y elaborar reportes con la información procesada. A últimas fechas, para llevar a cabo estas aplicaciones nos podemos auxiliar con uno de los equipos de procesamiento digital más versátil y funcional que existen en la actualidad, la computadora.

En la actualidad la computadora se ha convertido en un instrumento indispensable en la realización de las tareas humanas, su capacidad para almacenar grandes cantidades de información, su velocidad y precisión en cálculos numéricos, su bajo costo y su versatilidad, la han convertido en un producto de gran éxito (algunas de las compañías constructoras de computadoras figuran entre las más destacadas del mundo). Su influencia en todos los ámbitos del quehacer del hombre, ha sido tal, que resulta difícil mencionar alguno que no haya sido modificado con su incorporación.

La finalidad del presente proyecto es desarrollar un sistema de adquisición y despliegue de datos en tiempo real, que sea capaz de interactuar con una computadora tipo PC, y el cual se pueda aplicar a un proceso de suma importancia dentro de la industria dedicada a la manufactura de metales (como acero, cobre, fierro, aluminio, etc.), dicho proceso es la laminación de metales.

La laminación consiste en la reducción de la sección transversal de un material al hacerlo pasar entre dos rodillos cilíndricos que giran en sentido opuesto, y cuya separación es menor que el espesor inicial del lingote. Dichos rodillos producen la deformación plástica del material por medio de esfuerzos de compresión y corte. Los esfuerzos de compresión se deben a que según el material se introduce entre los rodillos, el espesor libre va siendo menor. Los esfuerzos cortantes se deben a que las fibras externas del material están sujetas a la fuerza de fricción de los rodillos. Por ser un proceso continuo, el material resultante se caracteriza por su sección transversal constante.

Vale la pena mencionar que el proceso de laminación sobre el cual se aplicará el sistema a desarrollar, se realiza con una laminadora Hille 25 tipo experimental que se ubica en el Laboratorio de Materiales de la División de Ingeniería Mecánica de la Facultad de Ingeniería; pero la cual no se diferencia en mucho de las laminadoras de tipo industrial usadas actualmente en la industria.

Siendo dentro del proceso de laminación las variables determinantes: la temperatura, la carga, la velocidad angular, el espesor del material y el par de laminación, se hace necesario que el sistema pueda adquirir datos en tiempo real para las cinco variables, o si se desea para una sola en particular. Para ésto el sistema deberá contar con tres módulos fundamentales: un acondicionador de señales, una arquitectura basada en un microcontrolador y la computadora tipo PC.

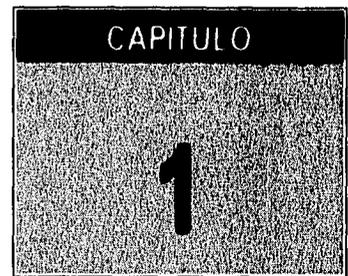
El acondicionador de señales deberá tomar las mediciones hechas por los transductores que se localizan en diversas partes de la laminadora y adaptarlas para que puedan ser introducidas a la arquitectura.

La arquitectura basada en un microcontrolador, tendrá como funciones principales: convertir los datos del acondicionador de señales del tipo analógico al tipo digital; controlar el flujo de dichos datos digitales y establecer la comunicación con la computadora para realizar la transmisión de los datos. Estas funciones variarán dependiendo de que si estamos trabajando con las cinco variables a un mismo tiempo o con una sola.

Por último, las funciones de la computadora, a grosso modo, se deberán enfocar principalmente al despliegue en tiempo real de valores decimales y gráficas que representen directamente los datos adquiridos por el sistema; sin olvidar el almacenamiento, la consulta y la emisión de reportes impresos de los datos adquiridos; ésto tanto al trabajar con las cinco variables como al trabajar con una sola. Es obvio que deberá existir un sistema de software, el cual también se deberá desarrollar, y que se encargue de realizar los despliegues y almacenamientos de datos.

En el presente trabajo primero se analizan y diseñan los acondicionadores de señales, posteriormente la arquitectura del sistema y por último el sistema de software que se implementará en la computadora tipo PC.

PLANTEAMIENTO DEL PROBLEMA Y CONSIDERACIONES DEL DISEÑO



El planteamiento de este proyecto en términos generales puede separarse en dos partes fundamentales, la primera consistirá en adquirir y acondicionar las señales analógicas suministradas por los transductores de cada una de las variables a trabajar: temperatura, carga, velocidad angular, par de laminación y espesor; y la segunda, que consistirá en procesar las señales analógicas (ya acondicionadas), para su posterior transmisión en forma de datos digitales, a una computadora tipo PC, la cual se encargará principalmente del despliegue y almacenamiento de los datos. Para ésto el Sistema de Adquisición y Despliegue de Información a desarrollar se valdrá de 3 módulos principales, uno de acondicionamiento (tarjeta de acondicionadores de señales), otro para controlar el flujo de la información (arquitectura basada en un microcontrolador) y el último constituido por el software desarrollado e implementado en la computadora tipo PC.

La tarea de adquirir las señales analógicas provenientes de los transductores encargados de medir cada una de las variables, le corresponderá al módulo acondicionador de señales; el cual posteriormente deberá efectuar el acondicionamiento de dichas señales para su inmediata transmisión al convertidor analógico-digital del microcontrolador de nuestro sistema. Al decir acondicionamiento nos referimos a amplificar, atenuar, filtrar o linealizar (con ayuda de componentes electrónicos de tipo analógico) la señal analógica obtenida en el transductor correspondiente para que ésta quede en el intervalo de valores que para señales analógicas maneja el convertidor analógico-digital de nuestro microcontrolador. Por lo tanto los transductores de cada una de las variables se deben conectar antes que nada al módulo de acondicionamiento de señales del Sistema de Adquisición y Despliegue de Información.

Las funciones del segundo módulo del sistema, o arquitectura basada en un microcontrolador, deberán ser principalmente tres:

- Realizar la conversión analógica-digital de la información proveniente del módulo acondicionador de señales (la cual se encuentra en forma de señales analógicas), para que la misma pueda ser transmitida y procesada por la computadora.

- Controlar el flujo de información tanto analógica (señales que entran al convertidor analógico-digital) como digital (información lista para ser transmitida a la computadora), con la finalidad de evitar pérdidas y mezclado de la información correspondiente a cada una de las variables.
- Transmitir la información digital en forma serial fijando una serie de especificaciones en la transmisión, como son la velocidad, paridad, bits de datos, etc., y siguiendo el estándar de comunicación serial RS-232 para computadoras tipo PC.

El tercer módulo constituido por la computadora tipo PC y el software que se encontrará funcionando en ella, el cual también es parte del Sistema de Adquisición y Despliegue de Información, tendrá las siguientes funciones principales:

- Procesar la información digital proveniente del microcontrolador.
- Hacer un despliegado por medio del monitor de la computadora, de dicha información en forma de datos y gráficas.
- Guardar la información en disco duro o en discos flexibles para su posterior análisis.
- Emitir reportes impresos de la información adquirida.

Así tenemos que de manera esquemática nuestro sistema se encuentra integrado de la siguiente forma:

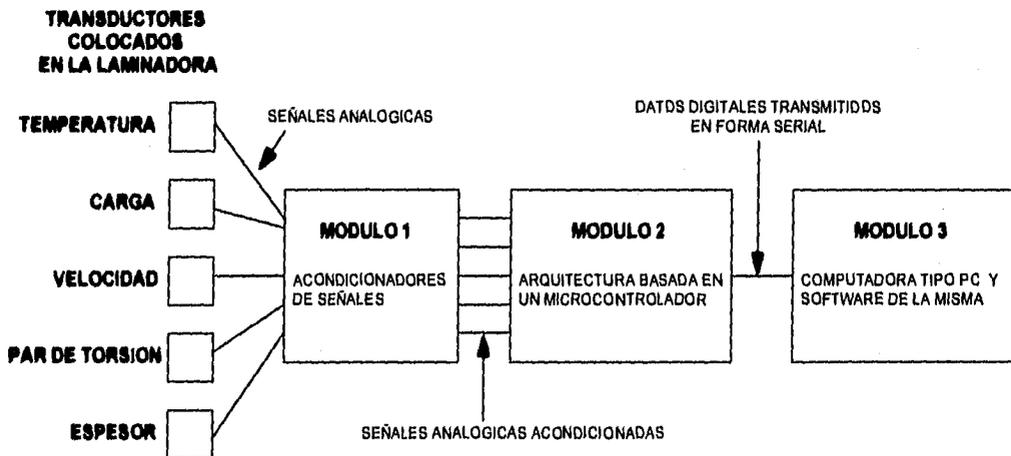


Figura 1. Diagrama esquemático del Sistema de Adquisición y Despliegue de Información para la Laminadora Hille 25

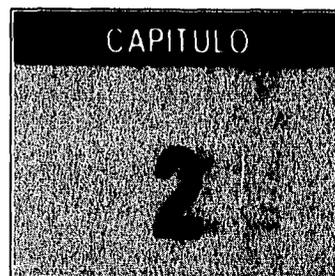
Dentro de las consideraciones que debemos tener en cuenta, una de las más importantes se refiere a los transductores a utilizar para cada una de las variables. Se debe buscar que dichos transductores tengan un grado aceptable de precisión, que abarquen un intervalo amplio de medición, un costo accesible, que de

preferencia puedan ser encontrados en el mercado nacional y que puedan ser fácilmente adaptados para ser compatibles con el resto del sistema.

Es importante mencionar que actualmente la laminadora cuenta con sistemas completos de adquisición y despliegue de datos para las variables Temperatura y Carga, el uso de los cuales no se debe descartar conforme se vaya desarrollando el nuevo sistema que englobe todas las variables, ya que dichos sistemas cuentan con un módulo que se encarga de adquirir, procesar y desplegar los datos de la variable en cuestión, además del transductor de la misma.

Los sistemas anteriormente mencionados fueron adquiridos por la División de Ingeniería Mecánica de la Facultad de Ingeniería, ya que anteriormente se tenía la idea de colocar en la laminadora un sistema de adquisición y despliegue de información por cada una de las variables que se pretende manejar; lo cual se dejó de hacer debido al alto costo de los sistemas, a lo difícil que es adaptar dichos sistemas a la laminadora y a lo complicado que se volvería observar a un sólo tiempo los cinco módulos encargados del despliegado de datos.

CONCEPTOS GENERALES



2.1 TRANSDUCTORES ELECTRICOS

En este breve apartado se explica de manera general qué se entiende por transductor, independientemente de la transformación de energía o información que haga; posteriormente nos enfocamos a los transductores del tipo eléctrico, ya que de este tipo serán los utilizados en el Sistema de Adquisición y Despliegue de Información; por último y como complemento hacemos una descripción general sobre las funciones de dichos transductores en el área enfocada a la medición de variables físicas.

2.1.1. DEFINICION DE TRANSDUCTOR.

Los transductores se definen en general como los dispositivos que convierten energía o información de una forma a otra. Se utilizan ampliamente en el campo de las mediciones porque no todas las cantidades que se necesitan medir se pueden observar tan fácilmente como otras. Normalmente se puede obtener una medición mejor de una cantidad si ésta se puede convertir a otra forma que sea más fácil o exacta de observar. Por ejemplo, el termómetro de mercurio convierte cambios de temperatura en cambios de longitud de una columna de mercurio. Puesto que los cambios en la longitud de la columna de mercurio son más fáciles de medir, el termómetro de mercurio se ha convertido en un dispositivo muy conveniente para sensar los cambios de temperatura.

2.1.2. TRANSDUCTORES ELECTRICOS.

Los transductores eléctricos son los encargados de convertir las mediciones hechas a cantidades físicas en señales eléctricas, lo cual los convierte en los más utilizados hoy en día. La razón de su popularidad es que las señales eléctricas poseen muchas cualidades que son convenientes en el campo de las mediciones. La primera de éstas es que la instrumentación eléctrica está tan desarrollada que generalmente existen varios métodos diferentes para convertir la mayoría de las cantidades físicas en señales eléctricas. Además,

si una señal débil se convierte a una señal eléctrica, se puede amplificar hasta que su amplitud sea lo suficientemente grande para observarse fácilmente. Finalmente, hay dispositivos para registrar y observar señales eléctricas que pueden seguir variaciones de muy alta frecuencia. Por tanto, una cantidad no eléctrica que tiene una variación de muy alta frecuencia (tal como las vibraciones de un sólido) se puede convertir a una señal eléctrica y registrarse exactamente. Comúnmente los transductores eléctricos se clasifican de acuerdo con la cantidad física a la cual responden.

2.1.3. LA FUNCION DE LOS TRANSDUCTORES EN LOS SISTEMAS DE MEDICION.

Existen, en general, tres elementos principales que son comunes a la mayoría de los sistemas de medición (figura 1).

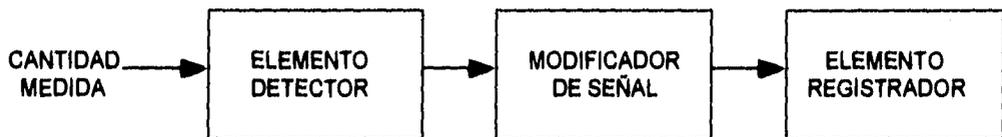


Figura 1. Elementos de un sistema general de medición

El primero de éstos es el elemento detector (o sensor). El propósito de este elemento es el de responder a la magnitud (o cambios en la magnitud) de la cantidad que se está midiendo. La respuesta del sensor es una señal de salida cuya magnitud es proporcional a la magnitud de la cantidad que se está midiendo.

El segundo elemento es el modificador de la señal. Este elemento recibe la señal de salida del elemento detector y la modifica ya sea amplificándola o cambiando su forma de onda. Cuando la señal emerge del modificador de señal, debe tener la forma apropiada para ser desplegada o registrada. El tercer elemento de un sistema de medición es el dispositivo registrador. En los sistemas eléctricos, los dispositivos registradores incluyen instrumentos tales como medidores, tubos de rayos catódicos, registradores de papel, grabadores, registradores X-Y y computadores digitales.

Si el sistema de medición es tal que una cantidad no eléctrica se va a medir convirtiéndola a una señal eléctrica, se utiliza un transductor eléctrico como elemento detector (o sensor). El transductor eléctrico se llama un transductor activo si es capaz de producir una señal sin necesidad de una excitación eléctrica. Si el transductor es capaz de producir una señal de salida únicamente cuando se usa con una fuente de excitación, el transductor es pasivo.

2.2 USO DE MICROCONTROLADORES EN EL DISEÑO DE SISTEMAS DE ADQUISICION DE DATOS

En la actualidad los sistemas enfocados a la instrumentación y control de procesos son de bajo costo, versátiles y flexibles, esto se debe en gran medida al empleo de modernos dispositivos electrónicos que simplifican enormemente las tareas, pero en especial se debe al empleo de un dispositivo electrónico revolucionario, el cual se denomina microcontrolador. Este dispositivo no es más que un circuito integrado de muy alta escala de integración (cientos de miles de transistores integrados en una pequeña pastilla de silicio), que contiene una unidad central de procesamiento (CPU) y una serie de periféricos interconectados entre sí. Por ejemplo, los microcontroladores de la familia M68HC11 de Motorola, contienen una unidad central de procesamiento, tres clases distintas de memorias (ROM, RAM y EEPROM), un convertidor tipo Analógico/Digital (A/D), interfaces para comunicación serial síncrona y asíncrona, un subsistema acumulador de pulsos, además de una serie de puertos de entrada y salida.

El desarrollo de sistemas fundamentados en el empleo de microcontroladores requiere el conocimiento y la coordinación de diversas herramientas. La medición de una sola variable de un proceso determinado, como puede ser la temperatura, requiere conocimientos sobre sensores, transductores y circuitería analógica, ya que la temperatura se debe transformar en una señal de voltaje que se encuentre en el intervalo de valores que admite el convertidor A/D del microcontrolador. Otro caso puede ser el control de procesos, en donde se requiere la activación de motores o relevadores por medio de instrucciones del microcontrolador, para esto se requieren, además de los conocimientos anteriormente mencionados, conocimientos de tecnología digital y de dispositivos semiconductores de potencia como transistores, MOSFET's, SCR's y triacs. Más importante, sin embargo, es tener los conocimientos, el ingenio y la habilidad para programar el microcontrolador, esto si es que deseamos que forme parte de la solución a un problema específico.

2.2.1 ALGUNOS PROBLEMAS RESOLUBLES CON AYUDA DE SISTEMAS BASADOS EN MICROCONTROLADORES.

A continuación se tiene una breve descripción de algunos problemas prácticos, de diversas áreas, que pueden llegar a ser resueltos implementando un sistema de adquisición de datos o de control, cuyo funcionamiento esté apoyado principalmente en un microcontrolador.

- En el área de la *Biología* se hace necesario controlar el nivel de iluminación de un microscopio sobre un espécimen a observar, con la finalidad de que la elevación de la temperatura debida a la iluminación, no dañe al espécimen.
- En *Ingeniería Mecánica*, se necesita saber a qué presión debe encontrarse la gasolina dentro de un motor para que se dé el proceso de combustión, ésto con la finalidad de medir el rendimiento de dicho motor.
- En *Ingeniería Civil* es necesario medir y controlar la temperatura del concreto, con la finalidad de garantizar su apropiada hidratación.
- En el campo de las ciencias *físicas* es necesario medir y controlar el campo magnético generador a través de electroimanes para alinear las partículas dentro de un acelerador de partículas.
- En una *granja* es necesario medir y controlar la humedad relativa de los granos dentro de un silo con la finalidad de prevenir una posible explosión.

Cabe mencionar que ligados al microcontrolador y como parte de la solución a los problemas anteriormente mencionados, debe haber una serie de circuitos (analógicos, digitales, de potencia, etc.) tanto para señales de entrada, como de salida.

2.2.2 PROCEDIMIENTO PARA EL DISEÑO DE UN SISTEMA DE ADQUISICION DE DATOS O DE CONTROL BASADO EN UN MICROCONTROLADOR

La mayoría de los problemas relacionados con la adquisición de datos y el control de procesos, comienzan con la medición de una cantidad física, siempre y cuando se realice con el sensor apropiado. La salida analógica del sensor es la que debe acondicionarse o incluso transformarse con ayuda de algún transductor, para poder ser admitida por el microcontrolador, el cual posteriormente con ayuda del convertidor A/D la convertirá en un valor digital. Los valores digitales que representan las mediciones son analizados y procesados por la unidad central de procesamiento del microcontrolador, ésto se hace con ayuda de programas previamente grabados, para que finalmente a través de alguno de los puertos de comunicación exista una respuesta a las mediciones hechas. Las respuestas son variadas y pueden ir desde prender un tablero luminoso de LED's hasta activar un motor o una impresora.

Así tenemos que de manera general, los pasos a seguir para diseñar un sistema de adquisición de datos o de control, que se fundamente en un microcontrolador, son los siguientes:

1. Seleccionar un transductor o sensor apropiado para medir la cantidad física deseada y que de preferencia nos proporcione como salida una señal eléctrica.

2. Diseñar un circuito acondicionador de la señal eléctrica que sea capaz de adecuarla de acuerdo a las especificaciones del convertidor A/D integrado en el microcontrolador. En otras palabras, el acondicionador es la interfaz entre el transductor y el convertidor A/D.
3. Escribir el programa que sea capaz de adquirir datos, procesarlos y que proporcione una respuesta adecuada a través de alguno de los puertos de salida.
4. Diseñar los circuitos que sirvan de interfaz entre la respuesta digital del microcontrolador y los dispositivos periféricos que se deseen activar.

A continuación se muestra en forma de diagrama en bloques un sistema completo de adquisición de datos y de control, en donde el microcontrolador juega un papel principal en el correcto funcionamiento del sistema. En este ejemplo el microcontrolador utilizado es de la familia M68HC11 de Motorola, el cual digitaliza los valores medidos por el transductor, con ayuda del convertidor A/D; después procesa dichos valores, los comunica a una computadora y finalmente como respuesta acciona con ayuda de una interfaz digital un motor de DC.

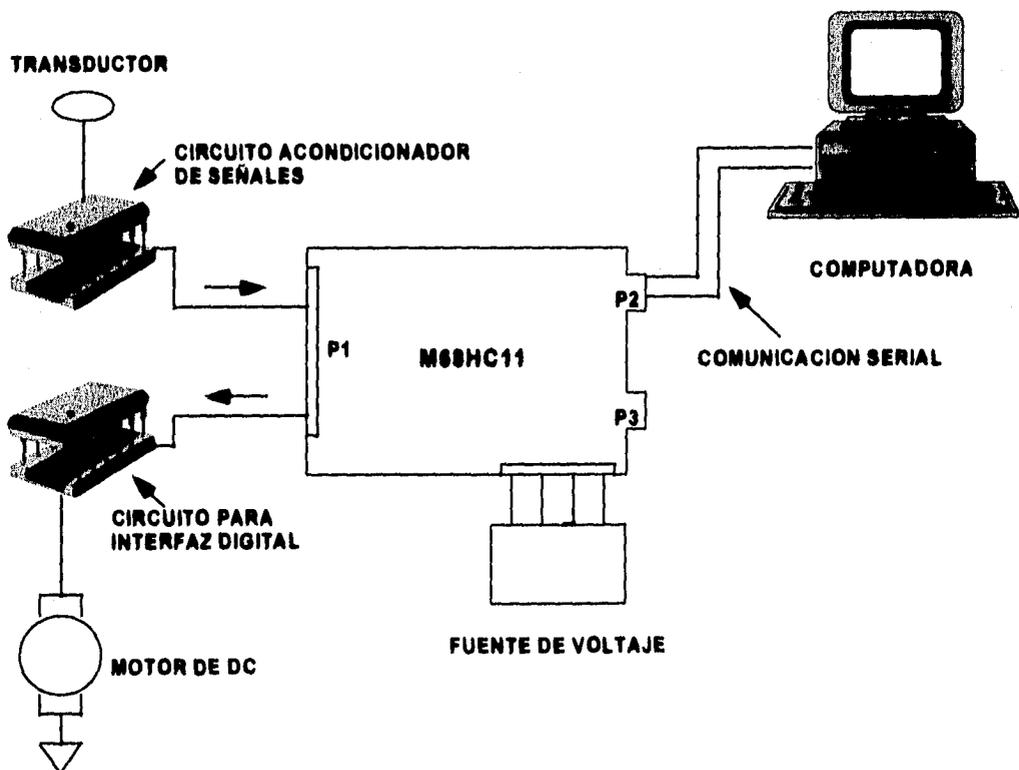


Figura 2. Diagrama de bloques de un sistema completo de adquisición de datos y control basado en un microcontrolador de la familia M68HC11.

2.3 TRANSMISION DE DATOS DIGITALES.

En este apartado se tratarán los aspectos básicos sobre las comunicaciones digitales haciendo énfasis principalmente en las características de la transmisión de datos digitales a través del puerto serie de la computadora (RS - 232C).

2.3.1 VELOCIDAD DE TRANSMISION: BPS Y BAUDS.

La primera confusión que aparece en el área de las comunicaciones digitales, y en específico, en la transmisión de datos digitales, es la referente a lo que se conoce como "velocidad de transmisión". La velocidad de transmisión, indicada generalmente en bits por segundo, es el término que debe tomarse como correcto, aunque a menudo aparezca erróneamente denominada en bauds.

Entonces, si "velocidad en bauds" y "velocidad en bps" no significan lo mismo y a menudo se utilizan como sinónimos para definir la "velocidad de transmisión", ¿dónde radica la diferencia?. Un baud es igual al número de cambios de la señal por segundo; mientras que el término "bps" expresa la velocidad de cambios en la señal, constituye una medida de la velocidad de transmisión y es igual al número de bits de datos por señal, multiplicado por los bauds.

Baud = cambios de la señal por segundo.

bps = (grupo de bits de datos) x velocidad en baudios

2.3.2 TRANSMISION SERIE Y PARALELO.

Los dispositivos digitales como son la computadoras operan en grupos de ocho bits, llamados bytes, estos paquetes de bits pueden intercambiarse con varios dispositivos (como pueden ser modems e impresoras), a través de un "interfaz". La interfaz es una placa de circuito impreso con una base para conector. Dentro del entorno PC existen dos interfaces estándar: Paralelo y Serie.

Cuando los datos se transmiten bit por bit a través de un sólo cable, se le denomina transmisión serial, si por el contrario, los datos se transmiten por medio de 8 cables (un bit por cable), la transmisión se le denomina paralela.

La **transmisión paralela** utiliza una línea adicional, además de las ocho de datos, denominada línea de reloj o de sincronización. Los datos son enviados de acuerdo con esta señal, y el dispositivo receptor utiliza este reloj para sincronizarse con los datos que se le están enviando en paralelo. La mayoría de los equipos PC vienen equipados con puertos paralelo unidireccionales, que sólo pueden enviar datos, ésto debido a que están pensados para ser utilizados con impresoras. Una prestación interesante y que constituye una ventaja sobre la transmisión serial, es la capacidad de transmitir datos a una velocidad más elevada que en un momento dado puede llegar a ser la velocidad máxima del dispositivo con el que esta conectado. Una desventaja de este tipo de transmisión es el elevado costo que involucra, ya que se requieren 11 cables como mínimo sin incluir las señales de control. Por otra parte las distancias permisibles para esta transmisión están muy restringidas, debido a que dentro del ducto (bus) de comunicación, los 8 respectivos cables se comportarán como un filtro paso bajas, entorpeciendo la transmisión a altas frecuencias.

En la **transmisión serial** los ocho bits de datos pasan por el canal serie de uno en uno, en un orden determinado y a una velocidad predeterminada mucho menor a la que se maneja en la comunicación paralela. Pero como se mencionó anteriormente, en la transmisión paralelo necesitamos 11 cables como mínimo, mientras que para la comunicación serial solamente necesitamos 3 cables sin incluir las señales de control, esta limitación es la que ha dado lugar a que en la actualidad la transmisión paralelo solamente se use con impresoras. La interfaz para comunicación serial de los equipo PC utiliza exclusivamente la transmisión asíncrona, en la que los bits de datos se transmiten por un sólo canal de datos de forma consecutiva, uno tras otro. La transmisión asíncrona necesita un entorno estable, si se produce cualquier irregularidad en la línea, y le sucede algo a un bit de datos, los bits siguientes pueden ser leídos erróneamente.

Las características tanto de las comunicaciones asíncronas como de las síncronas serán discutidas en el apartado siguiente, junto con las ventajas y desventajas de cada una.

2.3.3 COMUNICACION SINCRONA Y ASINCRONA

Estos términos nos describen el cómo se realiza la transmisión digital.

La **comunicación asíncrona** es un tipo de comunicación serial utilizada para transmitir datos entre dispositivos. "Asíncrono" se refiere a que la temporización de la transmisión de un caracter es independiente del resto de caracteres. También se le puede llamar ASYNC.

En este tipo de comunicación se pueden distinguir cuatro bloques fundamentales:

- 1) Bit de inicio.
- 2) Palabra de mensaje o bits de datos.
- 3) Bit de paridad.
- 4) Bit de paro.

En la figura 3 se puede observar un ejemplo de transmisión de datos en forma asíncrona.

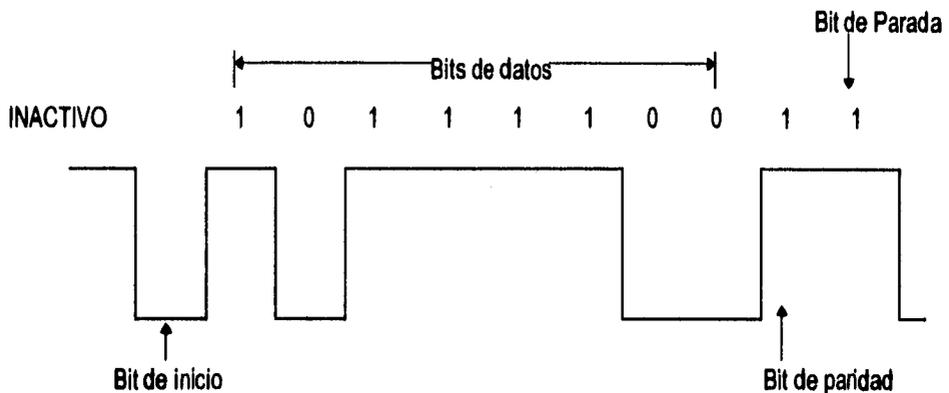


Figura 3. Bits de datos Asíncronos

Bit de inicio. El bit de inicio es el primer bit que precede cada palabra de datos (transmisión de datos o caracter transmitido). Si recordamos que en las comunicaciones seriales básicas y asíncronas, cada caracter se transmite de forma independiente. Esto significa que cada caracter transmitido necesita un bit de inicio. Inmediatamente después del bit de inicio, aparecen los bits de datos en serie.

Bit de Parada. En términos reales no es un bit, sino un medio o un pulso completo, en que no se transmite absolutamente nada, este tiempo muerto tiene como finalidad que el dispositivo receptor tenga tiempo suficiente para direccionar, almacenar o procesar el paquete transmitido, además de indicarnos que el conjunto de datos ya ha sido enviado.

Bit de paridad. Para asegurarse de que un caracter es recibido correctamente, se añade un bit inmediatamente después de los bit de datos, al cual denominamos bit de paridad. No se utiliza siempre, pero cuando se llega a utilizar, la paridad puede ser par o impar, ésto se verifica contando el número de 1's incluidos en los bits de datos (los bits de inicio y parada no se tienen en cuenta). Cuando recibimos un caracter, se cuenta el número de 1's y se compara con el bit de paridad, verificando así si algunos de los bits

de datos han cambiado durante la transmisión. Si la paridad no coincide, se indica el error, y se puede producir la retransmisión de los datos.

La **comunicación síncrona** es un sistema de transmisión de datos dependiente de la sincronización precisa de los datos transmitidos y de los mecanismos de transmisión y de envío. La gran diferencia que notaremos en este tipo de comunicación es la existencia de dos señales adicionales (reloj de transmisión y reloj de recepción), en las cuales se encuentra la información de temporización. Los bits y caracteres de datos se envían a una velocidad fija, estando sincronizados los dispositivos emisor y receptor. Esto elimina los elementos de inicio y final, y ofrece una mayor eficiencia. Como consecuencia de todo lo anterior, la comunicación no depende del largo ni de la composición de la transmisión. Este tipo de comunicación es en esquema más eficiente y más rápida que la asíncrona, pero el hecho de poseer relojes sincronizados entre sí en ambos lados de la transmisión da como resultado un tipo de comunicación de alto costo.

2.3.4 PROTOCOLOS.

La comunicación de datos demanda el intercambio de información entre dispositivos antes y después de enviar un nuevo mensaje. Al inicio de una transmisión el dispositivo transmisor debe alertar al receptor de su intención de enviar un mensaje y espera una respuesta por parte del dispositivo receptor de estar en posibilidad de recibir dicho mensaje, a este proceso se le denomina protocolo de comunicación.

De manera más formal, un protocolo es un conjunto de reglas que regulan algún tipo de comunicación entre dispositivos, por lo general computadoras. Las computadoras poseen protocolos que anuncian el envío de grandes o pequeños paquetes de información, y si éstos fueron recibidos satisfactoriamente por el dispositivo receptor. Un claro ejemplo de este tipo de protocolos son los protocolos de verificación de errores, los cuales controlan cierta información enviada con cada bloque de datos; si los datos recibidos no concuerdan con la información de verificación, el sistema notifica al emisor que se ha producido un error, y solicita la retransmisión de los datos. Este proceso continúa hasta que se reciben todos los datos (señal que también forma parte del protocolo), o la transferencia es interrumpida por alguna razón.

Los protocolos se dividen en dos categorías principales de acuerdo a la técnica utilizada para enmarcar el mensaje. Estas dos técnicas son el protocolo orientado a carácter y el protocolo orientado a bit.

PROTOCOLO ORIENTADO A CARACTER. El protocolo orientado a carácter se basa en el código binario de un conjunto de caracteres. El código más comúnmente utilizado es el ASCII. Es un código de 7

bits, con un total de 128 caracteres, de los cuales 95 son caracteres alfanuméricos y 33 son caracteres de control. Los caracteres que controlan la transmisión se denominan "Caracteres de Control de Comunicación". Dichos caracteres se enumeran a continuación, cada uno de los caracteres tiene un código de 7bits y se reconocen con un símbolo de 3 letras. El papel de cada uno de los caracteres en el control de transmisión de datos es enunciado brevemente en la columna "Función" de la tabla 1.

CODIGO	SIMBOLO	SIGNIFICADO	FUNCION
0010110	SYN	ESPERA SINCRONIA	ESTABLECE EL SINCRONISMO
0000001	SOH	ARRANQUE DEL TITULO	TITULO DEL BLOQUE DE MENSAJE
0000010	STX	ARRANQUE DEL TEXTO	PROCEDE EL BLOQUE DEL TEXTO
0000011	ETX	FINAL DEL TEXTO	TERMINA EL BLOQUE DEL TEXTO
0000100	EOT	FINAL DE LA TRANSMISION	CONCLUYE LA TRANSMISION
0000110	ACK	RECONOCIMIENTO	RECONOCIMIENTO AFIRMATIVO
0010101	NAK	NO RECONOCIMIENTO	RECONOCIMIENTO NEGATIVO
0000101	ENQ	PREGUNTA	PREGUNTA SI EL TERMINAL ESTA ENERGIZADO
0010111	ETB	FINAL DE LA TRANSMISION	FINAL DEL BLOQUE DE DATOS
0010000	DLE	ESCAPE DEL ENLACE DE DATOS	CARACTER DE CONTROL ESPECIAL

Tabla 1. Caracteres de Control de Comunicación.

SYN	SYN	SOH	ENCABEZADO	STX	TEXTO	EOX	BCC
-----	-----	-----	------------	-----	-------	-----	-----

Figura 4. Formato típico de mensaje de un Protocolo Orientado a Caracter.

PROTOCOLO ORIENTADO A BIT. El protocolo orientado a bit no utiliza caracteres en su campo de control y es independiente de cualquier código en particular. Permite la transmisión sucesiva de información en forma de paquetes de cualquier longitud sin la implicación de fronteras de carácter. Los mensajes son organizados en un formato específico denominado "un marco". Además del campo de información, un marco contiene una dirección, control y campos de verificación de error. Los límites del marco son determinados de acuerdo a un número especial de 8 bits denominado "bandera". Ejemplos de los protocolos orientados a bit

son SDLC, HDLC y ADCCP. El formato de un marco para el protocolo orientado a bit es el que a continuación se muestra en la figura 5.

BANDERA	DIRECCION	CONTROL	INFORMACION	VERIFICACION	BANDERA
01111110	OCHO BITS	OCHO BITS	N BITS	DE MARCO	01111110

Figura 5. Formato de marco para un Protocolo Orientado a Bit

Un marco comienza con una bandera de 8 bits 01111110 seguida por una secuencia de dirección y control. El campo de información no está restringido en formato o contenido y puede tener cualquier longitud. El campo de verificación de marco es una secuencia CRC (Verificación de Redundancia Cíclica) utilizada para detectar errores en la transmisión. La bandera finalizadora indica a la estación receptora que los 16 bits justamente recibidos constituyen los bits de CRC. El marco de finalización puede ser seguido de otro marco, otra bandera, o una secuencia de 1's consecutivos. Cuando los dos marcos siguen el uno al otro, la bandera que interviene es simultáneamente la bandera de finalización del primer marco y la bandera de comienzo del marco siguiente.

2.3.5 INTERFAZ DE LINEA RS-232C.

La interfaz RS - 232C es una interfaz electrónica estándar que se utiliza para la conexión de componentes a un sistema tales como impresoras, modems y computadoras.

El estándar fue establecido por la Asociación de Industrias Electrónicas (EIA, Electronic Industries Association), una organización de comercio industrial. La RS - 232C define el camino para una señal a través de 25 conductores en base a 18 circuitos con un retorno común a través de tierra. El estándar también define los voltajes (los intervalos de uno y cero lógicos) usados en todos los circuitos y previene la posibilidad de que se produzca un daño físico en éstos como resultado de un corto circuito entre los cables o conectores; ésto asegura que no habrá ningún daño mientras se experimente.

Como se mencionó, el estándar RS - 232 cuenta con 25 conductores, debido a que fue definido por un comité que tenía que cubrir todas las necesidades de una industria entera, sin embargo, ésto no significa que deben usarse todos los conductores en la mayoría de los casos. Como consecuencia de ésto, surgieron dos variantes de dicho estándar, la que maneja los 25 conductores (con conectores tipo DB25) y la que solamente maneja 9 conductores (con conectores tipo DB9), que conforme pasa el tiempo es utilizada con

mayor frecuencia y aceptación en los equipos nuevos, dejando cada vez más en el olvido a los 25 conductores que se utilizaron en un principio.

En una computadora personal, las señales lógicas se conforman en base a lo que conocemos como niveles TTL, un estándar para la interconexión de circuitos integrados. En esta tecnología un voltaje entre 2 y 5 volts representa un "1" lógico, y un voltaje entre 0 y 0.8 volts representa un "0" lógico. Sin embargo, estos niveles no se usarán fuera de la computadora personal a causa de su insuficiente inmunidad al ruido electrónico. En su lugar se usa el esquema de la figura 6, en donde el voltaje es quien define el estado lógico de un circuito, no la presencia o ausencia de corriente, como en los sistema telegráficos.



Figura 6. Niveles de Voltaje del estándar RS-232C

Una computadora personal, así como la mayoría del equipo de comunicaciones, producen señales de salida en un intervalo de -12 a +12 volts. No obstante, una entrada de -3 a +3 volts es suficiente para definir un estado lógico. La figura 7a muestra una señal limpia entrando por un cable de 30 metros de longitud. La señal ruidosa que se produce por el otro extremo aparece en la figura 7b y la figura 7c indica como aparece la señal dentro de la computadora personal receptora.

Obsérvese que los umbrales en el extremo receptor son distintos a los del extremo emisor, debido a la degradación de la señal en el cable, nótese como una señal bajo el estándar RS-232C tiene la polaridad invertida respecto a la señal TTL en el interior de la computadora personal; un "1" lógico TTL de 5 volts equivale a un "1" lógico (o marca) de -12 volts.

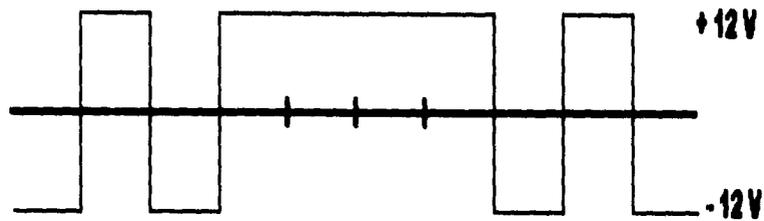


Figura 7a. Señal Transmitida

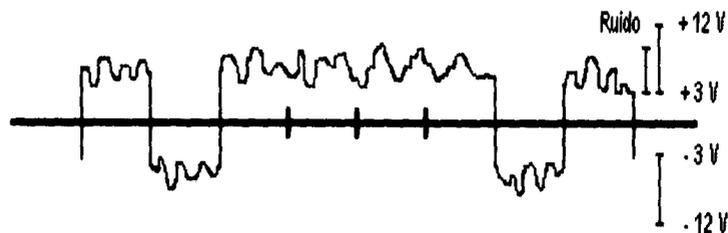


Figura 7b. Señal Recibida



Figura 7c. Señal TTL Reconstruida

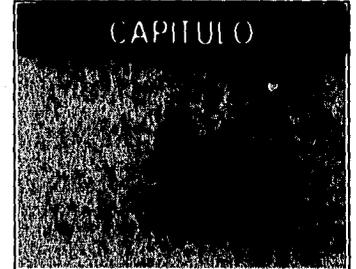
Los cables terminan en conectores, cuya constitución física no está definida por el estándar. Los conectores tienen terminales, y las terminales están numeradas del 1 al 25 (para el caso de DB25) o del 1 al 9 (para el caso de DB9). El estándar RS-232C se refiere a números de las terminales. Las terminales de la RS-232C que se utilizan en un puerto serie típico (DB25) para una computadora personal, se definen en la tabla 2 que se muestra a continuación, así como su correspondencia con las terminales del conector de 9 terminales (DB9).

En la mayoría de las tareas de comunicación de datos, los circuitos más importantes son los de transmisión y recepción de datos, terminales 2 y 3 respectivamente para el tipo de conector DB25. Estos son los conductores por los que se envían los datos y reciben simultáneamente, el resto de los conductores con excepción de la tierra (terminal 7), son circuitos de control.

Nº DE TERMINAL DB25	Nº DE TERMINAL DB9	SEÑAL	DESCRIPCIÓN
1	No hay equivalencia	GND	Tierra de Protección
2	3	TxD	Salida de datos
3	2	RxD	Entrada de datos
4	7	RTS	Request to Send Petición para inicio de envío de datos
5	8	CTS	Clear to Send Listo para recibir datos
6	6	DSR	Data Set Ready Datos listos para ser transferidos
7	5	GND	Tierra de la Señal
8	1	CD	Carry detect Detector de portadora
20	4	DTR	Data terminal Ready Aviso de comunicación de datos activa

Tabla 2. Terminales importantes de la interfaz RS - 232C para DB25 y DB9

CARACTERIZACION DE LOS TRANSDUCTORES A UTILIZAR



La primera etapa del diseño para el Sistema de Adquisición y Despliegue de Información consiste en elegir los transductores y/o sensores que se utilizarán, ya que de acuerdo con las características de éstos, se diseñará la etapa de acondicionamiento de señales que requiera el sistema.

3.1. CRITERIOS PARA LA SELECCION Y USO DE TRANSDUCTORES.

Cuando la medición de una cantidad no eléctrica se va a llevar a cabo convirtiendo la cantidad a una señal eléctrica, se debe seleccionar un transductor (o una combinación apropiada de ellos) para llevar a cabo la conversión. El primer paso en el proceso de selección es el de definir claramente la naturaleza de la cantidad a medir. Esto también incluye el conocimiento del intervalo de magnitudes y frecuencias que se espera que la cantidad exhiba. Cuando el problema se ha establecido, se deben examinar los principios fundamentales de funcionamiento del transductor aconsejable para el tipo de medición. En caso de no existir ningún transductor capaz de producir una señal satisfactoria, debemos decidir entre construir uno o utilizar alguno de los disponibles en el mercado, realizando las adaptaciones y/o combinaciones pertinentes.

Cuando las especificaciones de un transductor particular se van a examinar, los siguientes puntos se deben considerar para determinar su conveniencia para una medición:

- 1. Intervalo.** El intervalo de valores del transductor debe ser lo suficientemente grande para abarcar todas las magnitudes esperadas de la cantidad a medir.
- 2. Sensibilidad.** Para obtener un dato significativo, el transductor debe producir una señal de salida suficiente por unidad de la entrada medida.

3. **Características de la salida eléctrica.** Las características eléctricas (tales como la impedancia de salida, la respuesta en frecuencia y la respuesta en el tiempo) de la señal de salida del transductor deben ser compatibles con el dispositivo registrador y el resto del equipo del sistema de medición.

4. **Ambiente físico.** El transductor seleccionado debe ser capaz de soportar las condiciones ambientales a las cuales puede estar sujeto mientras hace las mediciones. Parámetros tales como la temperatura, humedad y químicos corrosivos pueden dañar algunos transductores pero no otros.

5. **Errores.** Los errores inherentes en la operación del transductor o aquellos causados por las condiciones ambientales deben ser lo suficientemente pequeños o controlables, de tal forma que no sean significativos en los datos medidos.

Una vez que el transductor se ha seleccionado e incorporado dentro del diseño del sistema de medición, se deben observar los siguientes criterios para incrementar la exactitud de las mediciones:

1. **Calibración del transductor.** La salida del transductor se debe calibrar con respecto a algún patrón conocido mientras se utiliza bajo las condiciones que va a trabajar. Esta calibración se debe efectuar regularmente.

2. Los cambios en las condiciones ambientales del transductor se deben registrar continuamente. Si se sigue este procedimiento, los datos obtenidos se pueden corregir más tarde tomando en cuenta los cambios en las condiciones ambientales.

3. Se pueden reducir los posibles errores de un transductor controlando artificialmente el medio ambiente. Ejemplos del control artificial del medio ambiente del transductor incluyen su encapsulamiento en una caja o gabinete o temperatura controlada o aislar el dispositivo de las vibraciones y choques externos.

3.2. TRANSDUCTOR A UTILIZAR PARA LA VARIABLE TEMPERATURA.

Con la finalidad de medir la temperatura en los materiales a laminar, la división de Ingeniería Mecánica adquirió con anterioridad un sistema digital de medición, el cual consta de un sensor de lectura indirecta, un transductor del tipo termopila y una unidad de acondicionamiento y despliegue de información digital con pantalla de cuarzo. Debido a que cuando se nos asignó el proyecto, el sistema de medición para la

temperatura ya se encontraba funcionando, tuvimos que adaptarnos a la utilización del transductor de dicho sistema en lugar de hacer todo un análisis para seleccionar uno nuevo. La ventaja es que las características eléctricas del sistema adquirido son óptimas para utilizarse en el Sistema de Adquisición y Despliegue de Información a desarrollar.

El sensor de temperatura es un pirómetro de radiación que no necesita estar en contacto con el material al cual se va a medir la temperatura, ésto es una ventaja ya que los termómetros convencionales pierden su linealidad o inclusive su forma física al tener contacto con materiales cuya temperatura sea superior a los 500 ° C. En este tipo de sensores el límite inferior depende de la compensación de la junta fría de la termopila y el límite superior no existe. Se entiende por termopila al conjunto de termopares conectados en serie, dispuestos de tal forma que las juntas calientes se encuentran en el campo de radiación a la entrada del sensor; y todas las juntas frías están en contacto térmico con la caja del pirómetro, de tal forma que éstas se encuentran a temperatura ambiente.

Para funcionar, el sensor enfoca la radiación mediante un lente sobre el elemento térmico sensible (termopila), en donde la elevación de temperatura en dicho elemento es directamente proporcional a la radiación total que reciba. La termopila, por otra parte, entregará una diferencia de potencial proporcional a la temperatura que le haya sido radiada; dicha diferencia de potencial posteriormente se amplifica para ser registrada y desplegada en la pantalla del sistema.

A continuación se enumeran algunas de las características del sistema de medición de temperatura que lo hacen conveniente para su aplicación en la laminadora y en el Sistema de Adquisición y Despliegue de Información:

- Amplo intervalo de medición: 200 a 1200 ° C (400 a 2200 ° F) , sin tener contacto con el material
- Un corto tiempo de respuesta de 400 milisegundos
- Impedancia de salida de 50 ohms
- Salida analógica con resolución de 1 mV por grado (C o F)
- Intervalo de corriente de 4 a 20 mA para la salida analógica
- Opción de comunicación digital tipo serial bajo el estándar RS 232
- Temperatura para la operación normal del transductor: 0 a 65 ° C
- Temperatura para la operación normal de la unidad de acondicionamiento y despliegue: 0 a 40 °C
- Inmunidad a vibraciones externas con valores de frecuencia entre 11 y 200 Hz.

Todas estas características hacen que sea posible la incorporación de este sistema para medir temperaturas al Sistema de Adquisición y Despliegue de Información de la Laminadora Hille 25, haciendo los acondicionamientos y adaptaciones pertinentes tanto al sistema como a las señales eléctricas.

3.3. TRANSDUCTOR A UTILIZAR PARA LA VARIABLE CARGA.

En el caso del transductor para la variable carga, existe un caso similar al de la variable temperatura, la División de Ingeniería Mecánica adquirió un sistema completo para medir la carga que en toneladas se ejerce sobre el material al momento de laminarlo. Dicho sistema cuenta con cuatro celdas de carga colocadas en los principales puntos de apoyo de los rodillos de laminación, las cuales funcionan como sensor y transductor ya que dichas celdas al ser excitadas con un voltaje exterior, nos entregan una señal eléctrica proporcional a la carga ejercida sobre la celda. Además de las celdas de carga, el sistema cuenta con una unidad sumadora, en la cual se suman las señales eléctricas proporcionadas por las cuatro celdas, con la finalidad de obtener una señal promedio proporcional a la carga total que se ejerce sobre el material al momento de laminarlo. Por último existe una pantalla digital en el cual se muestra el valor de la carga total ejercida al momento de laminar y la forma en que varía a lo largo del proceso.

Es importante mencionar que la carga máxima que puede ejercer la laminadora sobre un material al momento de laminarlo, es de 30 toneladas y que la carga de laminación que por lo general se utiliza experimentalmente es de 20 toneladas; por lo tanto los transductores deben tener como característica principal que la carga máxima que soporten sea mayor a 20 y menor o igual a 30 toneladas

A continuación se muestran las principales características que hacen de estas celdas de carga los transductores adecuados para la laminadora y para el Sistema de Adquisición y Despliegue de Información a desarrollar.

- Máxima carga medible 50,000 lbs (22680 Kgs)
- Linealidad: 0.2 %
- Histéresis: 0.2 %
- Señal de salida: 2 mV / V
- Voltaje de excitación: 10 V
- Resistencia del arreglo tipo puente: 350 ohms (Las cuatro celdas de carga se colocan en un arreglo tipo puente como el de la figura 1)

- Amplio intervalo de temperatura de operación: - 65 ° F a 250 ° F

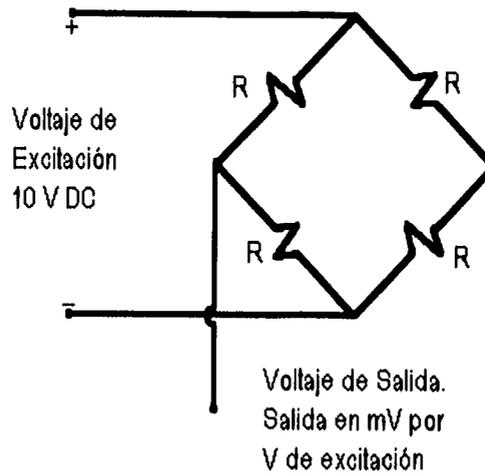


Figura 1. Configuración tipo puente (4 celdas de carga)

A pesar de que las celdas de carga instaladas no alcanzan a medir la máxima carga que puede ejercer la laminadora sobre un material (30 toneladas), su rango de medición se adecua a las necesidades de los experimentadores de la División de Ingeniería Mecánica, ya que por lo general la carga de laminación utilizada no excede las 20 toneladas. Tomando en cuenta lo anterior y las demás características, tanto eléctricas como mecánicas, de las celdas de carga, podemos afirmar que son los transductores adecuados para trabajar con la variable CARGA del Sistema de Adquisición y Despliegue de Información, sin olvidar hacer los acondicionamientos necesarios.

3.4. TRANSDUCTOR A UTILIZAR PARA LA VARIABLE VELOCIDAD.

Para poder medir la velocidad angular de la flecha del motor al momento de laminar, se deben tener en cuenta las siguientes consideraciones:

- La velocidad de la flecha del motor, en r.p.m., va desde 0 hasta 1640.
- El proceso de laminado experimental se realiza a velocidades pequeñas.
- Debido a que en este proceso las piezas mecánicas que intervienen siempre deben estar perfectamente lubricadas, existe gran cantidad de grasa en el medio y por lo tanto se debe pensar en un sensor y/o transductor, cuyo funcionamiento no se vea afectado por la grasa y lubricantes propios de la laminadora.

- Como en este proceso se imprime una gran fuerza sobre el material a laminar, ésto se traduce en vibraciones que producen ruido y en consecuencia se debe tener un transductor con cierta inmunidad al ruido externo generado.

Tomando en cuenta las consideraciones anteriormente descritas, el transductor a utilizar deberá ser libre de mantenimiento, que no le afecten las vibraciones del medio y que los lubricantes y grasas del ambiente no afecten su funcionamiento.

Por lo tanto, se propone utilizar como transductor un sensor de efecto Hall, tipo interruptor, el cual produce una señal eléctrica al incidir sobre él un campo magnético. Una de las grandes ventajas de este sensor, es que con una polarización única puede detectar la presencia de diversos tipos de campos magnéticos, ésto es, el sensor al detectar un campo tipo NORTE nos dará a la salida el voltaje de polarización del mismo, mientras que al detectar un campo tipo SUR nos dará cero volts. Dichos campos se pueden producir mediante pequeños imanes que se coloquen alrededor de la flecha principal de transmisión de la laminadora; el número de imanes que se coloquen en la flecha dependerá de las velocidades que se empleen al momento de laminar, las cuales suelen ser muy pequeñas por lo general. A fin de cuentas, los imanes colocados en la flecha producirán una serie de pulsos que deberán ser transmitidos a uno de los registros de nuestro microcontrolador, el cual mediante algún algoritmo deberá obtener la velocidad angular en radianes por segundo, para posteriormente transmitir los datos a la computadora tipo PC.

En el mercado existe un sensor que cubre todas las necesidades expuestas anteriormente, se trata del TL170C, su costo no excede los N\$ 20 y además se puede polarizar con 5 volts, lo cual lo hace compatible con el microcontrolador que se utilizará en el sistema. En lo que a los imanes se refiere, existen en el mercado los que se componen de tierras raras, los cuales se caracterizan por su muy reducido tamaño y gran campo magnético que generan, siendo éstos los más apropiados para ser colocados alrededor de la flecha principal de la laminadora; su costo no excede los N\$ 50.

El costo del sensor y de los imanes, así como la facilidad de implementación; hacen que sea más factible diseñar el transductor para medir la velocidad angular de los rodillos de la laminadora, en base al TL170C y los imanes compuestos de tierra raras, que adquirir alguno de los ya existentes en el mercado.

3.5 TRANSDUCTOR A UTILIZAR PARA LA VARIABLE PAR DE LAMINACION.

En la actualidad, gracias a los avances alcanzados en el campo de la instrumentación electrónica y medición de variables físicas, podemos encontrar en el mercado sistemas de medición con un alto grado de sofisticación, encaminados generalmente a la medición de variables muy complejas. En la mayoría de los casos, dichos sistemas vienen acompañados de un elevado costo en dólares y un largo periodo de espera para su entrega e instalación, ésto se debe a que se trata de sistemas totalmente de importación y a que la misma complejidad de la variable a medir no los hace muy comerciales.

Tal es el caso de la variable par de laminación, la cuarta variable, y tal vez la más difícil, sobre la cual pretende trabajar el Sistema de Adquisición y Despliegue de Información a desarrollar; ya que con los sistemas capaces de medir directamente una variable de este tipo, además del alto costo y la tardanza en su entrega, existen problemas relacionados con el montaje de dichos sistemas en la laminadora y con la poca variedad de los mismos en el mercado. A continuación analizamos los principales factores que hacen extremadamente difícil la utilización de algún transductor y/o sistema que trabaje directamente con esta variable.

Analizando el factor económico, encontramos que el costo de los sistemas, los cuales se encuentra integrados por un transductor y un módulo de despliegue de información digital, oscila entre los 1500 y los 5500 dólares, dependiendo del intervalo de medición deseado, existiendo la posibilidad de adquirir solamente el transductor, el cual tiene un costo que varía desde los 1000 y hasta los 4000 dólares, lo cual sigue siendo un costo bastante elevado.

Antes de analizar el problema del montaje haremos una breve descripción de cómo se transmite la fuerza desde el motor de DC hasta los rodillos para llevar a cabo el proceso de laminación. Primero tenemos que la flecha del motor de DC se acopla directamente a un reductor de velocidad, posteriormente la flecha de salida de dicho reductor de velocidad se acopla a una caja de transmisión con engranes 1 a 1, de tal forma que a la salida de esta caja de transmisión tenemos dos tomas de fuerza, una para cada rodillo, las cuales con ayuda de flechas o ejes propulsores ajustables se unen a los rodillos y proporcionan la fuerza necesaria para llevar a cabo el proceso de laminación; cabe mencionar que todo ésto se encuentra montado en una mesa de trabajo especialmente diseñada para la laminadora. En la figura 2 se muestra una descripción esquemática de la laminadora Hille 25 con las partes anteriormente mencionadas.

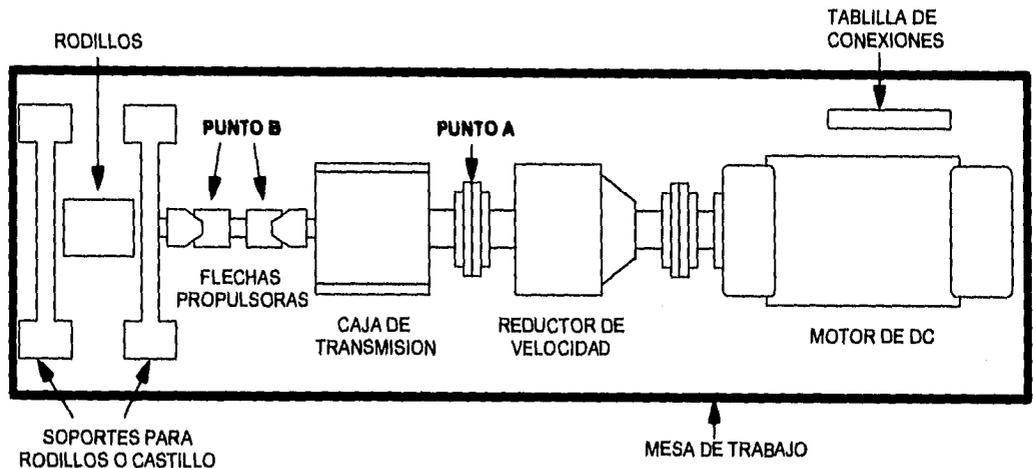


Figura 2. Descripción esquemática de la Laminadora Hille 25 (Vista de planta)

Ahora, el problema del montaje surge debido a que el transductor que emplean los sistemas encargados de trabajar con esta variable, se encuentra representado por un medidor de tensión física que para su funcionamiento debe acoplarse directamente a la flecha o eje principal que impulsa la transmisión, o en su defecto, en alguna de las flechas propulsoras provenientes de la misma transmisión. Esto implicaría colocar dicho transductor en dos puntos posibles.

1. Entre el reductor de velocidad y la caja de transmisión, exactamente en el punto A señalado en la figura 2. Esto hace necesario desplazar hacia la parte trasera de la mesa de trabajo el motor y el reductor de velocidad o desplazar hacia la parte delantera la caja de transmisión y los rodillos, con todo y la base metálica que los sostiene; todo esto con la finalidad de generar un espacio suficiente en el punto A para colocar el transductor.
2. Entre alguno de los rodillos y la caja de transmisión, punto B señalado en la figura 2. Esto debido a que la fuerza en la entrada de la caja de transmisión es la misma que a la salida de la misma, solamente que duplicada en dos tomas de fuerza que se unen a los rodillos. Para llevar a cabo esto, es necesario cortar alguna de las flechas propulsoras que se encargan de unir la caja de transmisión con los rodillos, generando con dicho corte el espacio suficiente para colocar el transductor.

Ambas opciones son sumamente difíciles de llevar a cabo; por ejemplo, la primera implica modificar casi la mitad de las perforaciones que sirven para sujetar los componentes de la laminadora a la mesa de trabajo, como consecuencia de esto, se hace necesario volver a alinear perfectamente dichos componentes, además existe el riesgo de que el espacio generado en el punto A no sea suficiente, ya que la mesa de trabajo cuenta

con espacios muy reducidos en las partes trasera y delantera. En lo que respecta a la opción dos, cortar alguna de las flechas de propulsión de los rodillos puede ocasionar problemas que pueden ir desde el mismo proceso que se lleve a cabo para cortarla y hacerle las modificaciones pertinentes para colocar el transductor, hasta fallas en el funcionamiento de dicho transductor, debido a la gran cantidad de grasa que siempre existe en este lugar, ésto sin mencionar las posibles pérdidas de fuerza que se pueden originar debido al debilitamiento de la flecha propulsora cortada.

Debido a todo el análisis anterior, descartamos la posibilidad de colocar un transductor y/o sistema de medición capaz de trabajar directamente con la variable par de laminación, procediendo mejor a investigar si era posible calcular el valor del par de laminación empleando algunas otras variables, aunque no estuvieran contempladas por el Sistema de Adquisición y Despliegue de Información, pero cuyos transductores fueran más fáciles de adquirir y colocar.

Con la ayuda de la División de Ingeniería Mecánica se pudo averiguar que conociendo el valor de algunos parámetros eléctricos, propios del motor de DC, y la velocidad angular de los rodillos, es posible llegar a conocer el valor del par de laminación durante todo el proceso. A continuación se explica de qué manera se pueden llegar a obtener los valores de dichos parámetros y, como consecuencia, el par de laminación.

Primero es necesario conocer el valor de la potencia consumida por el motor, ésto se hace con la siguiente expresión:

$$W_M = V_A I_A$$

donde:

V_A es el voltaje de armadura, e I_A es la corriente de armadura.

Posteriormente se tiene que la potencia consumida en la laminación será igual a la consumida por el motor, menos las pérdidas tanto eléctricas como mecánicas, quedando entonces:

$$W_L = W_M - W_V$$

donde:

W_V es la potencia consumida por la marcha en vacío de la laminadora.

Finalmente tenemos que el par de laminación se calcula como:

$$T_L = W_L / \omega$$

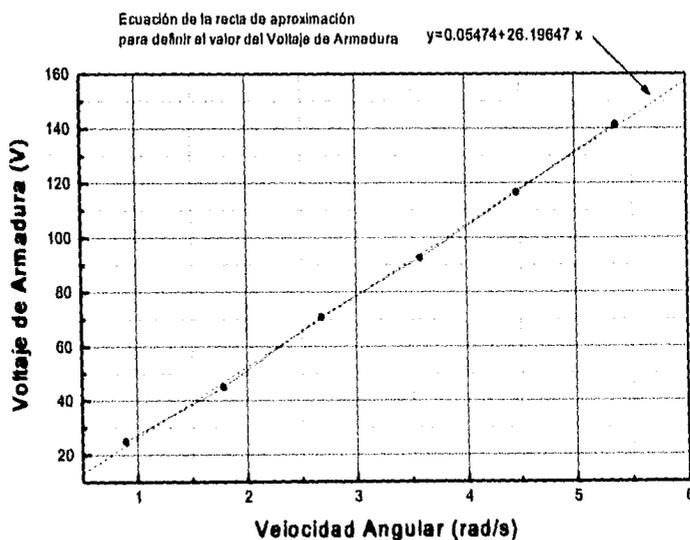
donde ω es la velocidad angular de los rodillos de laminación.

Analizando las variables anteriormente mencionadas, así como el comportamiento de las mismas durante el proceso de laminación, y con ayuda de los ingenieros de la División de Ingeniería Mecánica pudimos llegar a las siguientes conclusiones:

- El voltaje de armadura solamente varía con respecto a la velocidad periférica de los rodillos, leída directamente en el tablero de control de la laminadora; en consecuencia dicho voltaje varía con respecto a la velocidad angular, como se muestra a continuación en la tabla y gráfica 1. Cabe mencionar que la velocidad angular en este caso se calculó con base a la velocidad periférica de los rodillos pero la idea es que el Sistema de Adquisición y Despliegue de Información calcule directamente, con ayuda del correspondiente transductor y circuito acondicionador, el valor de dicha velocidad angular.

Velocidad Periférica (ft / min)	Velocidad Angular (rad/s)	Voltaje de Armadura (V)
10	0.8935	25
20	1.7871	45
30	2.6807	71
40	3.5743	92.7
50	4.4678	116.7
60	5.3614	141.5

Tabla 1. Variación del voltaje de armadura con respecto a la velocidad angular.



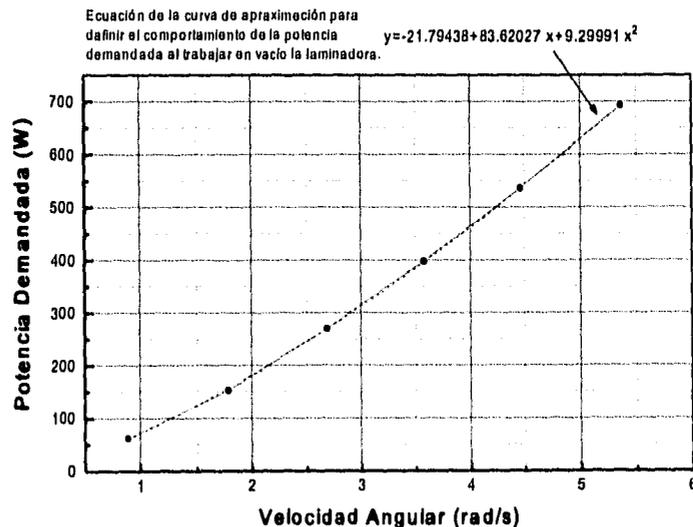
Gráfica 1. Comportamiento del voltaje de armadura con respecto a la velocidad angular.

Por lo tanto, con ayuda de la ecuación que define la gráfica 1, podemos llegar a conocer el valor del voltaje de armadura, solamente conociendo el valor de la velocidad angular.

- El valor de la corriente de armadura está en función de la velocidad angular y de la potencia que sea demandada al motor cuando la laminadora se encuentra trabajando en vacío o laminando algún material. Al trabajar en vacío, se pudo notar que la corriente sólo varía en función de la velocidad angular de los rodillos de laminación, cosa que no sucede cuando se lamina algún material, ya que dependiendo de la carga en los rodillos y de la dureza del material a laminar, aumentará y variará considerablemente el valor de la potencia demandada al motor y, en consecuencia, el de la corriente también. Tomando mediciones de la corriente de armadura para los mismos valores de velocidad periférica que en los voltajes de armadura y trabajando con la laminadora en vacío, podemos obtener el valor de la potencia de vacío en función de la velocidad periférica y, como consecuencia, también en función de la velocidad angular.

Velocidad Periférica (ft/min)	Velocidad Angular (rad/s)	Voltaje de Armadura (v)	Corriente de Armadura (A)	Potencia Demandada (W)
10	0.8973	25	2.5	62.5
20	1.7871	45	3.4	153
30	2.6807	71	3.8	269.8
40	3.5743	92.7	4.3	398.61
50	4.4678	116.7	4.6	436.82
60	5.3614	141.5	4.9	693.35

Tabla 2. Variación de la potencia demandada con respecto a la vel. angular al trabajar la laminadora en vacío.



Gráfica 2. Comportamiento de la potencia demandada con respecto a la velocidad angular.

Al igual que con el voltaje de armadura, nos podemos valer de la ecuación de segundo grado que define la curva de la gráfica 2 para llegar a conocer el valor de la potencia de vacío, para lo cual sólo es necesario saber el valor de la velocidad angular

- Por lo tanto, tenemos que con base en la velocidad angular y en las ecuaciones obtenidas en las gráficas 1 y 2 podemos llegar a obtener el valor de los parámetros eléctricos *voltaje de armadura* y *potencia consumida por la marcha en vacío*, faltando sólo por conocer el valor de la *corriente de armadura*, el cual nos ayudará a obtener finalmente el valor de la *potencia consumida en la laminación*, y como consecuencia, el valor del *par de laminación*. Debido a que los valores de la corriente de armadura varían bastante dependiendo de la velocidad angular, de que la laminadora se encuentre trabajando en vacío o laminando, de la carga ejercida sobre los rodillos y de la dureza del material a laminar, se hace difícil en extremo definir con alguna ecuación el comportamiento de dicha corriente de armadura, por lo cual lo más factible es estar siempre adquiriendo datos relativos a esta variable que nos ayuden a obtener el valor real del par de laminación.

Analizando los puntos anteriores, podemos afirmar que es necesario integrar al Sistema de Adquisición y Despliegue de Información la variable conocida como *corriente de armadura*, la cual conjuntamente con la velocidad angular, nos ayudará a calcular el valor de la variable par de laminación sin necesidad de colocar un transductor que trabaje directamente con dicha variable.

Para ésto necesitamos seleccionar y/o diseñar un transductor que trabaje directamente con la variable corriente eléctrica y que cumpla, principalmente, con las siguientes características.

- Un intervalo de medición que abarque desde 0.5 y hasta 60 amperes (como mínimo).
- De fácil colocación en la línea de alimentación principal del motor de DC.
- De preferencia nos entregue una señal de voltaje proporcional a la corriente.
- Tenga una considerable inmunidad al ruido.

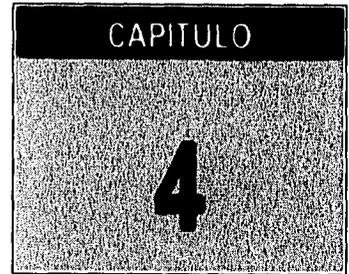
Considerando todo lo anterior se decidió colocar un amperímetro de gancho capaz de medir valores de corriente de hasta 600 amperes, el cual solamente se debe "engancharse" en uno de los conductores (ya sea en el que representa al polo negativo o al positivo) de la línea principal de alimentación del motor de DC, esto último es importante, ya que de colocarse en donde existen dos o más conductores la lectura en el amperímetro siempre será cero. Este instrumento nos despliega digitalmente en una pantalla de cuarzo el valor de corriente medido directamente en la línea de alimentación del motor de DC, por lo cual puede en un

momento dado sustituir a los amperímetros analógicos colocados tanto en el tablero de control de la laminadora como en el alimentador principal de la misma.

Aunado a lo anterior tenemos que el funcionamiento de este instrumento se basa en el principio de inducción eléctrica de los transformadores. Tenemos que la corriente eléctrica que circula a través de un conductor (devanado primario del transformador) produce un campo eléctrico, el cual es inducido al gancho con núcleo de hierro del amperímetro (devanado secundario del transformador), lo cual produce en la salida de dicho núcleo una diferencia de potencial de AC de algunos milivolts, proporcional a la corriente que circula por el conductor en donde se encuentra "enganchado" el amperímetro. Si esta diferencia de potencial de AC se rectifica y amplifica adecuadamente nos debe dar como resultado una señal de voltaje de DC compatible con el Sistema de Adquisición y Despliegue de Información y directamente proporcional a la corriente de armadura que circula por los conductores de alimentación principal del motor de DC. Además de lo anterior tenemos la ventaja de que podemos obtener la diferencia de potencial de AC del núcleo de fierro del gancho sin alterar el funcionamiento del amperímetro al momento de desplegar en la pantalla de cuarzo el valor de la corriente de armadura.

Finalmente concluimos que para efectuar la medición de la variable par de laminación necesitamos, además de la velocidad angular, un transductor representado por el amperímetro de gancho, para medir el valor de la corriente de armadura.

DISEÑO DE LOS ACONDICIONADORES DE SEÑALES



Una vez que son seleccionados los transductores eléctricos más adecuados para cada una de las variables físicas a trabajar (temperatura, carga, velocidad y par de laminación), se hace necesario modificar las señales eléctricas que nos entregan dichos transductores, con la finalidad de hacer que las señales sean compatibles con el convertidor Analógico/Digital (A/D) del microcontrolador de la arquitectura a desarrollar. Mencionaremos que aunque aún no se tenga seleccionado un microcontrolador específico para la arquitectura del sistema, se tiene en mente utilizar alguno de la familia M68HC11 de Motorola, los cuales manejan en la entrada de su convertidor A/D integrado un intervalo de voltajes que va de 0 a 5 volts. Esta característica no se debe olvidar al momento de diseñar los acondicionadores de señales.

Al proceso de modificar las señales eléctricas se le conoce como acondicionar y a los elementos que llevan a cabo el proceso se les denomina acondicionadores. Al decir acondicionar nos podemos referir, por ejemplo, a amplificar, atenuar, filtrar o linealizar una señal eléctrica con la finalidad de adecuarla a nuestras necesidades; para ésto los acondicionadores de señales se diseñan, por lo general, con base en componentes electrónicos de tipo analógico, como pueden ser amplificadores operacionales, transistores, diodos, reguladores, etc.

A continuación se hace un análisis del diseño de cada uno de los acondicionadores que se implementarán en el Sistema de Adquisición y Despliegue de Información para cada una de las variables.

4.1. ACONDICIONADOR DE SEÑAL PARA LA VARIABLE TEMPERATURA.

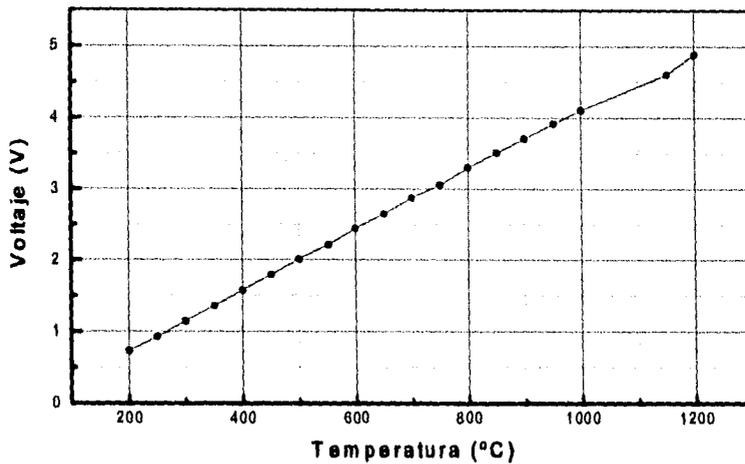
Para esta variable tenemos la ventaja de que el sistema de medición adquirido con anterioridad, cuenta con una salida tipo analógica, la cual nos proporciona un voltaje directamente proporcional a la temperatura del material a laminar. El voltaje de salida varía desde 0.73 V hasta 4.89 V aproximadamente (como se verá más adelante), para un intervalo de temperaturas que va de 200 a 1200 ° C (siendo esta última la máxima temperatura medible por el sensor); este voltaje de salida es compatible con el convertidor A/D, pero con la

inconveniencia de que dicho voltaje al ser la temperatura menor a 200 ° C se incrementa hasta 6 volts, lo cual dañaría el convertidor de nuestro microcontrolador.

Así tenemos que para proceder a realizar el diseño de este acondicionador, primero se realizaron las mediciones de voltaje en la salida analógica del sistema, ésto a diversas temperaturas y con distintos coeficientes de emisividad. Este coeficiente de emisividad depende del material al cual se le pretenda medir la temperatura, ésto debido a que el sensor no tiene contacto con el material y la emisión de las radiaciones de calor y la captación de las mismas por el pirómetro de radiación no es la misma para todos los materiales. A continuación se muestra la tabla de valores obtenida, así como la gráfica correspondiente.

Temperatura (° C)	Emisividad				Promedio (V)
	0.95	1	0.5	0.3	
200	0.739	0.746	0.73	0.72	0.73375
250	0.942	0.936	0.935	0.918	0.93275
300	1.155	1.148	1.144	1.134	1.14525
350	1.366	1.365	1.36	1.35	1.36025
400	1.589	1.578	1.574	1.564	1.57625
450	1.813	1.799	1.795	1.782	1.79725
500	2.028	2.016	2.009	2.003	2.014
550	2.22	2.215		2.216	2.217
600			2.453	2.441	2.447
650			2.655	2.65	2.6525
700			2.881	2.872	2.8765
750			3.056	3.063	3.0595
800			3.31	3.301	3.3055
850			3.51		3.51
900			3.71		3.71
950			3.92		3.92
1000			4.11		4.11
1150			4.61		4.61
1200			4.89		4.89

Tabla 1. Valores de voltaje proporcionados por el transductor de la variable temperatura



Gráfica 1. Comportamiento del transductor para la variable temperatura

Como puede observarse en la gráfica, el comportamiento de la señal eléctrica de voltaje proporcionada por el transductor es netamente lineal y se encuentra en el intervalo de valores que maneja el convertidor A/D del microcontrolador, lo cual la hace compatible con el sistema. Solamente existen dos problemas a resolver, el primero se refiere al ajuste de estos valores con respecto a cero, ya que en 200°C (el valor mínimo que puede ser medido) el voltaje es de 0.73375 V , debiendo ser cero; el segundo se ha mencionado con anterioridad y se refiere a que cuando la temperatura a medir es menor a 200°C el voltaje que se obtiene a la salida es de aproximadamente 6 volts, debiendo ser cero también, ya que el sensor se encuentra fuera del intervalo de medición.

Para solucionar el problema del ajuste podemos colocar un amplificador operacional implementado como restador de voltaje, con los componentes y función de transferencia que se muestran a continuación.

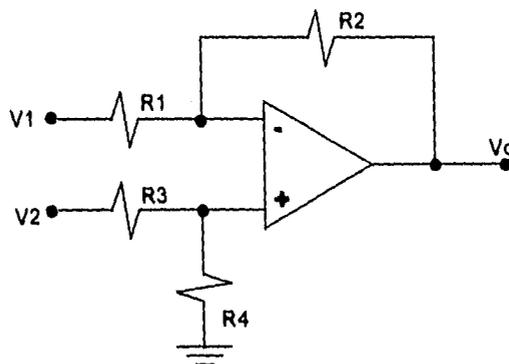


Figura 1. Circuito restador de voltaje analógico.

$$\text{Si } V_1 \neq 0 \text{ y } V_2 = 0, \text{ entonces } V_o = -\frac{R_2}{R_1} V_1$$

Luego, si $V_1 = 0$ y $V_2 \neq 0$, entonces

$$V_o = \left[\frac{R_4}{R_3 + R_4} + \frac{R_2}{R_1} \frac{R_4}{R_3 + R_4} \right] V_2$$

Por lo tanto, nuestra función de transferencia total es

$$V_o = -\frac{R_2}{R_1} V_1 + \left[\frac{R_4}{R_3 + R_4} + \frac{R_2}{R_1} \frac{R_4}{R_3 + R_4} \right] V_2$$

Si hacemos $R_1 = R_2 = R_3 = R_4$, tenemos que:

$$V_o = -V_1 + V_2 \text{ ó } V_o = V_2 - V_1$$

Análisis 1. Función de Transferencia para el circuito Restador de Voltaje.

Por lo tanto, si tomamos V_2 como la entrada de la señal proveniente del sistema de medición para la variable temperatura y en V_1 fijamos un voltaje aproximadamente igual a 0.73375 (voltaje promedio medido a 200° C), a la salida del restador obtendremos $V_2 - V_1$, o mejor dicho los valores de voltaje ajustados al cero establecido en 200 ° C.

Como se observa en la figura 1, se consideró una ganancia unitaria al restador y como consecuencia, los valores de todas las resistencias (R_1 , R_2 , R_3 y R_4) se fijaron en 10 K Ω . Por otra parte para obtener el voltaje de ajuste (0.73375) para la entrada V_1 se propone colocar un diodo y una resistencia variable en un arreglo como el de la figura 2.

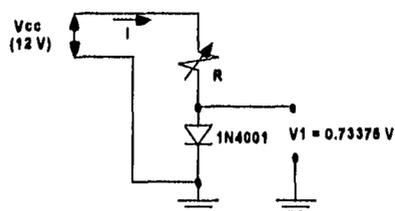


Figura 2. Arreglo para obtener un voltaje de ajuste de 0.73375 V.

Partiendo de la suma de voltajes para el circuito anterior tenemos que:

$$V_{cc} = V_R + V_D \quad \text{entonces:}$$

$$V_R = V_{cc} - V_D$$

Sustituyendo la expresión que define a V_R podemos despejar R quedando:

$$R = \frac{V_{cc} - V_D}{I}$$

Considerando una corriente I de 75 mA, por especificaciones del fabricante, y sustituyendo valores, tenemos que:

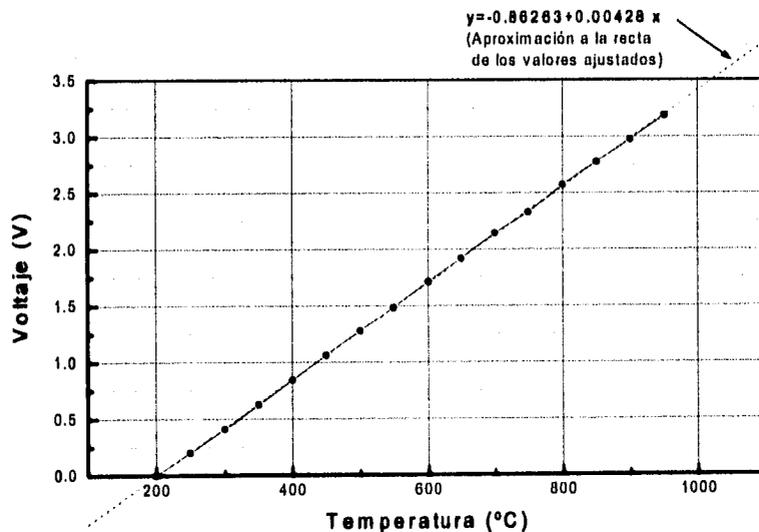
$$R = 150 \text{ ohms (Aprox.)}$$

Análisis 2. Cálculos para encontrar el valor de R en el circuito de la figura 2.

La tabla y gráfica que se muestran a continuación corresponden a los valores de voltaje promediados ya ajustados con ayuda del restador. En la gráfica podemos ver con una línea continua la representación correspondiente a los valores de voltaje ajustados, mientras que con línea punteada podemos apreciar la aproximación hecha a una línea recta de dichos valores. Como complemento se muestra en la parte superior de la gráfica la ecuación de la recta de aproximación, la cual se obtuvo realizando una regresión lineal.

TEMPERATURA (°C)	VOLTAJE AJUSTADO	TEMPERATURA (°C)	VOLTAJE AJUSTADO
200	0	600	1.714
250	0.19975	650	1.9195
300	0.41225	700	2.1435
350	0.62725	750	2.3265
400	0.84325	800	2.5725
450	1.06425	850	2.777
550	1.281	900	2.977
550	1.484	950	3.187

Tabla 2. Valores de Voltaje ajustados a cero con ayuda del restador.



Gráfica 2. Representación gráfica de los valores ajustados a cero y aproximación a una línea recta de dichos valores

De ahora en adelante, el motivo de hacer aproximaciones de las gráficas de comportamiento de los transductores a una línea recta, obedecerá a que la ecuación de dicha aproximación nos ayudará más tarde

a interpretar y a procesar los valores digitales que llegarán a la computadora, los cuales no son más que una representación particular de la información analógica obtenida por los transductores de la laminadora.

Para solucionar el problema del voltaje que se eleva a niveles superiores de los aceptados por el convertidor A/D del microcontrolador, necesitamos un elemento capaz de proporcionar una cierta señal cuando el voltaje suministrado por la salida analógica del sistema para medir temperatura, sea mayor al tolerado por el convertidor A/D y, de preferencia, una señal de otro tipo cuando el voltaje se encuentra en el intervalo de valores permisibles por dicho convertidor. Esta especie de señal de control debe ser capaz de habilitar o deshabilitar otro dispositivo que se encargue de interrumpir el paso del voltaje cuando la señal del comparador así lo indique.

El primer elemento puede ser un comparador de niveles de voltaje con histéresis, el cual basa su funcionamiento en un amplificador operacional. Este comparador lo que hace es analizar una señal de voltaje a la entrada respecto a un voltaje medio (derivado de un voltaje de referencia), de tal forma que cuando dicho voltaje de entrada sobrepasa al medio, en la salida del comparador tenemos el voltaje de saturación (ya sea negativo o positivo) del amplificador operacional utilizado, por otra parte cuando dicho voltaje es menor al medio, se tiene a la salida del comparador el voltaje de saturación contrario al anterior. La histéresis se refiere a la propiedad que el circuito exhibe cuando cambia de un primer estado a un segundo estado con una cierta señal y entonces regresa del segundo al primer estado con otra señal de entrada diferente, impidiendo falsos cambios de estado en la salida del comparador debido al ruido que acompaña a la señal de entrada.

En lo que se refiere al dispositivo que interrumpa el paso del voltaje cuando una señal del comparador así lo indique, lo más factible a utilizar es un interruptor electrónico compatible con los voltajes de saturación que nos proporciona el comparador. Tomando en cuenta esto, se decidió utilizar un interruptor que utiliza tecnología CMOS en su fabricación y que al recibir una señal con bajo nivel de voltaje se comporta como un interruptor abierto y con señales de alto nivel de voltaje se comporta como un interruptor cerrado. Se entiendo por bajo nivel de voltaje de 0 a 0.5 volts y por nivel alto, el voltaje de polarización del circuito, el cual se encuentra en el intervalo de 3 a 15 volts. Dicho interruptor electrónico es el CD4016.

Partiendo del funcionamiento del interruptor electrónico CD4016 tenemos que nuestro comparador debe proporcionarnos un voltaje de saturación positivo (que mantenga el interruptor cerrado) siempre y cuando el voltaje a la entrada del comparador sea menor o igual al voltaje medio; dicho voltaje medio lo fijará el voltaje máximo que puede aceptar el convertidor A/D del microcontrolador, o sea 5 volts, con una histéresis de 1 volt (0.5 hacia arriba y 0.5 hacia abajo de 5 volts) para prevenir que el ruido de la señal de entrada provoque un

cambio de estado no deseado en el comparador. Por otra parte cuando el voltaje de entrada sea mayor al voltaje medio, el comparador nos proporcionará a la salida el voltaje de saturación negativo, el cual será utilizado para abrir el interruptor electrónico, como se verá más adelante, impidiendo el paso de voltajes superiores a 5 volts. A continuación se muestran la configuración del circuito comparador de voltaje, la gráfica de comportamiento y el análisis de dicho comparador.

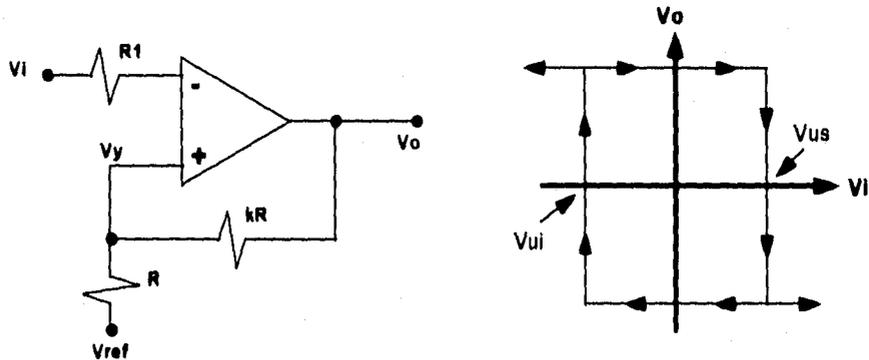


Figura 3. Circuito y gráfica de comportamiento para el Comparador con histéresis.

Si primero analizamos el voltaje en el punto V_y tenemos

$$V_y = \frac{R}{R + kR} V_o + \frac{kR}{R + kR} V_{ref}$$

Ahora, si suponemos que $V_o = V_{sat(-)}$ para un $V_i > V_y$

Entonces necesitamos para que conmute un $V_i < V_y$, por lo que sustituyendo condiciones y simplificando la ecuación anterior, tenemos:

$$V_i < \frac{1}{1+k} V_{sat(-)} + \frac{k}{k+1} V_{ref}$$

De esta ecuación obtenemos los voltajes de umbral V_{us} y V_{ui}

$$V_{ui} = \frac{1}{1+k} V_{sat(-)} + \frac{k}{k+1} V_{ref}; \quad V_{us} = \frac{1}{1+k} V_{sat(+)} + \frac{k}{k+1} V_{ref}$$

Si ahora calculamos el voltaje de histéresis tenemos que:

$$V_H = V_{us} - V_{ui} = \frac{1}{1+k} [V_{sat(+)} - V_{sat(-)}]$$

Además si consideramos $V_{sat(+)} = V_{cc}$ y $V_{sat(-)} = -V_{cc}$, tenemos:

$$V_H = \frac{2 V_{cc}}{1+k}, \text{ por último obtenemos el voltaje medio como:}$$

$$V_m = \frac{V_{us} + V_{ui}}{2} = \frac{k}{k+1} V_{ref}$$

Análisis 3. Ecuaciones para obtener los parámetros y el valor de los componentes del comparador.

Si para nuestro caso consideramos un voltaje medio (V_m) de 5 V, un voltaje de histéresis (V_H) de 1 V, el voltaje de polarización (V_{cc}) igual a 12 V, y fijamos R_1 y R en 10 k Ω ; tenemos que el valor de k en la resistencia de realimentación del circuito será igual a 23, por lo que dicha resistencia debe ser de 230 k Ω ; a su vez, este valor de k nos ayuda a obtener el voltaje de referencia (V_{ref}), el cual será igual a 5.2173. En la figura 4 se muestra el arreglo usado para obtener un voltaje aproximadamente igual al de referencia, basándonos en la utilización de un diodo zener comercial con un voltaje de 5.1 volts.

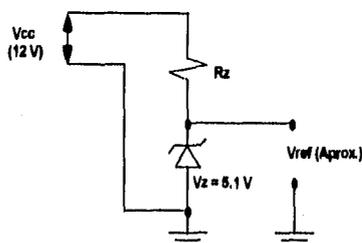


Figura 4. Arreglo para obtener un voltaje de referencia aproximadamente de 5.2173 V.

Partiendo de la suma de voltajes para el circuito tenemos que:

$$V_{cc} = V_{Rz} + V_z \quad \text{entonces:}$$

$$V_{Rz} = V_{cc} - V_z$$

Sustituyendo la expresión que define a V_{Rz} podemos despejar Rz quedando:

$$Rz = \frac{V_{cc} - V_z}{I}$$

Considerando una corriente I de 40mA, por especificaciones, y sustituyendo los demás valores, tenemos que:

$$Rz = 172 \text{ ohms (Aprox.)}$$

Análisis 4. Cálculos para encontrar el valor de Rz en el circuito de la figura 4.

Finalmente, haciendo uso de todos los elementos y arreglos anteriormente descritos, tenemos que nuestro circuito acondicionador de señales para la variable temperatura, queda integrado como se muestra a continuación en la figura 5. Cabe mencionar que este circuito consta a la entrada de un elemento, hasta ahora no mencionado, denominado acoplador, el cual no es más que un amplificador operacional realimentado directamente en la terminal inversora; su función principal es acoplar el circuito acondicionador con el sistema de medición para la variable temperatura, la ventaja es que esta función la realiza sin invertir la polaridad de la señal de entrada. Por otra parte, también podemos observar otro elemento que no se había mencionado con anterioridad; se trata de un diodo colocado a la salida del comparador de voltaje, el cual

tiene como función principal entregar un nivel bajo de voltaje que se encargue de abrir el interruptor electrónico cuando exista un voltaje de saturación negativo en el comparador.

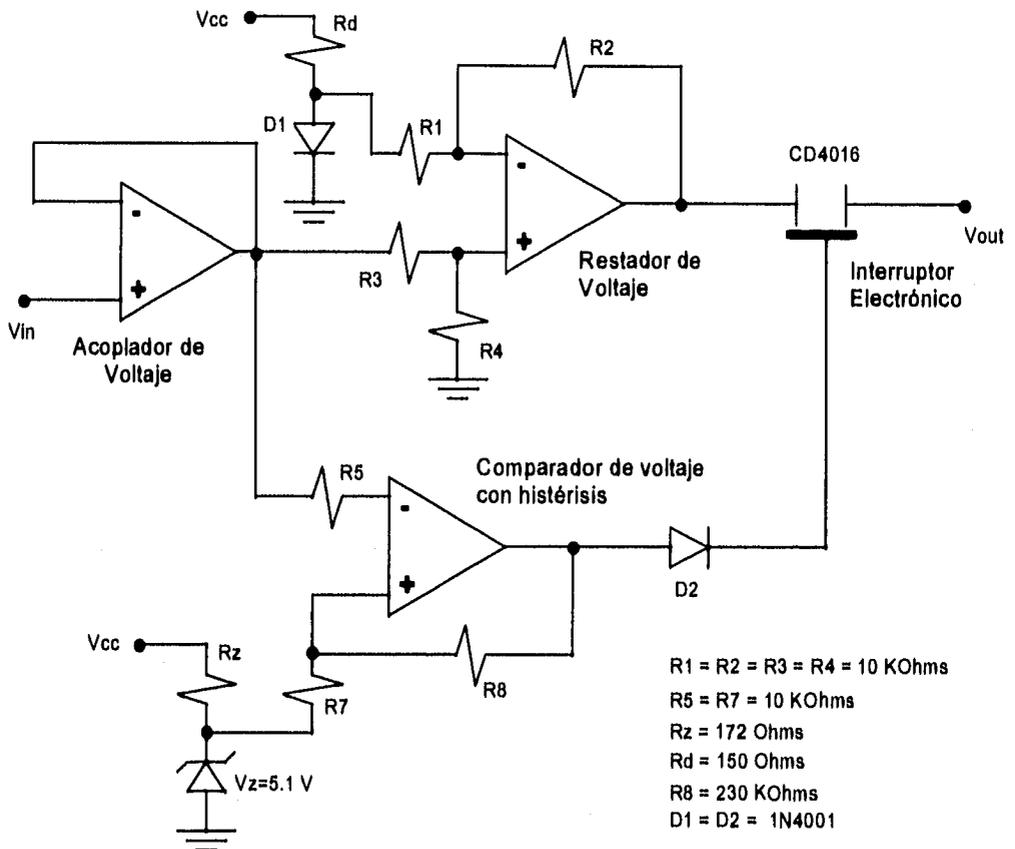


Figura 5. Circuito Acondicionador de Señal para la variable Temperatura.

4.2 ACONDICIONADOR DE SEÑAL PARA LA VARIABLE CARGA.

En el caso de ésta variable, también contamos con un sistema de medición completo, el cual se encuentra integrado, como se mencionó en el capítulo anterior, de 4 celdas de carga, una unidad sumadora y promediadora de carga y una unidad encargada del despliegue de información en forma digital.

Después de analizar diversos lugares para efectuar mediciones de voltaje, tanto en la unidad sumadora de carga como en la unidad de despliegue de datos, se llegó a la conclusión de que el mejor lugar para llevar a cabo dichas mediciones es en la salida de la unidad sumadora, ya que en este punto contamos con la

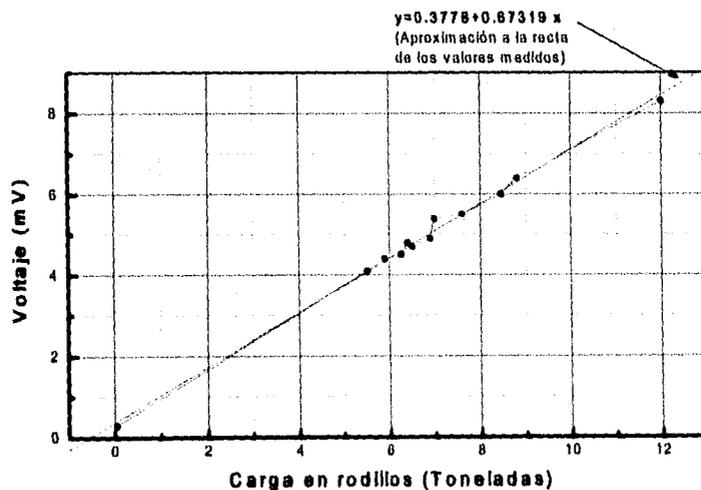
señal sumada y promediada que se transmite directamente hacia la unidad de despliegue de datos; la desventaja de esta señal es que su magnitud solamente es de unos cuantos milivolts.

A continuación se muestran las mediciones de carga efectuadas directamente en la unidad de despliegue de datos digital y su correspondiente valor de voltaje, medido en la unidad sumadora.

Carga ejercida sobre el material a laminar (Toneladas)	Voltaje medido a la salida de la unidad sumadora (mV)
0.043	0.3
5.5	4.1
5.9	4.4
6.26	4.5
6.4	4.8
6.5	4.7
6.9	4.9
7	5.4
7.6	5.5
8.45	6
8.8	6.4
12	8.3

Tabla 3. Valores de voltaje proporcionados por la unidad sumadora al momento de aplicar carga en los rodillos

La representación gráfica y la aproximación a una línea recta de los valores medidos se presenta a continuación.



Gráfica 3. Representación gráfica de los valores medidos en la unidad sumadora.

Analizando las mediciones hechas a esta señal, así como la gráfica que las representa, podemos observar que su comportamiento es netamente lineal pero con la única desventaja de que su magnitud es solamente de unos cuantos milivolts. Tomando ésto en cuenta, podemos suponer que una parte primordial en el acondicionamiento de esta señal, será la amplificación de la misma, ya que su magnitud se encuentra totalmente fuera del intervalo de valores que maneja el convertidor A/D del microcontrolador. Para conocer cuánto debemos amplificar la señal, primero debemos saber cuál es el máximo valor de carga que se puede presentar en los rodillos de laminación y qué voltaje se tendrá en la unidad sumadora con dicha carga.

Utilizando la ecuación de la recta de aproximación y suponiendo la carga máxima que puede soportar la laminadora, la cual es de aproximadamente 30 toneladas, tenemos que el voltaje en la unidad sumadora debe ser aproximadamente de:

$$V = 0.3778 + 0.67319 (30) mV$$

$$V = 20.5735 mV$$

Si este voltaje máximo, medible en la unidad sumadora, se relaciona con el voltaje máximo que permite el convertidor A/D del microcontrolador (5 volts), podemos obtener el valor de la ganancia para el amplificador, de la siguiente forma:

$$\text{Ganancia} = \frac{V_o}{V_i} = \frac{5 V}{20.5735 mV} = 243.03$$

Como amplificador se utilizará la configuración conocida como amplificador inversor, la cual también se fundamenta en la utilización de amplificadores operacionales. Dentro de las características más importantes de este amplificador tenemos que posee una alta impedancia de entrada (para que no desaparezca la señal), una impedancia de salida casi igual a cero (para acoplar más etapas), puede amplificar señales de AC o DC sin distorsionarla, amplio rango de frecuencias útil, además sus características no varían significativamente con la temperatura, frecuencia o tipo de carga. A continuación se muestra el circuito que representa al amplificador inversor, así como el análisis encaminado a obtener su función de transferencia.

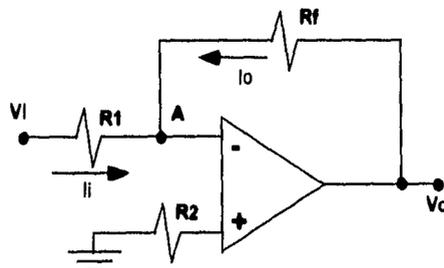


Figura 6. Circuito Amplificador Inversor.

Si realizamos suma de corrientes en el punto A tenemos que: $i_o + i_i = 0$

Si ahora definimos i_o con la siguiente ecuación:

$$i_o = \frac{V_o - V(-)}{R_f}$$

Por tierra virtual tenemos que $V(-) = 0$, entonces:

$$i_o = \frac{V_o}{R_f}$$

Ahora tenemos que i_i se define como: $i_i = \frac{V_i}{R_i}$

Si sustituimos i_o e i_i en la ecuación inicial, tenemos:

$$\frac{V_o}{R_f} + \frac{V_i}{R_i} = 0$$

Reacomodando, se tiene la función de transferencia que define el comportamiento y ganancia del amplificador inversor.

$$V_o = - \frac{R_f}{R_i} V_i$$

Por último R_2 se coloca para minimizar los desajustes debidos a la corriente de polarización y es igual a la resistencia paralela de R_1 y R_f .

Análisis 5. Función de transferencia para el circuito Amplificador Inversor.

Cuando en este tipo de amplificadores se trabaja con ganancias grandes (mayores a 100 por ejemplo), se recomienda que la amplificación se haga en etapas conectadas en cascada, en donde la ganancia total es igual al producto de las ganancias de cada una de las etapas. Tomando en cuenta lo anterior, así como la ganancia calculada para el amplificador, podemos llegar a obtener esta última en dos etapas, la primera con una ganancia de 15 y la segunda de 16. El producto de estas dos etapas nos dará como resultado una ganancia total de 240, lo cual es un valor bastante aproximado al requerido de 243.03. En la figura 7 se muestra como quedaría configurado el amplificador en dos etapas.

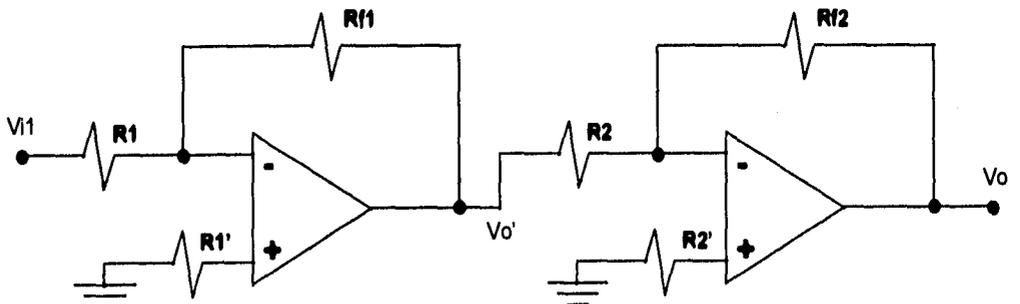


Figura 7. Circuito Amplificador de dos etapas para una ganancia de 240.

Para la primera etapa tenemos los siguientes cálculos:

$$-\frac{V_{o'}}{V_i} = \frac{R_{f1}}{R_1} ; \text{Si } -\frac{V_{o'}}{V_i} = 15, \text{ entonces } R_{f1} = 15 R_1$$

Si fijamos $R_1 = 10 \text{ Kohms}$, se tiene que $R_{f1} = 150 \text{ Kohms}$

Finalmente, si R_1' es igual a $R_1 \parallel R_{f1}$, entonces $R_1' = 9375 \text{ ohms}$

Ahora para la segunda etapa tenemos que:

$$-\frac{V_o}{V_{o'}} = \frac{R_{f2}}{R_2} ; \text{Si } -\frac{V_o}{V_{o'}} = 16, \text{ entonces } R_{f2} = 16 R_2$$

Si fijamos $R_2 = 10 \text{ Kohms}$, se tiene que $R_{f2} = 160 \text{ Kohms}$

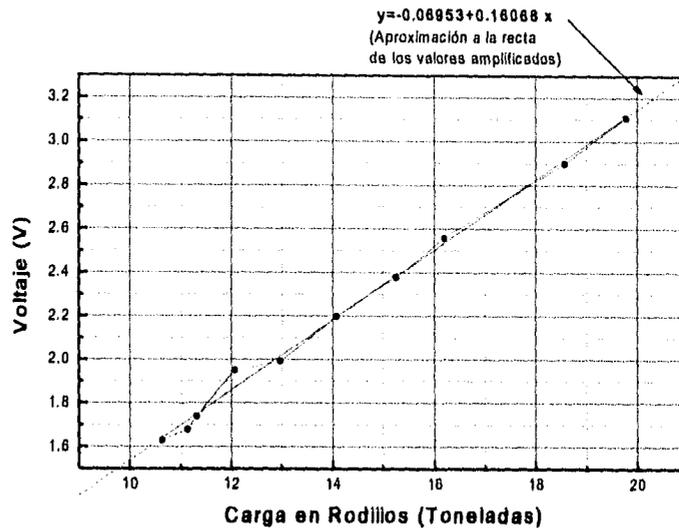
Finalmente, si R_2' es igual a $R_2 \parallel R_{f2}$, entonces $R_2' = 9411 \text{ ohms}$

Análisis 6. Cálculos para encontrar el valor de los componentes del circuito amplificador de la figura 7.

Una vez implementado el amplificador, éste se conecta a la salida de la unidad sumadora, obteniéndose los valores de voltaje amplificados que se muestran en la tabla 4, esto para diversos valores de carga ejercida sobre los rodillos.

Carga ejercida sobre el material a laminar (Toneladas)	Voltaje medido a la salida del amplif. conectado a la unidad sumadora (V)
10.641	1.63
11.12	1.68
11.297	1.74
12.058	1.95
12.971	1.993
14.072	2.20
15.252	2.38
16.200	2.56
18.571	2.90
19.774	3.11

Tabla 4. Valores de voltaje medidos en el amplificador conectado en la unidad sumadora al momento de aplicar carga en los rodillos



Gráfica 4. Representación gráfica de los voltajes amplificados de la unidad sumadora.

Como puede observarse en la gráfica 4, el comportamiento de los voltajes amplificados sigue siendo lineal, pero con la gran ventaja de que ahora su magnitud se encuentra en el intervalo de valores de entrada que maneja el convertidor A/D del microcontrolador. Por lo tanto podemos concluir que para acondicionar la señal obtenida de la unidad sumadora de cargas solamente necesitaremos del amplificador de dos etapas de la figura 7.

4.3 ACONDICIONADOR DE SEÑAL PARA LA VARIABLE VELOCIDAD.

Como se mencionó en el capítulo anterior, el transductor a utilizar para medir la variable velocidad angular, es el sensor de efecto Hall TL170C, de tipo interruptor, el cual tiene la ventaja de poderse polarizar con 5 volts. Este transductor al detectar un cierto campo magnético nos proporciona en su salida el voltaje de polarización del mismo, mientras que al detectar el campo magnético contrario, tenemos cero volts. Debido a que los voltajes de polarización del transductor y del microcontrolador son exactamente los mismos, existe una compatibilidad del 100 % entre la señal que nos proporciona el transductor, al momento de detectar los campos magnéticos, y el intervalo de valores de voltaje que maneja a la entrada el contador de eventos del microcontrolador. Por lo tanto podemos concluir que si polarizamos correctamente el sensor TL170C, no tendremos la necesidad de emplear acondicionador de señales alguno; siendo el único arreglo ajeno al sensor, el filtro que el fabricante recomienda colocar en la terminal de salida, esto ayudará a evitar distorsiones que puedan llegar a afectar el comportamiento de la señal.

fundamentados en la utilización de amplificadores operacionales y diodos. Como en nuestro caso buscamos convertir una señal de AC con una amplitud y frecuencia muy reducidas, en su equivalente de DC, nos conviene utilizar un circuito denominado *convertidor de AC a DC*, el cual se encuentra integrado básicamente por un rectificador de precisión y un capacitor de alto valor, el cual servirá como filtro para la señal rectificada. Este circuito básicamente lo que hace es calcular el valor promedio del voltaje de AC rectificado. Un circuito rectificador de onda completa también se conoce como circuito de valor absoluto y puesto que un valor promedio se puede denominar valor medio, el convertidor de AC a DC también se conoce como *circuito de valor medio absoluto*.

El circuito que representa a dicho convertidor de AC a DC o circuito de valor medio absoluto se muestra en la figura 9, inmediatamente después se hace un breve análisis sobre su funcionamiento.

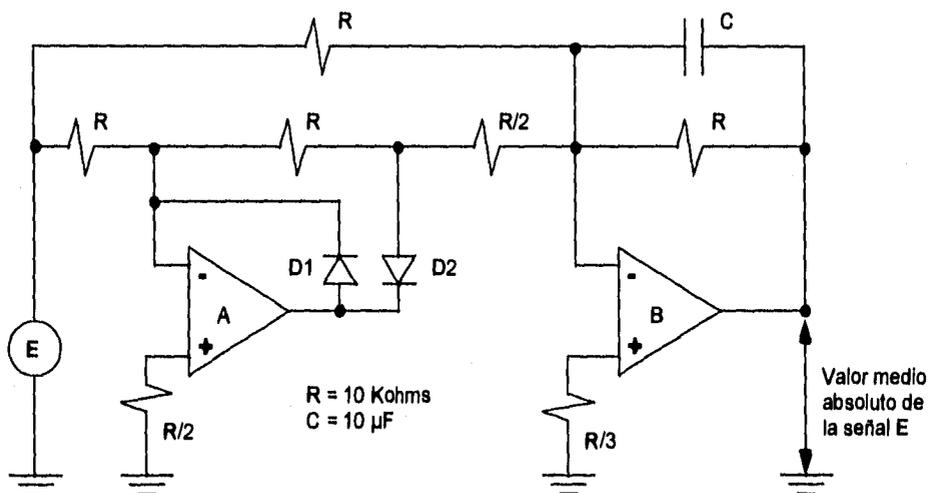


Figura 9. Circuito convertidor de AC a DC.

Para **entradas positivas** de una señal de AC (denominada E), el circuito de la figura 9 funciona de la siguiente manera: El amplificador operacional A se comporta como un amplificador inversor de ganancia unitaria, esto último debido a que las resistencias de entrada y realimentación son iguales, por lo tanto a la salida de dicho amplificador tenemos la señal E pero con la polaridad invertida ($-E$). Posteriormente el amplificador operacional B, el cual se comporta como un inversor sumador, suma la salida del amplificador A con la señal E, obteniéndose una salida $V_o = E$:

Para el sumador inversor tenemos que:

$$V_o = - \left[\frac{E}{R/2} + \frac{E}{R} \right] R_f$$

Si $R = R_f$, entonces: $V_o = 2E - E$

Por lo tanto $V_o = E$

Análisis 7. Salida del sumador inversor para entradas positivas

Ahora, para **entradas negativas** de la misma señal de AC, tenemos que en el amplificador operacional A no se permite el paso de la señal $-E$ hacia la resistencia de entrada $R/2$ del sumador Inversor (ésto se logra con ayuda de los diodos), entonces tenemos que el amplificador operacional B se comporta como un amplificador inversor de ganancia unitaria que nos produce una salida $V_o = -(-E)$, o sea $V_o = E$.

Así tenemos una señal completamente rectificada que por último debe pasar a través del capacitor de alto valor y baja fuga ($10 \mu\text{F}$ de tantalio), el cual nos sirve para obtener el promedio de la salida rectificada del amplificador operacional B; ésto se debe a que toma cerca de 50 ciclos del voltaje de entrada al capacitor cargarse antes de que su voltaje de salida se establezca en una lectura final.

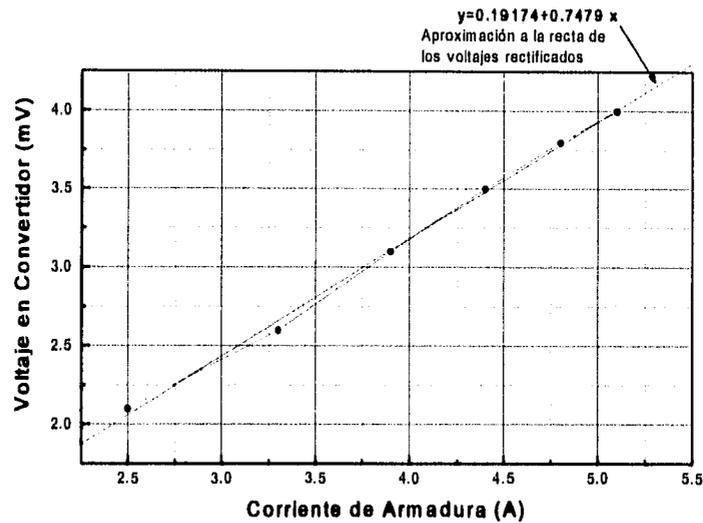
Cabe mencionar que, como en el caso de la variable carga, en este caso también se colocaron resistencias en las terminales no inversoras de los amplificadores operacionales A y B para minimizar los efectos de la corriente de polarización.

Una vez implementado el convertidor de AC a DC, el siguiente paso a seguir es hacer las mediciones de voltaje de DC a la salida del convertidor, ésto para diversos valores de corriente, los cuales a su vez fueron variando en base a la velocidad periférica de los rodillos. Así fue como llegamos a obtener la siguiente tabla de valores.

Velocidad Periférica (Ft / min)	Corriente de Armadura (A)	Voltaje a la salida del convertidor de AC a DC (mV)
10	2.5	2.1
20	3.3	2.6
30	3.9	3.1
40	4.4	3.5
50	4.8	3.8
60	5.1	4.0

Tabla 5. Valores de corriente de armadura medidos en la salida del convertidor de AC a DC.

A continuación se muestra la representación gráfica de estos valores, así como la ecuación de la recta que posteriormente será utilizada para calcular el valor de la ganancia que debe tener el amplificador que complementará la etapa de acondicionamiento para la variable corriente de armadura.



Gráfica 5. Representación gráfica de los valores medidos en la salida del convertidor.

Si ahora utilizamos la ecuación de aproximación a la recta de la gráfica anterior y suponemos una corriente máxima de 40 amperes, tenemos que el voltaje en la salida del convertidor debe ser aproximadamente de:

$$V = 0.19174 + 0.7479 (40) \text{ mV}$$

$$V = 30.1077 \text{ mV}$$

Si relacionamos este voltaje máximo, medible en la salida del convertidor, con el voltaje máximo que permite el convertidor A/D del microcontrolador (5 volts), podemos obtener el valor de la ganancia para el amplificador de la siguiente forma:

$$\text{Ganancia} = \frac{V_o}{V_i} = \frac{5 \text{ V}}{30.1077 \text{ mV}} = 166.07$$

Como se trata de un valor de ganancia grande, se optó por llevar a cabo la amplificación en dos etapas utilizando, como en el caso de la variable carga, la configuración conocida como amplificador inversor. Así tenemos que para obtener una ganancia de 166 podemos colocar una primera etapa con ganancia de 12 y una segunda con ganancia de 14. El producto de estas dos etapas nos dará como resultado una ganancia total de 168, el cual es un valor bastante aproximado al requerido de 166.07. A continuación se muestra como quedaría el circuito para el amplificador en dos etapas.

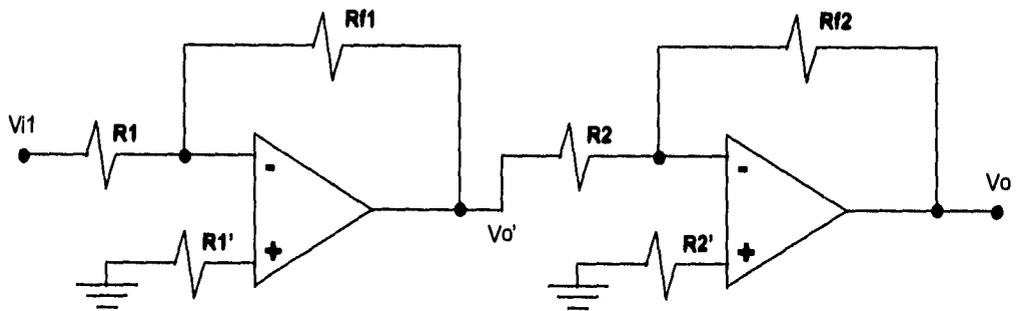


Figura 10. Circuito Amplificador de dos etapas para una ganancia de 168.

Para la primera etapa tenemos los siguientes cálculos:

$$-\frac{V_{o'}}{V_i} = \frac{R_{f1}}{R_1} ; \text{Si } -\frac{V_{o'}}{V_i} = 12, \text{ entonces } R_{f1} = 12 R_1$$

Si fijamos $R_1 = 10 \text{ Kohms}$, se tiene que $R_{f1} = 120 \text{ Kohms}$

Finalmente, si $R_{1'}$ es igual a $R_1 \parallel R_{f1}$, entonces $R_{1'} = 9230 \text{ ohms}$

Ahora para la segunda etapa tenemos que:

$$-\frac{V_o}{V_{o'}} = \frac{R_{f2}}{R_2} ; \text{Si } -\frac{V_o}{V_{o'}} = 14, \text{ entonces } R_{f2} = 14 R_2$$

Si fijamos $R_2 = 10 \text{ Kohms}$, se tiene que $R_{f2} = 140 \text{ Kohms}$

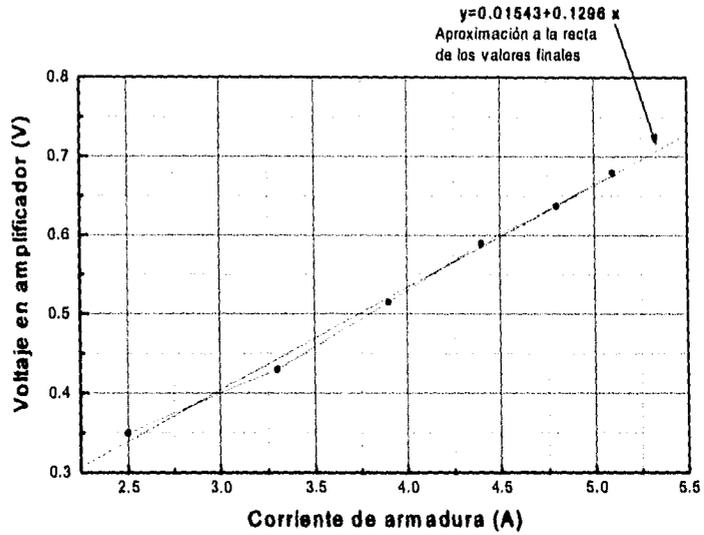
Finalmente, si $R_{2'}$ es igual a $R_2 \parallel R_{f2}$, entonces $R_{2'} = 9333.33 \text{ ohms}$

Análisis 8. Cálculos para encontrar el valor de los componentes del circuito amplificador de la figura 10.

Una vez implementado el amplificador, éste se conecta en la salida del convertidor de AC a DC, para posteriormente proceder a realizar nuevamente las mediciones de voltaje para los mismos valores de corriente de armadura de la tabla 5; así obtenemos los valores que se muestran en la tabla 6.

Velocidad Periférica (Ft / min)	Corriente de Armadura (A)	Voltaje a la salida del amplificador (V)
10	2.5	0.350
20	3.3	0.430
30	3.9	0.515
40	4.4	0.590
50	4.8	0.638
60	5.1	0.680

Tabla 6. Valores de voltaje medidos en la salida del amplificador conectado al convertidor de AC a DC.



Gráfica 6. Representación gráfica de los voltajes finales obtenidos en el amplificador.

El comportamiento de los voltajes amplificados sigue siendo aproximado a una línea recta, pero con la ventaja de que ahora su magnitud se encuentra plenamente en el intervalo de valores que maneja a la entrada el convertidor A/D del microcontrolador a utilizar en el sistema. Así tenemos que el circuito acondicionador final es la conjunción del convertidor de AC a DC con el amplificador de dos etapas.

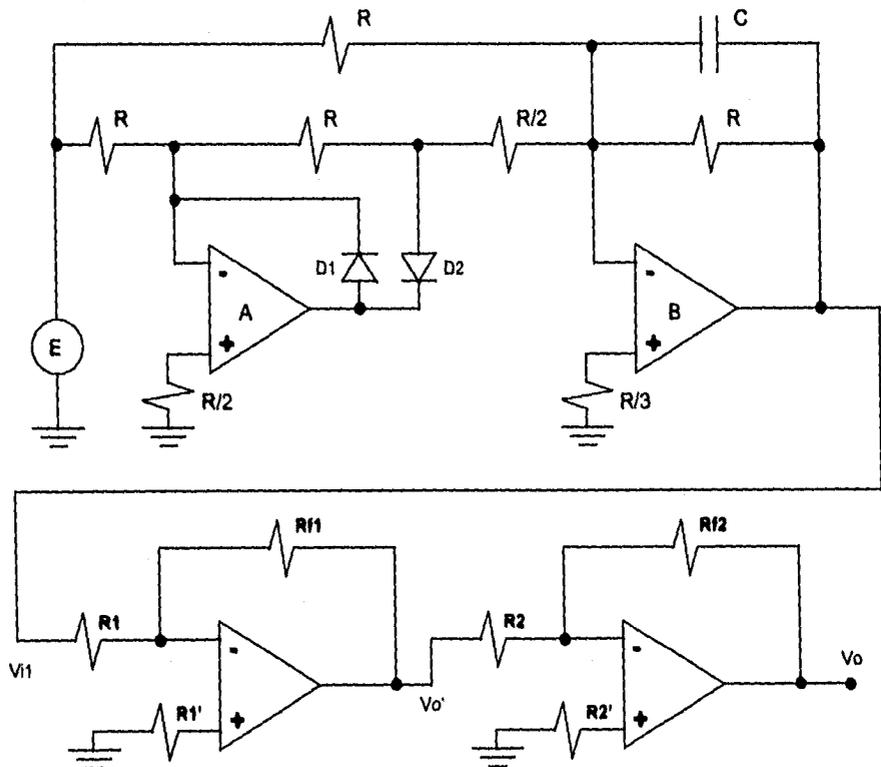
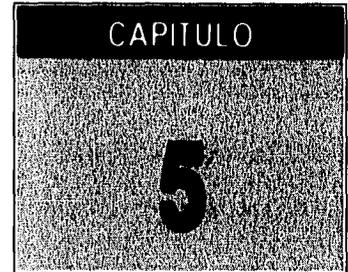


Figura 11. Circuito acondicionador de señal para la variable Corriente de Armadura.

DISEÑO DE LA ARQUITECTURA DEL SISTEMA



Siempre que se tiene planeado utilizar un microcontrolador como parte fundamental en el desarrollo de un sistema, ya sea de adquisición de datos o de control, se hace necesario diseñar una plataforma de trabajo para dicho microcontrolador, independientemente del que se tenga planeado utilizar. Dicho plataforma es mejor conocida como Arquitectura y no es más que una tableta de circuito impreso que contiene, además del microcontrolador, una serie de dispositivos periféricos auxiliares (algunos en forma de circuitos integrados), los cuales pueden ser memorias (RAM, ROM y EEPROM), interfaz de comunicaciones (MAX 232), controladores para el flujo de datos (Latches y Buffers), terminales para la entrada y salida de datos, etc.

En este capítulo describimos el diseño de la arquitectura utilizada en el Sistema de Adquisición y Despliegue de Información. Como se explicó en el capítulo 1, la arquitectura constituye el módulo intermedio (módulo 2) del sistema, ya que se encarga de controlar el flujo de información del módulo acondicionador de señales (módulo 1) hacia la computadora tipo PC (módulo 3); la gran ventaja de utilizar un microcontrolador radica en que al mismo tiempo que controlamos el flujo de información, podemos convertir los valores analógicos provenientes de módulo acondicionador de señales, a valores digitales que puedan ser transmitidos e interpretados por la computadora tipo PC, ésto gracias al convertidor analógico/digital integrado al microcontrolador.

De manera anticipada mencionaremos que la arquitectura esta constituida básicamente por un microcontrolador configurado en su modo simple de operación, además de una interfaz de comunicación que se apega al estándar de comunicación serial RS-232.

5.1. SELECCION DEL MICROCONTROLADOR.

El primer paso a seguir en el diseño de una arquitectura, es seleccionar el microcontrolador que se adecúe a las necesidades del sistema que se desee implementar. Debido a que son diversas las marcas y versiones de microcontroladores existentes en el mercado, se hace necesario llevar a cabo un análisis de las

características de éstos y compararlas con los requerimientos del sistema, esto con la finalidad de hacer la elección más adecuada.

Para el diseñador es importante que el microcontrolador sea la parte básica de la arquitectura, esto se logra al poder concentrar dentro de él la mayor parte posible de dispositivos a utilizar como son memorias y periféricos.

5.1.1 REQUERIMIENTOS DEL SISTEMA.

A continuación mencionaremos los requerimientos de mayor relevancia que se deben tomar en cuenta para la selección del microcontrolador.

- **Tamaño de palabra del microcontrolador de 8 bits.** Debido a que el sistema será utilizado con fines de adquisición de datos, las restricciones físicas como son precisión e intervalo dinámico se satisfacen al trabajar con un microcontrolador que tenga un tamaño de palabra de 8 bits.
- **Periféricos integrados.** El tener una arquitectura pequeña y con bajo consumo de energía es, sin lugar a duda, un requerimiento importante del sistema, el cual se cubre al contar con un microcontrolador que tenga periféricos integrados, tales como un convertidor A/D o un temporizador.
- **Convertidor analógico-digital.** Debido a que existen cuatro señales de tipo analógico provenientes del módulo encargado del acondicionamiento, se hace necesaria la presencia de un convertidor analógico/digital que adecue dichas señales analógicas a un formato digital que permita su posterior transmisión a la computadora tipo PC.
- **Interfaz de comunicación serial.** Este es un requerimiento importante ya que una vez que la información es convertida de su forma analógica a su forma digital, ésta debe ser transmitida, utilizando comunicación serial asíncrona, a la computadora tipo PC para ser procesada.
- **Temporizador.** Este es otro dispositivo que se debe considerar ya que puede facilitar enormemente la medición de la velocidad angular de los rodillos de laminación.
- **Memorias.** Es necesario considerar dos tipos de memorias: *memoria de datos (RAM)* y *memoria para el programa (EEPROM)*. Debido a que prácticamente no será necesario registrar o almacenar gran información, ya que ésta será transmitida inmediatamente, se puede considerar una memoria RAM de tamaño pequeño. Por otro lado se calcula que el programa monitor ocupará alrededor de 500 bytes, sin

considerar futuras modificaciones encaminadas a la optimización del sistema, por lo tanto el tamaño de la memoria EEPROM debe ser de tamaño considerable (1 Kbyte ó más).

- **Gran inmunidad al ruido.** Debido a que el lugar en el que trabajará la arquitectura existe un ambiente ruidoso por la presencia de diversas máquinas, incluyendo los motores de la laminadora, el microcontrolador debe contar con una gran inmunidad al ruido.

Considerando lo anterior y tomando en cuenta la familiaridad y experiencia en el manejo de los microcontroladores de la familia M68HC11 de Motorola, decidimos utilizar el microcontrolador MC68HC811E2 para implementarlo en su modo simple de operación en la arquitectura a desarrollar.

A continuación se hará una breve descripción de las principales características que hacen del MC68HC811E2 el microcontrolador adecuado para desarrollar la arquitectura del Sistema de Adquisición y Despliegue de Información.

5.1.2 BREVE DESCRIPCIÓN DEL MICROCONTROLADOR MC68HC811E2.

El MC68HC811E2 es un microcontrolador con un gran avance y sofisticación en sus capacidades periféricas. Utiliza novedosas técnicas de diseño que le permiten desarrollar una velocidad nominal de procesamiento de 2 MHz. Tiene la ventaja de contar con un diseño estático completo, el cual le permite su operación a bajas frecuencias para DC, además de reducir el consumo de potencia. En adición a lo anterior, la tecnología HCMOS utilizada en el desarrollo de este microcontrolador combina un tamaño reducido y altas velocidades de procesamiento con un bajo consumo de potencia y una gran inmunidad al ruido.

En lo que a memorias integradas se refiere, este microcontrolador incluye una memoria EEPROM de 2 Kbytes y una RAM de 256 bytes. Además en cuestión de periféricos cuenta con los siguientes dispositivos:

- Un convertidor Analógico/Digital (A/D) de 8 canales con una resolución de 8 bits.
- Una interfaz de comunicación serial asíncrona (SCI).
- Una interfaz periférica serial síncrona (SPI).
- Un sistema temporizador principal de libre corrimiento de 16 bits con 3 líneas de captura de entrada.
- Una función de interrupción en tiempo real.
- Un subsistema acumulador de pulsos de 8 bits que puede ser configurado para medir periodos externos o como contador de eventos.

Para complementar el funcionamiento de dichos periféricos el microcontrolador cuenta con 5 puertos diferentes de entrada y salida de datos (puertos A, B, C, D y E respectivamente), cuyas funciones dependen del periférico a utilizar, así como también del modo de operación seleccionado.

Aunado a todo esto, el microcontrolador escogido cuenta con lo siguiente:

- Un circuito de monitoreo o control para protección contra errores del sistema.
- Un sistema guardián COP que protege al microcontrolador contra fallas en el software.
- Un sistema controlador del reloj que genera un restablecimiento sistematizado (RESET) en caso de que el reloj se pierda o funcione demasiado lento.
- Dos modos de control llamados poder-salvado (WAIT y STOP), que se habilita para conservar energía adicional.

Como se puede observar, con todas las características del MC68HC811E2 anteriormente mencionadas podemos cubrir perfectamente los requerimientos de la Arquitectura del Sistema de Adquisición y Despliegue de Información, por lo que el siguiente paso a seguir es la implementación de dicha arquitectura.

5.2 ARQUITECTURA DEL SISTEMA.

Como se mencionó al principio del presente capítulo, la arquitectura estará formada básicamente por el microcontrolador, en este caso el MC68HC811E2, configurado en su modo simple de operación y por la interfaz de comunicación para niveles RS232. Se trata de una arquitectura sencilla debido a que el microcontrolador escogido incluye los periféricos y memorias necesarios para cubrir las necesidades del Sistema de Adquisición y Despliegue de Información. En la figura 1 se presenta el diagrama electrónico de dicha arquitectura con todos los componentes que la integran.

La mayoría de las conexiones para la operación en modo simple del microcontrolador y de la arquitectura, pueden verse claramente en la figura 1; solamente no podemos apreciar bien las conexiones para el circuito de restablecimiento (RESET) del sistema. En la figura 2 se pueden apreciar claramente las conexiones de dicho circuito de RESET utilizando como componentes principales dos inversores de los seis que contiene el circuito integrado 74HC14.

5.2.1 PERIFERICOS.

La adquisición de las señales analógicas acondicionadas para las variables temperatura, carga, par y espesor, se hará a través de las terminales PE0, PE1, PE2 y PE3 del puerto E, las cuales corresponden a los cuatro primeros canales del convertidor analógico/digital integrado al microcontrolador. Una ventaja que proporciona dicho convertidor es la de contar con otros cuatro canales de conversión, lo cual sin duda será de gran utilidad en el caso de que se desee adquirir información para otras variables.

Para manejar la variable velocidad angular, se utilizará la terminal PA2 del puerto A como vía de adquisición de la señal correspondiente. Esta terminal forma parte del sistema temporizador integrado al microcontrolador y junto con la función de captura de entrada, nos ayudará a poder calcular el período de la señal y en consecuencia la velocidad angular en los rodillos de laminación.

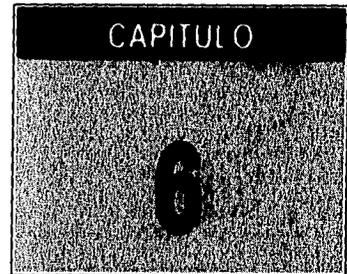
La transmisión y recepción de información con la computadora tipo PC se llevará a cabo mediante la interfaz de comunicación serial integrada en el microcontrolador; el acceso a dicha interfaz se hace a través de las terminales PD0 (terminal RxD) y PD1 (terminal TxD) del puerto D, las cuales a su vez deben conectarse a la interfaz de comunicación para niveles RS 232.

5.2.2 INTERFAZ PARA NIVELES RS 232.

Debido a que los niveles de voltaje que maneja el microcontrolador en su interfaz de comunicación serial integrada no son compatibles con los que maneja la computadora tipo PC en su puerto de comunicación serial, se hace necesario agregar a la arquitectura una interfaz que se encargue de acoplar los niveles de voltaje del microcontrolador (niveles TTL/CMOS) con los de la computadora tipo PC (niveles RS 232). Esta interfaz se encuentra representada por el circuito integrado MAX232, el cual convierte niveles de 0 a 5 volts (niveles apropiados para lógicas TTL y CMOS) a niveles RS 232 y viceversa. Para lograr la compatibilidad de los niveles de voltaje, las terminales TxD y RxD de la interfaz de comunicación serial del microcontrolador se conectan al MAX232, el cual a su vez se conecta a las terminales TxD y RxD provenientes de la computadora tipo PC (figura 1). Cabe mencionar que los capacitores que se encuentran conectados al MAX232 sirven para configurar dicho circuito y en consecuencia para que lleve a cabo la conversión de voltajes de la mejor manera posible.

El siguiente paso en el diseño de la arquitectura es el que se refiere a la programación del microcontrolador. Dicha programación se explica en el siguiente capítulo enfocado al diseño de la programación en alto y bajo nivel del Sistema de Adquisición y Despliegue de Información.

DISEÑO DE LA PROGRAMACION DEL SISTEMA



Cuando dentro de un sistema existen componentes a nivel de hardware, siempre detrás de dichos componentes existe una ardua labor de programación de los mismos con la finalidad de lograr su más alto desempeño dentro del sistema del cual forman parte. Dicha labor de programación se puede llevar a cabo a varios niveles que pueden ir desde el más bajo, como programar directamente a nivel de código - máquina los circuitos de procesamiento (microprocesadores o microcontroladores) que forman parte del hardware del sistema; hasta los más altos, haciendo uso de lenguajes de programación para computadora.

Dentro de los lenguajes de programación existen varias clasificaciones, por generaciones, por niveles, por categorías, etc. La que se muestra a continuación se basa en categorías y niveles

- Lenguajes ensambladores. Menor Nivel.
- Lenguajes para aplicación de sistemas. Nivel Medio. C, BCPL, Macro-ensamblador.
- Lenguajes estáticos de Alto Nivel. COBOL y FORTRAN.
- Lenguajes de Alto Nivel estructurados por bloques. ALGOL, PASCAL, ADA, MODULA.
- Lenguajes dinámicos de Alto Nivel. APL, PROLOG, LISP.

Para la elección de un lenguaje de programación se deben tomar en cuenta varios factores que pueden ir desde la plataforma de hardware con que se cuente, la transportabilidad, la facilidad de mantenimiento, la aplicación del sistema, la eficiencia del compilador, hasta el tiempo disponible para desarrollar el trabajo y la actitud y disponibilidad del programador.

El Sistema de Adquisición y Despliegue de Información para la Laminadora Hille 25 se encuentra integrado por dos componentes a nivel de hardware para poder funcionar completamente, dichos componentes son:

- Una arquitectura basada en un microcontrolador MC68HC11 tipo E2, en modo simple de operación.
- Una computadora tipo PC.

Dichos componentes constituyen la plataforma de trabajo, a nivel de hardware, sobre la cual se debe llevar a cabo el trabajo de programación del Sistema de Adquisición y Despliegue de Información. Como consecuencia de esto, surge la necesidad de desarrollar la labor de programación en dos partes:

Parte 1. Programación del microcontrolador utilizado en la arquitectura.

Parte 2. Programación de la computadora tipo PC.

Para llevar a cabo la tarea de programación de la parte 1 sólo existe la opción de utilizar un lenguaje de bajo nivel que permite el manejo de instrucciones a nivel de código - máquina, siendo dicho código el empleado por los microcontroladores y microprocesadores. Así tenemos que el lenguaje utilizado es el de tipo ensamblador.

Para la programación de la parte 2, de entre los lenguajes de programación de niveles alto y medio disponibles, como pueden ser C, Pascal, Basic e incluso Clipper, se escogió el C en su versión Turbo C debido a varias ventajas sobre los demás como son:

- Su fácil acceso a los puertos de comunicación de la computadora.
- La gran disponibilidad que tiene para el manejo de ambientes gráficos en color y monocromáticos.
- Permite estructurar el código fuente mediante el uso de funciones y procedimientos definidos por el usuario.
- Existen muchas funciones ya predefinidas por el lenguaje mismo, en comparación con el ANSI C.
- El tiempo requerido para la compilación del código fuente es reducido.
- Permite la generación de archivos ejecutables en poco tiempo y de tamaño reducido considerando la cantidad de código fuente.
- Las rutinas de más bajo nivel desarrolladas en el ANSI C pueden ser utilizadas en esta versión Turbo.
- Tiene transportabilidad con otras versiones de lenguaje C, como son Turbo C++, Borland C++ for Windows, etc.
- Permite la manipulación de bits, bytes y direcciones (elementos básicos con los que funciona la computadora).
- Con la estructuración del código fuente se hace fácil el mantenimiento del sistema.

A continuación, se analizan por separado cada una de las partes que conforman la labor de programación del Sistema de Adquisición y Despliegue de Información para la laminadora Hille 25.

6.1 PROGRAMACION Y DISEÑO DEL SOFTWARE A UTILIZARSE EN LA ARQUITECTURA DEL SISTEMA.

Después del diseño y construcción de la arquitectura del sistema basada en un microcontrolador, se hace necesario llevar a cabo una labor de programación de la misma. Este proceso nos debe dar como resultado el desarrollo de un programa monitor, el cual tenga como funciones principales la adquisición, conversión y transmisión de las señales provenientes del modulo acondicionador de señales.

6.1.1 PROGRAMA MONITOR.

La figura 1 que a continuación se presenta tiene como finalidad mostrar el diagrama de flujo general del programa monitor, dicho diagrama nos da una visión de muy alto nivel de los procesos realizados dentro del microcontrolador de nuestra arquitectura. Para entender más fácilmente cada uno de los procesos, haremos en seguida una descripción más amplia de cada uno de ellos.

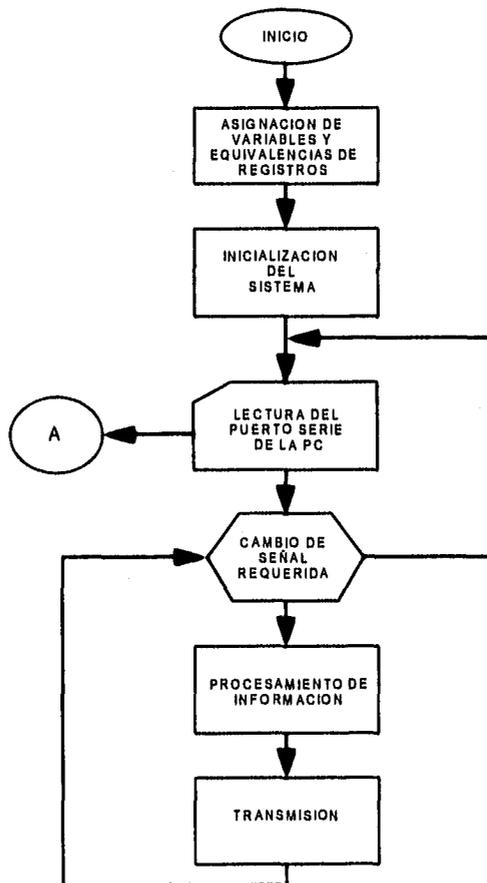


Figura 1 Diagrama de flujo general del programa monitor.

6.1.1.1 ASIGNACION DE VARIABLES Y EQUIVALENCIA DE REGISTROS.

Este primer bloque del programa monitor tiene como funciones principales el indicar la equivalencia de cada registro a utilizar con su correspondiente dirección, así como también la asignación de las variables a la memoria RAM.

Los registros inicializados son aquellos involucrados con el convertidor analógico-digital, la interfaz de comunicación serial asincrónica, la función de captura de entrada del temporizador y algunas interrupciones mascarables. Por otro lado, la poca memoria RAM utilizada en el programa es ocupada por algunas variables que intervienen, principalmente, en el registro del periodo de la señal conocida como velocidad angular.

6.1.1.2 INICIALIZACION DEL SISTEMA.

Este bloque se refiere básicamente a la inicialización de cada uno de los periféricos del microcontrolador que van a ser utilizados en el sistema.

Para inicializar al convertidor analógico-digital se cuenta con los registros *OPTION* y *ADCTL*. La habilitación de la bomba de carga del convertidor se logra escribiendo un "1" en el bit *ADPU* del registro *OPTION*, después de lo cual es necesario programar un retardo para que la bomba de carga y todos los circuitos de comparación se estabilicen. Para lograr que el convertidor realice conversiones continuas en cuatro canales, es necesario escribir una serie de 1's en los bits *MULT* (canal múltiple) y *SCAN* del registro *ADCTL*, finalmente para seleccionar los primeros cuatro canales del convertidor se escriben 0's en los bits *CD*, *CC*, *CB* y *CA* del mismo registro *ADCTL*.

En lo que respecta a las habilitaciones del temporizador, en su función de captura de entrada, y de las interrupciones mascarables; éstas se llevarán a cabo en el bloque de procesamiento de información referente a la variable velocidad angular, debido a que si se inicializaran en este bloque el programa no funcionaría correctamente como se comprenderá más adelante.

Para inicializar la interfaz de comunicación serial (SCI), la cual es un sistema de transmisión y recepción asincrónica (UART) con formato estándar de no retorno a cero (NRZ), es necesario trabajar sobre los registros *BAUD*, *SCCR1* y *SCCR2*. El registro *BAUD* es el encargado de controlar de baudaje de la comunicación, el

cual para nuestro caso deberá estar inicializado a 2400 bits/seg; ésto se logra escribiendo un \$32 (valor hexadecimal) en dicho registro, quedando involucrados los bits *SCP1, SCP0, SCR2, SCR1* y *SCR0*. Mediante el registro *SCCR1* se programa el formato de datos para la comunicación, el cual quedará configurado, en nuestro caso, para un tamaño de palabra de ocho bits, un bit de inicio y un bit de paro, limpiando dicho registro. Por último, la habilitación del transmisor y receptor se logra escribiendo 1's en los bits *TE* y *RE* del registro *SCCR2*.

6.1.1.3 LECTURA DEL PUERTO SERIAL DE LA PC.

Una vez inicializado el sistema, es necesario saber con qué señal se desea trabajar, ésto con la finalidad de efectuar la adquisición, procesamiento y transmisión adecuadas. Para ésto, el primer paso a seguir es verificar que existe comunicación entre la computadora tipo PC y el SCI del microcontrolador, lo cual se logra leyendo el registro de estados del SCI, si éste nos indica que el registro de recepción de datos está lleno se procede a leer el registro de comunicación. El software en la computadora PC deberá estar diseñado para transmitir un caracter determinado al microcontrolador dependiendo de cuál sea la señal que se desea observar, dicho caracter está relacionado de la siguiente manera:

Señal	Caracter	Equivalencia
		Hexadecimal
TEMPERATURA	T	\$ 54
CARGA	C	\$ 43
PAR	P	\$ 50
ESPESOR	E	\$ 45
VELOCIDAD	V	\$ 56
GLOBAL	G	\$ 47

Tabla 1. Relación entre los caracteres transmitidos y las variables.

Después de haber sido leído dicho caracter por el SCI, se hace una comparación para saber de qué señal se trata, y proceder a su adquisición, procesamiento y transmisión. Cabe mencionar que si el caracter leído es una letra "G", ésto quiere decir que se desea un monitoreo global, lo cual significa procesar una señal tras otra dentro de un ciclo. A continuación se muestra el diagrama de flujo encaminado a explicar esquemáticamente cómo funciona la lectura del puerto serie (figura 2).

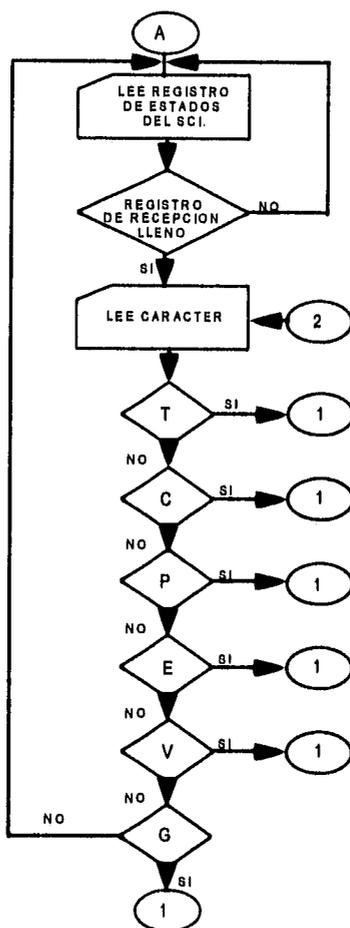


Figura 2. Diagrama de flujo para la lectura del puerto serie de la computadora PC.

6.1.1.4 PROCESAMIENTO DE INFORMACION.

Esta parte del programa consiste en la adquisición de cada una de las señales después de haber pasado por el periférico correspondiente. Como hemos mencionado anteriormente, las señales para las variables temperatura, carga, par de laminación y espesor pasan por el convertidor analógico-digital, por lo que su procesamiento se hace en forma similar. Un caso especial es el de la señal correspondiente a la variable par de laminación ya que en este caso carecemos de un transductor capaz de medir directamente el par (debido a su alto costo y difícil acoplamiento a la laminadora). En su lugar la señal que pasará por el convertidor A/D es la correspondiente a la corriente de armadura, la cual debe ser transmitida conjuntamente con la de velocidad angular, para poder determinar finalmente, en forma indirecta y mediante software en la PC, el valor del par de laminación.

La figura 3 nos muestra la forma en que se adquieren o registran las señales de temperatura, carga y espesor, cuyos canales del convertidor correspondientes son el 1,2 y 4 respectivamente.

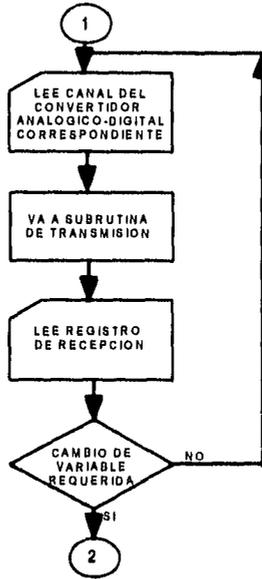


Figura 3. Diagrama de flujo del procesamiento de las señales de temperatura, carga y espesor.

A continuación se muestra en la figura 4 el diagrama de flujo que explica cómo se lleva a cabo la transmisión de las dos señales requeridas para calcular el valor del par de laminación.

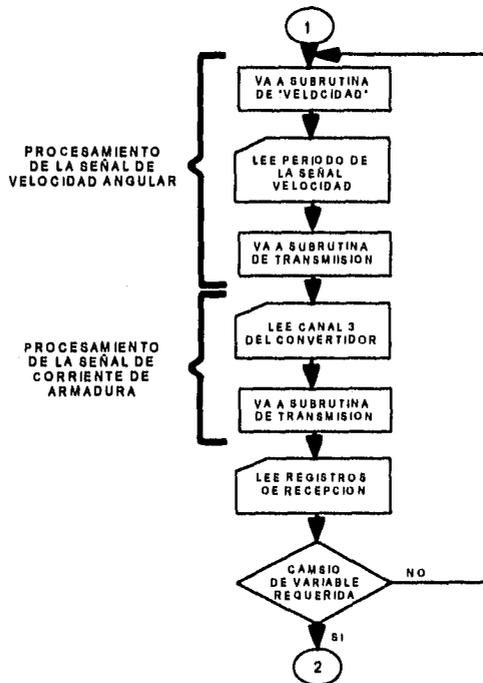


Figura 4. Diagrama de flujo para el procesamiento de las señales Velocidad Angular y Corriente de Armadura

En lo que se refiere a la señal de la variable velocidad angular, el dato adquirido por la subrutina de velocidad (el cual posteriormente se transmite a la computadora PC) es el correspondiente a la duración del periodo de dicha señal. Este dato queda almacenado en la subrutina en una variable de ocho bits (*OVCNT1*), la cual representa el número de veces que se llenó el registro de 16 bits *TIC1* con los periodos de reloj del microcontrolador transcurridos ($E = 500 \text{ ns}$), finalmente la duración del periodo de la señal, se calcula mediante el software de la PC como:

$$T = OVCNT1 \times 65,535 \times E$$

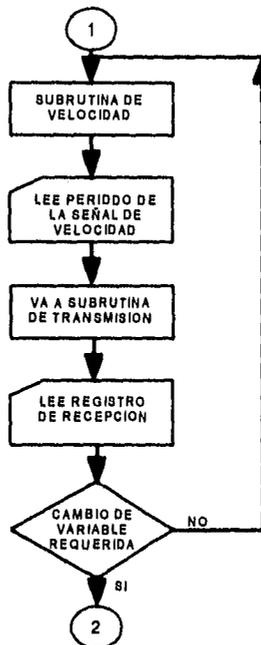


Figura 5. Diagrama de flujo para el procesamiento de la señal velocidad angular.

6.1.1.4.1 SUBROUTINA DE VELOCIDAD.

Para poder adquirir el dato correspondiente a la señal de la variable velocidad angular, hubo la necesidad de desarrollar una subrutina independiente que nos proporcionara dicho dato con la mayor exactitud posible. La subrutina de velocidad captura el tiempo transcurrido entre dos niveles altos de la señal (los cuales son generados por los imanes colocados en la flecha principal de la laminadora), por lo cual fue necesario trabajar con la función de captura de entrada del temporizador y las interrupciones *IC1* (Input Capture 1 Interrupt) y *TOI* (Timer Overflow Interrupt). El siguiente diagrama muestra esquemáticamente como es que funciona dicha subrutina.

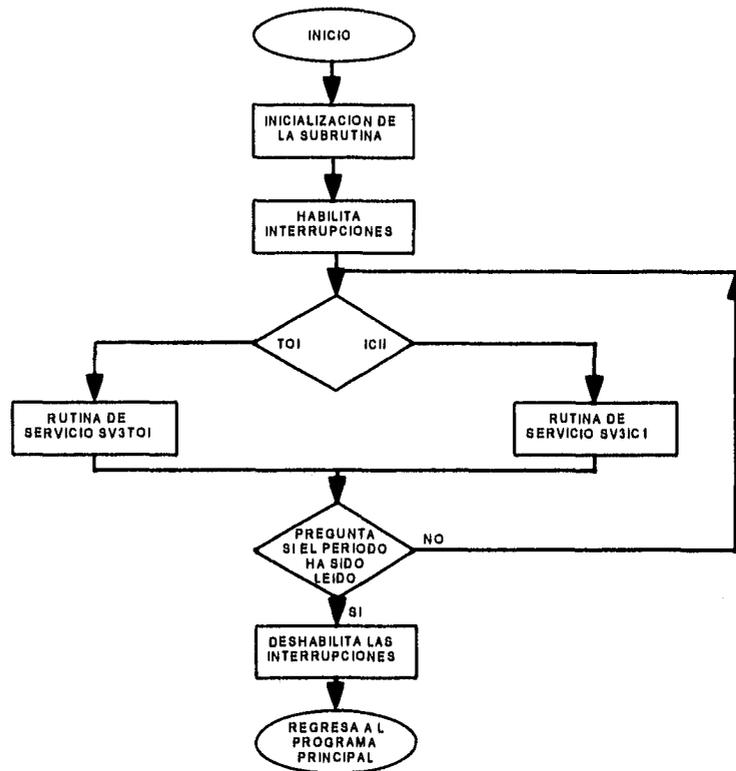


Figura 6. Diagrama de flujo de la subrutina de velocidad.

El primer bloque de la subrutina inicializa al temporizador para trabajar con los niveles altos de la señal, lo cual se logra al escribir el equivalente binario de un \$10 hexadecimal en el registro dos de control del temporizador (*TCTL2*).

El siguiente paso consiste en limpiar las banderas de interrupción *IC1F* y *TOI*, escribiendo un "1" en los bits tres y siete de los registros *TFLG1* y *TFLG2* respectivamente. Consecuentemente se habilitan las interrupciones *IC1I* y *TOI* mediante los bits tres y siete de los registros *TMSK1* y *TMSK2* respectivamente.

Posteriormente, cada vez que el registro *TIC1* se llena, se levanta la bandera *TOI*, por lo que el programa llama a la rutina de servicio *SV3TOI* y si la medición del periodo está en progreso, entonces el contador *OVCNT1* se incrementa.

Por otra parte, la rutina de servicio *SV3IC1* es llamada inicialmente cuando el primer nivel alto es detectado, registrando el tiempo transcurrido hasta entonces. La segunda llamada se presenta con el

segundo nivel alto, y de igual manera se registra su tiempo. El periodo de tiempo transcurrido entre los dos niveles altos, se determina restando el primer registro del segundo.

Finalmente, si el periodo de la señal ha sido medido, no queda otra cosa que hacer más que deshabilitar las interrupciones y regresar al programa principal. Esta subrutina es capaz de contar un máximo de 16,777,215 ciclos de reloj, aproximadamente 8.38 segundos y un mínimo de 70 ciclos de reloj.

6.1.1.4.2 MONITOREO GLOBAL

El monitoreo global implica adquirir y transmitir una por una las señales correspondientes a cada una de las cinco variables con las que trabajará el Sistema de Adquisición y Despliegue de Información. El proceso es similar al utilizado para la variable par de laminación pero con la diferencia de que ahora, además de la velocidad angular y la corriente de armadura, intervienen la temperatura, la carga y el espesor. En la figura 7 se muestra el diagrama que describe esquemáticamente este proceso.

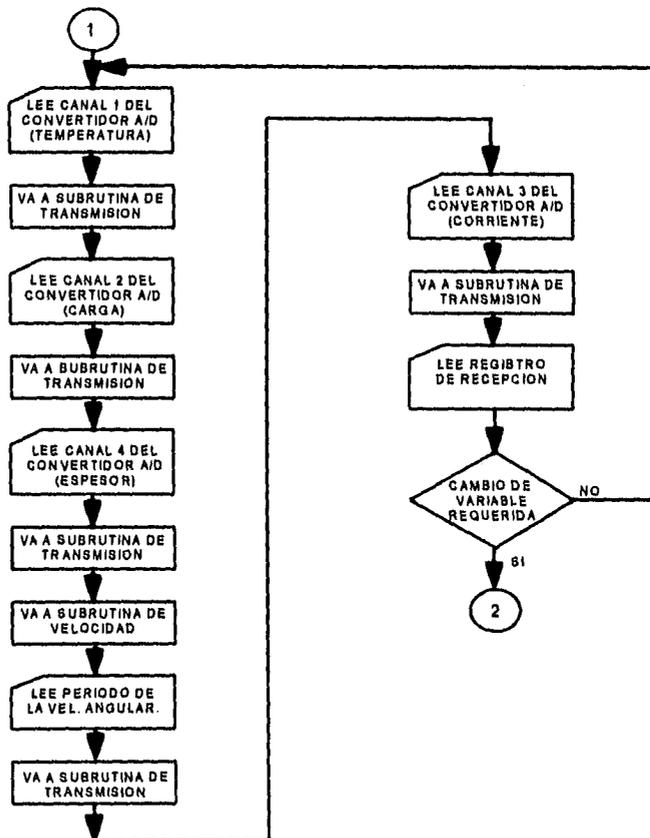


Figura 7. Diagrama de flujo para llevar a cabo el monitoreo global.

6.1.1.5 TRANSMISION.

Este último bloque está constituido por la subrutina de transmisión, la cual primero que nada verifica que el registro de transmisión se encuentre vacío, para después proceder a mandar el dato requerido hacia el puerto serial de la computadora tipo PC. Finalmente verifica por medio del bit seis del registro de estados del SCI que la transmisión se haya completado, para así poder regresar al programa principal.

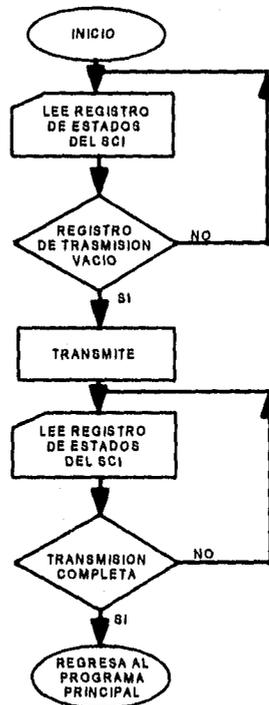


Figura 8. Diagrama de flujo de la subrutina de transmisión.

6.2. PROGRAMACION Y DISEÑO DEL SOFTWARE A UTILIZARSE EN LA COMPUTADORA TIPO PC.

6.2.1 CICLO DE VIDA DEL SOFTWARE.

Al igual que los sistemas a gran escala, independientemente del área sobre la cual trabajen, los grandes sistemas de software requieren de un tiempo considerable para su desarrollo y deben permanecer en uso durante un tiempo aún mayor. En este periodo de desarrollo y uso pueden identificarse varias etapas, que al juntarlas constituyen lo que se conoce como ciclo de vida del software. A continuación se citan y explican las etapas de dicho ciclo de vida del software:

- a) **ANALISIS Y DEFINICION DE NECESIDADES.** Los servicios, restricciones y objetivos del sistema se establecen consultando con los usuarios. Una vez acordados, deben definirse de una manera comprensible, tanto para los usuarios como para el personal de desarrollo. Esta etapa se puede conocer a través de la aplicación de cuestionarios y encuestas a los usuarios finales.
- b) **DISEÑO DEL SISTEMA Y DEL SOFTWARE.** Partiendo de su definición, las necesidades se dividen en sistemas de hardware y sistemas de software. A este proceso se le llama diseño del sistema. El diseño de software es el proceso de representar las funciones de cada sistema de software a fin de poderlo transformar con facilidad en uno o más programas de computación.
- c) **APLICACION Y PRUEBAS DE UNIDADES.** Durante esta etapa el diseño del software se realiza como un conjunto de programas o unidades de programas escritos en algún lenguaje de programación ejecutable. Las pruebas de unidades implican la comprobación de que cada unidad cumple con su especificación.
- d) **PRUEBAS DEL SISTEMA.** Las unidades de programa individuales o los programas se integran y prueban como a un sistema completo para asegurar que se cubren las necesidades del software. Después de las pruebas, el sistema de software se envía al cliente.
- e) **OPERACION Y MANTENIMIENTO.** Esta fase suele ser (aunque no necesariamente) la más larga del ciclo de vida. Se instala el sistema y se pone en uso práctico. La actividad de mantenimiento implica corregir errores que no se descubrieron en las primeras etapas del ciclo de vida, mejorar la aplicación de las unidades del sistema y aumentar los servicios de éste a medida que se perciben nuevas necesidades.

Llevando a cabo esta metodología se pretende llegar a desarrollar sistemas de software de calidad, confiabilidad y fácil mantenimiento.

El código fuente debe implementarse sistemáticamente, de acuerdo con las etapas de la Ingeniería de la Programación, acompañado de su documentación respectiva, como los principios de operación, el manual de usuario, glosario de términos, etc.

6.2.2 DEFINICION DE REQUERIMIENTOS.

El proceso de establecer los servicios que debe proporcionar el sistema y las restricciones con las cuales debe operar se denomina Análisis y Definición de Requerimientos. Esta suele ser la primera fase importante del ciclo de vida del software.

Es importante distinguir entre las necesidades y los requisitos del usuario. Se debe reunir y analizar la información acerca del problema que se va a resolver, y producir una definición completa de éste. A partir de la definición, se puede diseñar y aplicar la solución del software.

También es importante distinguir entre los objetivos y los requisitos del sistema. En esencia, un requisito es algo que puede probarse, mientras que un objetivo es una característica más general que debe exhibir el sistema.

En la Definición de Requerimientos se deben contemplar los siguientes puntos:

INTRODUCCION. Debe describir las necesidades del sistema y colocarlo dentro de contexto, describiendo con brevedad sus funciones presentando un fundamento del sistema de software.

HARDWARE. Si el sistema se va aplicar en un hardware especial, éste y sus interfaces deben describirse; es necesario exponer las configuraciones mínima y óptima en las que puede ejecutarse el sistema.

EL MODELO CONCEPTUAL. El modelo conceptual del sistema es una visión de alto nivel del sistema que muestra los principales servicios proporcionados por el software y las relaciones de unos con otros.

REQUISITOS FUNCIONALES. En esta sección se deben describir los servicios proporcionados al usuario. Dependiendo de la naturaleza de estos requisitos, la notación empleada puede ser el lenguaje natural, un lenguaje semiformal, un lenguaje formal o una mezcla de todas estas notaciones.

REQUISITOS NO FUNCIONALES. Deben expresarse y relacionarse con los requisitos funcionales, es decir, las restricciones en operación bajo las cuales operará el software.

INFORMACION PARA MANTENIMIENTO. Debe describir las suposiciones fundamentales sobre las cuales se basa el sistema y los cambios anticipados debidos a la evaluación del hardware, cambios en las necesidades del usuario, es decir, que el diseño se debe contemplar con vistas futuristas, por ejemplo, crear el sistema con posibilidad de funcionar en un ambiente de red, aunque por el momento no se requiera.

A continuación se presenta la definición de requerimientos correspondiente al sistema de software que se desarrolló para el Sistema de Adquisición y Despliegue de Información.

INTRODUCCION.

El Sistema de Adquisición y Despliegue de Información tiene como objetivo mostrar el comportamiento de las 5 variables físicas que a consideración de los experimentadores encargados de trabajar con la Laminadora Hille 25, son las más importantes para caracterizar el proceso de laminación de diversos materiales. Las variables a trabajar son las siguientes:

- a) La temperatura del material a laminar.
- b) La carga ejercida en los rodillos de laminación
- c) La velocidad angular de los rodillos al laminar
- d) El par en los rodillos de laminación.
- e) El espesor del material laminado.

El presente software lo que pretende es dar un apoyo confiable al Sistema de Adquisición y Despliegue de Información, al mostrar en el monitor de una computadora tipo PC el comportamiento de las 5 variables físicas antes mencionadas de manera confiable y real, con opción de poder mostrar el comportamiento de una sola variable o de las 5 a la vez.

Para poder mostrar dicho comportamiento, primeramente la computadora deberá hacer la adquisición de datos, para ésto deberá establecer una comunicación, por medio de uno de los puertos de comunicación serial, con el microcontrolador del sistema. Al mismo tiempo el microcontrolador se encuentra conectado al

módulo de acondicionamiento de señal, el cual a su vez se encuentra conectado a los transductores de la laminadora encargados de medir en forma real los valores de cada una de las variables físicas.

Una vez establecida la comunicación con el microcontrolador, para observar el comportamiento de una sola variable, o de las 5 a un mismo tiempo, lo que se desea es que sean desplegados en el monitor de la computadora, en tiempo real, los valores medidos por los transductores de la laminadora, los cuales son adquiridos por el microcontrolador y transmitidos hacia la computadora en forma de datos binarios. Dichos datos transformados a números decimales constituirían una representación numérica del comportamiento de la ó las variables físicas. Una vez adquiridos los datos por la computadora, se pretende graficarlos contra el tiempo real en que fueron medidos y adquiridos, con la finalidad de observar una representación gráfica real de dicho comportamiento. Es importante considerar la posibilidad de que el sistema al mismo tiempo que va adquiriendo los datos, los vaya graficando, con la finalidad de observar directamente una representación gráfica del comportamiento de las variables y ahorrar un poco de tiempo.

Como parte complementaria del sistema se deberán tener las opciones de: poder guardar en archivos los datos adquiridos para un determinado proceso de laminación, poder abrir en un momento dado dichos archivos para su posterior análisis, poder cambiar el puerto serial por el cual se pretende establecer la comunicación, y por último, la opción de imprimir los datos adquiridos así como las gráficas obtenidas.

Todos estas opciones de procesamiento y despliegue de información se deberán poder aplicar tanto al observar el comportamiento de una sola variable, o de las 5 a un mismo tiempo.

HARDWARE.

Tomando en cuenta las necesidades anteriormente mencionadas, tenemos que éstas las podemos cubrir empleando para el desarrollo e implantación del sistema una computadora personal tipo IBM PC compatible, que cuente de preferencia con un disco duro, en donde se puedan instalar el sistema, los archivos auxiliares que necesita para su completo funcionamiento y los archivos en donde se van a ir guardando los datos obtenidos cada vez que se hagan experimentos de laminación; además el disco duro nos proporciona una velocidad de ejecución óptima del sistema. En caso de no haber disco duro, se recomienda tener una unidad de diskette, ya sea de 3.5 o de 5.25 pulgadas, pero de alta densidad, para no tener problemas de falta de espacio en diskette. Es recomendable que el sistema se implante, como mínimo, en una máquina que trabaje con un microprocesador 286; por una parte debido al tipo de unidad de diskette requerida y por otra, debido a las velocidades de acceso que puede alcanzar el disco duro.

Algo que es esencial en la computadora tipo PC a usar, es que cuente con un monitor VGA (en color o monocromático) y la tarjeta de video correspondiente; ya que después de analizar distintos tipos de monitores de video (como el HERCULES, CGA, EGA, etc.) éste es el que nos brinda la resolución y la actualidad adecuadas para el despliegue de gráficas, pantallas, colores e incluso textos, y sobre todo porque se tiene planeado que el sistema se desempeñe en un ambiente puramente gráfico. Como consecuencia de este ambiente puramente gráfico, es deseable, más no indispensable, que la computadora tipo PC tenga un coprocesador matemático 80X87 para que el sistema pueda desplegar sin problemas las gráficas, pantallas y menús correspondientes. En caso de no existir coprocesador, será necesario instalar un programa emulador del mismo para que el sistema funcione correctamente.

Por último, es indispensable que la computadora tenga por lo menos un puerto para comunicación tipo serial, con conector tipo DB-9 o DB-25, ya que por medio de este tipo de puerto se lleva a cabo la adquisición de datos conectando la computadora al módulo del microcontrolador por medio de un cable específicamente hecho para el Sistema de Adquisición y Despliegue de Información. En caso de querer imprimir gráficas o datos, es necesario que la computadora cuente con un puerto de comunicación paralela para poder conectar una impresora.

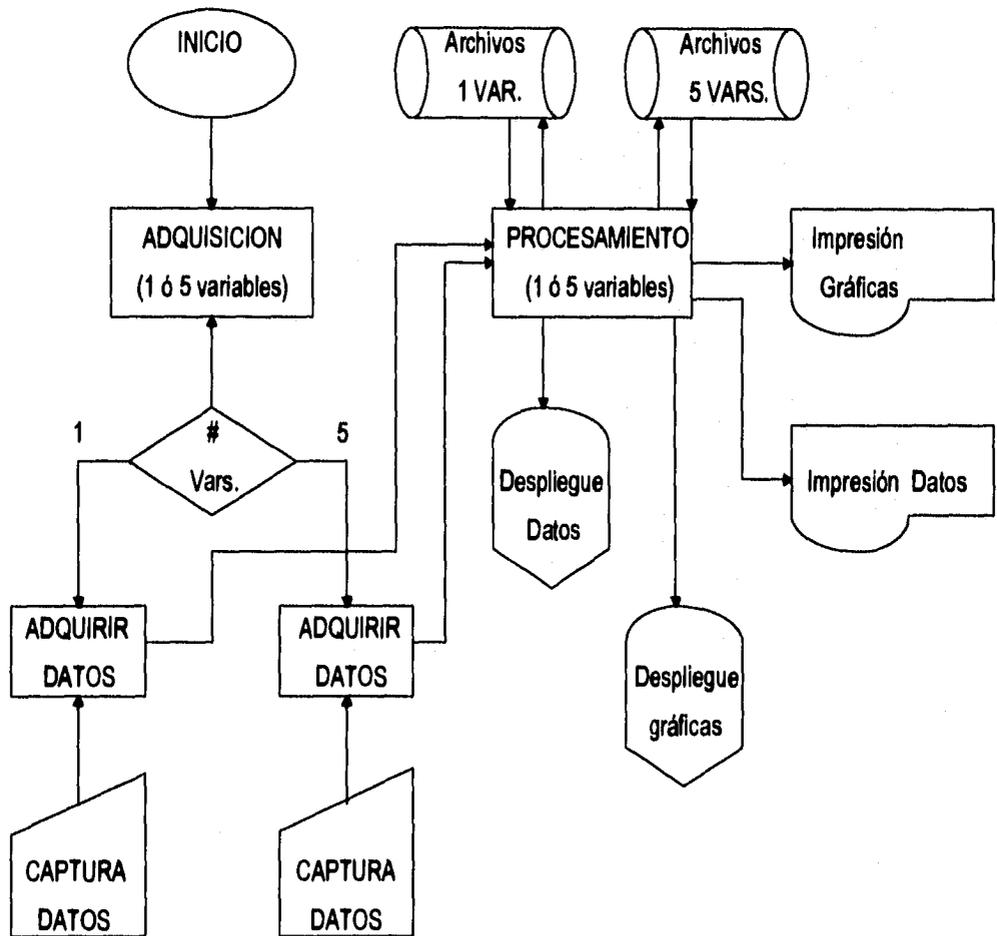
MODELO CONCEPTUAL.

Como se mencionó anteriormente, este modelo es una visión de muy alto nivel del sistema en la que se identifican los servicios principales al usuario y se documentan sus relaciones.

Para sistemas pequeños, este modelo puede existir solamente en la mente del diseñador del sistema, pues comprende los sistemas y sabe qué funciones debe proporcionar, al igual que las restricciones a las operaciones de esas funciones. Para cualquier sistema mayor, sin embargo, quizá resulte inadecuado un modelo mental, pues como el sistema que se modela tiene una complejidad inherente, los modelos mentales tienden a ser incompletos y crean conflictos. Es necesario establecer un modelo explícito del sistema, definido de manera precisa en una etapa inicial, y usar este modelo para comprender el sistema.

Las notaciones más efectivas para describir el modelo conceptual de un sistema son las notaciones gráficas. La razón es que las imágenes y diagramas suelen ser comprensibles para los usuarios que no tienen conocimiento de la ingeniería de software.

En la figura 9 se encuentra el modelo conceptual del sistema mostrando la relación existente entre cada uno de los principales módulos de procesamiento (despliegue, graficación, impresión, etc.) de los datos adquiridos.



La captura de datos es representada por la adquisición de datos por el puerto serie.

Figura 9. Modelo Conceptual del Sistema

REQUISITOS FUNCIONALES.

Los requisitos funcionales del sistema son aquellos servicios que el usuario espera del sistema. En general, al usuario no le interesa cómo se aplican dichos servicios, así que se debe evitar la inclusión de conceptos de aplicación en la presente sección de requerimientos.

Los requisitos funcionales de un sistema deben ser completos y consistentes. Por completos se entiende que todos los servicios requeridos por el usuario deben especificarse, y la consistencia significa que ninguna definición de requisitos debe contradecir a otra.

Comúnmente hay tres maneras de expresar los requisitos funcionales de un sistema:

1. Por medio de un lenguaje natural.
2. En un lenguaje estructurado sin una semántica rigurosa.
3. En un lenguaje formal con una sintaxis y semántica rigurosamente definidas.

De estas notaciones, la más utilizada es el lenguaje natural, por la simple razón de que es más expresivo y porque puede ser comprendido tanto por los usuarios como por los desarrolladores del sistema.

Es muy difícil evitar la mezcla de los requerimientos funcionales y no funcionales y de los objetivos del sistema cuando se utiliza el lenguaje natural para la definición de requisitos. La razón es que no hay una distinción notacional entre ellos, y la separación sólo se puede lograr con ayuda de cuidadosas revisiones de los requisitos. Sin embargo, es necesario usar este lenguaje cuando se formulan requisitos de alto nivel porque la generalidad de éstos no puede expresarse en una notación más restringida.

El desarrollo y aplicación del Sistema de Adquisición y Despliegue de Información como un sistema completo que involucra acondicionamiento de señales, control y digitalización de las mismas, adquisición y despliegue de datos, surge a partir de la necesidad real de automatizar la operación de la laminadora HILLE 25 ubicada en el Laboratorio de Mecánica de la Facultad de Ingeniería; como se menciona en los capítulos de Introducción y Planteamiento del problema.

El sistema de software a desarrollar, como se ha venido mencionando a lo largo de este capítulo, pretende apoyar y complementar el funcionamiento del Sistema de Adquisición y Despliegue de Información, al mostrar, gracias a una computadora tipo PC, de manera confiable y real el comportamiento de las 5 variables físicas que más interesan en el proceso de laminación.

Tanto al observar el comportamiento de una sola variable física o de las 5 a un mismo tiempo, se requiere que los datos directamente leídos por los transductores sean acondicionados y digitalizados con ayuda del módulo de acondicionamiento y del microcontrolador respectivamente, para luego ser transmitidos a la computadora, que los desplegará en el monitor y posteriormente (o al mismo tiempo, si se desea), los graficará contra el tiempo en que fueron adquiridos.

Además existe la necesidad de proporcionar una opción que permita al usuario almacenar los datos adquiridos para un proceso de laminación en particular. Dicho almacenamiento deberá hacerse en archivos que puedan ser abiertos para análisis posteriores, tanto numérico como gráfico, de los datos adquiridos.

Por último, se requiere que el sistema presente en forma de reportes impresos las representaciones numérica y gráfica del comportamiento real de las variables físicas, tanto al adquirir datos referidos a las 5 variables físicas o solamente a una, según la conveniencia del usuario y del experimentador. La finalidad de estos reportes es la de facilitar el análisis de las variables que intervienen en un proceso de laminación, bajo condiciones particulares en la laminadora HILLE 25.

REQUISITOS NO FUNCIONALES.

Un requisito no funcional es aquel que no se necesita de inmediato, pero se debe de pensar que en un momento dado éste requisito que no era funcional se convierta en una necesidad, convirtiendo a este requisito no funcional en un requisito funcional e indispensable. También se entiende por requisito no funcional, aquel requisito que aunque el usuario final no lo pide formalmente, el diseñador del sistema lo debe de contemplar.

De este modo, un requisito no funcional es una restricción u obligación impuesta al servicio del sistema. Un ejemplo claro de los requisitos no funcionales son las obligaciones impuestas a los tiempos de respuesta del sistema, las limitaciones en la cantidad de memoria que ocupará el software y las restricciones en la representación de los datos del sistema.

Aunque tanto los requisitos funcionales como los no funcionales están sujetos a cambios, los requisitos no funcionales se ven especialmente afectados por los cambios en la tecnología del hardware. Puesto que el tiempo de desarrollo de un gran sistema puede ser de varios años, es probable que el hardware al concluir el proyecto sea más potente que el disponible cuando se concibió el proyecto. Además, el hardware evolucionará a través del tiempo de vida del software desarrollado y los requisitos no funcionales se modificarán mientras el software esté en uso.

Uno de los requisitos no funcionales del sistema de software se encuentra reflejado en su manejo, ya que por el momento las opciones del menú principal y de los submenús correspondientes son escogidas por medio de la primera letra que le da nombre a la opción; o más fácilmente, colocando el marco de desplazamiento, con ayuda de las flechas del teclado (o flechas de desplazamiento), en la opción deseada y

posteriormente presionando la tecla "entrar". Posteriormente no se debe descartar la posibilidad de que el manejo del sistema se facilita más con ayuda de un mouse que permita acceder directamente menús y submenús.

Otro requisito de este tipo lo podemos encontrar en lo que a automatización de la laminadora se refiere, ya que por el momento el software y la computadora solamente sirven como un medio de adquisición y despliegue de datos y gráficas. Posteriormente no debemos descartar la posibilidad de que la computadora y el software, por medio de una serie de especificaciones iniciales, actuadores y un protocolo de comunicación más elaborado entre la computadora y el microcontrolador, sea la encargada de controlar y manipular las variables que intervienen en el proceso de laminación a placer del experimentador o del usuario.

INFORMACION PARA MANTENIMIENTO.

El término "mantenimiento" generalmente se ha aplicado al proceso de modificar un programa cuando ya se ha entregado y está en uso. Estas modificaciones pueden implicar cambios sencillos para corregir errores de codificación, cambios mayores para corregir errores de diseño o reescrituras drásticas para corregir errores de especificación o introducir nuevos requisitos.

Sin embargo, el término mantenimiento debe considerarse para denotar la modificación de un programa con el fin de corregir errores y proporcionar nuevas posibilidades. Hay tres tipos de mantenimiento de software:

1. Mantenimiento de Perfeccionamiento.
2. Mantenimiento Adaptativo.
3. Mantenimiento Correctivo.

El mantenimiento de perfeccionamiento comprende los cambios solicitados por el usuario o por el programador del sistema; el mantenimiento adaptativo se debe a cambios en el ambiente del programa y el mantenimiento correctivo es la corrección de errores del sistema no descubiertos.

Durante la vida del sistema de software, sus requisitos tenderán a ser modificados con la finalidad de reflejar necesidades cambiantes, el ambiente del sistema cambiará y surgirán otros errores ocultos, no descubiertos durante la comprobación del sistema. Como quiera que sea el mantenimiento es inevitable, los sistemas se deben diseñar y aplicar de modo que se minimicen los problemas en este aspecto.

Los costos de mantenimiento son muy difíciles de estimar con anticipación. La evidencia de los sistemas existentes muestra que los costos de mantenimiento son, en mucho, los más cuantiosos del desarrollo y uso del sistema. En general, esos costos fueron muy subestimados al diseñar y aplicar los sistemas. Es cierto que los costos de mantenimiento varían mucho de una aplicación a otra, pero, en promedio, representan alrededor de cuatro veces los costos de desarrollo en grandes sistemas de software.

Para reducir los costos de mantenimiento se deben especificar claramente todos los requisitos desde su fase de planeación del sistema, ya que de no ser así se caería en la necesidad de un mantenimiento frecuente y específicamente en un mantenimiento correctivo; este tipo de mantenimiento es el más costoso y requiere de un tiempo considerablemente mayor para que se lleve a cabo porque afecta directamente al diseño original del sistema, ya que implica un drástico rediseño y reescritura de una o más unidades del programa.

Algunos otros factores que se podrían mencionar para reducir los costos de mantenimiento, son los factores técnicos. Entre estos factores se encuentran la "Interdependencia de los módulos" que se refiere básicamente a que al modificar una unidad del programa se pueda realizar sin afectar a ninguna otra unidad. Otro de ellos es la "Comprobación y prueba del Programa", es decir, que en cuanto más tiempo y esfuerzo se dedique a la comprobación del diseño y a la prueba del programa, menos errores tendrá éste y, en consecuencia, los costos de mantenimiento debidos a la corrección de errores serán menores.

6.2.3. ESTUDIO DE FACTIBILIDAD.

Primeramente, es preciso entender que un sistema de computación, en sí mismo, no es un sistema de información. Sin embargo, puede ser un instrumento básico que incremente la efectividad del sistema y proporcione los medios para llevar a cabo ciertas actividades que de otra forma parecerían imposibles. Para nuestro caso al decir sistema de información nos referimos al sistema completo de Adquisición y Despliegue de Información para la Laminadora Hille 25, el cual tiene como finalidad principal, automatizar el proceso de adquisición y despliegue de información, la cual es generada al momento de que la laminadora se encuentra trabajando. Es necesario señalar que no todos los sistemas formales de información necesitan de la computadora y su tecnología, que la computadora no necesariamente mejora un sistema y que no todos los reportes que produce una computadora se pueden calificar como información. El hecho de computarizar el sistema actual de manejo de información, o un proceso anterior, no necesariamente acrecentará su efectividad; más bien sucede lo contrario: la instalación de una computadora en un sistema de información erróneamente diseñado, normalmente no hará sino incrementar los errores y fallas existentes en forma

exponencial. Lo que debe hacerse primero es llevar a cabo un análisis formal y apropiado del trabajo de sistemas, incorporando la computadora y otros medios semejantes sólo en el caso de que con ello se mejore la efectividad del sistema. Dicho de otro modo, debe pensarse primero en el sistema de información o, como en nuestro caso, el proceso a automatizar, y después en los dispositivos y equipo necesarios para ponerlo en marcha.

En esta sección se analizará el proceso que actualmente se sigue en el Laboratorio de Mecánica para la adquisición y despliegue de la información, la cual corresponde a algunas de las variables físicas que más interesan e intervienen al momento de llevar a cabo la laminación de materiales. Esto se hace con el fin de ver si es o no necesario el implantar un sistema computarizado, ya que puede darse el caso de que al hacer un simple cambio en el manejo de la información sea más que suficiente para mejorar y corregir las deficiencias que padezca el proceso, ésto es a lo que se llama alternativa 0.

Actualmente la laminadora sólo cuenta con dos sistemas independientes de adquisición y despliegue de información, como se ha mencionado en el capítulo 1 al momento de definir el problema. Uno de estos sistemas es para la temperatura del material a laminar y otro para la carga ejercida en los rodillos al momento de laminar, dichos sistemas se encuentran integrados, cada uno, por un sensor, que también sirve como transductor; y una unidad de acondicionamiento, digitalización y despliegue de información, con una pequeña pantalla digital y algunos controles para especificar condiciones e intervalos de funcionamiento.

Para poder observar el comportamiento de las otras tres variables (velocidad, par y espesor) se pensó en adquirir 3 sistemas similares a los dos que actualmente se encuentran conectados a la laminadora, uno para cada variable en específico. Esta decisión es discutible por dos sencillas razones: la primera se relaciona con la cuestión costo, ya que cada uno de estos sistemas tiene como principal característica un elevado precio en dólares; y la segunda, es que tendríamos cinco unidades de acondicionamiento, digitalización y despliegue de información, y en consecuencia, cinco pantallas que observar, posiblemente a un mismo tiempo.

Con ayuda de una computadora tipo PC, podemos observar en un sólo monitor, a colores y con una resolución mayor a la que maneja cualquiera de los sistemas por separado, el comportamiento de cada una de las variables que intervienen en el proceso de laminación. Con ésto podríamos solucionar el problema de las cinco pantallas. En cuanto al costo, éste disminuiría notablemente, ya que en el peor de los casos sólo se tendrían que adquirir los sensores para medir cada variable, y en caso de ser necesario, un monitor de video VGA (a color o monocromático) con su respectiva tarjeta controladora de video, que en un momento dado se puede conectar a cualquier CPU que como mínimo tenga un microprocesador 286. El costo de todo ésto,

comparado con el de adquirir tres sistemas sigue siendo menor, sin tomar en cuenta que algunos sensores pueden ser fabricados y que la medición de algunas variables puede hacerse indirectamente. Este punto se analiza en otros capítulos del presente trabajo.

Una vez analizando y observando el manejo que se requiere de la información, se llegó a la conclusión de que es necesario implantar un sistema de manejo de información computarizado, entendiendo por manejo la adquisición y despliegue de la misma. Por otro lado, el sistema que se va a implantar también va a ser de gran ayuda en cuanto a las opciones de impresión, almacenamiento y consulta posterior de la información se refiere.

En definitiva, las computadoras y la tecnología con ellas asociada pueden ser de gran ayuda si son utilizadas adecuadamente, y analizadas de acuerdo con la relación costo/beneficio, en muchos casos pueden acrecentar en gran medida la efectividad de los sistemas de información sobre todo cuando el volumen, la complejidad y el cálculo son elevados.

6.2.4. DISEÑO DEL SOFTWARE.

El diseño del software es un proceso creativo que requiere del diseñador ciertas cualidades ya que el diseño final suele ser una repetición de varios diseños preliminares. El diseño debe practicarse y aprenderse mediante la experiencia y el estudio de sistemas ya existentes. Un buen diseño es la clave de una Ingeniería de Software efectiva, ya que un sistema bien diseñado es fácil de aplicar y mantener, además de ser comprensible y confiable.

Dada una definición de requisitos, el diseñador debe utilizarla para desarrollar el diseño de un sistema de programación que satisfaga esos requisitos. Esto se realiza en varias etapas:

1. Deben establecerse los subsistemas que componen el sistema de programación.
2. Cada sistema debe dividirse en componentes individuales y ha de establecerse la especificación de los subsistemas definiendo la operación de esos componentes.
3. Después, cada programa se puede diseñar a base de subcomponentes que actúen recíprocamente.
4. Se tiene que refinar cada componente como una jerarquía de subcomponentes.

Además de estas etapas, el programador tiene que diseñar las estructuras de los archivos así como las estructuras de datos que se utilicen en sus programas. También se deberán diseñar las estructuras necesarias para contemplar los cambios anticipados debidos a la evolución del hardware, cambios en las necesidades del usuario, etc.; como es el caso del diseño del sistema con visiones futuristas, es decir, crear el sistema con posibilidad de funcionar interactivamente con actuadores que permitan automatizar completamente el proceso de laminación.

Tomando en cuenta los requerimientos del sistema, el hardware a usar, el modelo conceptual planteado, los requisitos funcionales y los requisitos no funcionales, tenemos en la figura 10 un diseño estructurado con una visión de alto nivel del sistema de software a desarrollar.

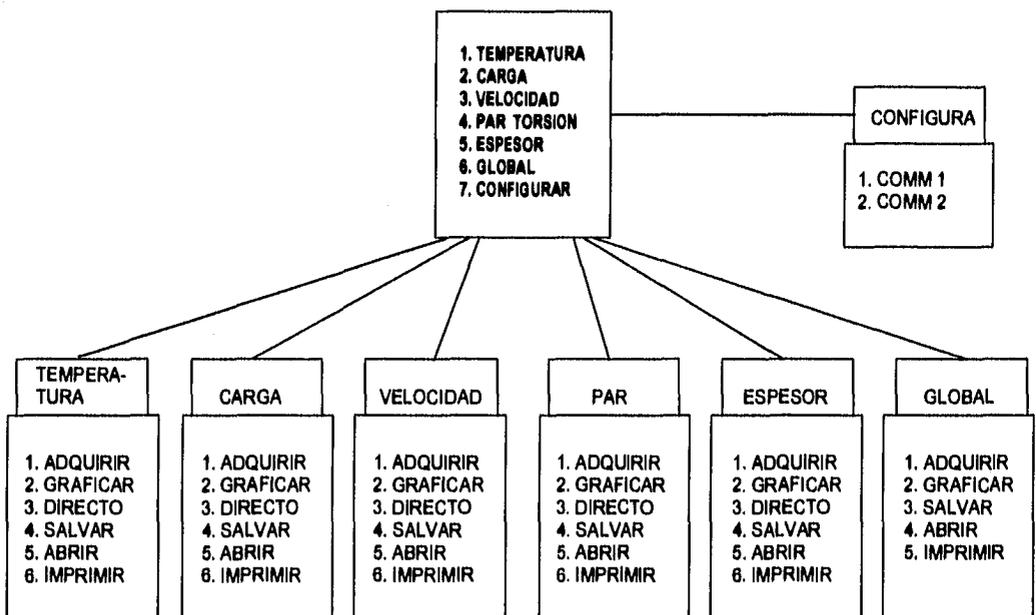


Figura 10. Diseño estructurado del sistema (visión de alto nivel)

Basándonos en esta visión de alto nivel podemos notar que nuestro sistema se puede dividir en tres módulos principales, aunque aparentemente sean siete. Esto se debe a que las opciones existentes (y los procesos ligados a dichas opciones) que nos permiten observar el comportamiento de una sola variable, cualquiera que ésta sea, son exactamente las mismas, por lo que en lugar de tener un módulo particular para cada variable, podemos tener un módulo general que maneje las 6 opciones y que en base al paso de parámetros que nos especifique qué variable se pretende observar, nos sirva para las cinco variables.

El módulo dos es el que nos sirve para adquirir datos correspondientes a las cinco variables, a un mismo tiempo, para la graficación posterior de cada una por separado. La diferencia principal entre este módulo y el que nos sirve para observar el comportamiento de una sola variable radica en que en este último existe una opción llamada Directo, que permite a un mismo tiempo adquirir y graficar los datos correspondientes a la variable.

Por último se tiene el módulo en donde se llevan a cabo los procesos que nos permiten especificar al sistema por cuál de los puertos seriales se va a llevar a cabo la comunicación con el microcontrolador del sistema.

Cada módulo se encuentra ligado con una serie de programas y subprogramas (que en el entorno de la programación de Turbo C se representan por funciones o procedimientos) que hacen funcionar cada una de las seis opciones que conforman este módulo y que, como consecuencia, nos permiten observar el comportamiento de las variables. En la figura 11 se puede observar:

- 1) En dónde se ubica dicho módulo dentro del sistema de software.
- 2) Cuáles submódulos conforman dicho módulo.
- 3) Qué tareas se llevan a cabo en dichos submódulos.
- 4) De cuáles funciones principales (programadas en Turbo C) se valen para realizar dichas tareas.

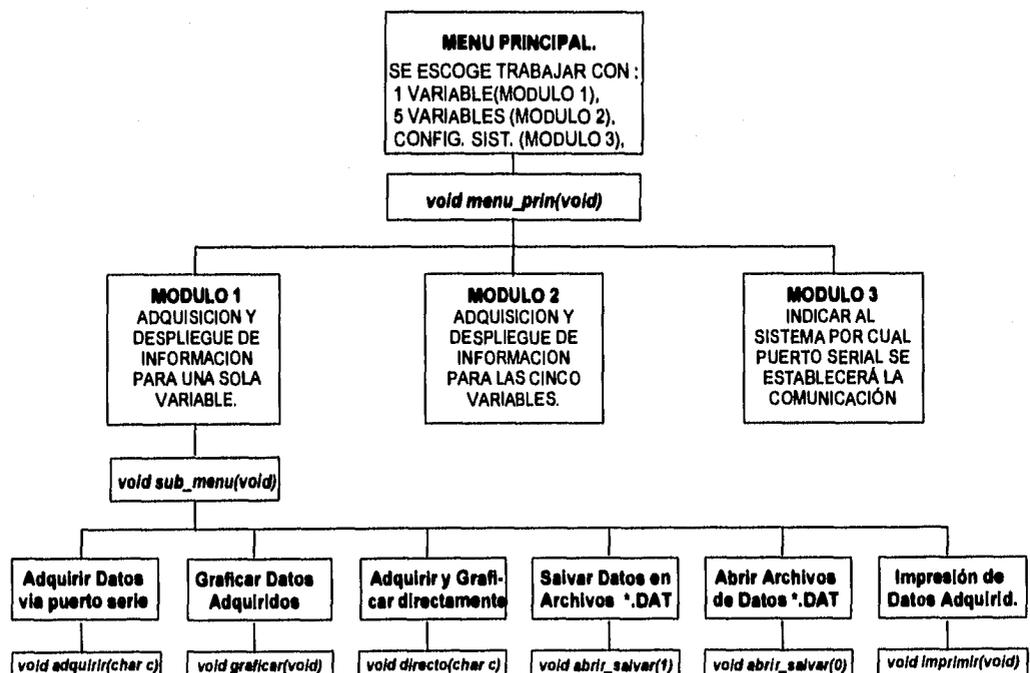


Figura 11. Componentes Módulo 1 (Adquisición y Despliegue de información 1 variable) del Sistema de Software

El módulo 2 se encuentra formado de una manera muy similar, como se podrá apreciar en la figura 12.

Incluso, el funcionamiento de este módulo es muy similar al del módulo 1, pero con dos diferencias esenciales: La primera se relaciona con la inexistencia en el módulo 2 de una opción de despliegue directo; y la segunda con la manipulación de la información, ya que el módulo debe manipular datos de las 5 variables a un mismo tiempo sin perder la coherencia y el origen de cada dato.

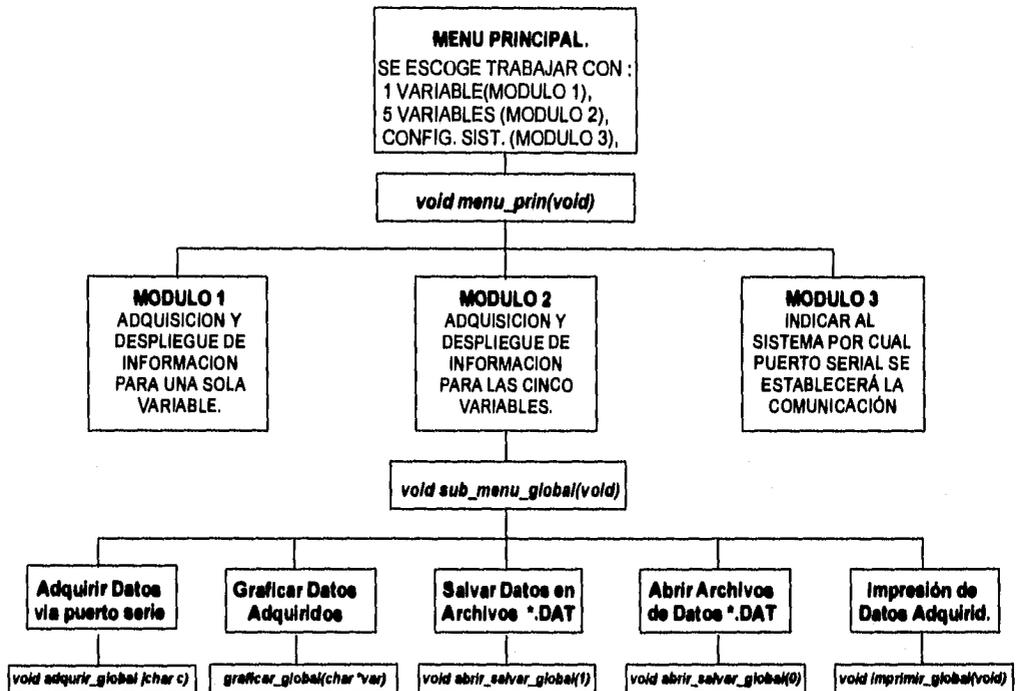


Figura 12. Componentes Módulo 2 (Adquisición y Despliegue de Información 5 variables) del Software.

El último módulo es el que se refiere a la configuración del sistema en lo que a comunicación vía puerto serie se refiere, es decir, mediante las funciones asociadas a este módulo podemos escoger por cuál puerto serie se quiere entablar la comunicación, independientemente de que sea conector tipo DB9 o DB25. En la figura 13 se muestran los componentes de dicho módulo y su ubicación dentro del sistema.

Además de las funciones principales que se mencionan en las figuras, deben existir otra serie de funciones que, a pesar de que no se mencionan, no dejan de tener importancia dentro del funcionamiento del Sistema de Software. Estas funciones pueden ser, por ejemplo, las que se utilizan para la presentación de: pantallas, ventanas para los submenús o mensajes de advertencia. También se tienen las funciones que

sirven como auxiliares en el proceso de graficación como: trazado de líneas, ejes coordenados, puntos, etc; o para el proceso de impresión: imprimir encabezado, imprimir número, imprimir letrero, etc.

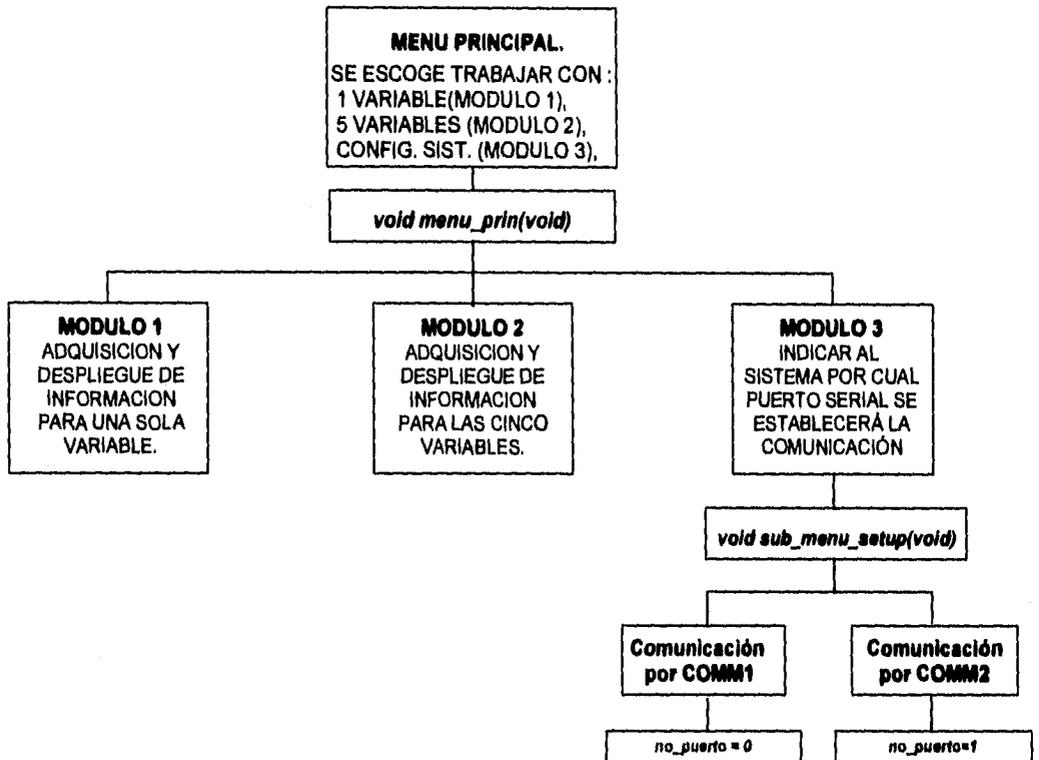


Figura 13. Componentes Módulo 3 (Configuración del puerto serie) del Sistema de Software.

6.2.5. PRUEBAS.

La prueba consiste en ejecutar el programa utilizando datos similares a los datos reales que habrán de ser procesados por el programa, observar los resultados y deducir la existencia de errores o insuficiencias del programa a partir de las anomalías de ese resultado.

A menudo se piensa que la prueba y depuración de programas es una misma cosa, aunque están muy relacionadas en realidad son procesos distintos; ya que la prueba es el proceso de establecer la existencia de errores en el programa y la depuración es el proceso de localizar donde se produjeron dichos errores y corregir el código incorrecto. Es muy importante comprender que la prueba nunca demuestra que un programa es correcto ya que pueden existir errores aún después de la prueba más completa. La prueba de programas sólo puede demostrar la presencia de errores en un programa, no su ausencia.

La prueba de un programa es un proceso destructivo. Se diseña para hacer que el comportamiento de un programa sea distinto al que esperaba su diseñador.

La depuración es el proceso de identificar las áreas del programa que causan errores y modificarlas para corregir el error. El proceso de depuración implica dos etapas. Primero, se deben localizar aquellas partes incorrectas del código del programa y, segundo, el programa se debe modificar de manera que cumpla con sus requisitos. Después de esto, debe repetirse la prueba del programa para garantizar que la modificación se ha hecho de manera correcta.

PROCESO DE PRUEBAS Y DEPURACION DEL SISTEMA DE SOFTWARE

A continuación se presenta el proceso de prueba y depuración correspondiente al sistema, dicho proceso se presenta listando algunos de los errores principales encontrados así como la depuración necesaria para corregir éstos.

El proceso de prueba del sistema fue dividido en dos etapas, la primera de ellas fue realizada por el diseñador del sistema, ya que se consideró conveniente entregar el sistema a los usuarios finales sin tener el problema de que se presenten insuficiencias del mismo; en la segunda etapa la prueba del sistema se realizó con personas que desconocieran totalmente el sistema y con los usuarios finales, pero que de preferencia tuvieran los conocimientos más esenciales sobre el manejo de computadoras y sistemas de software; esto fue con la finalidad de observar con mayor facilidad aquellos errores inherentes al diseñador del sistema. Este tipo de prueba es muy conveniente, ya que el diseñador conoce el sistema y sabe cuáles son los pasos a seguir exactamente para el manejo del mismo; en cambio un usuario sin experiencia alguna en el manejo del sistema y que nunca lo hubiese visto funcionar, puede cometer errores que para el diseñador nunca pudiesen ocurrir por lo que conlleva a que el sistema falle irremediablemente, es por esta razón que los sistemas deben ser probados por usuarios sin experiencia y por los usuarios finales antes de poner en función cualquier sistema.

Durante la primera etapa algunos de los errores encontrados fueron los siguientes:

- Al momento de adquirir datos, tanto para una sola variable como para las cinco, si existía un error de comunicación, como puede ser la falta del cable entre el microcontrolador y el puerto serial de la computadora o simplemente la interrupción de la comunicación, el sistema fallaba totalmente,

conduciendo al usuario inmediatamente al ambiente de sistema operativo. Esta falla se pudo solucionar introduciendo una función de revisión del estado del puerto serie, la cual al momento de detectar un error en la comunicación, muestra al usuario un mensaje en el que indica que la comunicación se ha perdido y en consecuencia ya no puede continuar la adquisición de datos; después presionando otra tecla se regresa al submenú correspondiente para reiniciar la adquisición de datos o graficar los ya adquiridos.

- En la opción de graficación, tanto al trabajar con una sola variable como con las cinco, si antes no se habían adquirido datos con la opción monitoreo, el sistema presentaba nuevamente una falla total, dejando al usuario del sistema en el ambiente de sistema operativo, con una serie de mensajes que indicaban a qué se debía el error y sin restaurar en pantalla el modo de texto. Esta falla se corrigió de manera semejante a la anterior, se introdujo una función que verificara que los archivos donde se guardan los datos adquiridos no estuvieran vacíos y en caso de que lo estuvieran, solamente desplegara un mensaje que lo indicara, y que le permitiera al usuario regresar al submenú correspondiente a adquirir los datos correspondientes.
- En la opción de impresión, si la impresora se encontraba fuera de línea o había un error en la comunicación a través del puerto paralelo (LPT1), el sistema no indicaba al usuario porqué no se habían impreso los datos, solamente se salía de la opción al submenú correspondiente. Para solucionar este problema se agregó una función que en caso de que la impresora y la comunicación estén correctas, indica al usuario que los datos se están imprimiendo y que espere un lapso de tiempo pertinente, dependiendo de la impresora con que se esté trabajando; en caso contrario, se despliega un mensaje que indica al usuario revisar la conexión o la impresora ya que no hay respuesta de la misma a los datos enviados a impresión.
- Al imprimir los reportes de los datos adquiridos, existía un corrimiento en la impresión de los encabezados, y en consecuencia de los datos, debido a que el número de líneas que se manejan en una hoja tamaño carta no se ajustaba al número de líneas que estaba manejando el sistema. Este error se corrigió fácilmente igualando el número de líneas que estaba manejando el sistema con la que se manejan en una hoja tamaño carta en modo de impresión normal.
- Por tratarse de una adquisición de datos en tiempo real y debido a la velocidad de transmisión que se manejará en el puerto serie (2400 bits/seg.), en un intervalo corto de tiempo tendremos en el monitor de la computadora una cantidad considerablemente grande de datos. El problema surge al graficar estos datos

ya que nunca se tomó en cuenta que debido a la resolución que se manejará en el monitor, solamente podremos graficar un máximo de 600 datos en el eje X de la gráfica. Para solucionar este problema se tomaron dos medidas. La primera consiste en aplicar un factor de escalación en el eje X; para ésto el usuario deberá escoger en un submenú de la opción Graficar, el factor de escalación que él crea conveniente para ajustar la gráfica dependiendo del número de datos adquiridos. La segunda medida consiste en realizar una especie de muestreo con los datos adquiridos. Esto nos llevará a que no se desplieguen en el monitor de la computadora todos los datos transmitidos por el microcontrolador sino que de conjuntos de n datos transmitidos solamente tomaremos uno, así si definimos grupos de 10 datos podremos reducir diez veces el número de datos a desplegar. Como consecuencia de este muestreo se perderán algunos datos pero aún así, debido a la velocidad de transmisión, los datos que lleguen a ser desplegados representan realmente el comportamiento de la o las variables que maneja el sistema.

- Como a cada uno de los datos enteros que transmite el microcontrolador se les debe aplicar la ecuación del convertidor A/D, todos estos datos se vuelven reales. La finalidad de esta ecuación es la de indicarnos cuál es el valor analógico que nos genera el valor digital. Una vez que se tiene este valor analógico, le debemos aplicar la ecuación que defina el comportamiento de la variable para la cual se están adquiriendo los datos. El problema surgió cuando se quisieron desplegar en el monitor los datos reales ya que la función de despliegue de datos era solo para datos enteros, por lo que hubo que cambiar el formato de despliegue para datos reales

En la segunda etapa como se mencionó anteriormente, se realizó con usuarios que desconocieran totalmente el funcionamiento, y en consecuencia, el manejo del sistema, y como se esperaba se obtuvieron algunos otros errores, siendo algunos de los más representativos los siguientes:

- En el rectángulo de color brillante que se encuentra rodeando las opciones del menú principal y submenús y que nos sirve como barra de desplazamiento para escoger las opciones de los mismos, notamos que cuando el usuario lo desplazaba en sentido horizontal en el menú principal para escoger alguna opción del mismo, no existían límites, por lo cual se salía dicho cuadro de la barra de menú e incluso de la pantalla; por otro lado en los submenús, sucedía lo mismo pero en sentido vertical, saliéndose nuevamente el cuadro para escoger las opciones de los límites del submenú e incluso de la pantalla dejando a lo largo de la misma lo que se conoce como "basura". Para corregir esta falla de presentación, se agregaron límites en forma de coordenadas, en los procedimientos que se encargan de

manejar el menú principal y sus correspondientes submenús evitando así que nuestra barra de desplazamiento se saliera de su barra de menú correspondiente o de la pantalla.

- El menú principal y sus submenús además de tener la barra de desplazamiento, tienen la posibilidad de que si escogemos la primera letra de la opción, que se encuentra en color rojo y con mayúsculas, podemos entrar directamente a dicha opción. El error consistía en que al escoger la opción con esta segunda posibilidad, la barra de desplazamiento se quedaba estática sin moverse a la opción que habíamos escogido con la letra, pudiéndose dar el caso de estar imprimiendo, por ejemplo, y la barra se encontraba señalado la opción de monitoreo, o en el caso del menú principal, podíamos estar trabajando con la variable carga y la barra se encontraba señalando temperatura. Este problema se corrigió agregando unas cuantas líneas de código a los procedimientos que manejan el menú principal y sus correspondientes submenús, para que al recurrir a la segunda posibilidad para escoger opciones, se trazara nuestra barra de desplazamiento en la opción escogida y así siempre saber con cuál variable u opción estamos trabajando.
- En la opción de guardar los datos adquiridos en algún archivo, nos olvidamos de modificar el formato para que los datos almacenados fueran reales, por lo tanto al desear el usuario consultar alguno de estos archivos, lo único que se desplegaba en el monitor de la computadora eran símbolos incoherentes. Este problema se solucionó fácilmente modificando las instrucciones que se encargan de guardar los datos en los archivos de un formato de número entero a uno de número real.
- Como consecuencia de lo anterior, si el usuario deseaba imprimir los datos adquiridos, lo único que se obtenía en la impresora eran números enteros sin ninguna interpretación coherente. Este problema se solucionó de manera semejante que el anterior; se modificaron las instrucciones que convertían los números enteros en cadenas de caracteres imprimibles a instrucciones que hicieran exactamente lo mismo pero con números de tipo real.
- Por último, dentro de la opción de abrir archivos con datos adquiridos con anterioridad, había un error que consistía en que si el nombre del archivo que se quería abrir no existía, el sistema se quedaba en un ciclo de ejecución abriendo y creando un archivo inexistente; la única forma de salir de dicho ciclo era romper intencionalmente la ejecución del sistema. Esta falla se solucionó agregando una función que verifica si el archivo que se desea abrir existe realmente en el subdirectorio de trabajo del sistema, si existe abre dicho

archivo y carga los datos para su observación, en caso contrario despliega un mensaje que indica al usuario que el archivo no existe y lo regresa al submenú correspondiente.

Para todos estos errores detectados por el usuario se hicieron las correcciones necesarias en la parte del código correspondiente según del error que se tratara, las correcciones a hacer se explicaron a groso modo en cada uno de los errores que se mencionan en esta etapa de pruebas. Una vez hechas las modificaciones necesarias se concluye la parte de pruebas y depuración del sistema.

6.2.6. DOCUMENTACION (DICCIONARIO DE DATOS).

El diccionario de datos es un documento que proporciona detalles de todas y cada una de las entidades relevantes para el sistema que se describe. Los diccionarios de datos son valiosos para los usuarios del sistema como para los que participan en el mantenimiento del mismo. El diccionario de datos consta de las siguientes partes:

- a) **Descripción.** Corresponde al nombre del campo y preferentemente debe representar una descripción del contenido del mismo.
- b) **Tipo.** En esta sección se definen los tipos de campos, los cuales pueden ser de tipo: caracter, numérico, tipo fecha, lógico, calculado y de tipo texto.
- c) **Longitud.** Dentro de esta parte se dice cuál es la longitud o tamaño del campo en cuestión, aunque algunos campos son de longitud fija, tales como los de tipo fecha que su longitud es de 8 ó los de tipo lógico que su longitud es de 1.
- d) **Decimales.** En esta sección se especifica, en el caso de que el campo sea numérico, la cantidad de decimales del número definido.

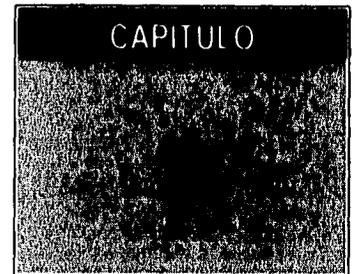
A continuación se presenta el diccionario de datos del sistema conteniendo las entidades más relevantes del mismo, es decir, las variables más frecuentemente utilizadas en el desarrollo de los programas fuentes. Se menciona el nombre, el tipo y longitud, así como una breve descripción de la función que realiza cada entidad.

Nombre	Tipo de Dato	Longitud (bits)	Descripción
controlador	numérico entero	16	Define el tipo de controlador de video a utilizar
modo_graf	numérico entero	16	Define el modo de video a utilizar por el controlador
maxx	numérico entero	16	El máximo valor que se puede tener en el eje X
maxy	numérico entero	16	El máximo valor que se puede tener en el eje Y
n_dato	numérico entero	16	Número de datos adquiridos para 1 o 5 variables
color_marco	numérico entero	16	Color de marcos en pantalla (gris claro)
color_fondo	numérico entero	16	Color principal de fondo en pantalla (azul)
color_fondo1	numérico entero	16	Color secundario de fondo en pantalla (azul claro)
color_fondo2	numérico entero	16	Color secundario de fondo en pantalla (cyan)
color_letra	numérico entero	16	Primer color usado para trazar letras (cyan claro)
color_letra1	numérico entero	16	Segundo color usado para trazar letras (negro)
color_letra2	numérico entero	16	Tercer color usado para resaltar letras (rojo)
color_gráfica	numérico entero	16	Color de fondo para resaltar gráficas (verde claro)
color_dato	numérico entero	16	Color correspondiente al número de dato (amarillo)
color_valor	numérico entero	16	Color correspondiente al valor adquirido (blanco)
num_temp	numérico entero	16	Nº datos adquiridos para var. temperatura (global)
num_carga	numérico entero	16	Nº datos adquiridos para var. carga (global)
num_vel	numérico entero	16	Nº datos adquiridos para var. velocidad (global)
num_par	numérico entero	16	Nº datos adquiridos para var. par (global)
num_esp	numérico entero	16	Nº datos adquiridos para var. espesor (global)
variable	puntero caracter	suficiente	Puntero para guardar el nombre de la variable
no_puerto	numérico entero	16	Especifica con qué puerto serie trabajará el sistema
driver	puntero caracter	suficiente	Puntero para el nombre del controlador de video
mode	puntero caracter	suficiente	Puntero que guarda nombre del modo de video
colores	numérico entero	16	Nº de colores que puede usar el controlador video
error_graf	numérico entero	16	Indicador de error al inicializar el sistema en modo de video gráfico.
inf_video	struct viewporttype		Estructura definida por los campos left, top, right, bottom (esquina sup. izq. e inf. derecha)
inf_palet	struct palettetype		Estructura definida por los campos que definen el número de paleta a usar y los colores de la misma
x1	numérico entero	16	Esquina sup. izq. barra de desplazamiento (eje X)
y1	numérico entero	16	Esquina sup. izq. barra de desplazamiento (eje Y)

Nombre	Tipo de Dato	Longitud (bits)	Descripción
x2	numérico entero	16	Esquina inf. der. barra de desplazamiento (eje X)
y2	numérico entero	16	Esquina inf. der. barra de desplazamiento (eje Y)
xs1m	numérico entero	16	Esquina sup. izq. submenú (eje X)
ys1m	numérico entero	16	Esquina sup. izq. submenú (eje Y)
xs2m	numérico entero	16	Esquina inf. der. submenú (eje X)
ys2m	numérico entero	16	Esquina inf. der. submenú (eje Y)
nombre	puntero caracter	suficiente	Nombre del archivo para guardar datos (global)
var	puntero caracter	suficiente	Nombre de la variable con que se trabaja (global)
b10	numérico entero	16	Dato leído por el puerto serie y desplegado en pantalla
ft	puntero archivo	suficiente	Puntero con el nombre del archivo que guardará los datos adquiridos al trabajar con una variable. Al trabajar con 5 variables guarda los datos adquiridos para la variable temperatura
fc	puntero archivo	suficiente	Puntero con el nombre del archivo que guardará los datos adquiridos para la variable carga
fv	puntero archivo	suficiente	Puntero con el nombre del archivo que guardará los datos adquiridos para la variable velocidad
fa	puntero archivo	suficiente	Puntero con el nombre del archivo que guardará los datos adquiridos para la variable par
fe	puntero archivo	suficiente	Puntero con el nombre del archivo que guardará los datos adquiridos para la variable espesor
fglob	puntero archivo	suficiente	Puntero con el nombre del archivo que guardará los datos adquiridos para las 5 variables
maximo	numérico real	32	Representa el dato que tenga el valor mayor de todos los que se van a graficar
minimo	numérico real	32	Representa el dato que tenga el valor menor de todos los que se van a graficar
intervalo_x	numérico entero	16	Es el valor con el cual se normaliza el eje X de graficación
intervalo_y	numérico real	32	Es el valor con el cual se normalizan en el eje Y los datos a graficación.

Nombre	Tipo de Dato	Longitud (bits)	Descripción
nuevo_x	numérico entero	16	Representa la siguiente posición en el eje X sobre la cual se graficará el siguiente valor.
nuevo_y	numérico real	32	Representa la siguiente posición en el eje Y sobre la cual se graficará el siguiente valor.
v_analog	numérico real	32	Es el valor que se obtiene al aplicarle al valor entero adquirido en el puerto serie, la ecuación del convertidor A/D.
valor_real	numérico real	32	Es el valor que se obtiene al aplicarle a v_analog la ecuación de comportamiento de cada una de las variables.
sumatoria	numérico real	32	Variable auxiliar que se utiliza para escalar en el eje X la gráfica de datos adquiridos.
promedio	numérico real	32	Variable auxiliar que se utiliza para escalar en el eje X la gráfica de datos adquiridos.
factor_escalay	numérico real	32	Variable global utilizada para escalar en el eje Y la gráfica de datos adquiridos.

PRUEBAS Y MEDICIONES



En este capítulo nos enfocaremos primero, a las pruebas y mediciones hechas por separado a cada uno de los componentes del módulo encargado del acondicionamiento de señales, posteriormente, a las pruebas efectuadas a la arquitectura basada en un microcontrolador, y por último a las pruebas hechas al Sistema de Adquisición y Despliegue de Información como un todo, es decir, acondicionadores, arquitectura y software para PC interconectados entre sí e integrados como sistema a la laminadora. En lo que se refiere al tercer módulo, constituido esencialmente por el software para la computadora tipo PC, queda excluido en éste capítulo debido a que en el capítulo 6 dicho software fue sometido a un exhaustivo proceso de pruebas encaminadas a poner al descubierto todas sus posibles fallas.

Una prueba consistirá en ejercitar los acondicionadores, la arquitectura o el Sistema de Adquisición y Despliegue de Información, según sea el caso, bajo condiciones reales de operación de la laminadora, siendo el objetivo principal observar los resultados, tomar mediciones, en el caso de los acondicionadores, y deducir la existencia de errores o insuficiencias. Dicha prueba deberá ser un proceso destructivo, el cual se diseña para hacer que el comportamiento del módulo o sistema probado sea distinto al que se esperaba.

En lo que a mediciones se refiere, sabemos que están jugando un papel importante en la sustentación de las leyes de la ciencia y que son esenciales para el estudio, desarrollo y control de muchos dispositivos y procesos. Sin embargo, el proceso de medición involucra algunos pasos a seguir antes de lograr un conjunto útil de información. Estos pasos son los siguientes:

1. Diseño de un conjunto eficiente de medición. Este paso incluye una apropiada selección del equipo disponible y una correcta interconexión de los diferentes componentes e instrumentos.
2. Operación inteligente de los aparatos de medición.
3. Registro de los datos en una forma clara y completa. La información registrada no debe presentar ambigüedad para futuras interpretaciones.
4. Estimación de la precisión de las mediciones y magnitud de los posibles errores causados.

5. Preparación de un informe el cual describe la medición y sus resultados para aquellos quienes pueden estar interesados en usarlos.

En el desarrollo del Sistema de Adquisición y Despliegue de Información el papel de las mediciones hechas a cada una de las variables cuando la laminadora se encontraba trabajando en condiciones reales de operación es fundamental, debido principalmente a que gracias a ellas pudimos obtener en forma de gráficas y ecuaciones matemáticas, el comportamiento de cada una de las variables que manejará el sistema. El análisis de las gráficas de comportamiento para cada una de las variables es lo que nos dió la pauta para escoger y diseñar los acondicionadores de señales más adecuados, los cuales se encargarían de efectuar las modificaciones pertinentes a las señales eléctricas. Por otra parte, gracias a las ecuaciones matemáticas que definen el comportamiento de cada una de las variables, se pudo dar una interpretación real en la computadora a los datos digitales transmitidos por el microcontrolador.

Cabe mencionar que en el caso del módulo encargado del acondicionamiento, los circuitos electrónicos encaminados a acondicionar las señales de cada una de las variables fueron implementados como prototipos y en consecuencia las pruebas y mediciones obtenidas en el capítulo 4 son de dichos prototipos. El objetivo es llevar dichos prototipos a una implementación real en una tableta fenólica con todos los componentes perfectamente acomodados, soldados e interconectados. Así tenemos que en esta parte del trabajo, las pruebas y mediciones serán realizadas con los circuitos reales implementados en tableta, siendo el objetivo principal verificar que el funcionamiento de los acondicionadores de dicha tableta sea igual al de los prototipos, utilizando como elementos de comparación, las mediciones del capítulo 4 y las que se obtendrán en el presente capítulo.

7. 1 PRUEBAS A LOS ACONDICIONADORES DE SEÑALES.

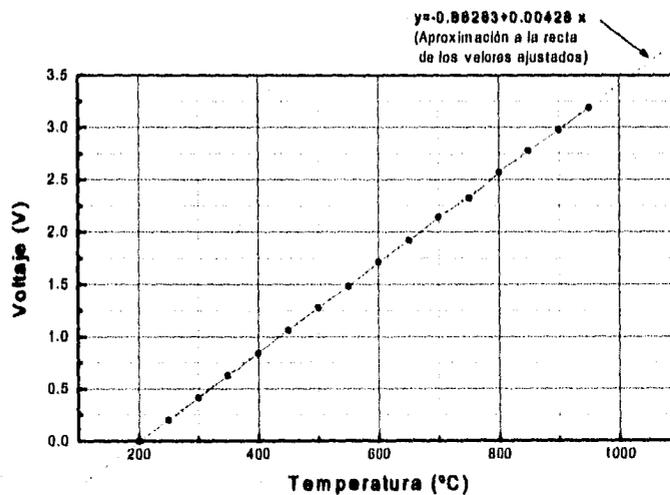
7.1.1 VARIABLE TEMPERATURA.

En este caso, para llevar a cabo las pruebas, primero se procedió a conectar el circuito acondicionador para la variable temperatura (figura 5 del capítulo 4) en la salida analógica de voltaje del sistema de medición correspondiente; posteriormente se colocó en la laminadora una placa de material de fierro, previamente calentada en un horno a una temperatura de 1000°C , de tal forma que el calor emitido por dicha placa fuera directamente captado por el transductor del sistema, el cuál se encuentra representado por un pirómetro de radiación. El objetivo de esta prueba es, además de obtener mediciones, verificar el comportamiento del

interruptor electrónico, el cuál debe permanecer cerrado cuando los voltajes ajustados por el circuito restador se encuentren en el intervalo de voltajes permisibles por el convertidor A/D del microcontrolador y abierto cuando se exceda el máximo valor permisible por dicho convertidor. Así fue como se obtuvieron las siguientes tabla y gráfica.

TEMPERATURA (°C)	VOLTAJE AJUSTADO	TEMPERATURA (°C)	VOLTAJE AJUSTADO
200	0	600	1.714
250	0.19975	650	1.9195
300	0.41225	700	2.1435
350	0.62725	750	2.3265
400	0.84325	800	2.5725
450	1.06425	850	2.777
550	1.281	900	2.977
550	1.484	950	3.187

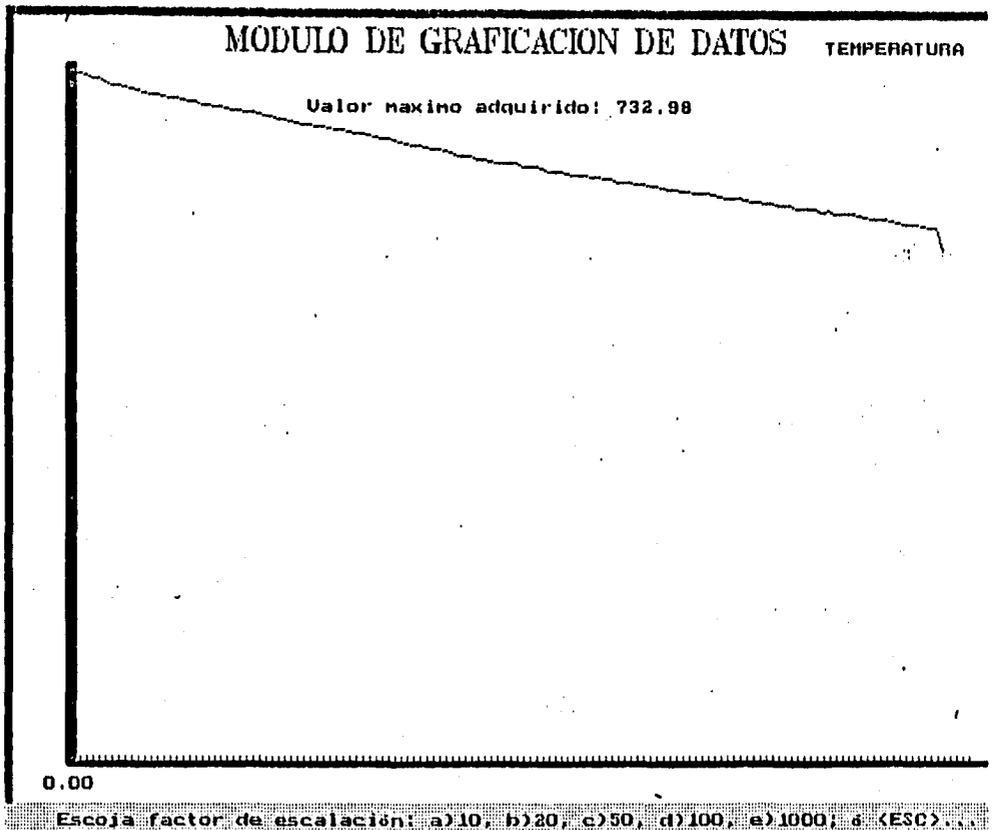
Tabla 1. Valores de Voltaje obtenidos a la salida del transductor para la variable temperatura.



Gráfica 1. Representación gráfica de los valores de voltaje obtenidos a la salida del transductor.

Es evidente que los valores obtenidos en el acondicionador prototipo del capítulo 4 son iguales a los obtenidos con el acondicionador implementado en forma real en la tableta fenólica, por lo tanto ésta etapa de la prueba se concluye satisfactoriamente.

Ahora, con ayuda de la ecuación de la recta de aproximación de la gráfica 1, de la ecuación del convertidor A/D del microcontrolador y del software del sistema, se puede obtener una representación gráfica real, en el monitor de la computadora, del comportamiento de la variable temperatura; dicho comportamiento puede observarse a continuación en la gráfica 2.



Gráfica 2. Representación gráfica del comportamiento de la variable temperatura obtenida por el software del Sistema con base en los datos reales adquiridos.

El comportamiento de esta variable no es difícil de predecir ya que conforme el material se va enfriando, disminuye el valor de la temperatura del mismo, de cualquier forma esto se pudo confirmar auxiliándonos del sistema de medición para la variable temperatura, en donde se pudo apreciar claramente como disminuye paulatinamente el valor de dicha variable. Por lo tanto, la representación gráfica obtenida en el monitor de la computadora, nos muestra claramente el comportamiento real de la variable temperatura.

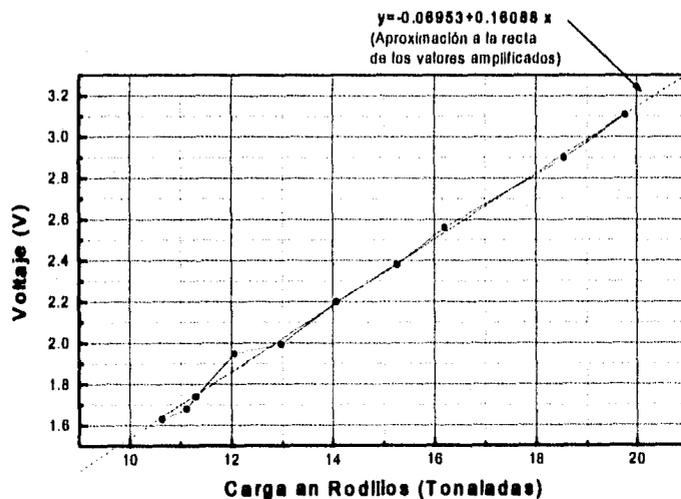
7.1.2 VARIABLE CARGA.

En el caso de esta variable, la cuál también cuenta con un sistema de medición completo, el circuito electrónico que representa al acondicionador de señales (Figura 7, capítulo 4) se conectó en la salida de la

unidad sumadora y promediadora de cargas del sistema. El objetivo principal que se persigue con ésta prueba, es comprobar que el comportamiento del circuito prototipo sea igual al del circuito que se implementó en la tableta, esto bajo condiciones reales de operación de la laminadora, es decir, al momento de laminar un material aplicando la respectiva carga en los rodillos. La forma de comprobar consiste en tomar mediciones y verificar que éstas sean iguales a las obtenidas en el capítulo 4 cuando se procedió a probar el circuito prototipo del acondicionador. A continuación se muestran las mediciones obtenidas, así como la representación gráfica de las mismas.

Carga ejercida sobre el material a laminar (Toneladas)	Voltaje medido a la salida del amplif. conectado a la unidad sumadora (V)	Carga ejercida sobre el material a laminar (Toneladas)	Voltaje medido a la salida del amplif. conectado a la unidad sumadora (V)
10.641	1.63	14.072	2.20
11.12	1.68	15.252	2.38
11.297	1.74	16.200	2.56
12.058	1.95	18.571	2.90
12.971	1.993	19.774	3.11

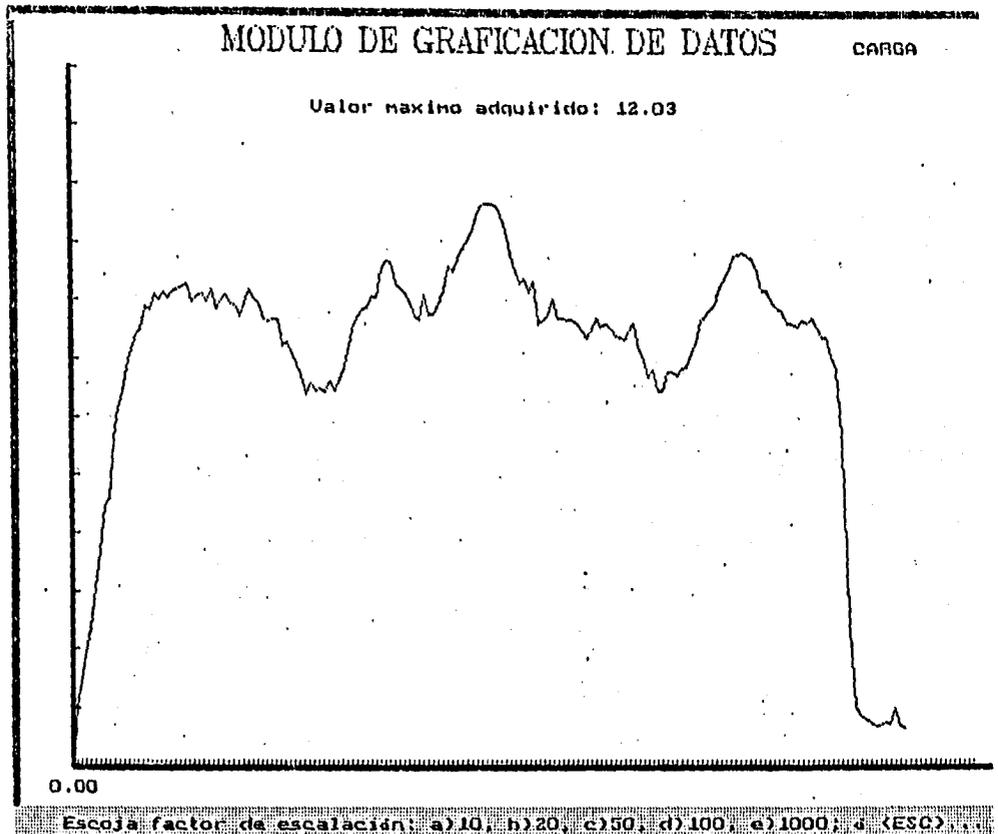
Tabla 2. Valores de voltaje medidos a la salida del acondicionador de señales para la variable carga.



Gráfica 3. Representación gráfica de los valores de voltaje obtenidos a la salida del transductor.

Podemos observar que las mediciones obtenidas con el circuito implementado en la tableta son iguales a las obtenidas con el prototipo, por lo tanto sólo resta verificar que la gráfica obtenida en la computadora con

los datos adquiridos y con ayuda de la ecuación de la recta de aproximación de la gráfica 3, sea una representación real del comportamiento de la variable carga al momento de laminar algún material. A continuación se muestra la gráfica obtenida en el monitor de la computadora.



Gráfica 4. Representación gráfica del comportamiento de la variable carga obtenida por el software del Sistema con base en los datos reales adquiridos.

La gráfica anterior nos muestra con detalle el comportamiento real de los valores de la carga ejercida por los rodillos sobre el material al momento de laminarlo. Las variaciones que se aprecian en dicha gráfica se deben a las formas irregulares del material, lo cual produce que en algunos puntos sea mayor la carga ejercida sobre el material que en otros. De cualquier forma, para verificar la validez de los datos adquiridos así como de la gráfica derivada de los mismos, nos auxiliamos de la unidad de despliegue de datos del sistema de medición para la variable carga, en donde se pudo apreciar que efectivamente los valores de la carga ejercida sobre el material muestran una notable variación dependiendo de que tan irregular es el material a laminar.

7.1.3 VARIABLE VELOCIDAD ANGULAR.

En el caso de esta variable el circuito que se utilizó para polarizar el sensor de efecto Hall, así como el filtro que por recomendación del fabricante debe colocarse para evitar distorsiones en la señal de salida, representan el circuito acondicionador de señales para el transductor (sensor de efecto Hall TL170C) que se utilizará para poder adquirir datos relativos al valor de la velocidad angular al momento de laminar algún material.

La señal proporcionada por el acondicionador de señales se encuentra formada por una serie de pulsos, espaciados uno del otro un diferente intervalo de tiempo; dichos pulsos entrarán al microcontrolador a través del temporizador previamente configurado en su función de captura de entrada, esto con la finalidad de medir el tiempo transcurrido entre dos pulsos y así posteriormente calcular el valor de la velocidad angular en ese preciso instante. Se debe tener especial cuidado en esto, ya que la velocidad angular varía dependiendo de las condiciones en que se encuentre operando la laminadora y en consecuencia el tiempo transcurrido entre dos pulsos puede variar en cualquier momento. Debido a lo anterior se desarrolló una rutina de programación que fuera capaz de contar directamente el número de pulsos de reloj transcurridos entre dos pulsos de la señal proporcionada por el acondicionador. Posteriormente dicho número de pulsos de reloj es transmitido en forma de dato a la computadora en donde con unas sencillas ecuaciones se procede a calcular el valor real de la velocidad angular en el momento preciso.

Antes de proceder a probar el funcionamiento del acondicionador para la variable velocidad angular con ayuda del software y la arquitectura del sistema, fue necesario probar el acondicionador con ayuda de un osciloscopio y simulando la flecha de la laminadora, pero ahora implementando dicho acondicionador permanentemente en la tableta fenólica donde se encuentran los demás acondicionadores. Para esto se colocaron los imanes a lo largo de una regla metálica (simulando la flecha de la laminadora en donde serían colocados), de tal forma que al pasar los imanes sobre el sensor, el acondicionador nos debe proporcionar una señal de salida que al ser visualizada en la pantalla del osciloscopio, se encuentra constituida por una serie de pulsos con un espaciamiento siempre distinto entre ellos. Esto último dependerá de la velocidad a la que pasen los imanes sobre el sensor.

Una vez concluida con éxito esta prueba, se procedió a conectar la salida del acondicionador al temporizador integrado del microcontrolador y éste último a la computadora, de tal forma que ahora colocando los imanes alrededor de la flecha principal de la laminadora, pueda observarse en el monitor de la computadora el comportamiento real de la variable velocidad angular. Vale la pena mencionar que dicho

comportamiento es constante siempre y cuando la laminadora se encuentre trabajando en vacío y a una velocidad periférica fija, ésto último se puede lograr apoyándonos en el control e indicador para la velocidad periférica que se encuentran en el panel principal de control de la laminadora. En cuanto se procede a laminar algún material, la velocidad angular tiende a variar dependiendo principalmente de la carga ejercida por los rodillos y de que tan irregular sea la constitución física del material a laminar.

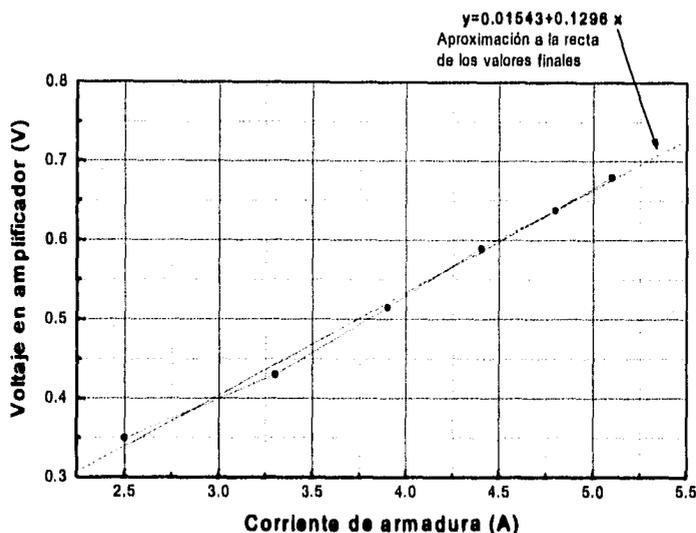
7.1.4 VARIABLE PAR DE LAMINACION.

Como se explicó con anterioridad, para conocer el valor de la variable par de laminación al momento de laminar algún material, nos auxiliaremos de las variables velocidad angular y corriente de armadura, las cuales nos ayudarán a calcular algunos parámetros eléctricos que finalmente nos proporcionen el valor del par de laminación. La variable velocidad angular no es problema ya que forma parte del conjunto de variables que manejará el Sistema de Adquisición y Despliegue de Información, en lo que a la variable corriente de armadura se refiere, se diseñó un acondicionador que fuera capaz de efectuar las modificaciones pertinentes a la señal proporcionada por un amperímetro de gancho, ésto para que al final tuviéramos una señal de voltaje compatible con el convertidor A/D del microcontrolador.

Lo primero que haremos será comprobar que el funcionamiento del acondicionador prototipo para la variable corriente de armadura sea igual al del implementado en forma fija en la tableta fénica. Para esto se deberán hacer mediciones y gráficas que representen el comportamiento de la variable, las cuales deben ser iguales a las obtenidas en el capítulo 4. A continuación se muestran las mediciones de voltaje obtenidas en la salida del acondicionador, así como la gráfica derivada de las mismas, en función de la corriente de armadura medida en el motor de la laminadora.

Corriente de Armadura (A)	Voltaje a la salida del acondicionador (V)
2.5	0.350
3.3	0.430
3.9	0.515
4.4	0.590
4.8	0.638
5.1	0.680

Tabla 3. Valores de voltaje medidos a la salida del acondicionador de señales para la variable carga.



Gráfica 5. Representación gráfica de los valores de voltaje obtenidos a la salida del transductor.

Como podemos apreciar, las mediciones obtenidas para la variable corriente de armadura utilizando el circuito acondicionador prototipo del capítulo 4 son iguales a las obtenidas con el circuito acondicionador real, por lo tanto solo resta verificar que el despliegue de datos y gráficas en el monitor de la computadora sea una representación real del comportamiento de la variable par de laminación. Para esto debemos emplear en el software del sistema las ecuaciones desarrolladas en el apartado 3.5 del capítulo 3, la ecuación de la recta de aproximación de la gráfica 5, así como los valores que el sistema haya adquirido para las variables velocidad angular y corriente de armadura, obteniendo como resultado valores numéricos reales de la variable par de laminación, los cuales nos ayudarán a obtener una representación gráfica del comportamiento de la variable.

Esta variable presenta un comportamiento constante siempre y cuando la laminadora se encuentre trabajando en vacío y a una velocidad angular constante, esto se debe a que bajo estas condiciones de funcionamiento el valor de la corriente de armadura siempre es el mismo y en consecuencia el valor del par de laminación resulta constante. Las variaciones más importantes se presentan cuando aumenta la velocidad angular y principalmente cuando se introduce el material a laminar y los rodillos comienzan a ejercer carga sobre él, ya que como consecuencia de esto la corriente eléctrica demandada para aumentar la potencia del motor es mayor y como consecuencia el valor del par de laminación siempre tiende a aumentar.

7. 2 PRUEBAS A LA ARQUITECTURA DEL SISTEMA BASADA EN UN MICROCONTROLADOR MC68HC811E2

La etapa de pruebas correspondiente a la Arquitectura del sistema será dividida en tres partes principales, cada una de las cuales se enfocará a verificar un aspecto diferente relacionado con el funcionamiento de la Arquitectura:

1. Continuidad de Conexiones.
2. Funcionamiento Básico.
3. Funcionamiento de Periféricos a Utilizar.

7.2.1 CONTINUIDAD EN CONEXIONES.

El objetivo de esta prueba es verificar que todas las conexiones hechas tanto al microcontrolador como a los circuitos auxiliares de la Arquitectura del sistema estén realizadas correctamente. Al decir esto, nos referimos a verificar la inexistencia tanto de fallas en la continuidad del circuito, ocasionadas por la falta de algunas conexiones o por falsos contactos en las mismas, como de cortos circuitos, ocasionados generalmente por conexiones equivocadas.

Las fallas en la continuidad de las conexiones generalmente derivan en un incorrecto funcionamiento de la arquitectura, mientras que los cortos circuitos pueden ocasionar desde un funcionamiento anormal de la arquitectura, hasta un daño irreparable en el microcontrolador y circuitos auxiliares que lo acompañan.

Gracias a estas pruebas se pudo detectar un pequeño problema en las conexiones del sistema encargado de reinicializar el funcionamiento del microcontrolador (reset), antes de proceder a realizar las pruebas de funcionamiento básico con toda la Arquitectura.

7.2.2 FUNCIONAMIENTO BASICO.

Para llevar a cabo el proceso de pruebas relativas al funcionamiento básico de la arquitectura, nos apoyamos en un par de sencillos programas grabados previamente en el microcontrolador.

El primer programa consistió en colocar un "1" lógico a través de uno de los bits del puerto C del microcontrolador, obteniéndose de esta manera en una de las líneas del puerto C una señal de salida de 5 volts, la cual tendría como función principal activar un diodo emisor de luz (led) permanentemente. Este programa sobresale por su simpleza, sin embargo nos sirvió enormemente para verificar el correcto funcionamiento del microcontrolador, al interpretar correctamente las instrucciones grabadas en él, y para verificar que por lo menos el puerto C se encontraba funcionando correctamente.

El segundo programa es una ampliación del primero y nos sirvió para reafirmar el correcto funcionamiento de la Arquitectura y principalmente del microcontrolador. La función principal de este programa fue hacer que el led ahora prendiera y apagara, permaneciendo en cada estado aproximadamente 1 segundo. Lo anterior se obtuvo creando, por medio de algunas instrucciones, una señal alterna fundamentada en los niveles lógicos "1" y "0" y en una subrutina de retardo de 1 segundo.

El resultado de estas pruebas fue satisfactorio, por lo que el siguiente paso a seguir es programar y probar individualmente cada uno de los periféricos, tanto de la Arquitectura como del microcontrolador, que serán utilizados en el Sistema de Adquisición y Despliegue de Información.

7.2.3 FUNCIONAMIENTO DE LOS PERIFERICOS A UTILIZAR.

Esta etapa se enfocó principalmente a la realización de pruebas con los periféricos del microcontrolador a utilizar, siendo dichos periféricos los siguientes: la interfaz de comunicación serial, el convertidor Analógico - Digital y el temporizador en su función de captura de entrada.

Para probar la **Interfaz de comunicación serial (SCI)** fue necesario utilizar dos programas; el primero de ellos consistía en transmitir un bloque de 256 datos (números del 0 al 255 generados internamente en el microcontrolador) a través del registro de datos de comunicación. Dichos datos fueron transmitidos en forma serial a la PC, en donde con ayuda del software desarrollado para el sistema, se desplegaron y graficaron en el monitor.

En el segundo programa se procedió a probar las dos funciones del SCI, la transmisión y la recepción. Este programa es un complemento del primero ya que se transmite el mismo bloque de datos pero no sin antes haber recibido de la computadora un dato, es decir, al microcontrolador debe recibir un dato específico generado y enviado por el software de la PC para consecuentemente enviar como respuesta el mismo bloque de 256 datos.

El **convertidor Analógico - Digital** integrado en el microcontrolador se probó con ayuda de un programa previamente grabado en el microcontrolador, un generador de funciones, un osciloscopio y el software para PC desarrollado para el Sistema de Adquisición. A continuación se explica el procedimiento que se siguió para probar el convertidor.

La señal proveniente del generador de funciones fue básicamente una señal con un voltaje menor o igual a 5 volts en la cual hubo que tener mucho cuidado al manejar el offset del generador ya que si la señal de voltaje que entrara por alguno de los canales del convertidor fuera negativo, este periférico se dañaría irremediablemente; para evitar esto nos fue de gran ayuda el osciloscopio ya que así se pudo analizar la señal generada antes de ser transmitida. La forma de la señal generada realmente no tuvo gran importancia.

La función del programa con el cual se probó la Arquitectura fue configurar el convertidor analógico - digital del microcontrolador para hacer conversiones continuas por un solo canal. Una vez hecha la conversión de la señal, los datos digitales generados fueron transmitidos a través del SCI en forma serial a la computadora en donde finalmente lo que se debía visualizar en el monitor al graficar los datos transmitidos, era la misma gráfica del generador de funciones observada en el osciloscopio.

El último periférico en ser probado fue el **temporizador en su función de captura de entrada**, para esto se desarrolló un programa que debía medir el tiempo transcurrido entre dos pulsos provenientes de un sensor de campos magnéticos, esto con el fin de poder determinar la velocidad angular en los rodillos de laminación.

La medición del tiempo mencionado anteriormente fue sin duda la que más inconvenientes presentó. En un principio se trató de hacer por medio del contador de eventos del temporizador, pero finalmente se vio la conveniencia de utilizar la función de captura de entrada, para lo cual hubo que usar algunas interrupciones en tiempo real.

En este caso se midió el tiempo transcurrido entre dos pulsos, quedando almacenado en un registro como el número de períodos de reloj del microcontrolador transcurridos. Finalmente dicho número de pulsos se transmitió en forma serial a la computadora con el fin de desplegar los datos y gráficas en el monitor.

Adicionalmente a todo lo anterior, se realizó una última prueba con los periféricos antes de conectar la Arquitectura a los acondicionadores de señales. Esta prueba consistió en que ahora con un solo programa y

3 generadores de funciones y el sensor de campos magnéticos se simulara un comportamiento global del sistema, es decir, se adquirieran datos para todas y cada una de las variables que manejará el sistema.

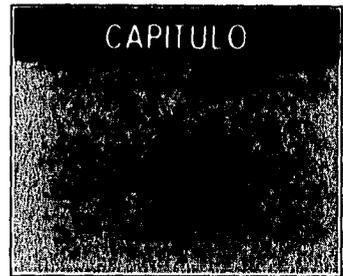
7.3 PRUEBA FINAL AL SISTEMA DE ADQUISICION Y DESPLIEGUE DE INFORMACION.

Esta prueba se enfocó principalmente a observar el comportamiento del sistema como un todo, es decir transductores, acondicionadores, arquitectura y software para computadora integrados, todo esto cuando la laminadora se encontraba trabajando bajo condiciones reales de temperatura, carga y velocidad angular. Así mismo fueron los usuarios finales quienes se encargaron de efectuar esta prueba final, poniendo al descubierto pequeñas fallas que inmediatamente fueron corregidas. A continuación se enlistan dichas fallas:

- Si el usuario se encontraba adquiriendo datos para una cierta variable y en un momento dado deseaba cambiar de variable, se hacía necesario reinicializar manualmente la Arquitectura con el botón de RESET para que ahora el microcontrolador cambiara ya sea de canal del convertidor A/D o al temporizador. Este problema se corrigió fácilmente agregando algunas instrucciones adicionales en el programa del microcontrolador para que automáticamente se reinicializara al terminar de adquirir datos para una cierta variable.
- Al momento de adquirir datos relativos a la variable corriente de armadura, se nos hizo la observación de que el valor de dicha variable oscilaba demasiado al calcular el valor del par de laminación. Esto se debía a que los diodos colocados en el circuito encargado de rectificar la señal proveniente del amperímetro de gancho tenían una respuesta demasiado lenta. Para solucionar este problema se cambiaron dichos diodos por unos de alta velocidad de respuesta.
- El último problema fue detectado en el modulo encargado del acondicionamiento de las señales, en específico en el correspondiente a la variable carga. Para esta variable pudimos notar que existía un sobrecalentamiento en los circuitos integrados, así como una obstrucción en el paso de la señal proveniente de la unidad sumadora y promediadora de cargas, esto repercutía al desplegar los datos en la computadora, ya que lo único que se podía observar al graficar los datos adquiridos era una señal ruidosa. Esto se debía a que la tableta se montó en una base de acrílico con unos soportes de aluminio, los cuales cortocircuitaban dos terminales del primer amplificador produciendo el sobrecalentamiento y como consecuencia la obstrucción en el paso de la señal. Este problema se solucionó cambiando de lugar el soporte de aluminio y sustituyendo el amplificador.

Los tres problemas mencionados anteriormente fueron los más relevantes en esta última etapa de pruebas, aunque cabe mencionar que se detectaron unos pequeños desajustes en los valores desplegados en pantalla en comparación con los valores desplegados por los sistemas de medición para las variables carga y temperatura. Estos desajustes se corrigieron rápidamente cambiando los parámetros que definen a las ecuaciones de comportamiento empleadas en el software del sistema para desplegar y graficar el valor real de los datos digitales transmitidos.

CONCLUSIONES



En la actualidad el perfeccionamiento de la técnica de integración de circuitos a muy grande escala (VLSI) ha permitido la fabricación de microprocesadores y microcontroladores muy baratos y poderosos, con ésto, las microcomputadoras están al alcance de cualquier proyecto de control siendo posible además realizar algoritmos más elaborados. Aunado a lo anterior tenemos los grandes adelantos que también ha alcanzado la electrónica analógica en lo que a desarrollo de transductores se refiere, lo cual permite que el comportamiento de prácticamente cualquier variable física pueda ser interpretada con ayuda de una señal eléctrica. Así tenemos que, gracias a los adelantos y tecnologías anteriormente mencionados, dentro del campo del control automático, los sistemas enfocados a la adquisición de datos pueden ser aplicados a cualquier proceso, ya sea industrial, biológico, agrícola, etc., siempre y cuando se haga un correcto análisis del proceso y sobre todo un óptimo diseño y selección de los componentes tanto analógicos como digitales que integrarán el sistema a desarrollar.

Es así como se decidió que para implementar un sistema de adquisición de datos, capaz de agilizar de manera significativa la medición de las variables Temperatura, Carga, Velocidad Angular y Par de Laminación, que intervienen directamente en el procesos de laminación experimental con ayuda de la laminadora Hille 25, debíamos tener, aparte de los sistemas de medición adquiridos con anterioridad para las variables Temperatura y Carga, tres módulos principales, el primero encargado del acondicionamiento de las señales eléctricas provenientes de los transductores colocados en la laminadora; el segundo (constituido por la arquitectura basada en un microcontrolador) encargado de convertir las señales analógicas a señales digitales, controlar el flujo de datos y llevar a cabo la transmisión de los mismos hacia el tercer módulo, el cual consiste en una computadora tipo PC y un software específicamente diseñado para desplegar, almacenar e imprimir los datos de las variables antes mencionadas.

El módulo encargado del acondicionamiento de señales constituye la parte analógica del sistema de adquisición de datos desarrollado y fue sin lugar a duda el modulo más difícil tanto de diseñar como de implementar. Esto de debió principalmente a que el comportamiento de los acondicionadores al ser probados

en el laboratorio bajo condiciones ideales de operación, distaba mucho de ser igual al obtenido cuando se conectaban dichos acondicionadores a los transductores de la laminadora y ésta última se hacía funcionar. En consecuencia se tuvieron que hacer una gran cantidad de pruebas con numerosos arreglos de amplificadores, comparadores, filtros, rectificadores, interruptores y seguidores de voltaje antes de poder asegurar que el funcionamiento de un acondicionador era el correcto.

En la actualidad la gran mayoría de los sistemas de adquisición de datos basan su funcionamiento en uno de los circuitos especiales de más versatilidad que existen en la actualidad, el microcontrolador. Esto se debe principalmente a que con ayuda de los periféricos que tiene integrados puede llevar a cabo una cantidad considerable de funciones como son la transmisión de datos en forma serial y paralela, la conversión de señales analógicas a digitales, la medición de intervalos de tiempo tanto largos como muy cortos, controlar el flujo de datos, etc. El sistema de adquisición desarrollado para la laminadora Hille 25 no constituye la excepción ya que también utiliza un microcontrolador, en este caso de la familia M68HC11, el cual se encarga entre otras cosas de convertir las señales analógicas acondicionadas en señales digitales, además controlar el flujo de datos para cada una de las variables y transmitir los datos a la computadora.

En lo que respecta al tercer módulo podemos afirmar que en el caso del sistema desarrollado para la laminadora Hille 25, la computadora PC con el software adecuado ayudará enormemente a llevar a cabo la labor de despliegue, almacenamiento, consulta e impresión de los datos adquiridos para cada una de las variables, permitiendo observar el comportamiento tanto de una variable como de todas a un mismo tiempo.

Es importante señalar que no todos los sistemas de adquisición de datos aplicados al proceso de laminación, ya sea industrial o experimental, necesitan de una computadora y su tecnología, que la computadora no necesariamente mejora un sistema y que no todos los reportes que pueda emitir se pueden calificar como información. El hecho de automatizar el proceso de laminación con ayuda de un sistema de adquisición de datos, no necesariamente acrecentará su efectividad, ya que si se realiza la instalación de una computadora en un sistema erróneamente diseñado, ocasionará que se incrementen las fallas y errores en forma exponencial. Dicho de otro modo, debe pensarse primero en el proceso y en el sistema a desarrollar como un todo y después en los dispositivos y equipo necesarios para poner en marcha dicho sistema de adquisición de datos, el cual debe adecuarse a las necesidades, mejorando claro está la efectividad y rapidez de la laminación.

Cabe mencionar que se tiene latente la posibilidad de integrar una quinta variable al sistema. Esta quinta variable es la que se definió al principio como Espesor; el motivo por el cual tuvo que quedar fuera del

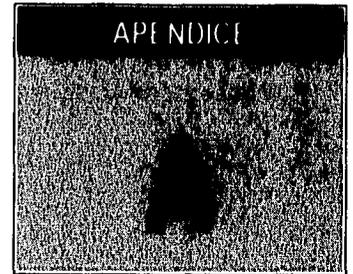
sistema obedeció a que los transductores capaces de medir espesor sin tener contacto con el material son extremadamente caros y el diseño e implementación de alguno es tan complejo que puede ser tema de otro trabajo. A pesar de todo esto, la programación del microcontrolador y el sistema de software para computadora PC consideran esta variable para que cuando se adquiera o desarrolle el transductor correspondiente solamente se diseñe, en caso de ser necesario, el acondicionador adecuado que se integre al módulo de acondicionadores de señales del Sistema de Adquisición y Despliegue de Información.

A continuación se enumeran solamente algunas de las ventajas que se pueden obtener al automatizar con ayuda del Sistema de Adquisición y Despliegue de Información el proceso de laminación que se realiza en forma experimental con ayuda de la laminadora Hille 25.

- La tecnología digital tiene un bajo costo.
- El consumo de potencia es bajo.
- El uso de señales digitales codificadas presentan las ventajas de que pueden ser almacenadas por un tiempo indefinido además de que pueden transmitidas con mayor confiabilidad mediante el uso de códigos de protección.
- Se logra un mejor funcionamiento empleando tecnología digital que con tecnología analógica.

Por último podemos mencionar que queda abierta la posibilidad de que otros procesos, tanto industriales como experimentales, también enfocados a la manufactura de metales puedan aprovechar las ventajas de la implantación de un sistema de adquisición de datos computarizado que permita observar y/o controlar de manera eficiente la o las variables que intervienen directamente en el proceso.

PROGRAMA FUENTE DEL SISTEMA PARA COMPUTADORA PC



```

#include <conio.h>
#include <math.h>
#include <stdio.h>
#include <bios.h>
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#include <dos.h>
#include <dir.h>
#include <time.h>
#include <graphics.h>

#define ESC 0x1b
#define ENTER 0xd
#define DERECHA 0x4d
#define IZQUIERDA 0x4b

/* VARIABLES GLOBALES DEL SISTEMA */
int controlador;
int modo_graf;
int maxx,maxy,n_datos;
float factor_escalay;
int color_marco,color_fondo,color_letra,color_letra1,color_letra2,color_fondo1,color_fondo2,color_grafica,color_datos,color_valor;
int num_temp,num_carga,num_vel,num_par,num_esp;
char *variable;
int no_puerto;
int muestreo;

/* DECLARACION DE LOS PROTOTIPOS DE FUNCIONES DEL SISTEMA */
void marco(int xinicial,int yinicial,int xfinal,int yfinal,int color);
void marco1(int xinicial,int yinicial,int xfinal,int yfinal,int color);
void borde(int xi,int yi,int xf,int yf);
void pantalla(int x,int y,int color_m,int color_f);
void imp_submenu(int xs1,int xs2,int xs1m,int ys1m,int xs2m,int ys2m,int color);
void imp_submenu_global(int x1,int x2,int x1m,int y1m,int x2m,int y2m,int color);
void imp_menu_prin(int x,int y);
void imp_submenu_setup(int x1,int x2,int x1m,int y1m,int x2m,int y2m,int color);
void inic_puerto(int n_puerto,unsigned char code);
lee_puerto(int n_puerto);
envia_puerto(int n_puerto,char c);
int gprintf(int *xloc,int *yloc,char *fmt, ... );
float mayor(FILE *f);
float menor(FILE *f);
void presenta(void);
void inicializa(void);
void menu_prin(void);
void sub_menu(int xs1,int xs2,int xs1m,int ys1m,int xs2m,int ys2m, char var);
void sub_menu_global(void);
void sub_menu_setup(void);
void imprimir();
void imprimir_global(void);
void imprime_numero(char cadena[5]);

```

```

void imprime_letrero(char *cadena);
void encabezado(int n_pagina);
void encabezado_global(int n_pagina);
void adquirir(char c);
void adquirir_global(char c);
void regresa_setup(int x1m,int y1b,int x2m,int y2b,int opcion);
void llama_graficas(int escalax);
graficar(int factor_escalax);
graficar_global(char *nombre,int numero,char *var,int factor_escalax);
void directo(char c);
void dibuja_ejes(float maximo,float minimo,int x0,int y0,int x1,int y1,int intervalo_x,float intervalo_y);
void dibuja_ejes_directo(float maximo,float minimo,int x0,int y0,int x1,int y1,float intervalo_yy);
void dibuja_ejes_global(float maximo,float minimo,int x0,int y0,int x1,int y1,int intervalo_x,float intervalo_y,int numero);
void abrir_salvar(int condicion);
void abrir_salvar_global(int condicion);
void mensaje(int color_fondo,int color_letra,char *msj);
void recuadro(int color_fondo,int color_marco,int color_letra,char *msj);
void finaliza(void);

```

```
/* MODULO PRINCIPAL DEL SISTEMA */
```

```

int main()
{
  Inicializa();
  presenta();
  menu_prin();
  restorecrtmode();
}

```

```
/*FUNCION PARA DETERMINAR LA CONFIGURACION DEL EQUIPO*/
```

```

void inicializa(void)
{
  union REGS r;
  struct viewporttype inf_video;
  struct linesettingstype inf_linea;
  struct fillsettingstype inf_fill;
  struct textsettingstype inf_texto;
  struct palettetype inf_palet;
  char *driver,*mode;
  int x,y;
  int colores;
  int error_graf;
  char *msj,*msj1,*msj2,*msj3;
  int xasp,yasp;
  unsigned eq;

  msj = "SISTEMA GRAFICO DISPONIBLE";
  controlador=DETECT;
  initgraph(&controlador,&modo_graf, " ");
  error_graf=graphresult();
  if (error_graf != grOk){
    printf(" Error en el sistema de Gráficos: %s\n", grapherrormsg(error_graf));
    exit(1);
  }
  muestreo=1;
  maxx=getmaxx();
  maxy=getmaxy();
  colores=getmaxcolor()+1;
  getviewsettings(&inf_video);
  getlinesettings(&inf_linea);
  getfillsettings(&inf_fill);
  gettextsettings(&inf_texto);
  getpalette(&inf_palet);
  color_marco=LIGHTGRAY;
  color_fondo=BLUE;

```

```

color_letra=LIGHTCYAN;
color_letra1=BLACK;
color_letra2=RED;
color_fondo1=LIGHTBLUE;
color_fondo2=CYAN;
color_grafica=LIGHTGREEN;
color_dato=YELLOW;
color_valor=WHITE;
x=142;
y=textheight(msj)+192;
driver=getdrivername();
mode=getmodename(modos_graf);
setlinestyle(SOLID_LINE,0,THICK_WIDTH);
marco(0,0,maxx,maxy,color_marco);
setfillstyle(SOLID_FILL,color_fondo);
floodfill(5,5,color_marco);
marco1(107,120,533,300,color_marco);
setfillstyle(SOLID_FILL,color_fondo1);
floodfill(128,160,color_marco);
setcolor(color_valor);
settextjustify(CENTER_TEXT, TOP_TEXT);
outtextxy(320,(textheight(msj)+160),msj);
setcolor(color_letra);
settextjustify(LEFT_TEXT,CENTER_TEXT);
gprintf(&x,&y,"Controlador de Graficos : %-15s (%d)",driver,controlador);
gprintf(&x,&y,"Modo actual de Graficos : %-15s (%d)",mode,modos_graf);
gprintf(&x,&y,"Resolucion de Pantalla : (0, 0, %d, %d)",getmaxx(),getmaxy());
gprintf(&x,&y,"No. Colores disponibles : %d",colores);
gprintf(&x,&y,"Color Actual      : %d",getcolor());
eq=blousequip();
eq>=1;
if(eq&1){
    setcolor(color_valor);
    settextjustify(CENTER_TEXT, TOP_TEXT);
    msj1="EXISTE UN COPROCESADOR MATEMATICO INSTALADO";
    msj2="EL SISTEMA FUNCIONARA CORRECTAMENTE";
    outtextxy(320,(textheight(msj1)+260),msj1);
    outtextxy(320,(textheight(msj2)+270),msj2);
}
else{
    setcolor(color_dato);
    settextjustify(CENTER_TEXT, TOP_TEXT);
    msj1="I M P O R T A N T E";
    msj2="NO EXISTE COPROCESADOR MATEMATICO INSTALADO";
    msj3="EL SISTEMA NO FUNCIONARA CORRECTAMENTE";
    outtextxy(320,(textheight(msj3)+250),msj1);
    outtextxy(320,(textheight(msj1)+260),msj2);
    outtextxy(320,(textheight(msj2)+270),msj3);
}
pausa(maxx,maxy);
}

/*PANTALLA DE PRESENTACION DEL SISTEMA*/
void presenta(void)
{
    struct viewporttype vp;
    char *msj1,*msj2,*msj3;

    msj1=" SOFTWARE DEL SISTEMA DE ADQUISICION Y ";
    msj2="DESPLIEGUE DE INFORMACION PARA LAMINADORA";
    msj3=" MARCA HILLE MODELO 25 TIPO EXPERIMENTAL ";
    clearviewport();
    getviewsettings(&vp);
    pantalla(maxx,maxy,color_marco,color_fondo);

```

```

marco1(34,107,604,291,color_marco);
setfillstyle(SOLID_FILL,color_fondo1);
floodfill(38,120,color_marco);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
setcolor(color_letra);
outtextxy(320,(textheight(msj1)+120),msj1);
outtextxy(320,(textheight(msj2)+160),msj2);
outtextxy(320,(textheight(msj3)+200),msj3);
pausa(maxx,maxy);
}

/*MENU PRINCIPAL DEL SISTEMA*/
void menu_prin(void)
{
union inkey {
char ch[2];
int i;
} c;
int x1,x2,y1,y2,x1m,x2m;
int xs1,xs2,xs1m,xs2m,ys1m,ys2m;

imp_menu_prin(maxx,maxy);
mensaje(color_marco,color_letra1,"Seleccione una opción del Menú Principal o <ESC> para salir...");
x1=14;
y1=14;
x2=107;
y2=31;
marco1(x1,y1,x2,y2,color_letra);
do {
while(!bioskey(1));
c.i=bioskey(0);
if(c.ch[0]){
switch(toupper(c.ch[0])) {
case 'r':{
if(x1==14){
xs1=23;
xs2=32;
xs1m=17;
ys1m=40;
xs2m=130;
ys2m=240;
variable="TEMPERATURA";
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'T');
break;
}
if(x1==101){
xs1=110;
xs2=118;
xs1m=102;
ys1m=40;
xs2m=215;
ys2m=240;
variable="CARGA";
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'C');
break;
}
if(x1==188){
xs1=195;
xs2=203;
xs1m=187;
ys1m=40;
xs2m=300;
ys2m=240;
variable="VELOCIDAD";
}
}
}
}

```

```

        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'V');
        break;
    }
    if(x1==275){
        xs1=280;
        xs2=288;
        xs1m=272;
        ys1m=40;
        xs2m=385;
        ys2m=240;
        variable="PAR";
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'P');
        break;
    }
    if(x1==362){
        xs1=365;
        xs2=373;
        xs1m=357;
        ys1m=40;
        xs2m=470;
        ys2m=240;
        variable="ESPESOR";
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'E');
        break;
    }
    if(x1==449){
        variable="GLOBAL";
        sub_menu_global();
        break;
    }
    if(x1==536){
        sub_menu_setup();
        break;
    }
}
case 'T': {
    marco1(x1,y1,x2,y2,color_marco);
    marco1(14,y1,107,y2,color_letra);
    variable="TEMPERATURA";
    xs1=23;
    xs2=32;
    xs1m=17;
    ys1m=40;
    xs2m=128;
    ys2m=240;
    sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'T');
    break;
}
case 'C': {
    marco1(x1,y1,x2,y2,color_marco);
    marco1(100,y1,194,y2,color_letra);
    variable="CARGA";
    xs1=110;
    xs2=118;
    xs1m=102;
    ys1m=40;
    xs2m=213;
    ys2m=240;
    sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'C');
    break;
}
case 'V': {
    marco1(x1,y1,x2,y2,color_marco);
    marco1(187,y1,281,y2,color_letra);

```

```

        variable="VELOCIDAD";
        xs1=195;
        xs2=203;
        xs1m=187;
        ys1m=40;
        xs2m=298;
        ys2m=240;
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'V');
        break;
    }
    case 'P': {
        marco1(x1,y1,x2,y2,color_marco);
        marco1(274,y1,368,y2,color_letra);
        variable="PAR";
        xs1=280;
        xs2=288;
        xs1m=272;
        ys1m=40;
        xs2m=383;
        ys2m=240;
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'P');
        break;
    }
    case 'E': {
        marco1(x1,y1,x2,y2,color_marco);
        marco1(361,y1,455,y2,color_letra);
        variable="ESPESOR";
        xs1=365;
        xs2=373;
        xs1m=357;
        ys1m=40;
        xs2m=468;
        ys2m=240;
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,'E');
        break;
    }
    case 'G': {
        marco1(x1,y1,x2,y2,color_marco);
        marco1(448,y1,542,y2,color_letra);
        variable="GLOBAL";
        sub_menu_global();
        break;
    }
    case 'S': {
        marco1(x1,y1,x2,y2,color_marco);
        marco1(535,y1,629,y2,color_letra);
        sub_menu_setup();
        break;
    }
    case ESC: {
        finaliza();
        closegraph();
        exit(1);
    }
}
else {
    switch(c.ch[1]) {
        case 77: {
            if(x1<=525){
                marco1(x1,y1,x2,y2,color_marco);
                x1=x1+87;
                x2=x2+87;
                marco1(x1,y1,x2,y2,color_letra);
            }
        }
    }
}

```

```

    }
    break;
}
case 75:{
    if(x1>=15){
        marco1(x1,y1,x2,y2,color_marco);
        x1=x1-88+1;
        x2=x2-88+1;
        marco1(x1,y1,x2,y2,color_letra);
    }
    break;
}
}
}
}while(c.ch[0]!='T' || c.ch[0]!='C' || c.ch[0]!='V' || c.ch[0]!='P' || c.ch[0]!='E' || c.ch[0]!='G' || c.ch[0]!='S' || c.ch[0]!='ESC');
}

/*SUBMENUS PARA CADA VARIABLE O GLOBAL*/
void sub_menu(int xs1,int xs2,int xs1m,int ys1m,int xs2m,int ys2m,char var)
{
    char regresa;
    union inkey {
        char ch[2];
        int i;
    } c;
    int y1b,y2b;

    imp_submenu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,color_fondo2);
    mensaje(color_fondo2,color_letra1,"Escoga una opción del Submenú o <ESC> para regresar al Menú Principal...");
    y1b=50;
    y2b=77;
    marco1(xs1m,y1b,xs2m,y2b,color_letra);
    do {
        while(!bioskey(1));
        c.=bioskey(0);
        if(c.ch[0]){
            switch(toupper(c.ch[0])) {
                case 'r':{
                    if(y1b==50){
                        adquirir(var);
                        do{
                            mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
                            regresa=getch();
                        }while(regresa!=ESC);
                        clearviewport();
                        imp_menu_prin(maxx,maxy);
                        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
                        break;
                    }
                }
                case 'e':{
                    if(y1b==76){
                        graficar(1);
                        do{
                            mensaje(color_letra,color_letra1,"Escoja factor de escalación: a)10, b)20, c)50, d)100; ó <ESC>...");
                            regresa=getch();
                            if(toupper(regresa)=='A') graficar(10);
                            if(toupper(regresa)=='B') graficar(20);
                            if(toupper(regresa)=='C') graficar(50);
                            if(toupper(regresa)=='D') graficar(100);
                        }while(regresa!=ESC);
                        clearviewport();
                        imp_menu_prin(maxx,maxy);
                        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
                        break;
                    }
                }
            }
        }
    }
}

```

```

if(y1b==102){
directo(var);
do{
mensaje(color_letra,color_letra1,"Presione <ESC> para regresar a submenú...");
regresa=getch();
}while(regresa!=ESC);
clearviewport();
Imp_menu_prin(maxx,maxy);
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
break;
}
if(y1b==128){
abrir_salvar(0);
initgraph(&controlador,&modo_graf, "");
Imp_menu_prin(maxx,maxy);
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
break;
}
if(y1b==154){
abrir_salvar(1);
initgraph(&controlador,&modo_graf, "");
Imp_menu_prin(maxx,maxy);
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
break;
}
if(y1b==180){
Imp_rimir();
do{
mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
regresa=getch();
}while(regresa!=ESC);
marco1(106,344,522,380,color_fondo);
setfillstyle(SOLID_FILL,color_fondo);
floodfill(160,350,color_fondo);
marco1(xs1m,y1b,xs2m,y2b,color_fondo2);
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
mensaje(color_fondo2,color_letra1,"Escoja una opción del submenú o <ESC> para regresar al menú principal...");
break;
}
}
case 'M': {
adquirir(var);
do{
mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
regresa=getch();
}while(regresa!=ESC);
clearviewport();
Imp_menu_prin(maxx,maxy);
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
break;
}
case 'G': {
graficar(1);
do{
mensaje(color_letra,color_letra1,"Escoja factor de escalación: a)10, b)20, c)50, d)100; ó <ESC>...");
regresa=getch();
if(toupper(regresa)=='A') graficar(10);
if(toupper(regresa)=='B') graficar(20);
if(toupper(regresa)=='C') graficar(50);
if(toupper(regresa)=='D') graficar(100);
}while(regresa!=ESC);
clearviewport();
Imp_menu_prin(maxx,maxy);
sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
}
}

```

```

        break;
    }
    case 'D': {
        directo(var);
        do{
            mensaje(color_letra,color_letra1,"Presione <ESC> para regresar a submenú...");
            regresa=getch();
        }while(regresa!=ESC);
        clearviewport();
        imp_menu_prin(maxx,maxy);
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
        break;
    }
    case 'A': {
        abrir_salvar(0);
        initgraph(&controlador,&modo_graf, "");
        imp_menu_prin(maxx,maxy);
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
        break;
    }
    case 'S': {
        abrir_salvar(1);
        initgraph(&controlador,&modo_graf, "");
        imp_menu_prin(maxx,maxy);
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
        break;
    }
    case '!': {
        marco1(xs1m,y1b,xs2m,y2b,color_fondo2);
        marco1(xs1m,180,xs2m,207,color_letra);
        imprimir();
        do{
            mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
            regresa=getch();
        }while(regresa!=ESC);
        marco1(xs1m,180,xs2m,207,color_fondo2);
        marco1(106,344,522,380,color_fondo);
        setfillstyle(SOLID_FILL,color_fondo);
        floodfill(160,350,color_fondo);
        marco1(xs1m,y1b,xs2m,y2b,color_fondo2);
        sub_menu(xs1,xs2,xs1m,ys1m,xs2m,ys2m,var);
        mensaje(color_fondo2,color_letra1,"Escoja una opción del submenú o <ESC> para regresar al menú principal...");
        break;
    }
    case ESC: {
        marco1(xs1m,ys1m,xs2m,ys2m,color_fondo);
        setfillstyle(SOLID_FILL,color_fondo);
        floodfill(xs1m+6,ys1m+6,color_fondo);
        if(var=='T') marco1(14,14,107,31,color_marco);
        if(var=='C') marco1(101,14,194,31,color_marco);
        if(var=='V') marco1(188,14,281,31,color_marco);
        if(var=='P') marco1(275,14,368,31,color_marco);
        if(var=='E') marco1(362,14,455,31,color_marco);
        menu_prin();
        mensaje(color_fondo,color_letra,"Seleccione una opción del Menú Principal o <ESC> para salir...");
        return;
    }
}
}
else {
    switch(c.ch[1]) {
        case 80:{
            if(y1b<=160){
                marco1(xs1m,y1b,xs2m,y2b,color_fondo2);
            }
        }
    }
}

```

```

        y1b=y1b+26;
        y2b=y2b+26;
        marco1(xs1m,y1b,xs2m,y2b,color_letra);
    }
    break;
}
case 72:{
    if(y1b>=60){
        marco1(xs1m,y1b,xs2m,y2b,color_fondo2);
        y1b=y1b+1-27;
        y2b=y2b+1-27;
        marco1(xs1m,y1b,xs2m,y2b,color_letra);
    }
    break;
}
}
} while(c.ch[0]!='M'||c.ch[0]!='G'||c.ch[0]!='A'||c.ch[0]!='D'||c.ch[0]!='S'||c.ch[0]!='I'||c.ch[0]!='ESC');
}

void sub_menu_global(void)
{
    char regresa;
    union lnkey {
        char ch[2];
        int i;
    } c;
    int x1,x2,x1m,y1m,x2m,y2m;
    int y1b,y2b,escala;
    char *nombre;
    char *var;

    x1=450;
    x2=458;
    x1m=442;
    y1m=40;
    x2m=555;
    y2m=200;
    Imp_submenu_global(x1,x2,x1m,y1m,x2m,y2m,color_fondo2);
    mensaje(color_fondo2,color_letra1,"Escoga una opción del Submenú o <ESC> para regresar al Menú Principal...");
    y1b=50;
    y2b=77;
    marco1(x1m,y1b,x2m,y2b,color_letra);
    do {
        while(!bioskey(1));
        c.i=bioskey(0);
        if(c.ch[0]){
            switch(tolupper(c.ch[0])) {
                case 'r': {
                    if(y1b==50) {
                        adquirir_global('G');
                        do{
                            mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
                            regresa=getch();
                        } while(regresa!='ESC');
                        clearviewport();
                        Imp_menu_prin(maxx,maxy);
                        sub_menu_global();
                        break;
                    }
                    if(y1b==76) {
                        llama_graficas(1);
                        do {
                            mensaje(color_letra,color_letra1,"Escoja factor de escalamiento: a)10, b)20, c)50, d)100; ó <ESC>...");

```

```

    regresa=getch();
    if(toupper(regresa)=='A') llama_graficas(10);
    if(toupper(regresa)=='B') llama_graficas(20);
    if(toupper(regresa)=='C') llama_graficas(50);
    if(toupper(regresa)=='D') llama_graficas(100);
} while(regresa!=ESC);
clearviewport();
imp_menu_prin(maxx,maxy);
sub_menu_global();
break;
}
if(y1b==102){
    abrir_salvar_global(0);
    initgraph(&controlador,&modo_graf,"");
    imp_menu_prin(maxx,maxy);
    sub_menu_global();
    break;
}
if(y1b==128){
    abrir_salvar_global(1);
    initgraph(&controlador,&modo_graf,"");
    imp_menu_prin(maxx,maxy);
    sub_menu_global();
    break;
}
if(y1b==154){
    imprimir_global();
    do{
        mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
        regresa=getch();
    }while(regresa!=ESC);
    marco1(106,344,522,380,color_fondo);
    setfillstyle(SOLID_FILL,color_fondo);
    floodfill(160,350,color_fondo);
    marco1(x1m,y1b,x2m,y2b,color_fondo2);
    sub_menu_global();
    mensaje(color_fondo2,color_letra1,"Escoja una opción del submenú o <ESC> para regresar al menú principal...");
    break;
}
}
case 'M':{
    adquirir_global('G');
    do {
        mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
        regresa=getch();
    } while(regresa!=ESC);
    clearviewport();
    imp_menu_prin(maxx,maxy);
    sub_menu_global();
    break;
}
case 'G':{
    llama_graficas(1);
    do {
        mensaje(color_letra,color_letra1,"Escoja factor de escalamiento: a)10, b)20, c)50, d)100; ó <ESC>...");
        regresa=getch();
        if(toupper(regresa)=='A') llama_graficas(10);
        if(toupper(regresa)=='B') llama_graficas(20);
        if(toupper(regresa)=='C') llama_graficas(50);
        if(toupper(regresa)=='D') llama_graficas(100);
    } while(regresa!=ESC);
    clearviewport();
    imp_menu_prin(maxx,maxy);
    sub_menu_global();
}

```

```

        break;
    }
    case 'A':{
        abrir_salvar_global(0);
        initgraph(&controlador,&modo_graf,"");
        imp_menu_prin(maxx,maxy);
        sub_menu_global();
        break;
    }
    case 'S':{
        abrir_salvar_global(1);
        initgraph(&controlador,&modo_graf,"");
        imp_menu_prin(maxx,maxy);
        sub_menu_global();
        break;
    }
    case 'I':{
        marco1(x1m,y1b,x2m,y2b,color_fondo2);
        marco1(x1m,154,x2m,181,color_letra);
        imprimir_global();
        do{
            mensaje(color_fondo1,color_letra,"Presione <ESC> para regresar a submenú...");
            regresa=getch();
        }while(regresa!=ESC);
        marco1(x1m,154,x2m,181,color_fondo2);
        marco1(106,344,522,380,color_fondo);
        setfillstyle(SOLID_FILL,color_fondo);
        floodfill(160,350,color_fondo);
        marco1(x1m,y1b,x2m,y2b,color_fondo2);
        sub_menu_global();
        mensaje(color_fondo2,color_letra1,"Escoja una opción del submenú o <ESC> para regresar al menú principal...");
        break;
    }
    case ESC:{
        marco1(x1m,y1m,x2m,y2m,color_fondo);
        setfillstyle(SOLID_FILL,color_fondo);
        floodfill(x1m+6,y1m+6,color_fondo);
        marco1(449,14,542,31,color_marco);
        menu_prin();
        mensaje(color_fondo,color_letra,"Seleccione una opción del Menú Principa o <ESC> para salir...");
        return;
    }
}
}
}
else {
    switch(c.ch[1]) {
        case 80:{
            if(y1b<=140){
                marco1(x1m,y1b,x2m,y2b,color_fondo2);
                y1b=y1b+26;
                y2b=y2b+26;
                marco1(x1m,y1b,x2m,y2b,color_letra);
            }
            break;
        }
        case 72:{
            if(y1b>=60){
                marco1(x1m,y1b,x2m,y2b,color_fondo2);
                y1b=y1b+1-27;
                y2b=y2b+1-27;
                marco1(x1m,y1b,x2m,y2b,color_letra);
            }
            break;
        }
    }
}
}

```

```

    }
  }
} while(c.ch[0]!='M' || c.ch[0]!='G' || c.ch[0]!='A' || c.ch[0]!='S' || c.ch[0]!='I' || c.ch[0]!='ESC');
}

void llama_graficas(int escalax)
(
  char *nombre;
  char *var;

  nombre="temperat.tmp";
  var="TEMPERATURA";
  graficar_global(nombre,num_temp,var,escalax);
  getch();
  nombre="carga.tmp";
  var="CARGA";
  graficar_global(nombre,num_carga,var,escalax);
  getch();
  nombre="espesor.tmp";
  var="ESPESOR";
  graficar_global(nombre,num_vel,var,escalax);
  getch();
  nombre="veloc.tmp";
  var="VELOCIDAD";
  graficar_global(nombre,num_par,var,escalax);
  getch();
  nombre="parto.tmp";
  var="PAR";
  graficar_global(nombre,num_esp,var,escalax);
}

/*SUBMENU PARA CONFIGURAR EL SISTEMA*/
void sub_menu_setup(void)
(
  char regresa;
  union lnkey {
    char ch[2];
    int i;
  } c;
  int x1,x2,x1m,y1m,x2m,y2m;
  int y1b,y2b;
  x1=535;
  x2=543;
  x1m=527;
  y1m=40;
  x2m=634;
  y2m=200;
  no_puerto=0;
  lmp_submenu_setup(x1,x2,x1m,y1m,x2m,y2m,color_fondo2);
  mensaje(color_fondo2,color_letra1,"Escoja una opción del submenú o <ESC> para regresar al Menú Principal...");
  y1b=50;
  y2b=77;
  marco1(x1m,y1b,x2m,y2b,color_letra);
  do {
    while(!bioskey(1));
    c.l=bioskey(0);
    if(c.ch[0]){
      switch(toupper(c.ch[0])) {
        case 'v':{
          if(y1b==50){
            recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA PUERTO SERIE 1 (COMM1)");
            no_puerto=0;
            regresa_setup(x1m,y1b,x2m,y2b,0);
            break;
          }
        }
      }
    }
  }
}

```

```
}
if(y1b==76){
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA PUERTO SERIE 2 (COMM2)");
    no_puerto=1;
    regresa_setup(x1m,y1b,x2m,y2b,0);
    break;
}
if(y1b==102){
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA ADQUIRIR TODOS LOS DATOS");
    muestreo=1;
    regresa_setup(x1m,y1b,x2m,y2b,0);
    break;
}
if(y1b==128){
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA ADQUIRIR CADA 10 DATOS");
    muestreo=10;
    regresa_setup(x1m,y1b,x2m,y2b,0);
    break;
}
if(y1b==154){
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA ADQUIRIR CADA 100 DATOS");
    muestreo=100;
    regresa_setup(x1m,y1b,x2m,y2b,0);
    break;
}
}
case '1':{
    marco1(x1m,y1b,x2m,y2b,color_fondo2);
    marco1(x1m,50,x2m,77,color_letra);
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA PUERTO SERIE 1 (COMM1)");
    no_puerto=0;
    regresa_setup(x1m,y1b,x2m,y2b,1);
    break;
}
case '2':{
    marco1(x1m,y1b,x2m,y2b,color_fondo2);
    marco1(x1m,76,x2m,103,color_letra);
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA PUERTO SERIE 2 (COMM2)");
    no_puerto=1;
    regresa_setup(x1m,y1b,x2m,y2b,2);
    break;
}
case '3':{
    marco1(x1m,y1b,x2m,y2b,color_fondo2);
    marco1(x1m,102,x2m,129,color_letra);
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA ADQUIRIR TODOS LOS DATOS");
    muestreo=1;
    regresa_setup(x1m,y1b,x2m,y2b,3);
    break;
}
case '4':{
    marco1(x1m,y1b,x2m,y2b,color_fondo2);
    marco1(x1m,128,x2m,155,color_letra);
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA ADQUIRIR CADA 10 DATOS");
    muestreo=10;
    regresa_setup(x1m,y1b,x2m,y2b,4);
    break;
}
case '5':{
    marco1(x1m,y1b,x2m,y2b,color_fondo2);
    marco1(x1m,154,x2m,181,color_letra);
    recuadro(color_letra,color_marco,color_letra1,"SISTEMA INICIALIZADO PARA ADQUIRIR CADA 100 DATOS");
    muestreo=100;
    regresa_setup(x1m,y1b,x2m,y2b,5);
}
```

```

        break;
    }
    case ESC:{
        marco1(x1m,y1m,x2m,y2m,color_fondo);
        setfillstyle(SOLID_FILL,color_fondo);
        floodfill(x1m+6,y1m+6,color_fondo);
        marco1(536,14,629,31,color_marco);
        menu_prin();
        mensaje(color_marco,color_tetra1,"Escoja una opción del Menú Principal o <ESC> para salir...");
        return;
    }
}
}
else {
    switch(c.ch[1]) {
        case 80:{
            if(y1b<=140){
                marco1(x1m,y1b,x2m,y2b,color_fondo2);
                y1b=y1b+26;
                y2b=y2b+26;
                marco1(x1m,y1b,x2m,y2b,color_tetra);
            }
            break;
        }
        case 72:{
            if(y1b>=60){
                marco1(x1m,y1b,x2m,y2b,color_fondo2);
                y1b=y1b+1-27;
                y2b=y2b+1-27;
                marco1(x1m,y1b,x2m,y2b,color_tetra);
            }
            break;
        }
    }
}
}while(c.ch[0]!='1'||c.ch[0]!='2');
}

void regresa_setup(int x1m,int y1b,int x2m,int y2b,int opcion)
{
    char regresa;

    do{
        mensaje(color_tetra,color_tetra1,"Presione <ESC> para regresar a submenú...");
        regresa=getch();
    } while(regresa!=ESC);
    if(opcion==1) marco1(x1m,50,x2m,77,color_fondo2);
    if(opcion==2) marco1(x1m,76,x2m,103,color_fondo2);
    if(opcion==3) marco1(x1m,102,x2m,129,color_fondo2);
    if(opcion==4) marco1(x1m,128,x2m,155,color_fondo2);
    if(opcion==5) marco1(x1m,154,x2m,181,color_fondo2);
    if(opcion==6) marco1(x1m,180,x2m,207,color_fondo2);
    marco1(106,344,521,380,color_fondo);
    setfillstyle(SOLID_FILL,color_fondo);
    floodfill(160,350,color_fondo);
    marco1(x1m,y1b,x2m,y2b,color_fondo2);
    sub_menu_setup();
    mensaje(color_fondo2,color_tetra1,"Escoja una opción del submenú o <ESC> para regresar al Menú Principal...");
}

/*ADQUISICION DE DATOS PARA UNA SOLA VARIABLE*/
void adquirir(char c)
{
    float v_analog,valor_real,b10,velocidad,corriente;

```

```

int Inc,x,y,valor,i;
char *till,*msj;
FILE *ft;
char siguiente,cad,regresa;
int n_datos,n_digital,n_real,n;

till=" MODULO DE ADQUISICION DE DATOS ";
msj="Dat# Val";
clearviewport();
pantalla(maxx,maxy,color_marco,color_fondo);
setcolor(color_letra);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
outtextxy(320,16,till);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy((maxx/1.15),(maxy/17.14),variable);
marco1(33,43,603,448,color_marco);
setfillstyle(EMPTY_FILL,color_fondo);
floodfill(37,120,color_marco);
valor_real=0.00;
n=0;
num_vel=0;
num_par=0;
n_real=0;
n_digital=0;
n_dato=0;
n_datos=0;
inic_puerto(no_puerto,163);
ft=fopen("temporal.tmp","wb");
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextj(justify(CENTER_TEXT, TOP_TEXT);
setcolor(color_letra);
x=maxx/32;
y=maxy/7.74;
outtextxy(x+(maxx/10),y-(maxy/36),msj);
mensaje(color_fondo2,color_letra1,"Presione cualquier tecla para cortar adquisición de datos...");
envia_puerto(no_puerto,c);
do{
    b10=lee_puerto(no_puerto);
    n_digital++;
    v_analog=b10*5/255;
    switch(c){
        case 'T': valor_real=(v_analog+0.86665)/0.004282;
            n_real++;
            break;
        case 'C': valor_real=(v_analog+0.10382)/0.21187;
            n_real++;
            break;
        case 'E': valor_real=v_analog;
            n_real++;
            break;
        case 'V': valor_real=b10*65535*0.000000500;
            n_real++;
            break;
        case 'P': if((n_digital==1||n_digital==(num_vel*2)+1)){
                num_vel++;
                velocidad=b10*65535*0.000000500;
            }
            if((n_digital==2||n_digital==(num_par*2)+2)){
                num_par++;
                valor_real=(v_analog-0.00929)/0.12045;
                n_real++;
            }
            break;
    }
}

```

```

if(n_real==muestreo*n+1) {
    n++;
    n_dato++;
    n_datos++;
    fwrite(&valor_real,sizeof(float),1,ft);
    setcolor(color_dato);
    x=x+(maxx/12);
    dprintf(&x,&y,"%d",n_dato);
    x=x+(maxx/18);
    setcolor(color_valor);
    rprintf(&x,&y,"%2f",valor_real);
    if(n_datos>0&&n_datos<=34) x=maxx/32;
    if(n_datos>34&&n_datos<=69) x=maxx/6.4;
    if(n_datos>69&&n_datos<=104) x=maxx/3.65;
    if(n_datos>104&&n_datos<=139) x=maxx/2.56;
    if(n_datos>139&&n_datos<=174) x=maxx/1.96;
    if(n_datos>174&&n_datos<=209) x=maxx/1.6;
    if(n_datos>209&&n_datos<=244) x=maxx/1.34;
    if(n_datos>244&&n_datos<=488){
        mensaje(color_fondo2,color_letra1,"Presione cualquier tecla para cortar adquisición de datos...");
        setfillstyle(EMPTY_FILL,color_fondo);
        floodfill(37,120,color_marco);
        n_datos=0;
        x=maxx/32;
    }
    y=y+(maxy/40);
    if(y>435){
        y=maxy/7.74;
        setcolor(color_letra);
        outtextxy(x+(maxx/10),y-(maxy/36),msj);
    }
}
}while(b10>=00);
fclose(ft);
}

```

/*ADQUISICION DE DATOS PARA LAS 5 VARIABLES*/

```

void adquirir_global(char c)
{
    int inc,x,y,valor,i;
    char *tlti,*msj,*var1;
    FILE *fp;
    FILE *ft,*fc,*fv,*fa,*fe,*fglob;
    int b10,n_datos,datos_renglon,datos_pantalla;
    float v_analog;
    float valor_real,velocidad,corriente;

    tlti=" MODULO DE ADQUISICION DE DATOS ";
    var1="TEMPERATURA CARGA ESPESOR VELOCIDAD PAR ";
    msj=" Dat# Val Dat# Val Dat# Val Dat# Val Dat# Val";
    clearviewport();
    pantalla(maxx,maxy,color_marco,color_fondo);
    setcolor(color_letra);
    settxtstyle(TRIPLEX_FONT,HORIZ_DIR,2);
    outtextxy(320,16,tlti);
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
    outtextxy(556,28,variable);
    marco1(33,43,603,448,color_marco);
    setfillstyle(EMPTY_FILL,color_fondo);
    floodfill(37,120,color_marco);
    num_temp=0;
    num_carga=0;
    num_vel=0;
    num_par=0;
}

```

```

num_esp=0;
n_dato=0;
n_datos=1;
datos_renglon=0;
datos_pantalla=1;
fglob=fopen("global.tmp","wb");
ft=fopen("temperat.tmp","wb");
fc=fopen("carga.tmp","wb");
fv=fopen("veloc.tmp","wb");
fa=fopen("parto.tmp","wb");
fe=fopen("espesor.tmp","wb");
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(CENTER_TEXT,TOP_TEXT);
setcolor(color_letra);
outtextxy(310,50,var1);
outtextxy(310,60,msg);
mensaje(color_fondo2,color_letra1,"Presione cualquier tecla para cortar comunicaci3n...");
x=68;
y=75;
inic_puerto(no_puerto,163);
envia_puerto(no_puerto,c);
do{
    b10=lee_puerto(no_puerto);
    v_analog=b10*(5/255);
    n_dato++;
    datos_renglon++;
    if(n_dato==1||n_dato==((num_temp*5)+1)){
        valor_real=(v_analog+0.86665)/0.004282;
        num_temp++;
        fwrite(&valor_real,sizeof(float),1,ft);
        fwrite(&valor_real,sizeof(float),1,fglob);
    }
    if(n_dato==2||n_dato==((num_carga*5)+2)){
        valor_real=(v_analog+0.10382)/0.16187;
        num_carga++;
        fwrite(&valor_real,sizeof(float),1,fc);
        fwrite(&valor_real,sizeof(float),1,fglob);
    }
    if(n_dato==3||n_dato==((num_esp*5)+3)){
        valor_real=(v_analog+0.86665)/0.004282;
        num_esp++;
        fwrite(&valor_real,sizeof(float),1,fe);
        fwrite(&valor_real,sizeof(float),1,fglob);
    }
    if(n_dato==4||n_dato==((num_vel*5)+4)){
        velocidad=b10;
        valor_real=velocidad*65535*0.000000500;
        num_vel++;
        fwrite(&valor_real,sizeof(float),1,fv);
        fwrite(&valor_real,sizeof(float),1,fglob);
    }
    if(n_dato==5||n_dato==((num_par*5)+5)){
        valor_real=(v_analog-0.00929)/0.12045;
        num_par++;
        fwrite(&valor_real,sizeof(float),1,fa);
        fwrite(&valor_real,sizeof(float),1,fglob);
    }
    setcolor(color_dato);
    dprintf(&x,&y,"%d",n_datos);
    x=x+35;
    setcolor(color_valor);
    rprintf(&x,&y,"%2f",valor_real);
    x=x+82;
    if(datos_renglon>=5){

```

```

y=y+10;
x=68;
n_datos++;
datos_pantalla++;
datos_renglon=0;
}
if(datos_pantalla>41) {
mensaje(color_fondo2,color_letra1,"Presione cualquier tecla para cortar comunicacion...");
setfillstyle(EMPTY_FILL,color_fondo);
floodfill(37,120,color_marco);
setcolor(color_letra);
outtextxy(310,50,var1);
outtextxy(310,60,msj);
datos_pantalla=1;
y=75;
}
}while(b10>=00);
fclose(ft);
fclose(fc);
fclose(fv);
fclose(fa);
fclose(fe);
fclose(fglob);
}

/*FUNCION PARA LA GRAFICACION DE DATOS ADQUIRIDOS*/
graficar(int factor_escalax)
{
int x,y;
int n_datos;
char *titl;
char cad_max[6];
FILE *ft;
int x0,y0,x1,y1,intervalo_x,nuevo_x,l;
float maximo,minimo,valor,maxima_y;
float promedio,max_aux,sumatoria;
float intervalo_y,nuevo_y,coord_y;

titl=" MODULO DE GRAFICACION DE DATOS ";
clearviewport();
sumatoria=0;
pantalla(maxx,maxy,color_marco,color_grafica);
setcolor(color_letra1);
settextstyle(TRIPLEX_FONT, HORIZ_DIR, 2);
outtextxy(320,7,titl);
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
outtextxy(556,19,variable);
ft=fopen("temporal.tmp", "rb");
maximo=mayor(ft);
minimo=menor(ft);
fclose(ft);
factor_escalay=1;
if((n_dato==0)||(n_dato==1){
recuadro(color_letra,color_marco,color_letra1,"No existen datos en archivos para graficar...");
return;
}
max_aux=0.00;
maxima_y=0.00;
x0=maxx/16;
y0=maxy/1.090;
x1=maxx/1.015;
y1=maxy/13.71;
intervalo_x=((x1-x0)*factor_escalax)/(n_dato);
intervalo_y=((y0-y1)/(maximo-minimo);

```

```

dibuja_ejes(maximo,minimo,x0,y0,x1,y1,intervalo_x,intervalo_y);
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
nuevo_x=x0;
moveo(x0,y0);
n_datos=1;
ft=fopen("temporal.tmp","rb");
while(!feof(ft)){
    while(n_datos<=factor_escalax){
        fread(&valor,sizeof(float),1,ft);
        sumatoria=sumatoria+valor;
        n_datos++;
    }
    promedio=sumatoria/(n_datos-1);
    if(promedio>max_aux) max_aux=promedio;
    if((nuevo_x>=x0)&&(nuevo_x<=610)){
        coord_y=y0-(promedio*intervalo_y*factor_escalay);
        lineo((nuevo_x+intervalo_x),coord_y);
        nuevo_x=nuevo_x+intervalo_x;
    }
    n_datos=1;
    sumatoria=0;
}
fclose(ft);
maxima_y=y0-(max_aux*intervalo_y*factor_escalay);
gcvt(maximo,5,cad_max);
outtextxy(x0+30,maxima_y-10,cad_max);
factor_escalax=0;
}

/*GRAFICACION DE LOS DATOS ADQUIRIDOS PARA TODAS LAS VARS.*/
graficar_global(char *nombre, int numero, char *var, int factor_escalax)
{
    int x,y,n_datos;
    char *titi;
    int x0,y0,x1,y1,intervalo_x,nuevo_x;
    float valor,maximo,minimo,intervalo_y,nuevo_y;
    float sumatoria,promedio;
    FILE *ft;

    titi=" MODULO DE GRAFICACION DE DATOS ";
    clearviewport();
    pantalla(maxx,maxy,color_marco,color_grafica);
    setcolor(color_letra1);
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 2);
    outtextxy(320,7,titi);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(556,19,var);
    ft=fopen(nombre,"rb");
    maximo=mayor(ft);
    minimo=menor(ft);
    fclose(ft);
    if(numero==0||numero==1){
        recuadro(color_letra,color_marco,color_letra1,"No existen datos en archivos para graficar...");
        return;
    }
    x0=maxx/16;
    y0=maxy/1.090;
    x1=maxx/1.015;
    y1=maxy/13.71;
    intervalo_x=((x1-x0)*factor_escalax)/numero;
    intervalo_y=(y0-y1)/(maximo-minimo);
    dibuja_ejes_global(maximo,minimo,x0,y0,x1,y1,intervalo_x,intervalo_y,numero);
    siguiente_grafica(maxx,maxy,color_dato,color_letra1);
    nuevo_x=x0;

```

```

moveto(x0, y0);
ft=fopen(nombre, "rb");
sumatoria=0.00;
n_datos=1;
while(!feof(ft)){
    while(n_datos<=factor_escalax){
        fread(&valor,sizeof(float),1,ft);
        sumatoria=sumatoria+valor;
        n_datos=n_datos+1;
    }
    promedio=sumatoria/(n_datos-1);
    lineto((nuevo_x+Intervalo_x),(y0-((promedio*intervalo_y*factor_escalay))));
    nuevo_x=nuevo_x+intervalo_x;
    n_datos=1;
    sumatoria=0;
}
fclose(ft);
}

/*ADQUISICION Y GRAFICACION DIRECTA DE UNA VARIABLE*/
void directo(char c)
{
    int x0,y0,x1,y1,Intervalo_x,nuevo_x,n,n_real,b10;
    float v_analog,valor_real,maximo,minimo,Intervalo_yy,f_escalax;
    char *titi;
    FILE *ft;

    titi=" MODULO DE GRAFICACION DIRECTA DE DATOS ";
    clearviewport();
    pantalla(maxx,maxy, color_marco, color_grafica);
    setcolor(color_letra1);
    setttextstyle(TRIPLEX_FONT, HORIZ_DIR, 2);
    outtextxy(320,16,titi);
    setttextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(590,28,variable);
    n=0;
    n_real=0.00;
    x0=40;
    y0=440;
    x1=630;
    y1=35;
    Intervalo_x=(x1-x0)/590;
    switch(c){
        case 'T':Intervalo_yy=(y0-y1)/40;
            dibuja_ejes_directo(40,0,x0,y0,x1,y1,Intervalo_yy);
            f_escalax=0.033;
            break;
        case 'C':Intervalo_yy=((y0-y1)/30);
            dibuja_ejes_directo(30,0,x0,y0,x1,y1,Intervalo_yy);
            f_escalax=1.00;
            break;
    }
    mensaje(color_dato, color_letra1, "Presione cualquier tecla para cortar adquisición de datos..");
    nuevo_x=x0;
    moveto(x0,y0);
    inic_puerto(no_puerto,163);
    envia_puerto(no_puerto,c);
    ft=fopen("temporal.tmp","wb");
    do{
        b10=lee_puerto(no_puerto);
        v_analog=b10*5/255;
        switch(c){
            case 'T':valor_real=(v_analog+0.86665)/0.004282;
                n_real++;

```

```

    break;
case 'C':valor_real=(v_analog+0.10382)/0.21187;
    n_real++;
    break;
}
if(n_real==muestreo*n+1){
    n++;
    lineto(nuevo_x+intervalo_x,y0-(valor_real*intervalo_yy*f_escal));
    nuevo_x=nuevo_x+intervalo_x;
    fwrite(&valor_real,sizeof(valor_real),1,ft);
}
}while(b10>=00);
fclose(ft);
}

/*FUNCIONES PARA LA IMPRESION DE LOS DATOS ADQUIRIDOS*/
void imprimir(void)
{
    unsigned eq;
    FILE *fuente;
    int n_datos,global,n_pagina;
    float car;
    char cadena1[10];
    char cadena2[5];

    fuente=fopen("temporal.tmp","rb");
    n_datos=0;
    global=0;
    n_pagina=1;
    eq=biosprint(2,0,0);
    eq>>=4;
    if(eq==9){
        recuadro(color_letra,color_marco,color_letra,"ESPERE UN MOMENTO, IMPRIMIENDO DATOS");
        encabezado(n_pagina);
        while(!feof(fuente)){
            fread(&car,sizeof(car),1,fuente);
            gcvt(car,6,cadena1);
            n_datos++;
            global++;
            ltoa(global,cadena2,10);
            if(n_datos>=108){
                biosprint(0,10,0);
                biosprint(0,10,0);
                biosprint(0,10,0);
                biosprint(0,10,0);
                biosprint(0,10,0);
                biosprint(0,10,0);
                n_pagina++;
                encabezado(n_pagina);
                n_datos=0;
            }
            biosprint(0,9,0);
            imprime_numero(cadena2);
            biosprint(0,9,0);
            biosprint(0,9,0);
            imprime_numero(cadena1);
            fread(&car,sizeof(car),1,fuente);
            gcvt(car,6,cadena1);
            global++;
            ltoa(global,cadena2,10);
            n_datos++;
            biosprint(0,9,0);
            biosprint(0,9,0);
            biosprint(0,9,0);

```

```

    imprime_numero(cadena2);
    biosprint(0,9,0);
    biosprint(0,9,0);
    imprime_numero(cadena1);
    biosprint(0,10,0);
}
marco1(115,345,521,380,color_fondo);
setfillstyle(SOLID_FILL,color_fondo);
floodfill(120,350,color_fondo);
}
else recuadro(color_letra,color_marco,color_letra1,"ERROR DE IMPRESION, CHECAR CONEXION O IMPRESORA");
fclose(fuente);
}

```

```
void encabezado(int n_pagina)
```

```

{
char *letrero1,*letrero2,*letrero3,*letrero4;
char cadena3[4];
time_t t;
struct time dos_time;
struct date dos_date;
struct tm *local;

getdate(&dos_date);
gettime(&dos_time);
t=dostounix(&dos_date,&dos_time);
local=asctime(localtime(&t));
letrero1=" Fecha y Hora: ";
letrero2=" Página #: ";
imprime_letrero(letrero1);
imprime_letrero(local);
imprime_letrero(letrero2);
ltoa(n_pagina,cadena3,10);
imprime_numero(cadena3);
biosprint(0,9,0);
imprime_letrero(" Variable: ");
imprime_letrero(variable);
biosprint(0,10,0);
letrero3=" SISTEMA DE ADQUISICION Y DESPLIEGUE DE INFORMACION HILLE 25";
letrero4=" Numero de Dato Valor adquirido Numero de Dato Valor adquirido";
imprime_letrero(letrero3);
biosprint(0,10,0);
biosprint(0,10,0);
imprime_letrero(letrero4);
biosprint(0,10,0);
biosprint(0,10,0);
}

```

```
void Imprimir_global(void)
```

```

{
unsigned eq;
char respuesta,caracter;
char *espacio1,*espacio2;
FILE *fuente;
int n_datos,global,n_pagina,datos_linea;
float car;
union REGS r;
char cadena1[10];
char cad_temp[5],cad_carga[5],cad_vel[5],cad_par[5],cad_esp[5];

num_temp=0;
num_carga=0;
num_vel=0;
num_par=0;

```

```

num_esp=0;
n_datos=0;
datos_linea=0;
global=0;
n_pagina=1;
fuente=fopen("global.tmp","rb");
eq=biosprint(2,0,0);
eq>=4;
if(eq==9){
    recuadro(color_letra,color_marco,color_letra1,"ESPERE UN MOMENTO, IMPRIENDO DATOS");
    encabezado_global(n_pagina);
    while(!feof(fuente)){
        fread(&car,sizeof(car),1,fuente);
        gcvt(car,6,cadena1);
        datos_linea++;
        n_datos++;
        global++;
        if(n_datos>=270){
            biosprint(0,10,0);
            biosprint(0,10,0);
            biosprint(0,10,0);
            biosprint(0,10,0);
            biosprint(0,10,0);
            biosprint(0,10,0);
            n_pagina++;
            encabezado_global(n_pagina);
            n_datos=0;
        }
        if(global==1||global==(num_temp*5)+1){
            num_temp++;
            ltoa(num_temp,cad_temp,10);
            espacio1=" ";
            imprime_letrero(espacio1);
            imprime_numero(cad_temp);
            biosprint(0,9,0);
            biosprint(0,32,0);
            biosprint(0,32,0);
            biosprint(0,32,0);
            imprime_numero(cadena1);
        }
        if(global==2||global==(num_carga*5)+2){
            num_carga++;
            ltoa(num_carga,cad_carga,10);
            biosprint(0,9,0);
            biosprint(0,32,0);
            biosprint(0,32,0);
            biosprint(0,32,0);
            imprime_numero(cad_carga);
            biosprint(0,9,0);
            biosprint(0,32,0);
            biosprint(0,32,0);
            imprime_numero(cadena1);
        }
        if(global==3||global==(num_esp*5)+3){
            num_esp++;
            ltoa(num_esp,cad_esp,10);
            biosprint(0,9,0);
            biosprint(0,32,0);
            biosprint(0,32,0);
            imprime_numero(cad_esp);
            biosprint(0,9,0);
            biosprint(0,32,0);
            imprime_numero(cadena1);
        }
    }
}

```

```

if(global==4||global==(num_vel*5)+4){
    num_vel++;
    itoa(num_vel,cad_vel,10);
    biosprint(0,9,0);
    biosprint(0,32,0);
    Imprime_numero(cad_vel);
    biosprint(0,9,0);
    imprime_numero(cadena1);
}
if(global==5||global==(num_par*5)+5){
    num_par++;
    itoa(num_par,cad_par,10);
    biosprint(0,9,0);
    Imprime_numero(cad_par);
    biosprint(0,9,0);
    imprime_numero(cadena1);
}
if(datos_linea>=5){
    biosprint(0,10,0);
    datos_linea=0;
}
}
marco1(115,345,521,380,color_fondo);
setfillstyle(SOLID_FILL,color_fondo);
floodfill(120,350,color_fondo);
}
else recuadro(color_letra,color_marco,color_letra1,"ERROR DE IMPRESION, CHECAR CONEXION O IMPRESORA");
fclose(fuente);
)

```

```

void encabezado_global(int n_pagina)

```

```

{
    char *letrero1,*letrero2,*letrero3,*letrero4,*letrero5;
    char cadena3[4];
    time_t t;
    struct time dos_time;
    struct date dos_date;
    struct tm *local;

    getdate(&dos_date);
    gettime(&dos_time);
    t=dostounix(&dos_date,&dos_time);
    local=asctime(localtime(&t));
    letrero1=" Fecha y Hora: ";
    letrero2=" Página #: ";
    Imprime_letrero(letrero1);
    Imprime_letrero(local);
    Imprime_letrero(letrero2);
    itoa(n_pagina,cadena3,10);
    imprime_numero(cadena3);
    biosprint(0,9,0);
    Imprime_letrero(" Variable: Global ");
    biosprint(0,10,0);
    letrero3=" SISTEMA DE ADQUISICION Y DESPLIEGUE DE INFORMACION HILLE 25 ";
    letrero4=" TEMPERATURA CARGA ESPESOR VELOCIDAD PAR ";
    letrero5=" #dato Valor #dato Valor #dato Valor #dato Valor #dato Valor";
    Imprime_letrero(letrero3);
    biosprint(0,10,0);
    biosprint(0,10,0);
    Imprime_letrero(letrero4);
    biosprint(0,10,0);
    Imprime_letrero(letrero5);
    biosprint(0,10,0);
}

```

```
void imprime_letrero(char *cadena)
{
    int i;
    char caracter;
    for(i=0; i<=strlen(cadena); ++i){
        caracter=cadena[i];
        biosprint(0,caracter,0);
    }
}
```

```
void imprime_numero(char cadena[5])
{
    int i;
    char caracter;
    for(i=0; i<=strlen(cadena); i++){
        caracter=cadena[i];
        switch(caracter){
            case '0': biosprint(0,48,0);
                break;
            case '1': biosprint(0,49,0);
                break;
            case '2': biosprint(0,50,0);
                break;
            case '3': biosprint(0,51,0);
                break;
            case '4': biosprint(0,52,0);
                break;
            case '5': biosprint(0,53,0);
                break;
            case '6': biosprint(0,54,0);
                break;
            case '7': biosprint(0,55,0);
                break;
            case '8': biosprint(0,56,0);
                break;
            case '9': biosprint(0,57,0);
                break;
            case '.': biosprint(0,46,0);
                break;
        }
    }
}
```

/*FUNCIONES PARA ABRIR Y CERRAR ARCHIVOS DE DATOS*/

```
void abrir_salvar(int condicion)
```

```
{
    struct fblk a;
    struct fblk b;
    register int fin;
    register int encuentra;
    int n_archivos,x,y;
    char respuesta,regresa;
    char nombre[12];
    FILE *entrada,*salida;
    float car;
```

```
    n_archivos=0;
    x=3;
    y=3;
    restorecrtmode();
    clrscr();
    fin=findfirst("*.dat",&a,0);
    textbackground(1);
```

```

window(1,1,80,25);
borde(1,1,80,25);
window(1,1,80,20);
borde(1,1,80,20);
while(!fin) {
    gotoxy(x,y);
    printf("%s\n",a.ff_name);
    fin=findnext(&a);
    n_archivos=n_archivos++;
    y=y++;
    if(n_archivos>=15) {
        x=x+16;
        y=3;
        n_archivos=0;
    }
    if(x>=83){
        gotoxy(7,19);
        printf("Presione cualquier tecla para ver más archivos ó <ESC> para salvar ");
        textbackground(0);
        respuesta=getch();
        if(respuesta==ESC){
            textbackground(1);
            window(1,21,80,24);
            gotoxy(7,1);
            printf("Introduzca nombre del archivo o <F> para regresar --> ");
            gets(nombre);
            if(!strcmp(strupr(nombre),"F"))
                return;
            else{
                if(condicion==1){
                    entrada=fopen("temporal.tmp","rb");
                    salida=fopen(nombre,"wb");
                    n_dato=0;
                    while(!feof(entrada)) {
                        n_dato=n_dato+1;
                        fread(&car,sizeof(car),1,entrada);
                        if(!feof(entrada)) fwrite(&car,sizeof(car),1,salida);
                    }
                }
                else{
                    encuentra=findfirst(nombre,&b,0);
                    if(encuentra!=-1){
                        gotoxy(5,3);
                        printf("No existe el archivo, Presione cualquier tecla para regresar a submenú");
                        getch();
                        return;
                    }
                    entrada=fopen(nombre,"rb");
                    salida=fopen("temporal.tmp","wb");
                    n_dato=0;
                    while(!feof(entrada)) {
                        n_dato=n_dato+1;
                        fread(&car,sizeof(car),1,entrada);
                        if(!feof(entrada)) fwrite(&car,sizeof(car),1,salida);
                    }
                }
            }
        }
        do{
            gotoxy(22,3);
            printf("Presione <ESC> para regresar a Submenú");
            regresa=getch();
        } while(regresa!=ESC);
    }
}
else {

```

```

    clrscr();
    x=3;
    textbackground(1);
    window(1,1,80,25);
    borde(1,1,80,25);
    window(1,1,80,20);
    borde(1,1,80,20);
}
}
}
window(1,21,80,24);
gotoxy(7,1);
cprintf("Introduzca nombre del archivo o <F> para regresar -> ");
gets(nombre);
if(!strcmp(strupr(nombre), "F"))
    return;
else{
    if(condicion==1){
        entrada=fopen("temporal.tmp", "rb");
        salida =fopen(nombre, "wb");
        n_dato=0;
        while(!feof(entrada)){
            n_dato=n_dato+1;
            fread(&car, sizeof(car), 1, entrada);
            if(!feof(entrada)) fwrite(&car, sizeof(car), 1, salida);
        }
    }
    else{
        encuentra=findfirst(nombre, &b, 0);
        if(encuentra== -1){
            gotoxy(5,3);
            cprintf("No existe el archivo, Presione cualquier tecla para regresar a submenú");
            getch();
            return;
        }
        entrada=fopen(nombre, "rb");
        salida =fopen("temporal.tmp", "wb");
        n_dato=0;
        while(!feof(entrada)){
            n_dato=n_dato+1;
            fread(&car, sizeof(car), 1, entrada);
            if(!feof(entrada)) fwrite(&car, sizeof(car), 1, salida);
        }
    }
}
do {
    gotoxy(22,3);
    cprintf("Presione <ESC> para regresar a submenú");
    regresa=getch();
} while(regresa!=ESC);
fclose(entrada);
fclose(salida);
}

void abrir_salvar_global(int condicion)
{
    struct fbik a;
    struct fbik b;
    register int fin;
    register int encuentra;
    int n_archivos, x, y;
    float car;
    char respuesta, regresa;
    char nombre[12];

```

```

FILE *entrada,*salida,*salida_t,*salida_c,*salida_v,*salida_p,*salida_e,*salida_glob;

n_archivos=0;
x=3;
y=3;
restorecrtmode();
clrscr();
fin=findFirst("**.dat",&a,0);
textbackground(1);
window(1,1,80,25);
borde(1,1,80,25);
wlnow(1,1,80,20);
borde(1,1,80,20);
while(!fin) {
    gotoxy(x,y);
    printf("%s\n", a.ff_name);
    fin=findnext(&a);
    n_archivos=n_archivos++;
    y=y++;
    if(n_archivos>=15) {
        x=x+16;
        y=3;
        n_archivos=0;
    }
    if(x>=83){
        gotoxy(7,19);
        cprintf("Presione cualquier tecla para ver más archivos ó <ESC> para salvar ");
        textbackground(0);
        respuesta=getch();
        if(respuesta==ESC){
            textbackground(1);
            window(1,21,80,24);
            gotoxy(7,1);
            cprintf("Introduzca nombre del archivo o <F> para regresar --> ");
            gets(nombre);
            if(!strcmp(strupr(nombre), "F"))
                return;
            else{
                if(condicion==1){
                    entrada=fopen("global.tmp","rb");
                    salida=fopen(nombre,"wb");
                    n_dato=0;
                    while(!feof(entrada)) {
                        n_dato=n_dato+1;
                        fread(&car,sizeof(car),1,entrada);
                        if(!feof(entrada)) fwrite(&car,sizeof(car),1,entrada);
                    }
                }
                else{
                    encuentra=findfirst(nombre,&b,0);
                    if(encuentra===-1){
                        gotoxy(5,3);
                        cprintf("No existe el archivo, Presione cualquier tecla para regresar a submenú");
                        getch();
                        return;
                    }
                    entrada=fopen(nombre,"rb");
                    salida_t=fopen("temperat.tmp","wb");
                    salida_c=fopen("carga.tmp","wb");
                    salida_v=fopen("veloc.tmp","wb");
                    salida_p=fopen("parto.tmp","wb");
                    salida_e=fopen("espesor.tmp","wb");
                    salida_glob=fopen("global.tmp","wb");
                    n_dato=0;
                }
            }
        }
    }
}

```

```

num_temp=0;
num_carga=0;
num_vel=0;
num_par=0;
num_esp=0;
while(!feof(entrada)) {
    n_dato=n_dato+1;
    fread(&car,sizeof(car),1,entrada);
    if((n_dato==1||n_dato==(num_temp*5)+1)){
        if(!feof(entrada)){
            num_temp++;
            fwrite(&car,sizeof(car),1,salida_t);
            fwrite(&car,sizeof(car),1,salida_glob);
        }
    }
    if((n_dato==2||n_dato==(num_carga*5)+2)){
        if(!feof(entrada)){
            num_carga++;
            fwrite(&car,sizeof(car),1,salida_c);
            fwrite(&car,sizeof(car),1,salida_glob);
        }
    }
    if((n_dato==3||n_dato==(num_esp*5)+3)){
        if(!feof(entrada)){
            num_esp++;
            fwrite(&car,sizeof(car),1,salida_e);
            fwrite(&car,sizeof(car),1,salida_glob);
        }
    }
    if((n_dato==4||n_dato==(num_vel*5)+4)){
        if(!feof(entrada)){
            num_vel++;
            fwrite(&car,sizeof(car),1,salida_v);
            fwrite(&car,sizeof(car),1,salida_glob);
        }
    }
    if((n_dato==5||n_dato==(num_par*5)+5)){
        if(!feof(entrada)){
            num_par++;
            putw(car,salida_p);
            putw(car,salida_glob);
        }
    }
}
do{
    gotoxy(22,3);
    cprintf("Presione <ESC> para regresar a Submenú");
    regresa=getch();
} while(regresa!=ESC);
}
else {
    clrscr();
    x=3;
    textbackground(1);
    window(1,1,80,25);
    borde(1,1,80,25);
    window(1,1,80,20);
    borde(1,1,80,20);
}
}
window(1,21,80,24);

```

```

gotoxy(7,1);
cprintf("Introduzca nombre del archivo ó <F> para regresar -> ");
gets(nombre);
if(!(strcmp(strupr(nombre),"F")))
    return;
else{
    if(condicion==1){
        entrada=fopen("global.tmp","rb");
        salida=fopen(nombre,"wb");
        n_dato=0;
        while(!feof(entrada)) {
            n_dato=n_dato+1;
            car=getw(entrada);
            if(!feof(entrada)) putw(car, salida);
        }
    }
    else{
        encuentra=findfirst(nombre,&b,0);
        if(encuentra==1){
            gotoxy(5,3);
            cprintf("No existe el archivo, Presione cualquier tecla para regresar a submenú");
            getch();
            return;
        }
        entrada=fopen(nombre,"rb");
        salida_t=fopen("temperat.tmp","wb");
        salida_c=fopen("carga.tmp","wb");
        salida_v=fopen("veloc.tmp","wb");
        salida_p=fopen("parto.tmp","wb");
        salida_e=fopen("espesor.tmp","wb");
        salida_glob=fopen("global.tmp","wb");
        n_dato=0;
        num_temp=0;
        num_carga=0;
        num_vel=0;
        num_par=0;
        num_esp=0;
        while(!feof(entrada)) {
            n_dato=n_dato+1;
            car=getw(entrada);
            if((n_dato==1||n_dato==((num_temp*5)+1)){
                if(!feof(entrada)){
                    num_temp++;
                    putw(car,salida_t);
                    putw(car,salida_glob);
                }
            }
            if((n_dato==2||n_dato==((num_carga*5)+2)){
                if(!feof(entrada)){
                    num_carga++;
                    putw(car,salida_c);
                    putw(car,salida_glob);
                }
            }
            if((n_dato==3||n_dato==((num_esp*5)+3)){
                if(!feof(entrada)){
                    num_esp++;
                    putw(car,salida_e);
                    putw(car,salida_glob);
                }
            }
            if((n_dato==4||n_dato==((num_vel*5)+4)){
                if(!feof(entrada)){
                    num_vel++;
                }
            }
        }
    }
}

```

```

        putw(car,salida_v);
        putw(car,salida_glob);
    }
}
if(n_dato==5||n_dato==(num_par*5)+5){
    if(!feof(entrada)){
        num_par++;
        putw(car,salida_p);
        putw(car,salida_glob);
    }
}
}
)
}
do {
    gotoxy(22,3);
    printf("Presione <ESC> para regresar a submenú");
    regresa=getch();
} while(regresa!=ESC);
fclose(entrada);
fclose(salida);
fclose(salida_t);
fclose(salida_c);
fclose(salida_v);
fclose(salida_p);
fclose(salida_e);
fclose(salida_glob);
}

```

```
void borde(int xi, int yi, int xf, int yf)
```

```

{
    register int i;
    gotoxy(1,1);
    putchar('+');
    for(i=0;i<=xf-xi;i++)
        putchar('-');
    gotoxy(xf, yi);
    putchar('+');
    gotoxy(1,yf-yi);
    putchar('+');
    for(i=0;i<=xf-xi-2;i++)
        putchar('-');
    gotoxy(xf,yf-1);
    putchar('+');
    for(i=2;i<=yf-yi-1;i++) {
        gotoxy(1,i);
        putchar('|');
        gotoxy(xf-xi+1,i);
        putchar('|');
    }
}
}

```

```
/* PROCEDIMIENTOS AUXILIARES PARA LA ADQUISICION Y GRAFICACION DE DATOS */
```

```
void inic_puerto(int n_puerto,unsigned char code)
```

```

{
    union REGS r;

    r.x.dx=n_puerto;
    r.h.ah=0;
    r.h.al=code;
    int86(0x14,&r,&r);
}

```

```
envia_puerto(int n_puerto,char c)
```

```

{
union REGS r;

r.x.dx = n_puerto;
r.h.al = c;
r.h.ah = 1;
int86(0x14, &r, &r);
if(r.h.ah & 128){
    recuadro(color_letra,color_marco,color_letra1,"Detectado error de transmisión en puerto serie");
}
}

lee_puerto(int n_puerto)
{
union REGS r;

while(!(chequea_estado(n_puerto)&256))
    if(kbhit()){
        return;
    }
r.x.dx=n_puerto;
r.h.ah=2;
int86(0x14, &r, &r);
if(r.h.ah & 128)
    recuadro(color_letra,color_marco,color_letra1,"Detectado error de lectura en el puerto serie");
return(r.h.al);
}

chequea_estado(int n_puerto)
{
union REGS r;

r.x.dx=n_puerto;
r.h.ah=3;
int86(0x14, &r, &r);
return(r.x.ax);
}

float mayor(FILE *ft)
{
float aux_max,valor_max;

valor_max=0;
while(!feof(ft)){
    fread(&aux_max,sizeof(float),1,ft);
    if(aux_max>valor_max) valor_max=aux_max;
}
return(valor_max);
}

float menor(FILE *ft)
{
float valor_min,aux_min;

valor_min=0;
fread(&aux_min,sizeof(float),1,ft);
while(!feof(ft)){
    if(aux_min<valor_min) valor_min=aux_min;
    fread(&aux_min,sizeof(float),1,ft);
}
if(valor_min<0) valor_min=0;
return(valor_min);
}

```

```

void dibuja_ejes(float maximo,float minimo,int x0,int y0,int x1,int y1,int intervalo_x,float intervalo_y)
{
    int ancho,inc,xi,yi;
    int alto;

    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    line(x0,y0,x0,y1);
    line(x0,y0,x1,y0);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
    alto=y0;
    ancho=x0;
    xi=x0;
    yi=y0+8;
    inc=0;
    while(inc<=n_datos){
        if((ancho<=600)&&(ancho>=x0)){
            line((ancho+intervalo_x),y0,(ancho+intervalo_x),435);
            ancho=ancho+intervalo_x;
        }
        inc=inc+1;
    }
    while(alto>=((maxy-(maximo*intervalo_y))-(maxy-y0))){
        line(x0,(alto-intervalo_y),(maxx/14.22),(alto-intervalo_y));
        alto=alto-intervalo_y;
    }
    rprintf(&xi,&yi,"%2f",minimo);
}

```

```

void dibuja_ejes_directo(float maximo,float minimo,int x0,int y0,int x1,int y1,float intervalo_yy)
{
    int xi,yi,ym,alto;

    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    line(x0,y0,x0,y1);
    line(x0,y0,x1,y0);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
    alto=y0;
    xi=x0;
    yi=y0+8;
    ym=y1-8;
    while(alto>=((maxy-(maximo*intervalo_yy))-(maxy-y0))){
        line(x0,(alto-intervalo_yy),(maxx/14.22),(alto-intervalo_yy));
        alto=alto-intervalo_yy;
    }
    if(maximo==30) rprintf(&xi,&ym,"%2f",maximo);
    if(maximo==40) rprintf(&xi,&ym,"%2f",maximo*30);
    rprintf(&xi,&yi,"%2f",minimo);
}

```

```

void dibuja_ejes_global(float maximo,float minimo,int x0,int y0,int x1,int y1,int intervalo_x,float intervalo_y,int numero)
{
    int ancho,inc;
    float alto;

    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    line(x0,y0,x0,y1);
    line(x0,y0,x1,y0);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
    alto=y0;
    ancho=x0;
    inc=0;
    while(inc<=numero){
        line((ancho+intervalo_x),y0,(ancho+intervalo_x),(maxy/1.103));
    }
}

```

```

inc=inc+1;
ancho=ancho+Intervalo_x;
}
while(alto>=((maxy-(maximo*intervalo_y*factor_escalay))-(maxy-y0))){
  line(x0,(alto-intervalo_y*factor_escalay),(maxy/14.22),((alto-intervalo_y)));
  alto=alto-intervalo_y*factor_escalay;
}
}

```

/*FUNCIONES AUXILIARES PARA LA PRESENTACION DE VENTANAS*/

```
void pantalla(int x,int y,int color_m,int color_f)
```

```
{
setlinestyle(SOLID_LINE,0,THICK_WIDTH);
marco(0,0,x,y,color_m);
setfillstyle(SOLID_FILL,color_f);
floodfill(5,5,color_m);
}

```

```
void marco(int xinicial,int yinicial,int xfinal,int yfinal,int color)
```

```
{
setcolor(color);
line(xinicial,yinicial,xinicial,yfinal);
line(xinicial,yinicial,xfinal,yinicial);
line(xfinal,yinicial,xfinal,yfinal);
line(xfinal,yfinal,xinicial,yfinal);
line(xinicial+3,yinicial+3,xinicial+3,yfinal-3);
line(xinicial+3,yinicial+3,xfinal-3,yinicial+3);
line(xfinal-3,yinicial+3,xfinal-3,yfinal-3);
line(xfinal-3,yfinal-3,xinicial+3,yfinal-3);
}

```

```
void marco1(int xinicial,int yinicial,int xfinal,int yfinal,int color)
```

```
{
setcolor(color);
line(xinicial,yinicial,xinicial,yfinal);
line(xinicial,yinicial,xfinal,yinicial);
line(xfinal,yinicial,xfinal,yfinal);
line(xfinal,yfinal,xinicial,yfinal);
line(xinicial,yinicial,xinicial,yfinal);
line(xinicial,yinicial,xfinal,yinicial);
line(xfinal,yinicial,xfinal,yfinal);
line(xfinal,yfinal,xinicial,yfinal);
}

```

```
int gprintf(int *xloc,int *yloc,char *fmt, ...)
```

```
{
va_list argptr;
char cad[140];
int cadena;

va_start(argptr,fmt);
cadena= vsprintf(cad,fmt,argptr);
outtextxy(*xloc,*yloc,cad);
*yloc += textheight("H")+2;
va_end(argptr);
return(cadena);
}

```

```
int dprintf(int *xloc,int *yloc,char *fmt, ...)
```

```
{
va_list argptr;
char cad[140];
int cadena;

```

```

va_start(argptr,format);
cadena= vsprintf(cad,fmt,argptr);
outtextxy("xloc,*yloc,cad);
va_end(argptr);
return(cadena);
}

int rprintf(int *xloc,int *yloc,char *fmt,...)
{
va_list argptr;
char cad[140];
float cadena;

va_start(argptr,format);
cadena=vsprintf(cad,fmt,argptr);
outtextxy("xloc,*yloc,cad);
va_end(argptr);
return(cadena);
}

pausa(int maxx,int maxy)
{
int c;
char *msj;

msj="Presione <ESC> para salir o cualquier tecla para continuar... ";
setcolor(color_marco);
line(0,462,maxx,462);
setfillstyle(SOLID_FILL,color_marco);
floodfill(38,466,color_marco);
setcolor(color_letra1);
settextstyle(DEFAULT_FONT,HORIZ_DIR, 1);
settextjustify(CENTER_TEXT,TOP_TEXT);
outtextxy(320,(textheight(msj)+(maxy-22)),msj);
c=getch();
if(ESC == c){
finaliza();
closegraph();
exit(1);
}
if( 0 == c ){
c = getch();
}
cleardevice();
}

void mensaje(int color_fondo,int color_letra,char *msj)
{
marco1(0,462,640,480,color_fondo);
setfillstyle(SOLID_FILL,color_fondo);
floodfill(38,466,color_fondo);
setcolor(color_letra);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(CENTER_TEXT,TOP_TEXT);
outtextxy(320,(textheight(msj)+460),msj);
}

void recuadro(int color_fondo,int color_marco,int color_letra,char *msj)
{
marco1(115,345,521,380,color_marco);
setfillstyle(SOLID_FILL,color_fondo);
floodfill(160,350,color_marco);
setcolor(color_letra);
outtextxy(314,360,msj);
}

```

```

}

siguiente_grafica(int maxx,int maxy,int color_f,int color_l)
(
char *msj;

msj="Presione cualquier tecla para pasar a la siguiente grafica...";
marco1(0,461,maxx,maxy,color_f);
setfillstyle(SOLID_FILL,color_f);
floodfill(38,466,color_f);
setcolor(color_l);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(CENTER_TEXT,TOP_TEXT);
outtextxy((maxx/2),(textheight(msj)+(maxy-(maxy/23))),msj);
}

void imp_menu_prin(int maxx,int maxy)
{
pantalla(maxx,maxy,color_marco,color_fondo);
marco1(16,14,628,32,color_marco);
setfillstyle(SOLID_FILL,color_marco);
floodfill(320,24,color_marco);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(LEFT_TEXT,BOTTOM_TEXT);
setcolor(color_letra2);
outtextxy(16,27,"T");
outtextxy(130,27,"C");
outtextxy(200,27,"V");
outtextxy(310,27,"P");
outtextxy(383,27,"E");
outtextxy(473,27,"G");
outtextxy(562,27,"S");
setcolor(color_letra1);
outtextxy(22,27,"emperatura");
outtextxy(138,27,"arga");
outtextxy(208,27,"elocidad");
outtextxy(318,27,"ar");
outtextxy(391,27,"spesor");
outtextxy(481,27,"lobal");
outtextxy(570,27,"etup");
}

void imp_submenu(int xs1,int xs2,int xs1m,int ys1m,int xs2m,int ys2m,int color)
{
marco1(xs1m,ys1m,xs2m,ys2m,color);
setfillstyle(SOLID_FILL,color);
floodfill(xs1m+6,ys1m+6,color);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(LEFT_TEXT,BOTTOM_TEXT);
setcolor(color_letra2);
outtextxy(xs1,70,"M");
outtextxy(xs1,96,"G");
outtextxy(xs1,122,"D");
outtextxy(xs1,148,"A");
outtextxy(xs1,174,"S");
outtextxy(xs1,200,"I");
setcolor(color_letra1);
outtextxy(xs2,70,"onitoreo");
outtextxy(xs2,96,"raficacion");
outtextxy(xs2,122,"irecto");
outtextxy(xs2,148,"brr Archivo");
outtextxy(xs2,174,"alvar Datos");
outtextxy(xs2,200,"mprimir");
}

```

```

void Imp_submenu_global(int x1,int x2,int x1m,int y1m,int x2m,int y2m,int color)
{
marco1(x1m,y1m,x2m,y2m,color);
setfillstyle(SOLID_FILL,color);
floodfill(x1m+6,y1m+6,color);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(LEFT_TEXT,BOTTOM_TEXT);
setcolor(color_letra2);
outtextxy(x1,70,"M");
outtextxy(x1,96,"G");
outtextxy(x1,122,"A");
outtextxy(x1,148,"S");
outtextxy(x1,174,"I");
setcolor(color_letra1);
outtextxy(x2,70,"onitoreo");
outtextxy(x2,96,"raficacion");
outtextxy(x2,122,"brir Archivo");
outtextxy(x2,148,"alvar Datos");
outtextxy(x2,174,"mprimir");
}

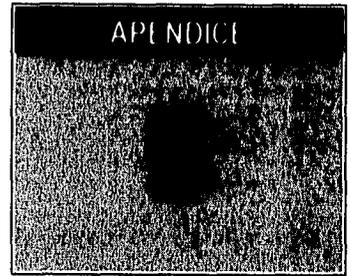
void Imp_submenu_setup(int x1,int x2,int x1m,int y1m,int x2m,int y2m,int color)
{
marco1(x1m,y1m,x2m,y2m,color);
setfillstyle(SOLID_FILL,color);
floodfill(x1m+6,y1m+6,color);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
settextjustify(LEFT_TEXT,BOTTOM_TEXT);
setcolor(color_letra2);
outtextxy(x1,70,"1");
outtextxy(x1,96,"2");
outtextxy(x1,122,"3");
outtextxy(x1,148,"4");
outtextxy(x1,174,"5");
setcolor(color_letra1);
outtextxy(x2,70,"(COMM 1)");
outtextxy(x2,96,"(COMM 2)");
outtextxy(x2,122," Adq.1 dt");
outtextxy(x2,148," Adq.10 dt");
outtextxy(x2,174," Adq.100 dt");
}

void finaliza(void)
{
char *msj1;

msj1=" FIN DEL SISTEMA ";
clearviewport();
pantalla(maxx,maxy,color_marco,color_fondo);
marco1(107,120,533,300,color_marco);
setfillstyle(SOLID_FILL,color_fondo1);
floodfill(128,160,color_marco);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,4);
setcolor(color_letra);
outtextxy(320,(textheight(msj1)+160),msj1);
getch();
}

```

PROGRAMA FUENTE PARA EL MICROCONTROLADOR



```
*****
*** PROGRAMA MONITOR PARA EL SISTEMA DE ***
*** ADQUISICION Y DESPLIEGUE DE      ***
*** INFORMACION DE UNA LAMINADORA    ***
*** HILLE DE 25 TONELADAS            ***
*****
```

** IGUALDADES DE REGISTROS

```
REGBAS EQU $1000 ;Origen de registros
PORTA EQU $00
PORTD EQU $08
DDRD EQU $09
TIC1 EQU $10
TCTL2 EQU $21
TMSK1 EQU $22
TFLG1 EQU $23
TMSK2 EQU $24
TFLG2 EQU $25
BAUD EQU $2B
SCCR1 EQU $2C
SCCR2 EQU $2D
SCSR EQU $2E
SCDR EQU $2F
ADCTL EQU $30
ADR1 EQU $31
ADR2 EQU $32
ADR3 EQU $33
ADR4 EQU $34
OPTION EQU $39
```

** ASIGNACION DE VARIABLES RAM

```
ORG $0000
IC1DUN RMB 1
IC1MOD RMB 1
OVCNT1 RMB 1
RES1 RMB 2
```

** PROGRAMA PRINCIPAL

```
INICIO: ORG $F800
        LDX #REGBAS
        LDS #$FF
```

```
LDAA  #$02
STAA  DDRD,X
LDAA  #$32
STAA  BAUD,X
CLR   SCCR1,X
LDAA  #$0C
STAA  SCCR2,X
BCLR  OPTION,X,$40
BSET  OPTION,X,$80
JSR   RET
LDAA  #$33
STAA  ADCTL,X
```

```
ET11: LDAA  #$20
      BITA  SCSR,X
      BEQ   ET11
```

* LECTURA DEL PUERTO SERIAL

```
DAT:  LDAA  SCSR,X
      LDAB  SCDR,X
      LDAA  SCSR,X
      CMPB  #$43
      BEQ   CARGA
      CMPB  #$45
      BEQ   ESP
      CMPB  #$47
      BEQ   GLOB
      CMPB  #$50
      BEQ   PAR
      CMPB  #$54
      BEQ   TEMP
      CMPB  #$56
      BEQ   VEL
      JMP   ET11
```

* MONITOREO DE TEMPERATURA

```
TEMP: LDAB  ADR1,X
      JSR   TRANS
      LDAA  SCDR,X
      CMPA  #$54
      BEQ   TEMP
      JMP   DAT
```

* MONITOREO DE CARGA

```
CARGA: LDAB  ADR2,X
      JSR   TRANS
      LDAA  SCDR,X
      CMPA  #$43
      BEQ   CARGA
      JMP   DAT
```

* MONITOREO DE PAR

```
PAR:  JSR   ICF
```

```
JSR  TRANS
LDAB  ADR3,X
JSR  TRANS
LDAA  SCDR,X
CMPA  #$50
BEQ  PAR
JMP  DAT
```

*** MONITOREO DE ESPESOR**

```
ESP:  LDAB  ADR4,X
      JSR  TRANS
      LDAA  SCDR,X
      CMPA  #$45
      BEQ  ESP
      JMP  DAT
```

*** MONITOREO DE VELOCIDAD**

```
VEL:  JSR  ICF
      JSR  TRANS
      LDAA  SCDR,X
      CMPA  #$56
      BEQ  VEL
      JMP  DAT
```

*** MONITOREO GLOBAL**

```
GLOB: LDAB  ADR1,X
      JSR  TRANS
      LDAB  ADR2,X
      JSR  TRANS
      LDAB  ADR4,X
      JSR  TRANS
      JSR  ICF
      JSR  TRANS
      LDAB  ADR3,X
      JSR  TRANS
      LDAA  SCDR,X
      CMPA  #$47
      BEQ  GLOB
      JMP  DAT
```

*** SUBROUTINA DE RETARDO**

```
RET:  PSHY
      LDY  #$FF
DELAY: DEY
      BNE  DELAY
      PULY
      RTS
```

*** SUBROUTINA DE CAPTURA DE ENTRADA**

ICF: LDAA # \$10
 STAA TCTL2,X
 LDAA # \$FF
 STAA IC1MOD
 CLR IC1DUN
 BCLR TFLG1,X,\$FB
 BCLR TFLG2,X,\$7F
 BSET TMSK1,X,\$04
 BSET TMSK2,X,\$80
 CLI

ETI3: LDAA IC1DUN
 BEQ ETI3
 SEI

LDD RES1
 CPD # \$00FF
 BLT ETI4
 INC OVCNT1
 ETI4: LDAB OVCNT1
 RTS

SV3TOF: TST IC1MOD ; Rutina de servicio (TOF)
 BMI OU3TOF
 INC OVCNT1
 OU3TOF: LDAA # \$80
 STAA REGBAS+TFLG2
 RTI

SV3IC1: LDX #REGBAS ; Rutina de servicio (ICF)
 INC IC1MOD
 BNE NO1ST3
 CLR OVCNT1 ; Primer nivel
 LDD TIC1,X
 STD RES1
 BMI OU3IC1
 LDAA TFLG2,X
 BPL OU3IC1
 DEC OVCNT1
 BRA OU3IC1

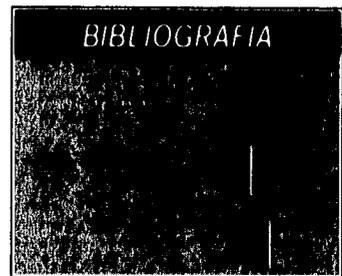
NO1ST3: LDD TIC1,X ; Segundo nivel
 BMI ARNOVI
 TST TFLG2,X
 BPL ARNOVI
 INC OVCNT1
 ARNOVI: SUBD RES1
 STD RES1
 BCC RES1OK
 DEC OVCNT1
 RES1OK: BCLR TCTL2,X,\$30
 LDAA #1
 STAA IC1DUN

OU3IC1: BCLR TFLG1,X,\$FB
RTI

* SUBROUTINA DE TRANSMISION

TRANS: LDAA SCSR,X
STAB SCDR,X
LDAA SCSR,X
TC: LDAA #\$40
BITA SCSR,X
BEQ TC
RTS

END



- **COUGHLIN - DRISCOLL**; "Amplificadores Operacionales y Circuitos Integrados Lineales"; Edit. Prentice - Hall; México 1993.
- **COUGHLIN - DRISCOLL - VILLANUCCI**; "Data Acquisition and Process Control with the M68HC11 Microcontroller"; Edit. Merrill; U. S. A. 1994.
- **DONALD - BAKER**; "Gráficas por computadora"; Edit. Prentice - Hall; México 1994.
- **DVORAK - ANIS**; "Telecomunicaciones para PC"; Edit. Borland - Osborne/McGraw-Hill; México 1992.
- **GILBERT - HELD**; "Data Communications Networking Devices"; Edit. Wiley; U. S. A. 1990.
- **KNOWLTON**; Manual Estándar del Ingeniero Electricista, Volúmenes I y II; Edit. Labor; España 1963
- **MALVINO, Albert**; "Principios de Electrónica"; Edit. McGraw - Hill; México 1991.
- **MANDADO, Enrique**; "Sistemas Electrónicos Digitales"; Edit. AlfaOmega; España 1989.
- **MOTOROLA**; "Fast and LS TTL Databook".
- **NATIONAL SEMICONDUCTOR**; "CMOS Logic Databook".

- **OGATA, Katsuhiko**; "Ingeniería de control Moderna"; Edit. Prentice - Hall; México 1984.
 - **RODRIGUEZ, Francisco**; "Dinámica de sistemas"; Edit. Trillas; México 1983.
 - **SCHILDT, Herbert**; "C, Guía para usuarios expertos"; Edit. Osborne/McGraw-Hill; España 1990.
 - **SCHILDT, Herbert**; "Programación en Turbo C"; Edit. Borland - Osborne/McGraw-Hill; España 1991.
 - **SCHILDT, Herbert**; "Turbo C/C++, Manual de Referencia"; Edit. Borland - Osborne/McGraw-Hill; México 1992.
 - **TORRES, Manuel**; "Microprocesadores y Microcontroladores Aplicados a la Industria"; Edit. Paranainfo; España, Madrid 1989.
 - **SOMMERVILLE, Ian**; "Ingeniería de Software"; Edit. Addison - Wesley Iberoamericana; México 1988.
 - **WOLF, Stanley**; "Guía para Mediciones Electrónicas y Prácticas de Laboratorio"; Edit. Prentice - Hall; México 1991
-