

T
621.380.422
A958



**ESCUELA SUPERIOR
POLITECNICA DEL LITORAL
FACULTAD DE INGENIERIA ELECTRICA**



**TELEMETRIA: Control digital remoto de la
velocidad de un motor DC usando comunicación
celular.**

**Proyecto de Tópico de Graduación:
"SISTEMAS CELULARES"**

**Previa a la obtención del Título de:
INGENIERO ELECTRICO
ESPECIALIZACION ELECTRONICA**

PRESENTADO POR:

**Ronald Avilés Lozano
Mauricio Alminate Vásquez
Giovana Minchala Aguirre
Raúl Santamaría Salazar**

GUAYAQUIL - ECUADOR

1996

INDICE

AGRADECIMIENTO _____

INTRODUCCION _____

CAPITULO 1 _____

OBJETIVOS Y GENERALIDADES _____ 9

OBJETIVOS _____ 9

GENERALIDADES _____ 10

CAPITULO II _____

MEDIOS DE COMUNICACION _____ 11

OPCIONES CONMUTADAS Y NO CONMUTADAS _____ 11

TABLA COMPARATIVA _____ 12

VELOCIDAD DE CANAL Y VELOCIDAD BINARIA _____ 13

ESQUEMA DE UNA COMUNICACIÓN _____ 14

CAPITULO III

EQUIPOS DE COMUNICACION 16

SISTEMAS SÍNCRONO - ASÍNCRONO	18
SIMPLEX, HALF Y FULL DUPLEX	19
PROTECCIÓN CONTRA ERRORES	19
PROCEDIMIENTO DE ENLACE	21

CAPITULO IV

CONTROL DE VELOCIDAD DE UN MOTOR DC 24

ANÁLISIS DEL CONTROL DE VELOCIDAD DE UN MOTOR DC	24
MÉTODO DE CONTROL DE VELOCIDAD POR ARMADURA	25
CONTROLADOR PROPORCIONAL INTEGRAL (PI)	25

CAPITULO V

CONFIGURACION DEL SOFTWARE DE COMUNICACION 28

INICIANDO UNA SESIÓN	29
CONFIGURANDO EL PUERTO DE COMUNICACIÓN	30
CONFIGURANDO LOS MODEMS	31
HACIENDO UNA LLAMADA	32

CAPITULO VI

MONTAJE DE LA PLATAFORMA DE HARDWARE 33

CAPITULO VII

<u>PROGRAMA DE CONTROL</u>	34
PROGRAMA PRINCIPAL	35
EJECUTANDO EL PROGRAMA DE CONTROL	53
CARÁTULA 2	54
CARÁTULA 3	55

CAPITULO VIII

<u>DETALLES DEL DISEÑO</u>	56
DISEÑO DIGITAL	57
DIAGRAMA ESQUEMÁTICO DEL CIRCUITO DE FUERZA	58
LISTA DE COMPONENTES	59
<u>CONCLUSIONES</u>	61
<u>OBSERVACIONES</u>	62
<u>BIBLIOGRAFIA</u>	63
<u>APENDICES</u>	64
CONVERTIDOR DIGITAL ANALOGICO DAC0830	64
CONVERTIDOR ANALOGICO DIGITAL DAC0804	65
INTERFACE PERIFERICA PROGRAMABLE 8255A	66

AGRADECIMIENTO

Para poder llegar a la culminación de este proyecto hicieron falta a más de los conocimientos aplicados de Ingeniería, el apoyo de personas que desinteresadamente contribuyeron especialmente en el préstamo de los equipos utilizados.

Es por esto que las personas que conformamos este grupo guardamos nuestro mayor agradecimiento a las siguientes personas:

Ing. Carlos Villafuerte, Decano de la Facultad de Ingeniería Eléctrica e Ing. Hugo Villavicencio, Director del Laboratorio de Microprocesadores, que nos facilitaron : el motor de paso, las tarjetas de interface y el protoboard donde se implementó el proyecto.

Ing. Blas Espinel: Que puso a nuestra disposición una de las computadoras personales, los equipos de comunicación y sus conocimientos del área de comunicación celular.

Ing. Roberto Valverde: Que nos facilitó un computador personal y los programas compiladores para la aplicación.

Carlos Mawyrín: Los modems para comunicación celular, sin los cuales nuestra aplicación en su forma remota no hubiese tenido la estabilidad conseguida.

Por demás está el agradecimiento a nuestras familias que nos apoyaron en todo momento y confiaron en nosotros.

A handwritten signature in black ink, appearing to be 'C. J. S.', written in a cursive style. The signature is located in the center of the page.



INTRODUCCION

La productividad y el rendimiento de las personas y organizaciones se ha visto engrandecido debido al gran impacto que tienen los computadores en la sociedad desde hace un par de décadas.

Cada vez más personas utilizan las redes de computadores para sus relaciones tanto profesionales, como para su vida doméstica. La tendencia actual, se acelera a medida que los usuarios descubren cada vez más beneficios que pueden ofrecer los diferentes equipos en el mercado.

Debido a lo mencionado anteriormente se decidió aplicar las ventajas ofrecidas en una aplicación práctica denominada :Telemetría ó Manejo remoto de dispositivos.

Accesaremos remotamente a una computadora que tendrá conectado mediante una tarjeta de interfaz de datos un circuito montado en un PPI. El diseño digital de éste nos permitirá controlar la velocidad de un motor de paso.

Presentamos una breve descripción teórica de los equipos utilizados, así como un repaso de la técnica de comunicación celular ya conocida por nosotros como último paso, el algoritmo de control, implementado en lenguaje Assembler que permite en su modo ejecutable el ingreso de

velocidades comprendidas entre 100 y 1200 rpm, de la manera más simple para el usuario final.

Cabe mencionar que dado que trabajamos con líneas celulares para transmisión de datos, nos vemos limitados a la velocidad que proporcionan los modems para este tipo de comunicación, es decir hasta 9.6 Kbps, que comparado con la velocidad que alcanzan en otras aplicaciones resulta bastante baja.

CAPITULO 1

OBJETIVOS Y GENERALIDADES



Objetivos

El proyecto Telemetría tiene como objetivo primordial permitir el control a distancia de dispositivos. Utilizar un recurso como éste, nos proporciona importantes ventajas que pudieran resumirse en los siguientes puntos:

- ↳ Las organizaciones modernas, suelen estar dispersas geográficamente, con sus oficinas situadas en diferentes lugares de un país, e inclusive del mundo. Una persona situada en un lugar lejano, gracias a la telemetría podría en determinada situación manejar, acceder a información y llegado el caso, accionar fácilmente dispositivos colocados en ubicaciones lejanas.

- ↳ El uso de la telemetría permite crear un entorno de trabajo flexible, dando lugar a que se pueda manejar ya sea procesos

simples, o dado el caso de emergencia procesar información crítica desde cualquier lugar, a puntos que puedan resultar de difícil acceso. Empleados que se desplacen a oficinas lejanas, pueden usar sus computadores portables, y haciendo uso del servicio telefónico pueden comunicarse rápidamente con sus compañías.

Generalidades

Este proyecto ha sido diseñado para controlar y monitorear la velocidad de un motor DC . Este hace uso de la tarjeta de interfaz de datos para enviar las señales de control al motor .

El programa tiene como objetivo ser una herramienta didáctica para los estudiantes de la Facultad de Ingeniería Eléctrica y Computación; y además presentar una iniciativa para realizar experimentos de control de cualquier tipo .

El proyecto está formado por programa de control en lenguaje ensamblador , un programa residente en memoria para establecer la comunicación entre computadoras llamado Carbon Copy . Además , una circuiteria externa constituida por la tarjeta de interfaz de datos ,el hardware que controla al motor y el motor DC .

CAPITULO II

MEDIOS DE COMUNICACION

Como muchos equipos terminales de comunicación de datos o de conmutación de ellos se conectan vía canal telefónico, y debido a que el medio de comunicación celular va a ser utilizado en este proyecto resulta útil dar un repaso de lo que es el sistema telefónico celular.

Opciones conmutadas y no conmutadas

Para establecer una comunicación permanente entre dos lugares, un abonado puede escoger entre una línea privada o una línea con dedicación exclusiva (que también se pueden conmutar vía centrales de conmutación privadas). Las líneas privadas no conmutadas son de mucha utilidad cuando los abonados no se pueden permitir retardos, o el bloqueo de una llamada cuando todos los circuitos están ocupados. Además los usuarios con una carga de tráfico de varias horas por conexión pueden ahorrar dinero con el uso de una línea con dedicación exclusiva. A continuación se enumeran las ventajas y desventajas del uso de líneas conmutadas normales y de los circuitos con dedicación exclusiva.



POLYTECNICA DEL LITORAL

BIBLIOTECA

CENTRAL

Tabla comparativa

	Conmutadas	No Conmutadas
Ventajas	Flexibles	Admite mayor volumen de tráfico
	Económicas para bajo volumen de tráfico	Posibilidad de mayor calidad
		No existe bloqueo, ni señal de ocupado
Desventajas	Respuesta lenta	Pérdida de flexibilidad cuando la línea se avería
	Baja calidad	
	Posibilidad de bloqueo	
	Costosa para alto volumen de tráfico	Costosa para bajos volúmenes de tráfico

Velocidad de canal y velocidad binaria

La forma más simple utilizada por los dispositivos para enviar un número binario por un canal de comunicaciones es conmutar la señal eléctricamente, o producir tensiones altas y bajas para representar unos y ceros. Sin tener en cuenta la forma de representar los datos mediante estados de conmutación, niveles de tensión o direcciones en el flujo de la corriente), el canal de comunicaciones se describe por su capacidad, es decir, el número de bits por segundo que se transmiten, que se representa bit/s, bps o bs. Cuando se habla de 4800 bit/s en línea, quiere decir que se envían 4800 bits por segundo por el canal. Un bit es simplemente la representación de un estado eléctrico, óptico o electromagnético de la línea. Para codificar un carácter de usuario se necesitan básicamente 7 u 8 bits, o lo que se conoce como un byte.

Un canal de comunicación de datos que utilice líneas telefónicas convencionales, es muy lento. A efectos de comparación, los canales se clasifican como de baja, media y alta velocidad:

Baja velocidad: 0-600 bps

Velocidad media: 600-4800 bps

Alta velocidad: 4800-9600 bps

Solo en los últimos años, han aparecido equipos capaces de transmitir con éxito a 9.6 Kbps por canal telefónico. Las velocidades típicas más allá de los 9600 bps son 14400, 19200, 56000 y 64000 bps y 1.544 Mbps y 2048 Mbps en Europa. El canal de 1.544 Mbps es muy empleado en canales digitales de alta velocidad y en centros de conmutación digitales.

Esquema de una comunicación

En todo sistema de comunicación, podemos diferenciar una función de transporte y otra de tratamiento de la información, distribuidas entre:

↳ ETTD (Equipo Terminal de Tratamiento de Datos): Es la fuente o destino de la información; incorpora el controlador de comunicaciones con interface con el ETCD; y

↳ ETCD (Equipo Terminal del Circuito de Datos): Es el encargado de adecuar la información procedente del ETTD a las características de la línea de transmisión, y dispone de una interface con el ETTD y otro con la línea (establece y libera el circuito). Los modems son un equipo ETCD.

La conexión entre dos ETCD forma el circuito de datos, mientras que la conexión entre los ETTD, por medio de sus respectivos ETCD, forma el enlace de datos. El circuito de datos puede ser punto a punto (conexión entre dos ETCD) o multipunto (un ETCD central se conecta a varios ETCD por una misma línea).

Del enlace de datos se encargan los controladores de comunicaciones, que definen las reglas (procedimientos de enlace) que gobiernan la comunicación. Deben saber qué hacer en caso de error, cómo codificar/decodificar los datos enviados /recibidos, cómo mantener el sincronismo con el equipo remoto, etc. Parte de estas funciones reciben el nombre de servicio, puesto que no se transmite información final, útil para el tratamiento, sino aquella necesaria para iniciar, mantener y liberar la conexión. Estas funciones de Servicio, suelen realizarse a través de códigos de control añadidos a la información de Interés.

La información puede enviarse caracter a caracter o en forma de bloques (como conjunto de caracteres). En este último caso, el bloque es la unidad de información con la que trabaja el controlador, de forma que si se produce un error, todo el bloque es rechazado.

Debemos primeramente diferenciar entre señales analógicas y señales digitales. Una señal digital es aquella que toma una serie de valores fijos de forma que entre ellos no hay ningún otro valor intermedio. En los ordenadores se trabaja con los valores de 0 y 1 (señal digital binaria). Una señal analógica es la que varía de forma continua, es decir, entre dos valores, siempre se puede encontrar otro intermedio.

CAPITULO III

EQUIPOS DE COMUNICACION

La línea telefónica (analógica) por sus características, no puede trabajar con las señales digitales binarias utilizadas por los ordenadores, por lo que es necesario transformarlas en señales analógicas compatibles con la línea. El equipo que realiza esta transformación es el **MODEM** (Modulador/DEModulador), un ETCD emisor/receptor.

Además, convierte la Información paralelo en Información secuencial. La señal emitida por el ETCD a la línea tarda un tiempo en llegar al receptor (tiempo de propagación).

La modulación transforma la señal digital binaria en analógica. La demodulación convierte la señal analógica en digital binaria. Para que dos ordenadores se puedan comunicar por línea telefónica, ambos deben estar ocupados con un MODEM.

El mensaje de datos (bits), codificado según el alfabeto ASCII, EBCDIC, etc) que pasa del ETD al ETCD debe ser transcodificado por este último para adecuarlo a la línea de comunicación. A cada símbolo (estados de la señal que representan la Información binaria) que se envía por la línea, se llama nivel. Estos no tienen por qué corresponderse una a uno con cada bit, ni tampoco la

decodificación de un símbolo depende solo de su valor. Puede asociarse varios bits con un solo nivel, y codificarse cada símbolo en función del precedente (decodificación diferencial). Al enviar los bytes, se suele empezar por los bits de mayor peso (del bit 7 al bit 0).

El ETCD emisor recibe el mensaje de datos del ETD, lo aleatoriza, codifica y modula, mientras que el ETCD receptor lo demodula y decodifica. Hay modems que codifican la información digital binaria en otra también digital, para ser enviadas por líneas de comunicación digitales. Estos ETCD digitales reciben el nombre de ETCD Banda Base, y pueden ser síncronos o asíncronos.

Pueden decodificar la señal digital del ordenador de distintas formas, siendo las más comunes:

NRZ
Bifase Diferencial
Miller
Bipolar de orden 1
Bipolar de orden 2
Bipolar de alta densidad
De valencia n

La señal digital se caracteriza por el período de bit (T) y por el tiempo entre dos transiciones eléctricas (t). La velocidad de una señal en bits por segundo es el inverso de t ($1/t$).

La trascodificación digital pretende: enviar el número máximo de bits por estado, compactar la banda de frecuencias de la señal; y facilitar la sincronización, detección de errores, etc,

Las señales analógicas se caracterizan por su amplitud, frecuencia y fase. Si pensamos en un sonido, la amplitud constituiría su intensidad o volumen y la frecuencia sería el tono. Para entender lo que es la fase, pensemos en un reloj con dos manecillas, cada una desplazándose a la misma velocidad. La fase vendría dada por la diferencia de ángulo que forman las manecillas en un instante dado.

La señal analógica es manipulable en tres variables, la amplitud, la frecuencia y la fase. Cuando enviamos la información digital binaria (0 y 1), existirá una correspondencia entre ésta y los valores de la señal analógica.

En el mercado hay equipos de 300, 1200, 2400, 4800 y 9600, siendo estos dos últimos sólo utilizables en líneas de comunicación especialmente diseñadas.

Sistemas Síncrono - Asíncrono

Los sistemas síncronos y asíncronos, pueden ser tanto serie como paralelo (vectorial). La mayoría de los modems, por utilizar línea telefónica conmutada, emplean un sistema asíncrono de comunicación. En estos sistemas cada dato (byte) se envía secuencialmente, precedido por un bit de arranque y seguido por un bit de stop. El tiempo entre dato y dato es variable.

El bit de arranque tiene por misión activar en el equipo receptor la lectura de los datos (bits) enviados. El bit de stop deja al receptor en un estado de espera.

Además del sistema asíncrono está el síncrono, cuya diferencia con el anterior radica en que tanto el ordenador emisor como el receptor quedan sincronizados, es decir, sus ciclos de lectura/escritura de datos (bits) son coincidentes. Además los bits se transmiten en grupos llamados tramas.

En las transmisiones asíncronas cada "palabra" enviada o recibida, está constituida por:

- Bit de arranque (Start)
- El dato (byte); de 5 a 8 bits
- Bit de paridad;
- 1 o 2 bits de stop.

Simplex, Half y Full Duplex

En el modo Simplex la comunicación tiene lugar sólo en una dirección; el receptor solo recibe. El modo half duplex indica que la comunicación es secuencial, es decir cuando un ordenador envía el otro recibe. El modo Full duplex implica una comunicación simultánea; de esta forma, emisor y receptor dialogan simultáneamente. Se puede utilizar líneas de dos o cuatro hilos. En este último caso es más caro, pero se reducen los tiempos de retorno.

Protección contra errores

En toda transmisión pueden aparecer errores. Se determina la tasa de error por la relación entre el número de bits erróneos y los bits

totales. Se denomina error residual al número de bits erróneos no corregidos en relación al total de bits enviados. Las señales emitidas pueden sufrir dos tipos de deformación; atenuación (reducción de su amplitud); y desfase, siendo ésta última la que más afecta a la transmisión. Otros factores que también afectan a la señal son: ruido blanco (por los componentes eléctricos de los transformadores); ruido impulsivo; ecos; diafonías; etc. Las distorsiones físicas de la señal las trata el ETCD y los problemas a nivel de bit los trata el ETD. En ambos casos se trabaja con datos binarios a través del enlace. Los errores se pueden detectar y/o corregir. La corrección la puede realizar el propio decodificador (corrección directa) o se realiza por retransmisión.

A los datos enviados se les asocian bits de control (se añade redundancia al mensaje). Estos se pueden calcular por cada bloque de datos, o en función de bloques recurrentes. Como ejemplos de procedimientos de control de error se pueden citar:

- Control de Paridad por caracter
- Control de Calidad por Matriz de caracteres
- Retransmisión con paro y espera (ARQ-ACK)
- Retransmisión Continua: (ARQ-NAK)
- Retransmisión con repetición selectiva

El rendimiento de un código de control viene dado por el número de bits de cada bloque, entre los bits del bloque más los bits de control.

Los modems suelen incluir ecualizadores (filtros) para reducir la interferencia entre símbolos (interferencia debida al efecto de otros símbolos adyacentes sobre el que se está recibiendo).

Procedimiento de enlace

Una vez establecido el circuito de datos entre los ETC, el intercambio de Información entre los ETD se gestiona a través del Enlace.

El enlace se define tanto a nivel físico (conexión con el circuito de datos), como lógico (gestión de la transmisión de la información). Además, actúa como interface entre la transmisión y el intercambio de datos. Los enlaces pueden ser:

↳ Punto a punto: Conecta dos ETD

↳ Multipunto: Un mismo enlace es usado para conectar varios ETD central (primario). Puede ser centralizado (la estación central decide con quién comunicar) o no centralizado (se va cediendo el control por un orden preestablecido a las estaciones secundarias y se llama hub polling)

↳ De bucle: Es un enlace multipunto en el que cada extremo del mismo se conecta al ETD central.

El intercambio de información sel enlace puede ser unidireccional (one way, solo en un sentido), bidireccional alternativo (en uno u otro sentido cada vez), o bidireccional simultáneo (en ambos sentidos a la vez). Las funciones que desempeña un enlace son:

↳ Estructuración de los Datos: En una transmisión asíncrona, los bits se organizan en caracteres. En las síncronas lo hacen en tramas (bloques de bits).

- ↳ Delimitar e Identificar los datos: La delimitación de un bloque puede realizarse por caracteres de control o por secuencias especiales de bits. El enlace debe ser transparente al código, no confundiendo los delimitadores con la información útil. Puede que grupos de bloques deban ser delimitados para formar mensajes a nivel de tratamiento. Los bloques suelen numerarse para identificarlos.

- ↳ Conocer origen y destino de la Información: Caso de enlaces multipunto, es necesario conocer a qué estación van, y de que estación vienen los datos. Para ello, cada estación es identificada con una dirección.

- ↳ Control del Enlace: Gestiona la transferencia de información; utiliza códigos de control o secuencias binarias específicas debiéndose cumplir el principio de transparencia. Se debe supervisar el enlace, detectar errores, conocer el estado de la conexión, etc. En transmisiones asíncronas el número de comandos de control es pequeño; en las síncronas es grande.

- ↳ Protección contra Errores: Detecta y/o corrige los errores que aparezcan. A esta función le corresponde generar los acuses de recepción positivos o negativos de los mensajes enviados de fuente a destino.

- ↳ Recuperación: Se encarga de recuperar fallos, tanto de los datos como de la comunicación (corte de conexión, etc).

↳ Interface con los Medios de Transmisión: El enlace debe adecuarse al medio: (equipos y canal) y a la velocidad de transmisión; sincronizar emisor y receptor (en modems síncronos la señal de reloj la da ETTD; también se utilizan códigos de sincronismo a través de bloque); y responder a errores. Debe gestionar la conexión entre ETTD y ETCD (Ej. V.24, V.28, V.35, V.10, V.25, RS-232, X.21, etc).

↳ Interface con el tratamiento de datos: El enlace debe adecuar los datos a la aplicación en curso. Puede ser conversacional, tratamiento por lotes (batch), de tiempo real, de transferencia de ficheros, transaccional, etc. Debe adecuarse al tráfico de datos (volumen, flujo, etc) y a los tiempos de respuesta (tiempo entre petición y respuesta).

CAPITULO IV

CONTROL DE VELOCIDAD DE UN MOTOR DC

Análisis del control de velocidad de un motor DC

Los motores de CC son especialmente indicados para aplicaciones que exigen muchos cambios de velocidad, por arriba y por debajo de los valores nominales, debido principalmente a su facilidad de control y gran versatilidad.

Este control de velocidad se realiza evidentemente a voluntad del operador quien realiza los ajustes necesarios de resistencia y de tensión de que dispone para que el motor varíe o no de velocidad con las alteraciones de la carga, para un ajuste dado de resistencia o de tensión. La regulación de la velocidad dependerá de estos parámetros y del tipo particular del motor utilizado.

La velocidad de un motor DC está definida por:

$$w = (E_a - R_a \cdot I_a) / K_a \cdot \phi$$

Existen dos métodos generales para gobernar la velocidad de un motor DC:

↳ Alterando la tensión en los terminales de la armadura

↳ Variando el flujo magnético

La segunda opción no ofrece una gama amplia de variación puesto que se encuentra dificultades de dimensionado del motor. Además, a medida que disminuye el campo de reacción de armadura existe la posibilidad de ocasionar inestabilidad. Por lo tanto la primera opción es una mejor manera de controlar la velocidad del motor.

Método de control de velocidad por armadura

Siendo la velocidad proporcional al voltaje de armadura, un control sobre este voltaje implica un control de la velocidad.

Se ha utilizado la realimentación para regular la velocidad. La señal realimentada de velocidad se compara con la velocidad referencial y de allí se obtiene una señal de error. Esto determina un mayor o menor voltaje en los terminales del motor, logrando un aumento o disminución de la velocidad del mismo.

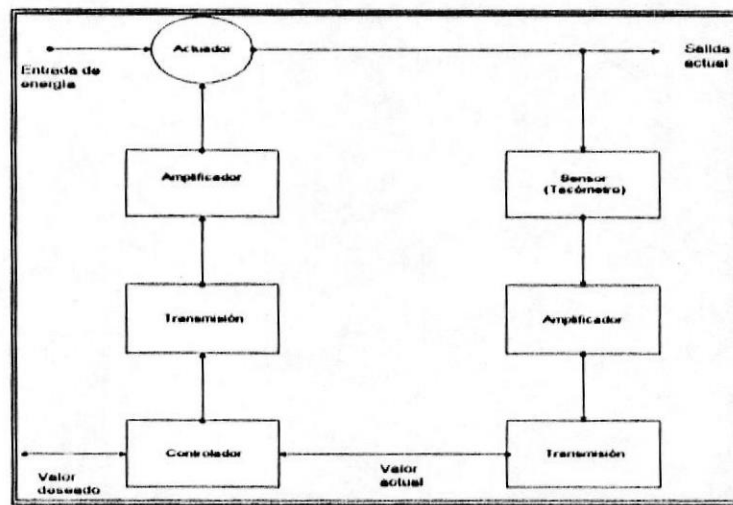
Controlador proporcional integral (PI)

El controlador provee la inteligencia del sistema. La figura muestra un sistema de control automático. El controlador tiene dos entradas. Una indica el valor deseado del parámetro (V_d), la otra señal de entrada indica el valor actual del parámetro (medida por un sensor, acondicionada y transmitida al controlador), es llamada la variable del proceso (V_p). La diferencia entre ellas se llama Error (E):

$$E = V_d - V_p$$

Es la tarea del controlador del proceso de actuar sobre el error y generar una señal que causará que el proceso sea modificado de tal manera que se mantenga el error igual a cero

Figura A



La respuesta del controlador a una señal de error es denominada su modo de control o ley de control. Para nuestro caso particular que se trata de un controlador digital, el término más apropiado es algoritmo de control.

Se ha seleccionado el modo de operación de un controlador proporcional integral; su expresión analítica es:

$$V_{out} = K_p \cdot V_{error} + K_i \int V_{error} \cdot dt + V_o$$

Para obtener la función de transferencia del controlador PI, tomamos la transformada de Laplace de la ecuación anterior:

$$\mathbf{V_{out} = K_p.V_{error} + (K_i .V_{error}) / S}$$

$$\mathbf{V_{out} / V_{error} = K_p + K_i / S}$$

Esta ecuación define la función de transferencia del controlador PI.

CAPITULO V

CONFIGURACION DEL SOFTWARE DE COMUNICACION



Para la instalación del software de comunicación escogido por el grupo (**Carbon Copy**), recomendamos el siguiente equipo:

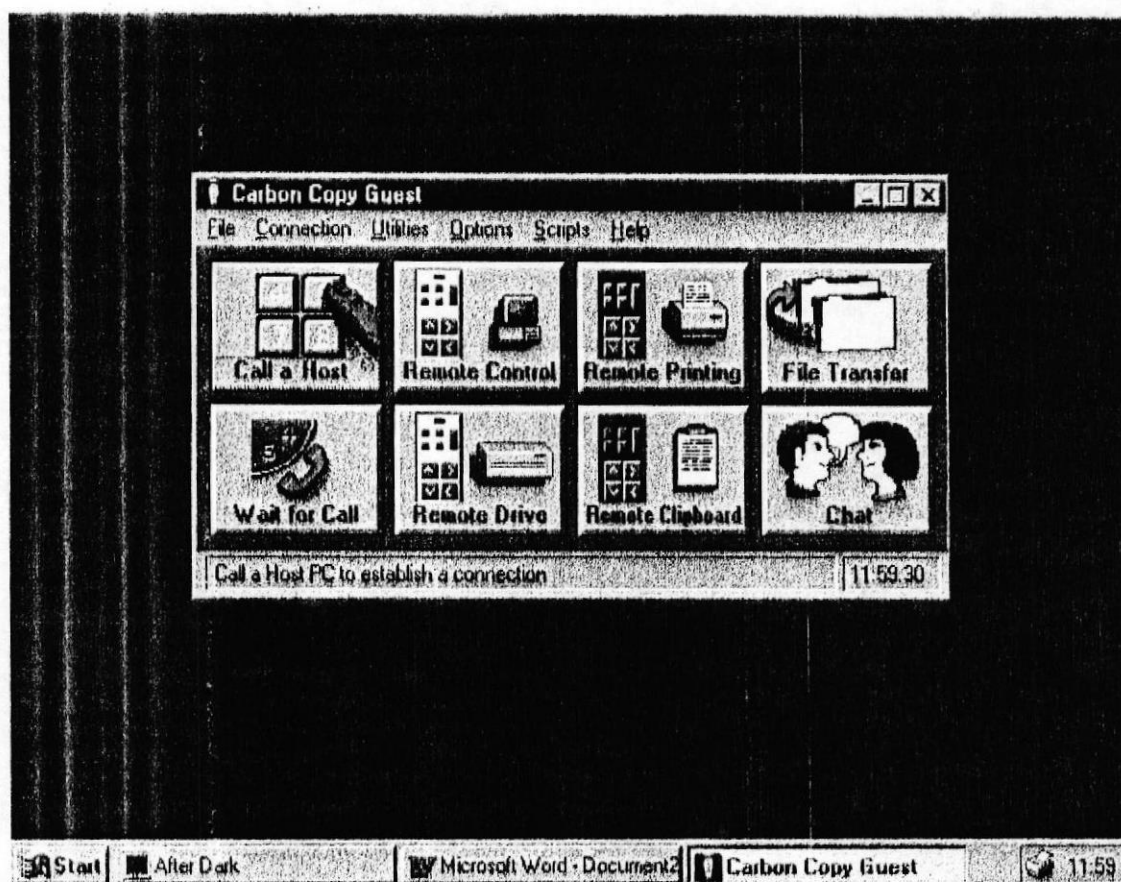
Computadores Personales con Procesador 80486; memoria RAM 8MB;
Monitores VGA, Espacio libre en Disco Duro 20MB

En el Disco duro de los dos computadores se debe instalar el programa de Comunicaciones Carbon Copy (son 4 diskettes de instalación de 3 ½ pulgadas).

El programa de Comunicaciones corre bajo ambiente Windows. En la máquina local se cargará de la ventana de Carbon Copy la parte correspondiente al **Guest**, y en la máquina remota la parte denominada **Host**.

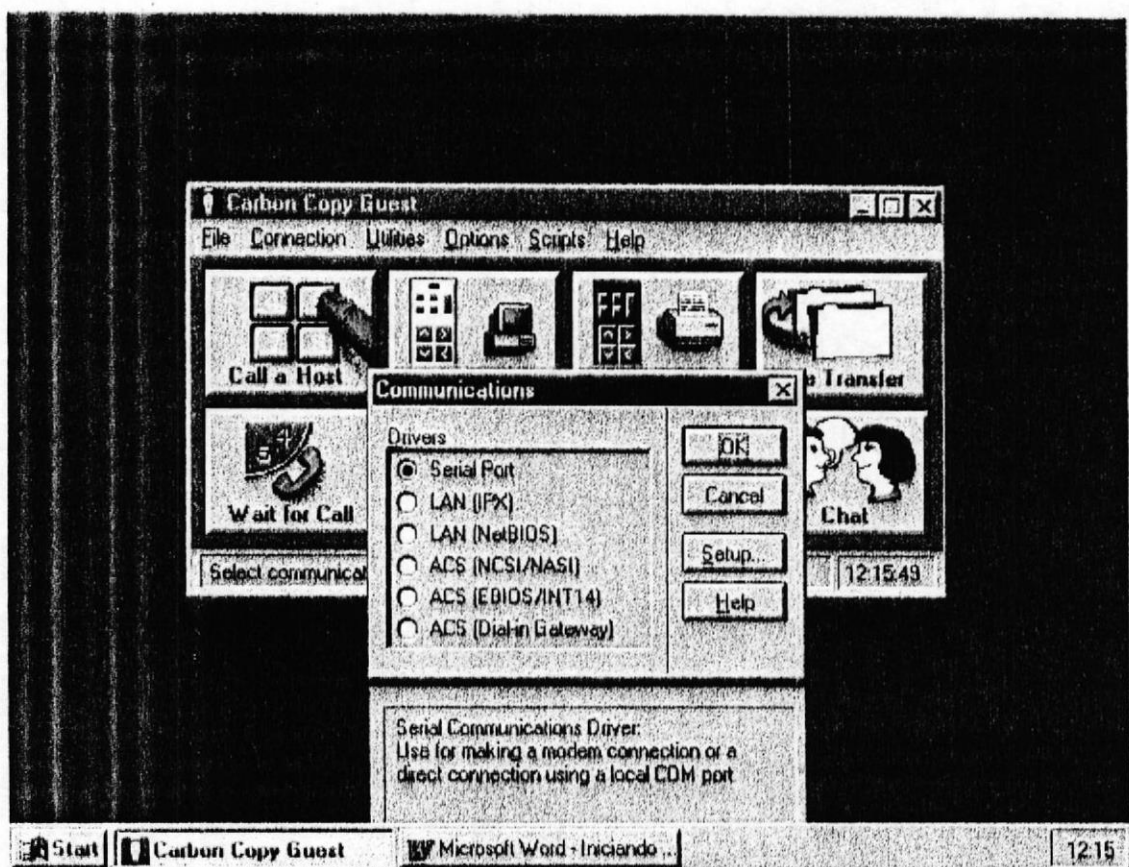
El programa de control del motor se encuentra instalado en la máquina que estará trabajando remotamente . Para correr el programa se deben seguir los siguientes pasos:

Iniciando una sesión



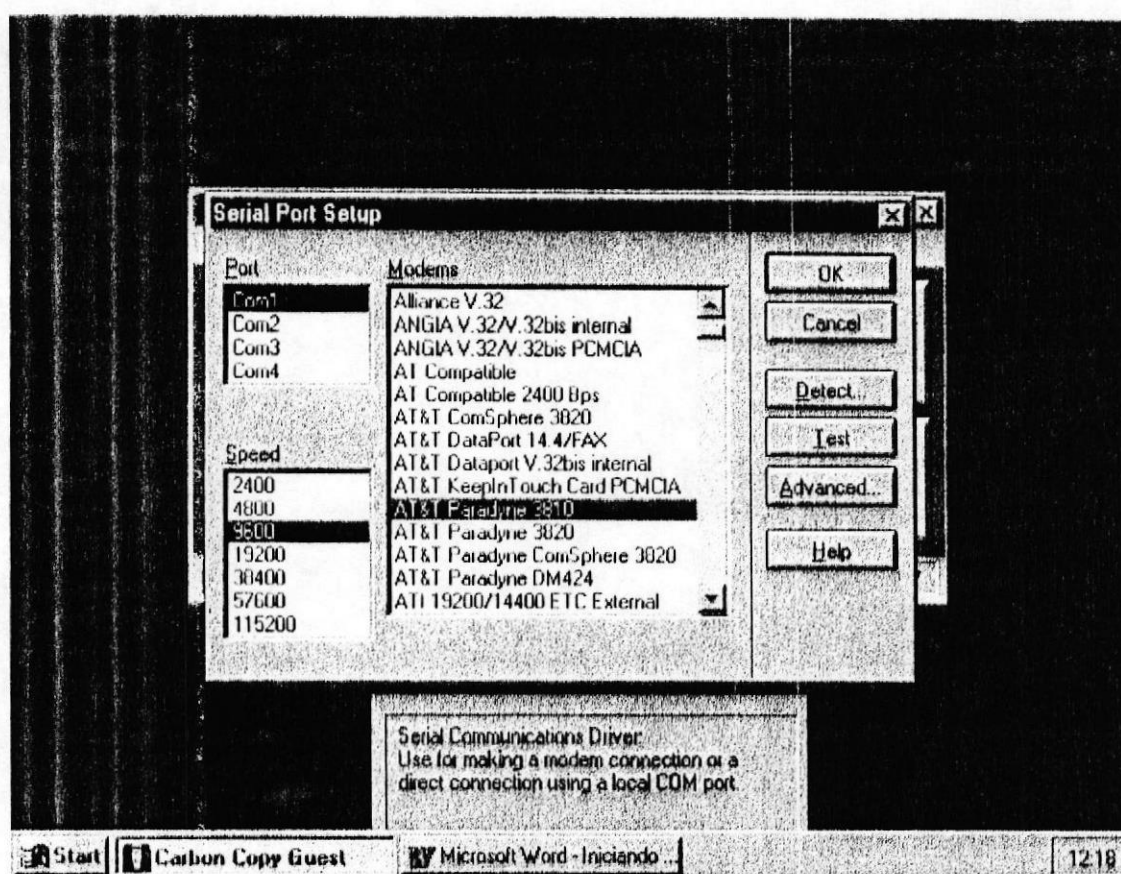
En la máquina local correr el Guest y ejecutar la parte denominada Call a Host, procediendo a marcar el número correspondiente al telular remoto.

Configurando el puerto de comunicación



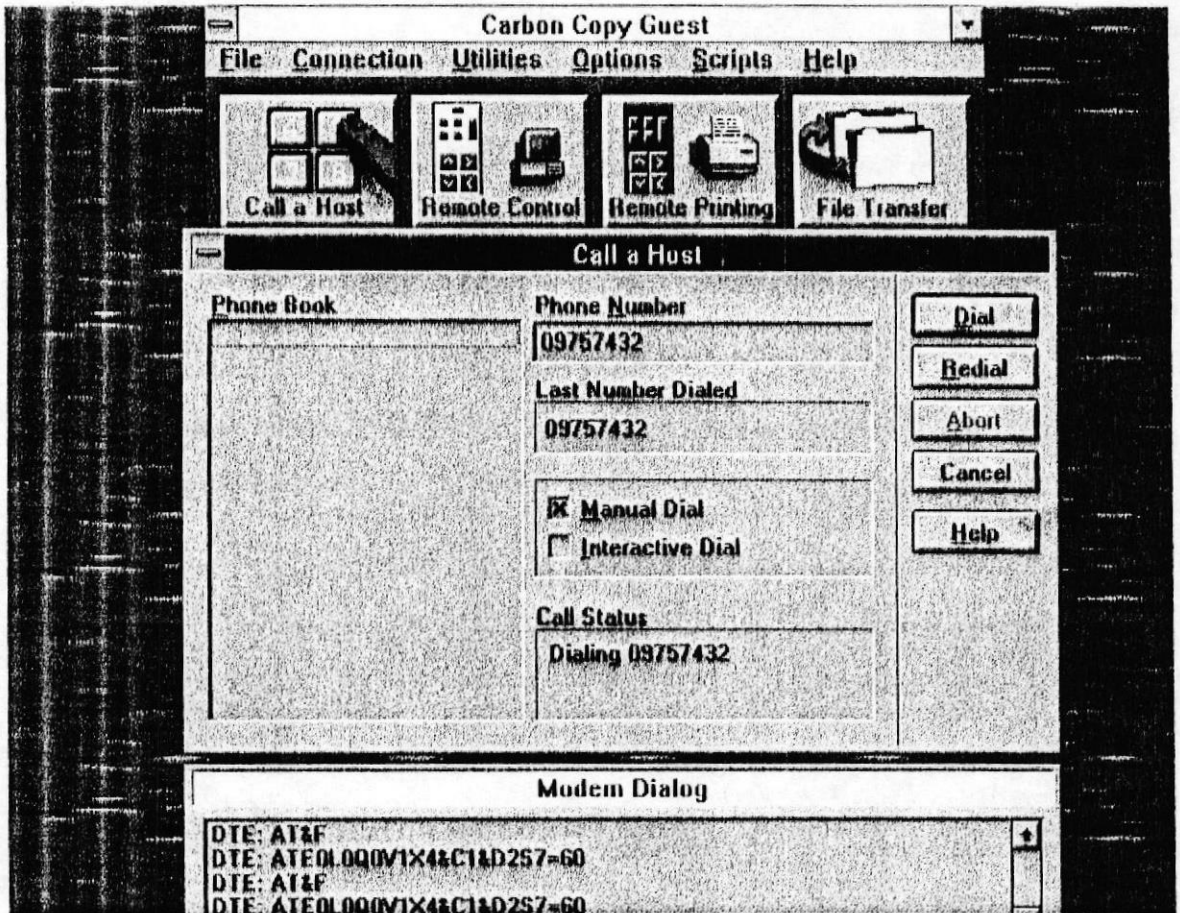
En esta ventana escoger *Serial Port* y presionar *Setup* para proceder a configurar los modems

Configurando los Modems



Configurando los modems: Escoger el modelo y la velocidad a la que a realizarse la comunicación.

Haciendo una llamada



Escogemos el número al cual nos deseamos conectar y seleccionamos *Dial*.

CAPITULO VI

MONTAJE DE LA PLATAFORMA DE HARDWARE

Para la implementación del proyecto de control del motor requerimos de los siguientes equipos:

1. Un Computador Personal con Procesador 80486; memoria RAM 8MB; Monitores VGA, Espacio libre en Disco Duro 20MB.
2. Una tarjeta de Interfaz de datos .
3. Una estación de Diseño Breadboarding (PB-88/4).
4. Un motor de paso DC.
5. Fuente de poder que suministre 15 Vdc. para la alimentación del motor.

Los elementos mencionados anteriormente son para la implementación y pruebas locales. Para la conexión remota necesitamos además:

6. Un segundo Computador Personal con características similares al anterior.
7. Dos Modems con capacidad para Comunicación Celular.
8. Dos Telulares.

CAPITULO VII

IMPLEMENTACION DEL PROGRAMA DE CONTROL

En este capítulo presentamos el detalle del programa concebido en lenguaje Assembler, así como los diagramas de estado en los que nos basamos para la implementación del circuito de control.

PROGRAMA PRINCIPAL

TITLE CONTROLADOR POR TELEMETRIA

```

:-----*
:*   VALORES ASCII DE TECLAS   *
:-----*

```

INTRO EQU 13
ESCAPE EQU 27

STACKSG SEGMENT PARA STACK 'STACK'
DW 32 DUP(?)
STACKSG ENDS

DATASG SEGMENT PARA 'DATA'

PORTA EQU 100H
PORTB EQU 101H
PORTC EQU 102H
REGISTRO EQU 103H

REFERENCIA DW 0
MUESTRA DW 2 DUP(0)
ERROR DW 0
INTEGRAL DW 0
K2 DW 26
K1 DW 4
DAC DW 0
VALOR DW 0
MUE DW 0
FUN DB 00
TECLA DB 00
CONT DB 00
ESCU DB 'ESCUELA SUPERIOR\$'
ELA DB 'POLITECNICA DEL LITORAL\$'
FACUL DB ' FACULTAD DE INGENIERIA ELECTRICAS '
NUME DB 'PROYECTO FINAL DE TOPICO DE GRADUACIONES '
PROFE DB 'Profesor: Ing. Washington Medina\$'
TEMA DB ' Tema: TELEMETRIA\$'
NAME1 DB 'Integrantes:ALMINATE VASQUEZ MAURICIO \$'
NAME2 DB ' AVILES LOZANO RONALD \$'
COLUMN1 DB 00
COLUMN3 DB 00
FILA3 DB 00
FILA1 DB 00
NAME3 DB ' MINCHALA AGUIRRE GIOVANA \$'
NAME4 DB ' SANTAMARIA SALAZAR RAUL \$'
PULSE DB 'PRESIONE UNA TECLA PARA CONTINUAR.....\$'
MENI DB 'CONTROL DE UN MOTOR DC USANDO TELEMETRIA\$'
MCOLUM DB 34

FILA2 DB 12

MEN2 DB 'MANUAL DEL USUARIO \$'
 MEN3 DB '1.-..... PRESIONE LA TECLA [C] PARA CONTINUAR.....\$'
 PASO1 DB ' "El motor empezara con una velocidad de 0 REV/MIN" \$'
 PASO2 DB '2.- DIGITE LA VELOCIDAD QUE SE QUIERE ALCANZAR CON EL
 MOTOR \$'
 PASO22 DB ' "La velocidad REFERENCIAL es el valor de estabilizacion\$'
 PASO23 DB ' y la velocidad ACTUAL es el valor medido en un instante"\$'
 PASO3 DB 'SI QUIERE FINALIZAR PRESIONE... [ESC]\$'
 REN DB 00
 COUNT DB 00
 ATRIB DB 00
 CHARACTER DB ?
 DATASG ENDS

;CODESG SEGMENT PARA 'CODE'

PRINCIPAL PROC FAR

ASSUME CS:CODESG, DS:DATASG, SS:STACKSG

PUSH DS

SUB AX,AX

PUSH AX

MOV AX,DATASG

MOV DS, AX

;LIMPIAR PANTALLA

MOV AX,0600H

MOV BH,07H

MOV CX, 0000H

MOV DX, 184FH

INT 10H

;PRESENTACION

CLRSCR

CALL CARATULA1

CALL CARATULA2

CALL VIDEO

;IMPRIMIR MENSAJES

MOV DX,0405H

MOV AH,02H

MOV BH,0

INT 10H

LEA DX, M0

MOV AH,09H

INT 21H

MOV DX,0E05H

MOV AH,02H

MOV BH,0

INT 10H

LEA DX,M1

MOV AH,09H

INT 21H

```
MOV DX,1005H
MOV AH,02H
MOV BH,0
INT 10H
LEA DX,M2
MOV AH,09H
INT 21H

MOV DX,1705H
MOV AH, 02H
MOV BH,0
INT 10H
LEA DX,MA
MOV AH,09H
INT 21H
MDOUBLE
; CARGAR REGISTRO DE CONTROL DEL 8255A
MOV DX,REGISTRO
MOV AL,98H
OUT DX,AL
; PONER EN 1 LOS BITS PC0 Y PC1 DEL PUERTO C DEL 8255A
MOV DX,REGISTRO
MOV AL,03H
OUT DX,AL
MOV AL,01H
OUT DX,AL
; MANDAR UNA ENTRADA DIGITAL AL DAC0830 DE MANERA QUE LA
VELOCIDAD ACTUAL
; DEL MOTOR SE INICIALICE EN CERO
MOV DX,PORTB
MOV AL,000
OUT DX,AL
; IMPRIMIR LA VELOCIDAD ACTUAL DEL MOTOR (INICIALMENTE CERO)
MOV DX,1040H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
MOV DL,30H
INT 21H
; A CONTINUACION SE IMPRIME LA VELOCIDAD REFERENCIAL VALOR INICIAL
DE CERO
; VALOR QUE DESPUES DEBERA SER VARIADO POR EL USUARIO A MEDIDA QUE
; AVANCE LA EJECUCION DEL PROGRAMA
MOV DX, 0E40H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
MOV DL,30H
INT 21H

; ALGORITMO P.I.
MOV REFERENCIA ,000
REPETIR: NOP
CALL MUESTREO
```

```
CALL VALACTUAL
MOV BX,REFERENCIA
SUB BX, MUESTRA
MOV ERROR ,BX
ADD INTEGRAL,BX
MOV DX,INTEGRAL
AND DX,8000H
CMP DX,0000H
JE ADEL12
MOV INTEGRAL,0
JMP ADEL13
ADEL12: CMP INTEGRAL,2800
JB ADEL13
MOV INTEGRAL,2800
ADEL13: NOP
```

```
MOV AX, INTEGRAL
MOV DX,0
MOV CX, K2
IDIV CX
MOV DAC, AX
MOV AX, ERROR
CWD
MOV CX, K1
IDIV CX
ADD DAC , AX
ADD DAC,49
MOV AX,DAC
AND AX,8000H
CMP AX,0000H
JE ADEL10
MOV DAC,0
JMP ADEL11
ADEL10: NOP
MOV AX,DAC
CMP DAC,255
JB ADEL11
MOV DAC,255
ADEL11: NOP
```

```
MOV DX,PORTB
MOV AX , DAC
OUT DX, AL
CALL TECLADO
JMP REPETIR
```

```
PRINCIPAL ENDP
```

TECLADO PROC NEAR

```
MOV AH,0BH
INT 21H
CMP AL,00H
JE FINAL
MOV AH, 06H
MOV DL,0FFH
INT 21H
CMP AL,1BH
JNE ADEL15
MOV AX,0600H
MOV BH,07H
MOV CX,0000H
MOV DX,184FH
INT 10H
MOV DX,1700H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,4CH
INT 21H
ADEL15:  CMP AL,63H
        JE REP2
        CMP AL,43H
        JE REP2
FINAL:  RET
REP2:   NOP
ADEL2:  NOP
        MOV DX,0E40H
        MOV AH,02H
        MOV BH,0
        INT 10H
        LEA DX,M4
        MOV AH,09H
        INT 21H
        MOV DX,0B05H
        MOV AH,02H
        MOV BH,0
        INT 10H
        LEA DX,M6
        MOV AH,09H
        INT 21H
        MOV DX,0E40H
        MOV AH,02H
        MOV BH,0
        INT 10H
        MOV POSCURSOR,0840H
        MOV REFERENCIA,0
        MOV SI,0
```

```
REP3:  NOP
      GETCH
      CMP SI,0
      JE ADEL3
      CMP SI,4
      JE ADEL4
      CMP AL,0DH
      JE ADEL4
ADEL3:  CMP AL,30H
      JL REP3
      CMP AL,39H
      JG REP3
      MOV AH,0
      MOV DIGITO,AX
      MOV POSCURSOR,0E40H
      ADD POSCURSOR,SI
      MOV DX,POSCURSOR
      MOV AH,02H
      MOV BH,0
      INT 10H
      MOV AH,02H
      MOV DX,DIGITO
      INT 21H
      SUB DIGITO,30H
      MOV AX,REFERENCIA
      MOV DX,0
      MOV CX,10
      MUL CX
      ADD AX,DIGITO
      MOV REFERENCIA,AX
      INC SI
      JMP REP3
ADEL4:  NOP
      CMP REFERENCIA,0000
      JE ADEL16
      CMP REFERENCIA,100
      JB ESPERA2
      NOP
      CMP REFERENCIA,1201
      JB ADEL16
ESPERA2: MOV DX,0B05H
      MOV AH,02H
      MOV BH,0
      INT 10H
      MOV CX,05BH
ESPERA3: NOP
      LEA DX,M5
      MOV AH,09H
      INT 21H
      LOOP ESPERA3
      JMP REP2
```


ADEL16: MOV AX,REFERENCIA

MOV DX,0
MOV CX,10
DIV CX
MOV REFERENCIA,AX
SUB AX,255
AND AX,8000H
CMP AX,8000H
JE ADEL9
MOV REFERENCIA,255

ADEL9: NOP

MOV DX,1840H
MOV AH,02H
MOV BH,0
INT 10H
RET

TECLADO ENDP

VALACTUAL PROC NEAR

MOV DX,1040H
MOV AH,02H
MOV BH,0
INT 10H
MOV AX,MUESTRA
CMP AX,1
JA REP17
MOV AX,0

REP17: MOV DL,10

MUL DL
MOV VALOR,AX
MOV DX,0
MOV CX,10000
DIV CX
MOV DIGITO,AX
ADD DIGITO,30H
MOV DX,DIGITO
MOV AH,02H
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,1000
DIV CX
MOV DIGITO,AX
MOV DX,1041H
MOV AH,02H
MOV BH,0
INT 10H
ADD DIGITO,30H

```
MOV AH,02H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,1000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,100
DIV CX
MOV DIGITO,AX
MOV DX,1042H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO, 30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,100
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,10
DIV CX
MOV DIGITO,AX
MOV DX,1043H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10
DIV CX
MOV AX,DX
MOV DIGITO,AX
MOV DX,1044H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
RET
VALACTUAL ENDP
```

IMPVALACTUAL PROC NEAR

```
MOV DX,1815H
MOV AH,02H
MOV BH,0
INT 10H
MOV AX,MUESTRA
MOV VALOR,AX
MOV DX,0
MOV CX,10000
DIV CX
MOV DIGITO,AX
ADD DIGITO,30H
MOV DX,DIGITO
MOV AH,02H
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,1000
DIV CX
MOV DIGITO,AX
MOV DX,1816H
MOV AH,02H
MOV BH,0
INT 10H
ADD DIGITO,30H
MOV AH,02H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,1000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,100
DIV CX
MOV DIGITO,AX
MOV DX,1817H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,100
DIV CX
```

```
MOV AX,DX
MOV DX,0
MOV CX,10
DIV CX
MOV DIGITO,AX
MOV DX,1818H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10
DIV CX
MOV AX,DX
MOV DIGITO,AX
MOV DX,1819H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
RET
IMPVALACTUAL ENDP

IMPINTEGRAL PROC NEAR
MOV DX,1835H
MOV AH,02H
MOV BH,0
INT 10H
MOV AX,INTEGRAL
MOV VALOR,AX
MOV DX,0
MOV CX,10000
DIV CX
MOV DIGITO,AX
ADD DIGITO,30H
MOV DX,DIGITO
MOV AH,02H
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,1000
DIV CX
```

```
MOV DIGITO,AX
MOV DX,1836H
MOV AH,02H
MOV BH,0
INT 10H
ADD DIGITO,30H
MOV AH,02H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,1000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,100
DIV CX
MOV DIGITO,AX
MOV DX,1837H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO, 30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,100
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,10
DIV CX
MOV DIGITO,AX
MOV DX,1838H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10
DIV CX
MOV AX,DX
MOV DIGITO,AX
MOV DX,1839H
MOV AH,02H
```

```
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
RET
IMPINTEGRAL ENDP
ERROR PROC NEAR
MOV DX,1825H
MOV AH,02H
MOV BH,0
INT 10H
MOV AX,ERROR
MOV VALOR,AX
MOV DX,0
MOV CX,10000
DIV CX
MOV DIGITO,AX
ADD DIGITO,30H
MOV DX,DIGITO
MOV AH,02H
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,1000
DIV CX
MOV DIGITO,AX
MOV DX,1826H
MOV AH,02H
MOV BH,0
INT 10H
ADD DIGITO,30H
MOV AH,02H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,1000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,100
DIV CX
MOV DIGITO,AX
MOV DX,1827H
MOV AH,02H
MOV BH,0
INT 10H
```

```
MOV DX,0
MOV CX,100
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,10
DIV CX
MOV DIGITO,AX
MOV DX,1828H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10
DIV CX
MOV AX,DX
MOV DIGITO,AX
MOV DX,1829H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
RET
IMPERROR ENDP
IMPDAC PROC NEAR
MOV DX,1845H
MOV AH,02H
MOV BH,0
INT 10H
MOV AX,DAC
MOV VALOR,AX
MOV DX,0
MOV CX,10000
DIV CX
MOV DIGITO,AX
ADD DIGITO,30H
MOV DX,DIGITO
MOV AH,02H
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10000
DIV CX
MOV AX,DX
MOV DX,0
```

```
MOV CX,1000
DIV CX
MOV DIGITO,AX
MOV DX,1846H
MOV BH,0
INT 10H
ADD DIGITO,30H
MOV AH,02H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,1000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,100
DIV CX
MOV DIGITO,AX
MOV DX,1847H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO, 30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,100
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,10
DIV CX
MOV DIGITO,AX
MOV DX,1848H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10
DIV CX
MOV AX,DX
MOV DIGITO,AX
MOV DX,1849H
```



```
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
RET
IMPDAC ENDP
IMPREFERENCIA PROC NEAR
MOV DX,1805H
MOV AH,02H
MOV BH,0
INT 10H
MOV AX,REFERENCIA
MOV VALOR,AX
MOV DX,0
MOV CX,10000
DIV CX
MOV DIGITO,AX
ADD DIGITO,30H
MOV DX,DIGITO
MOV AH,02H
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,1000
DIV CX
MOV DIGITO,AX
MOV DX,1806H
MOV AH,02H
MOV BH,0
INT 10H
ADD DIGITO,30H
MOV AH,02H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,1000
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,100
DIV CX
MOV DIGITO,AX
MOV DX,1807H
```

```

MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO, 30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,100
DIV CX
MOV AX,DX
MOV DX,0
MOV CX,10
DIV CX
MOV DIGITO,AX
MOV DX,1808H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
MOV AX,VALOR
MOV DX,0
MOV CX,10
DIV CX
MOV AX,DX
MOV DIGITO,AX
MOV DX,1809H
MOV AH,02H
MOV BH,0
INT 10H
MOV AH,02H
ADD DIGITO,30H
MOV DX,DIGITO
INT 21H
RET
IMPREFERENCIA ENDP

MUESTREO PROC NEAR
MOV SI,0
MOV MUESTRA,0
MOV MUESTRA+2,0
REP1: CALL OBTIENE
MOV AX,MUE
ADD MUESTRA ,AX
ADC MUESTRA+2,0
CMP SI,570
JE ADEL1
INC SI
JMP REP1

```



```
ADEL1:  NOP
        MOV AX,MUESTRA
        MOV DX,MUESTRA+2
        MOV CX,571
        DIV CX
        AND AX,0FFFEH
        MOV BX,REFERENCIA
        AND BX,0001H
        ADD AX,BX
        MOV MUESTRA,AX
;lineas adicionales
        MOV AX,MUESTRA
        CMP AX,1
        JGE ADEL17
        MOV AX,1
        CMP REFERENCIA,1
        JNE ADEL17
        MOV AX,0
ADEL17: NOP
        MOV MUESTRA,AX
        RET
MUESTREO ENDP

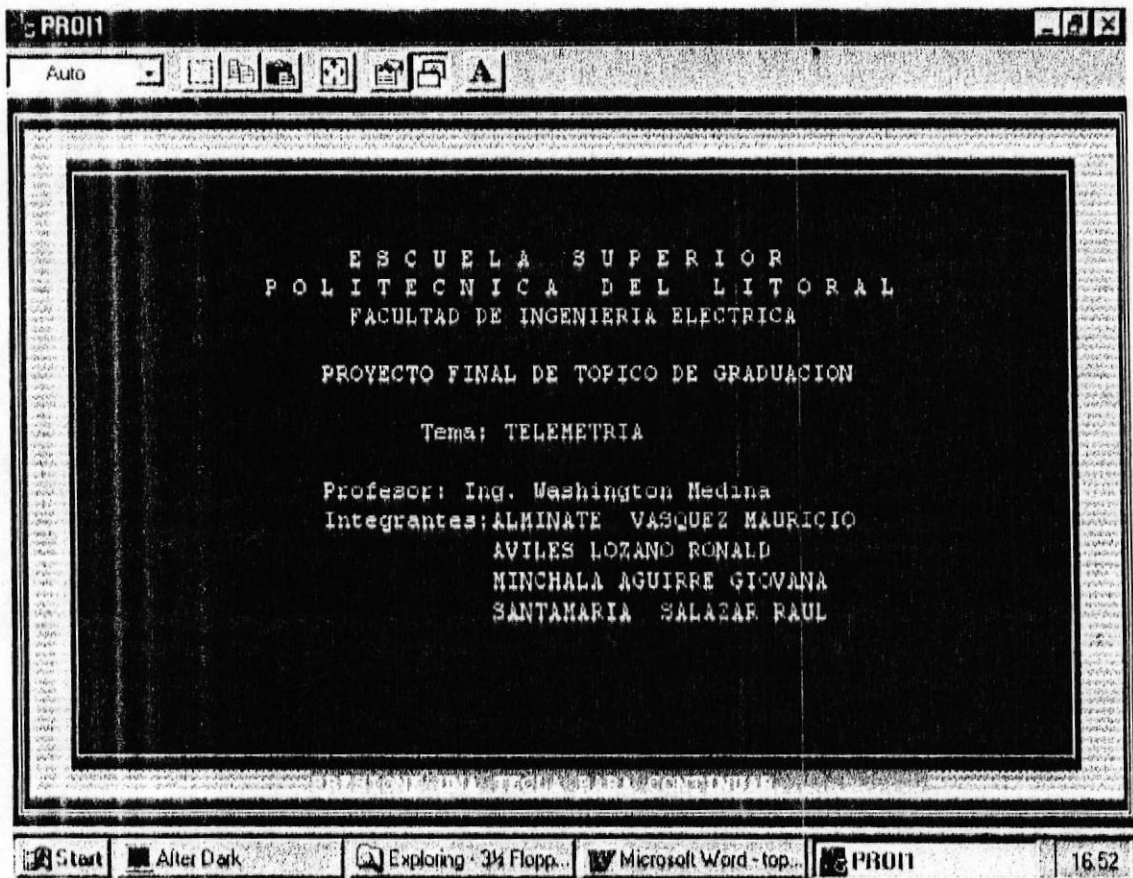
OBTIENE PROC NEAR
        MOV DX,REGISTRO
        MOV AL,03H
        OUT DX,AL
        MOV DX,REGISTRO
        MOV AL,01H
        OUT DX,AL
        MOV CX,5
REP11:  NOP
        LOOP REP11
        MOV DX,REGISTRO
        MOV AL,02
        OUT DX,AL
        MOV CX,20
REP10:  NOP
        LOOP REP10
        NOP
        MOV DX,REGISTRO
        MOV AL,03
        OUT DX,AL
REP5:   NOP
        MOV DX,PORTC
        IN AL,DX
        AND AL,10H
        MOV AH,0
        CMP AX,16
        JE REP5
        MOV CX,15
```

```
REP14:  NOP
        LOOP REP14
        MOV DX,REGISTRO
        MOV AL,00
        OUT DX,AL
        MOV CX,20
REP15:  NOP
        LOOP REP15
        MOV DX,PORTA
        IN AL,DX
        MOV AH,0
        MOV MUE,AX
        MOV DX,REGISTRO
        MOV AL,01
        OUT DX,AL
        RET
OBTIENE ENDP

CODESG ENDS
        END PRINCIPAL
```

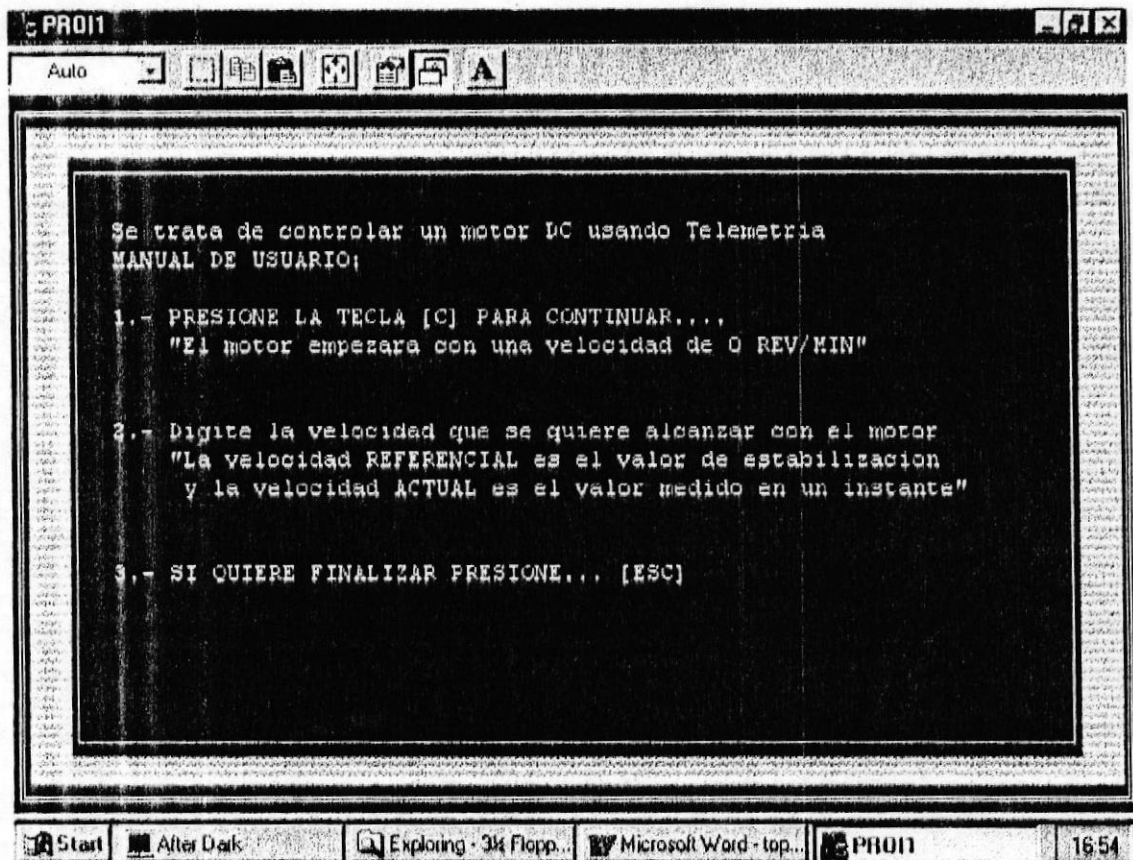
Ejecutando el Programa de Control

Cuando se ejecuta el programa CONTROL aparece la caratula 1 con el siguiente menú:



Carátula 2

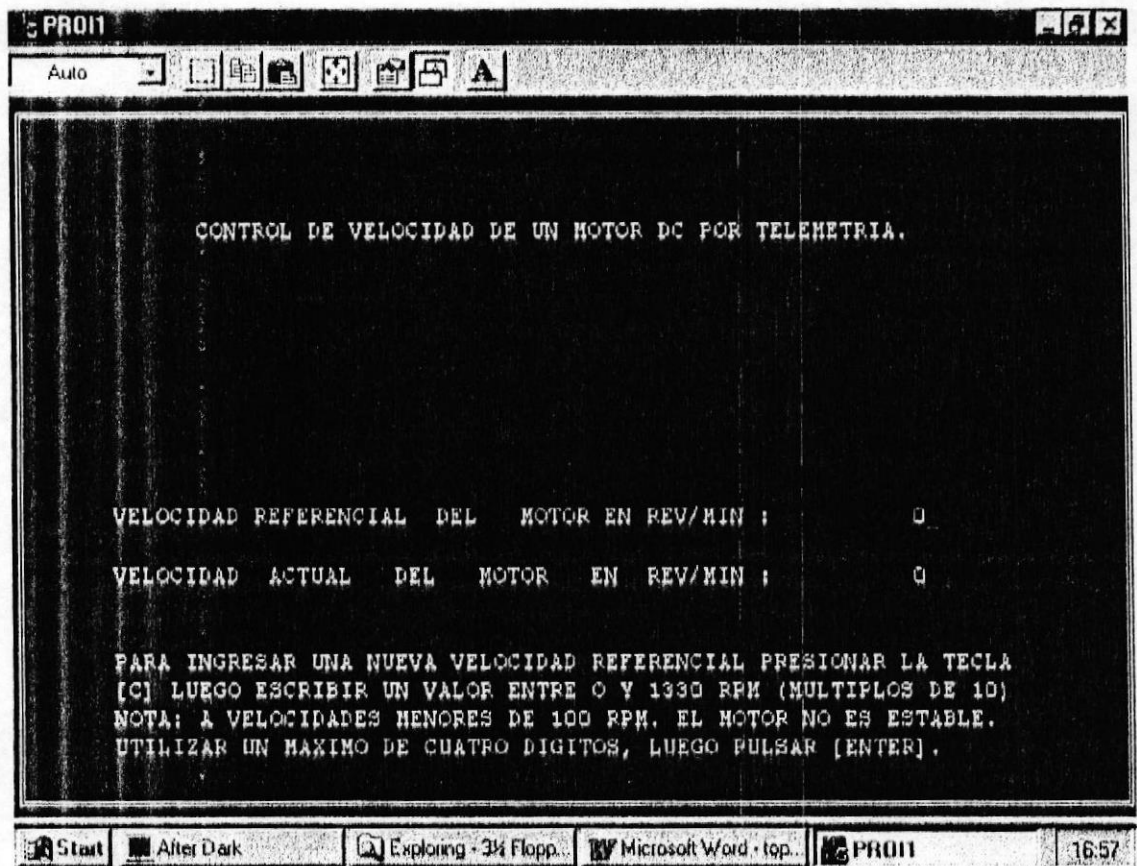
El usuario al presionar cualquier tecla inmediatamente ingresa a la siguiente pantalla que presenta el siguiente menú :



Este menú tiene dos opciones : Continuar y Salir . El usuario puede presionar la tecla C para pasar a la carátula 3 y así continuar con el programa , ó presionar la tecla ESCAPE para salir del programa .

Carátula 3

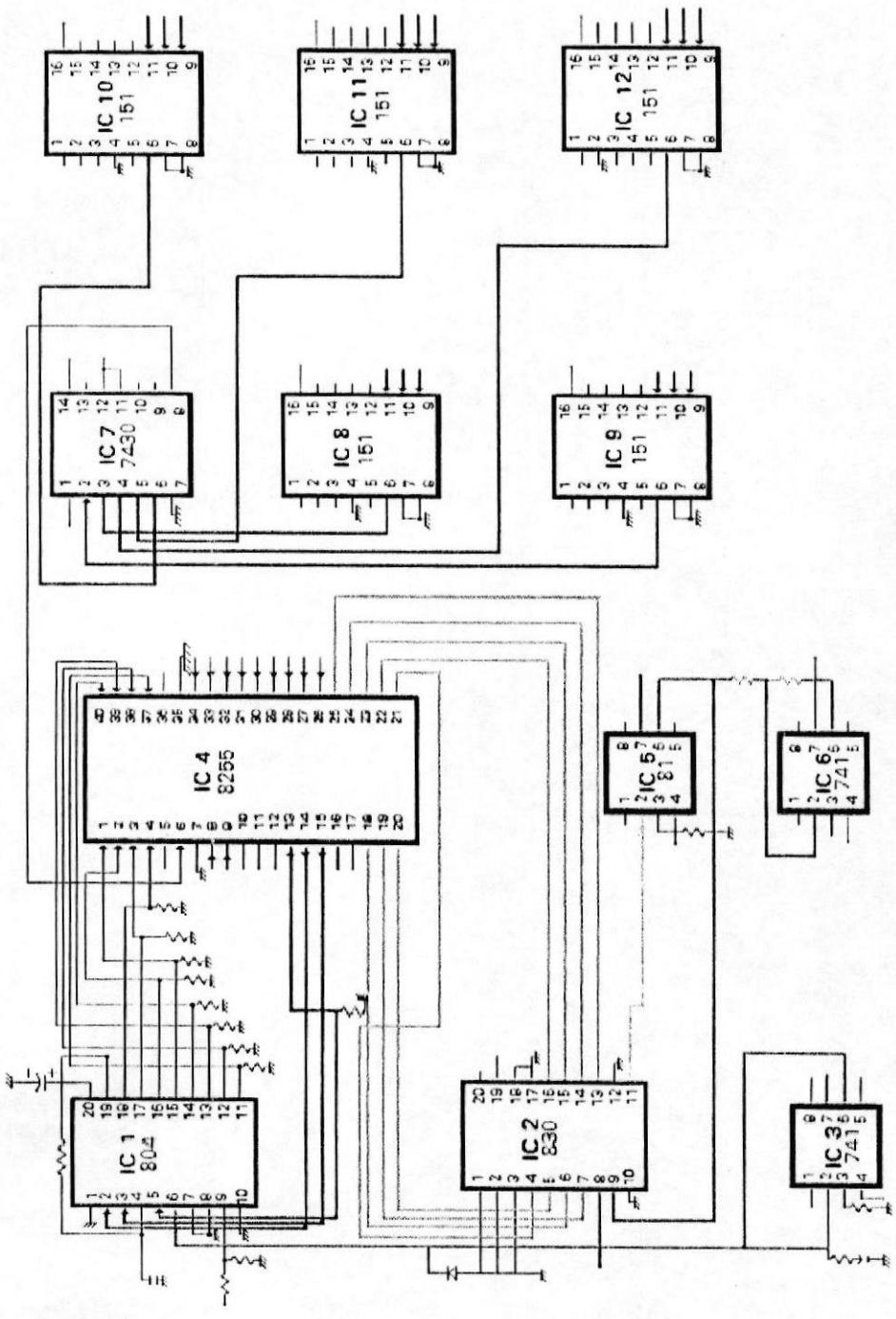
Si el usuario ha seleccionado la opción Continuar se presentará la pantalla que se muestra a continuación :



CAPITULO VIII

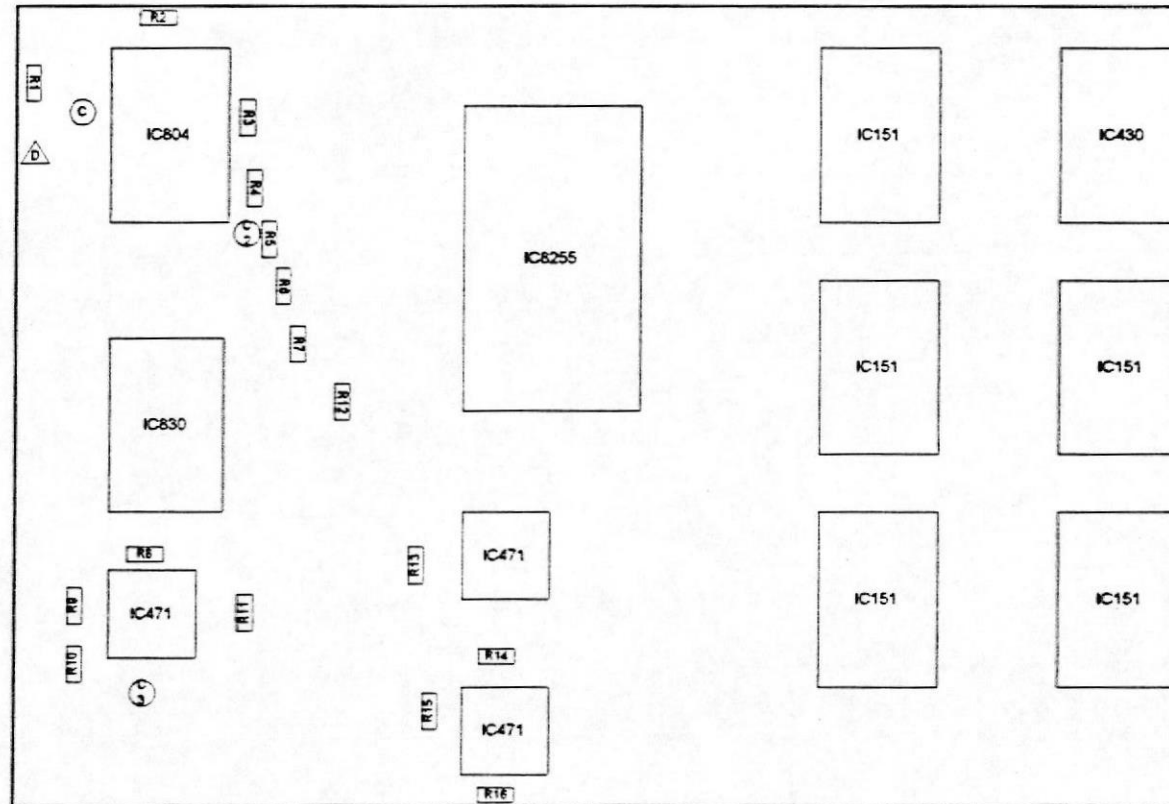
DETALLES DEL DISEÑO

En este capítulo mostraremos los diagramas correspondientes al diseño del Circuito Controlador de Velocidad del Motor de Paso, así como un detalle de los elementos utilizados.



1

Diagrama esquemático del circuito de fuerza



Lista de Componentes

COMPONENTE	DETALLES ADICIONALES
IC151	Multiplexores
7430	Puerta NAND de 8 entradas
R1,R3,R4,R5, R6, R7, R9, R11, R13, R14	Resistencias de 1K
R2	Resistencia de 13K
R8	Resistencia de 1.8K
R10	Resistencia de 2K
R15	Resistencia de 6.8K
C1	Capacitor de 150pF
C2, C3	Capacitor de 10uF
471	Opam
IC151	Multiplexor 8 a 1
ADC0804	Convertidor analógico digital
DAC0830	Convertidor digital analógico
8255A	Interface periférica programable

CONCLUSIONES

1. Para conseguir un acceso confiable hacia un dispositivo remoto es necesario considerar 4 parámetros : Software de Control , Hardware de Control , Software de Comunicaciones , Hardware de Comunicaciones .Al fallar uno de estos parámetros no se podrá realizar un control a distancia .
2. El language ensamblador utilizado en el software de control así como otros lenguajes de programación , son instrumentos muy útiles para mejorar el sistema de control en cualquier área ya sea ; de Comunicaciones , Industrial , etc.
3. Para controlar cualquier equipo mediante un computador es necesario implementar una interfase entre ellos . Esta interfase dependiendo del equipo a controlar debe poseer 3 partes importantes : la etapa de direccionamiento , la etapa de conversión de la señal y la etapa de potencia .
4. Hemos podido comprobar que en este tipo de control digital directo existe la desventaja de que si el software o el computador falla , todo el sistema de control también falla .
5. Mediante este proyecto se ha podido comprobar ciertos aspectos teóricos del control automático de velocidad de un motor DC . Entre los más notorios encontramos : la estabilidad y el tiempo de respuesta , son parámetros inversos, ya que al tratar de mejorar el primero de ellos, disminuía el otro . Por lo tanto , se hizo necesario hacer variaciones tanto en el software como en el hardware de control para poder lograr una

buena estabilidad en la velocidad del motor sin tener un tiempo de respuesta muy lento .

6. Después de realizar muchas pruebas con diferentes tipos de modems , se concluye que para que exista un enlace confiable de tipo celular ,es necesario, utilizar modems que posean entre sus características el soporte de comunicación de este tipo.
7. El programa de comunicación se inició con una velocidad de enlace de 7200 bps , más , después de hacer la negociación los modems siempre se conectaron a la velocidad de 4800 bps .
8. Después de hacer pruebas con dos diferente tipos de programas de comunicación, decidimos utilizar el programa CARBON COPY debido a su flexibilidad para manejarse en ambiente windows y debido a la facilidad en el momento de configurar las diferentes características de los modems
9. Una de las más importantes características de un Ingeniero debe ser la habilidad para investigar. En este proyecto se realizó una extensa investigación en las características de todos los dispositivos utilizados en la implementación del hardware de control y de comunicación. Además del software mas adecuado para la comunicación.
10. Este proyecto puede ser usado como guía practica para implementar otros sistemas de control a distancia mediante una comunicación celular . El material recopilado constituye una fuente de Información no solo estudiantil sino también a nivel profesional .

OBSERVACIONES

1. De acuerdo al programa de control implementado se puede lograr una estabilidad de la velocidad del motor variando el número de muestras de la integral en el controlador P.I. Se empezó con un número de muestras de 100, hasta concluir con un total de 840.
2. Debido al máximo valor de voltaje de entrada con que trabaja el convertidor analógico digital, se necesitó de una etapa limitadora que disminuya el voltaje entregado por el sensor de velocidad (tacómetro).
3. Para obtener una velocidad final e inicial confiable, trabajamos variando la frecuencia a través de la resistencia y el capacitor que acompañan al convertidor analógico digital, hasta lograr la mejor estabilidad.
4. Se realizaron pruebas de enlace utilizando dos programas de comunicación: PC ANYWHERE y CARBON COPY. Nos decidimos a trabajar con el último de éstos ya que gracias a su interface gráfica Windows facilita la tarea de enlace a los usuarios finales.
5. Para optimizar la comunicación entre los computadores y lograr acceder de un modo confiable al control del motor, se hizo necesario el uso de modems con capacidad para habilitar un modo de operación celular.
6. e mejora la velocidad de transmisión de la señal de control al mejorar las características de hardware de los computadores, es decir, aumentando ya sea la velocidad del procesador o la memoria RAM.

BIBLIOGRAFIA

ABEL PETER , IBM PC Assembly Language and Programming , New York ,
Prentice Hall , second edition 1991 .

NATIONAL SEMICONDUCTOR , Data Adquisition Databook , Santa Clara -
California , National Semiconductor , 1991 .

JACOB M . , Industrial Control electronics (Applications and Design) ,
New York , Prentice Hall , 1989 .

GONZALEZ DE LA GARZA M , MODEMS todo sobre telecomunicaciones
Madrid , ed . ParanInfo , 1992 .

UYLESS BLACK , REDES DE COMPUTADORES Protocolos , Normas e
Interfases , EE. UU Prentice Hall second edition , 1993 .

AT&T, Manual de Configuración de modems familia 3910

APENDICE

CONVERTIDOR DIGITAL-ANALOGICO DAC0830

Este dispositivo fabricado por la National Semiconductors, es un avanzado convertidor CMOS de SI-Cr de 20 pines, diseñado para trabajar en interfaces directamente con los microprocesadores 8080,8048,8085 y otros microprocesadores populares.

El doble almacenamiento de este convertidor digital-analógico le asigna a la salida un voltaje correspondiente a una palabra digital mientras se mantiene la próxima palabra digital. El DAC0830 tiene las siguientes especificaciones:

- ✓ Resolución : 8 bits
- ✓ Tiempo de estabilización : 1 us
- ✓ Baja disipación de potencia : 20 mW
- ✓ Fuente de alimentación : de 5 a 15 V

1	CS	VCC	20
2	WR1	ILE	19
3	GND	WR2	18
4	DI 3	XFER	17
5	DI 2	DI 4	16
6	DI 1	DI 5	15
7	LSB	DI 6	14
8	Vref	DI 7	13
9	RFB	IOU2	12
10	GND	IOU1	11

CONVERTIDOR ANALOGICO-DIGITAL ADC0804

Este dispositivo también es fabricado por la National Semiconductor; es un convertidor CMOS de aproximación sucesiva, que utiliza una escalera potenciométrica diferencial similar a la conformada por 256R. Este integrado de 20 pines es de fácil interface a todos los microprocesadores.

Las entradas de voltaje analógicas diferenciales en este convertidor, asignan un incremento del rechazo de modo común y compensan el valor de cero analógico de voltaje de entrada, esto permite en otras palabras un ajuste a cero a la entrada del convertidor. El ADC0804 presenta las siguientes especificaciones:

- ✓ Resolución : 8 bits
- ✓ Tiempo de conversión : 100us
- ✓ Precisión absoluta : ± 1 LSB (bit menos significativo)
- ✓ Fuente de alimentación : 5 V
- ✓ Rango de voltaje de entrada : de 0 a 5 V

1	CS	Vcc	24
2	RD	Clk R	23
3	WR	Clk IN	22
4	INTR		21
5	DB7		20
6	DB6		19
7	DB5		18
8	DB4	Vin +	17
9	DB3	Vin -	16
10	DB2	Gnd	15
11	DB1	Vref/2	14
12	DB0	Gnd	13

INTERFACE PERIFERICA PROGRAMABLE 8255A

Este dispositivo fabricado por Intel es una interface paralela; contiene un registro de control y tres puertos direccionables separados. El acceso a este puerto se lo realiza mediante el pin CS y la dirección del acceso es de acuerdo a las señales RD y WR. Las señales aplicadas en los pines A1 y A0 permiten el direccionamiento de los cuatro registros. Entonces estas cinco señales determinan si los datos ingresan al 8255A o salen de este, así :

A1	A0	RD	WR	CS	Descripción de la transferencia
0	0	0	1	0	del puerto A al bus de datos
0	1	0	1	0	del puerto B al bus de datos
1	0	0	1	0	del puerto C al bus de datos
0	0	1	0	0	del bus de datos al puerto A
0	1	1	0	0	del bus de datos al puerto B
1	0	1	0	0	del bus de datos al puerto C
x	x	x	x	1	D7-D0 van a un estado de alta impedancia
x	x	1	1	0	D7-D0 van a un estado de alta impedancia

