

**UNIVERSIDAD AUTÓNOMA DE BAJA  
CALIFORNIA  
ESCUELA DE INGENIERÍA  
UNIDAD ENSENADA**



**MÓDULO DE MEDICIÓN DE TIEMPO DE TRABAJO  
Y CONSUMO DE COMBUSTIBLE**

**MEMORIA DEL SERVICIO SOCIAL OBLIGATORIO  
COMO OPCIÓN A TITULACIÓN  
" PARTE A "**

**TRABAJO QUE PRESENTA :**

**SERGIO EDUARDO PERALTA TRASVIÑA**

**PARA OBTENER EL TÍTULO DE:**

**INGENIERO EN ELECTRÓNICA**

---

MÓDULO DE MEDICIÓN DE TIEMPO DE TRABAJO  
Y CONSUMO DE COMBUSTIBLE

MEMORIA DE SERVICIO SOCIAL

QUE PRESENTA

SERGIO EDUARDO PERALTA TRASVIÑA

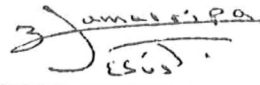
APROBADA POR :



M.C. HUMBERTO CERVANTES DE AVILA  
PRESIDENTE DEL JURADO  
Y DIRECTOR DEL TRABAJO



M.C. MANUEL CORONA DOMINGUEZ  
SINODAL



M.C. JOSÉ DE JESÚS ZAMARRIPA T.  
SINODAL



ING. JOSÉ ANTONIO MICHEL M.  
SINODAL



ING. JUAN DE DIOS SÁNCHEZ LÓPEZ  
SINODAL

---

**CONTENIDO**

	<b>página</b>
LISTA DE TABLAS	iii
LISTA DE FIGURAS	iv
RESUMEN	v
1 INTRODUCCIÓN	1
2 OBJETIVOS	2
3 ANTECEDENTES	3
4 JUSTIFICACIÓN	4
5 EQUIPO Y MATERIALES	5
6 FUNCIONAMIENTO DEL MÓDULO	
6.1 FUNCIONAMIENTO	7
6.2 CARACTERÍSTICAS Y ESPECIFICACIONES	10
7 CIRCUITERÍA	12
7.1 EL MICROCONTROLADOR	12
7.2 LAS MEMORIAS RAM Y EPROM	15
7.3 LA FUENTE DE ALIMENTACIÓN	15
7.4 LA COMUNICACIÓN CON EL USUARIO	16
7.5 CIRCUITOS MISCELÁNEOS	16
8 ADQUISICIÓN DE DATOS	17

9	LA PROGRAMACIÓN	18
10	EL CIRCUITO IMPRESO	21
11	DISCUSIONES	22
12	CONCLUSIONES	23
13	BIBLIOGRAFÍA	24
14	ANEXO A	26
15	ANEXO B	32

LISTA DE TABLAS

	<b>página</b>
<b>Relación del equipo utilizado (tabla I)</b>	<b>5</b>
<b>Relación del material utilizado (tabla II)</b>	<b>6</b>
<b>Relación de muestras del módulo (tabla III)</b>	<b>9</b>
<b>Mapa de memorias (tabla IV)</b>	<b>15</b>
<b>Mapa de direcciones (tabla V)</b>	<b>16</b>

LISTA DE FIGURAS

	página
Diagrama a bloques del módulo (figura No 1)	7
Diagrama a bloques del 80C51 (figura No 2)	13
Distribución de la memoria interna del 80C51 (figura No 3)	14
Diagrama de conexiones y tabla de funciones del 74HC123 (figura No 4)	17
Diagrama de flujo, programa principal en "turbo Pascal" (figura No 5)	18
Subrutinas del programa en "turbo Pascal" (figura No 6)	19
Subrutinas del programa en "turbo Pascal" (figura No 7)	20
Diagrama esquemático del módulo (figura No 8)	27
Distribución de los componentes en el circuito impreso, lado A de la tarjeta (figura No 9)	28
Distribución de los componentes en el circuito impreso, lado B de la tarjeta (figura No 10)	29
Distribución de las pistas y vías en el circuito impreso, lado A de la tarjeta (figura No 11)	30
Distribución de las pistas y vías en el circuito impreso, lado B de la tarjeta (figura No 12)	31

## RESUMEN

Se llevó a cabo el diseño, construcción y prueba a nivel experimental de un módulo electrónico que toma lecturas de niveles de gasolina y tiempo de uso de un vehículo. El módulo almacena los datos y cuenta con una interfaz para realizar la comunicación serie.

Se realizó también un programa para transferir los datos del módulo a una computadora personal compatible IBM. Además el programa es capaz de interpretar los datos y hacer reportes apropiados para servir como apoyo administrativo.

## 1 INTRODUCCIÓN

El avance de la tecnología actual ha permitido a la electrónica pasar del controlador común al microcontrolador. El diseño y realización del sistema de control en un proceso se ha simplificado enormemente gracias a la existencia de los microcontroladores. Actualmente, el adecuar señales eléctricas y realizar cualquier tipo de interfase digital con un sistema microcontrolador se ha convertido en una herramienta indispensable para el ingeniero en electrónica que pretenda desarrollar un sistema de control.

Existen diferentes instituciones públicas y privadas en las cuales el uso de vehículos, para el buen desarrollo de las mismas es indispensable; sin embargo, debido a la falta de información precisa sobre la utilización real de los vehículos, no es posible administrar su uso adecuadamente, lo cual ocasiona que los gastos de gasolina y mantenimiento, entre otros, sean muy elevados.

Con el desarrollo del presente proyecto se pretende contar con un módulo capaz de tomar lecturas con las cuales se pueda llevar control estadístico de la gasolina utilizada por un vehículo, así como también el control del tiempo de uso en horas. Se pretende que el prototipo experimental desarrollado se instale en los vehículos oficiales de la Escuela de Ingeniería de la UABC, siendo ésta la institución para la cual se desarrolló el proyecto, como Servicio Social.

El resumen de este trabajo es presentado en dos partes. En este trabajo se presenta la parte A, en la cual se tratarán solo las etapas en las que se tuvo participación directa, puesto que el desarrollo de este programa de Servicio Social se realizó como trabajo de conjunto.

## 2 OBJETIVOS

- Diseñar un módulo electrónico capaz de realizar mediciones sobre el gasto efectivo de combustible y el tiempo en horas del uso de un vehículo.
- Realizar un programa para la interpretación de los datos obtenidos por el módulo, de tal forma que estos puedan servir como un apoyo administrativo.

### 3 ANTECEDENTES

El control del gasto de combustible en los vehículos, así como el uso que se les da a estos en una empresa o en su caso en una institución pública se ha realizado de diferentes formas, una es contar con una persona encargada de verificar el kilometraje semanal o mensualmente, otra es la contabilización de los vales de gasolina que se les proporciona a los usuarios de los vehículos; sin embargo, estos procedimientos no aportan información sobre su uso real, por lo que no puede tenerse un verdadero control sobre el uso de los mismos.

#### 4 JUSTIFICACIÓN

El Servicio Social es la acción que fortalece los vínculos con la sociedad y permite al estudiante o pasante desarrollar, modificar y complementar los conocimientos adquiridos, mediante la aplicación directa de los mismos, como una culminación de su formación académica e inicio de su ejercicio profesional.

El desarrollo de este proyecto como Servicio Social cumple con tal objetivo, ya que la metodología utilizada para la realización del módulo, fué la aplicación directa de los conocimientos adquiridos durante la carrera, en materias como circuitos lógicos I, II y III, para el diseño de la parte digital, programación I y II, para el diseño del programa de alto nivel, laboratorio de estado sólido, electrónica I y II, para el diseño del acoplamiento analógico digital. Así como también se aplicaron los conocimientos de implementación de circuitos impresos, por el método de exposición de placa fotosensible.

Dicho módulo de medición de parámetros de automóvil le podrá servir a la Escuela de Ingeniería de la UABC para optimizar sus recursos.

## 5 EQUIPO Y MATERIALES

El equipo utilizado en el desarrollo del proyecto pertenece al laboratorio de la Escuela de Ingeniería, el cual consiste en equipo de medición y el laboratorio de circuitos impresos. La relación del equipo se muestra en la tabla I.

También se incluye una lista de los materiales que fueron utilizados para realizar el módulo los cuales se encuentran en la tabla II.

**TABLA I .- RELACIÓN DEL EQUIPO UTILIZADO**

<b>Cantidad</b>	<b>Descripción</b>	<b>Modelo</b>	<b>Compañía</b>
1	Computadora Personal	AT386	IBM o compatible
1	Impresora Láser	Laser Writer Pro 630	APPLE
1	Osciloscopio	2205	TEKTRONIX
1	Multímetro	E2373A	HEWLETT PACKARD
1	Programador Universal	JE680	JAMECO
1	Analizador de Estados Lógicos	3001 GPX	TEKTRONIX
1	Fuente de Poder	CPS250	TEKTRONIX
1	Tablero de Pruebas	JE26	JAMECO

Se utilizaron algunos paquetes de software para el diseño del circuito electrónico, de circuito impreso y de los programas, como lo son, OrCAD Schematic Design Tools y OrCAD PCB, el lenguaje Turbo Pascal, y paquetes como AVSIM51, Edit MSDOS, Word for Windows.

TABLA II .- RELACIÓN DEL MATERIAL UTILIZADO

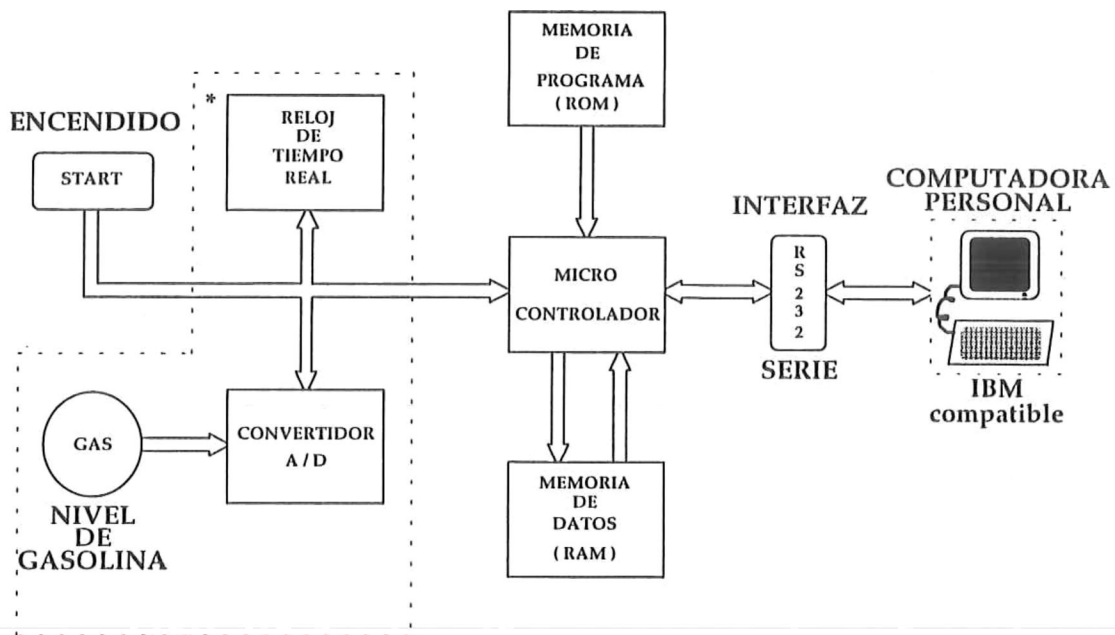
Referencia	Cantidad	Descripción	Número o Valor	Compañía
U1	1	Microcontrolador	80C31	INTEL
U7	1	Memoria RAM	62HC64	Texas Inst.
U3	1	Memoria EPROM	27HC64	Texas Inst.
U6	1	Decodificador de 3 a 8	74HC138	JAMECO
U2	1	Cerrojo 3 estados tipo D	74HC573	JAMECO
U8	1	Convertidor A/D	ADC0803	National Semiconductor
U9	1	Multivibrador monoestable de disparo múltiple	74HC123	JAMECO
U10	1	Compuerta AND	74HC08	JAMECO
U4	1	Reloj de tiempo real	7242	OKI
U5	1	Interfaz serie	MC145406	Motorola
U11	1	Regulador +5v	MAX666	MAXIM
U12	1	Regulador -5v	MAX660	MAXIM
XTAL2	1	Cristal	32768 Hz	JAMECO
XTAL1	1	Cristal	1.8432 MHz	JAMECO
D1,D2	2	Diodos	1N4001	NTE
C9,CD1↔CD10	11	Capacitores	0.1 $\mu$ F	NTE
C1↔C3,CV1	4	Capacitores	22 pF	NTE
C5	1	Capacitar	150 pF	NTE
C4	1	Capacitor	1 $\mu$ F	NTE
C6,C7 y C8	3	Capacitor	10 $\mu$ F	NTE
C10 y C11	2	Capacitores	200 pF	NTE
R1↔R3,R5,R8↔R10	7	Resistencias	10 K $\Omega$	NTE
R4 Y R5	2	Resistencias	5 K $\Omega$	NTE
R7	1	Resistencia	17 K $\Omega$	NTE
POT1↔POT3	3	Potenciómetros	1 K $\Omega$	NTE

## 6 FUNCIONAMIENTO DEL MÓDULO

### 6.1 FUNCIONAMIENTO

Las consideraciones hechas para el diseño del módulo fueron tanto de tipo técnico, como de tipo económico, tomando en cuenta la disponibilidad del material y equipo a utilizar para su desarrollo, de esta forma nos aseguramos que el microcontrolador que se utilizó estuviera disponible y a precios accesibles en el mercado regional, así como todos los dispositivos (circuitos integrados y componentes discretos) que complementan el sistema.

En la siguiente figura se muestra un diagrama a bloques del funcionamiento del módulo:



\* Explicación del diseño y sus características en la "parte B" de la memoria de Servicio Social.

FIGURA No. 1 DIAGRAMA A BLOQUES DEL MÓDULO

El módulo toma lecturas del nivel de gasolina, la hora y la fecha cada vez que el vehículo es encendido y almacena los datos en la memoria de acceso aleatorio (RAM), repitiendo el proceso cuando el automóvil es apagado, éste puede tomar un promedio de 18 muestras diarias, durante 31 días, 585 muestras en total (una muestra equivale a las lecturas de la fecha, hora y el nivel de gasolina en ese instante); posteriormente estos datos pueden ser analizados y ordenados mediante un programa de alto nivel realizado en Pascal, el cual se comunica directamente con el módulo por medio del puerto serie de una computadora, con esto podemos observar de manera estadística el uso que se le da al vehículo en relación a su gasto total de combustible, y el tiempo diario que estuvo en servicio.

Para probar la eficiencia y costeabilidad del módulo se hicieron pruebas de laboratorio; pruebas, tales como simular las variaciones de gasolina utilizando un tablero conectado al flotador de un tanque de gasolina, para esto, se activaba la interrupción del encendido del automóvil cada 5 minutos, en este instante se tomaban lecturas de la fecha y hora, los datos se pudieron recuperar satisfactoriamente, un ejemplo de esta prueba se muestra en la tabla III.

El módulo cuenta con una batería de respaldo de 9 volts, la cual tiene una duración limitada, y sirve para que en las ocasiones en que el módulo no se encuentre conectado con la batería del automóvil por cualquier razón mantenga el módulo sin que pierda información hasta por un período continuo de 50 horas.(la batería tiene una duración de 550 mA/H y el consumo del módulo es de 10 mA).

En la tabla III son especificados el día de la semana, el día el mes, el año, y la hora en la que se realizó la lectura así como el nivel de gasolina en ese momento, si el nivel tuvo incrementos con respecto al anterior (INC.), el gasto de combustible realizado en litros, el gasto económico en N\$ pesos, así como también si el automóvil fue encendido o apagado.

Tabla de muestras obtenidos de una prueba de laboratorio de módulo :

TABLA III .- RELACIÓN DE MUESTRAS EL MÓDULO

DÍA DE LA SEMANA	DÍA	MES	AÑO	HORA	LITROS	GASTO EN LITROS	GASTO EN N\$ PESOS	ENCENDIDO
LUNES	16	MAYO	1994	12:23:15	45.30	INC.	INC.	ACTIVO
LUNES	16	MAYO	1994	12:28:23	33.35	11.95	15.522	INACTIVO
LUNES	16	MAYO	1994	12:33:34	23.47	9.88	12.844	ACTIVO
LUNES	16	MAYO	1994	12:38:12	12.80	10.67	13.871	INACTIVO
LUNES	16	MAYO	1994	12:43:45	80.00	INC.	INC.	ACTIVO
LUNES	16	MAYO	1994	12:48:28	58.93	21.07	28.171	INACTIVO
LUNES	16	MAYO	1994	12:53:29	43.32	15.61	20.293	ACTIVO
LUNES	16	MAYO	1994	12:58:56	25.65	17.67	22.971	INACTIVO
LUNES	16	MAYO	1994	13:03:18	73.33	INC.	INC.	ACTIVO
LUNES	16	MAYO	1994	13:08:49	64.89	8.44	10.972	INACTIVO
LUNES	16	MAYO	1994	13:13:26	58.21	6.68	8.684	ACTIVO
LUNES	16	MAYO	1994	13:18:38	31.56	26.65	34.645	INACTIVO
GASTO TOTAL DE COMBUSTIBLE EN LITROS						128.70		
GASTO TOTAL EN N\$ PESOS							167.973	

## 6.2 CARACTERÍSTICAS Y ESPECIFICACIONES

Entre las características más relevantes del módulo se pueden enumerar las siguientes:

- Es alimentado por la batería del automóvil.
- Tiene almacenamiento de datos en memoria de estado sólido.
- Tiene comunicación externa a través de un puerto serie RS-232.
- La hora y fecha son totalmente programables.
- Es de bajo costo.

Además se tienen las siguientes especificaciones :

### CONVERTIDOR ANALÓGICO - DIGITAL

Dispositivo	ADC0803.
Resolución	8 bits.
Exactitud	$\pm 1/2$ LSB.
Tiempo de conversión	100 $\mu$ S.

### ALMACENAMIENTO DE DATOS

Medio	Memoria de Edo. sólido
Dispositivo	62C64
Capacidad	8 Kilo octetos
Núm. Máx. de muestras	585

PROGRAMACIÓN

Fecha	Día, Mes, Año y día de Semana
Hora	Segundos, Minutos y Hora

ENERGÍA

Medio	Batería de 12 volts y Batería de respaldo de 9 volts.
-------	--

CONSUMO

Consumo Constante	10 mA a 25 °C.
-------------------	----------------

TEMPERATURA

Temp. Máxima	0°C a 70 °C (En general).
--------------	---------------------------

## 7 CIRCUITERÍA

El módulo es básicamente un sistema de adquisición de datos, que se conforma de 12 circuitos, como se ilustra en la figura 8.

Se utilizó el microcontrolador 80C31 por su versatilidad, así como por la cantidad de información (data books, hojas de programación, etc.) disponible de éste, desarrollándose un sistema mínimo con todos los dispositivos necesarios.

Para la alimentación se utilizan dos reguladores los cuales nos proporcionan los voltajes requeridos (+5 y -5). En la parte de acoplamiento se utilizó un convertidor analógico-digital con su arreglo correspondiente. Para el control del tiempo se añadió el circuito generador de tiempo real. La memoria para el almacenamiento de la información utilizada es la RAM de 8k octetos puesto que en ella puede guardar la lectura de datos hasta por 30 días aproximadamente. Para la comunicación serie se utilizó el circuito de interfase que permite conectar directamente al microcontrolador con la computadora por medio del puerto serie.

A continuación se describen brevemente algunos de los elementos que conforman el módulo.

### 7.1 EL MICROCONTROLADOR

Para el control del módulo se eligió un sistema basado en el microcontrolador en su modo mínimo, ya que el consumo de éste en estado de reposo (Power Down State) baja hasta 0.050 mA y se puede mantener con una alimentación 2 Volts, y aunque en el prototipo final del módulo no se hizo uso de esta característica, se prefirió este microcontrolador para tener más facilidades en futuras modificaciones o mejoras al sistema.

Algunas de las características de este microcontrolador es que es de bajo consumo así como de gran facilidad manejo y programación. Bajo su supervisión se ejecutan las diferentes tareas de control. Algunas de sus principales ventajas que este tiene líneas de recepción/transmisión para comunicación serie, 128 octetos RAM interna, 2 temporizadores, 32 I/O líneas individuales y bidireccionables, 2 interrupciones externas, oscilador interno y 1 puerto en su modo mínimo o 4 puertos en modo mono pastilla.

El microcontrolador tiene espacio separados de direcciones, para almacenar los programas y datos como se muestra en las figuras 2 y 3. Esta separación de memoria de programa y datos permite que el espacio de memoria correspondiente a los datos pueda ser accedido con direcciones de 8 bits, con lo cual este puede ser manipulado y rápidamente accedido por un CPU de 8 bits.

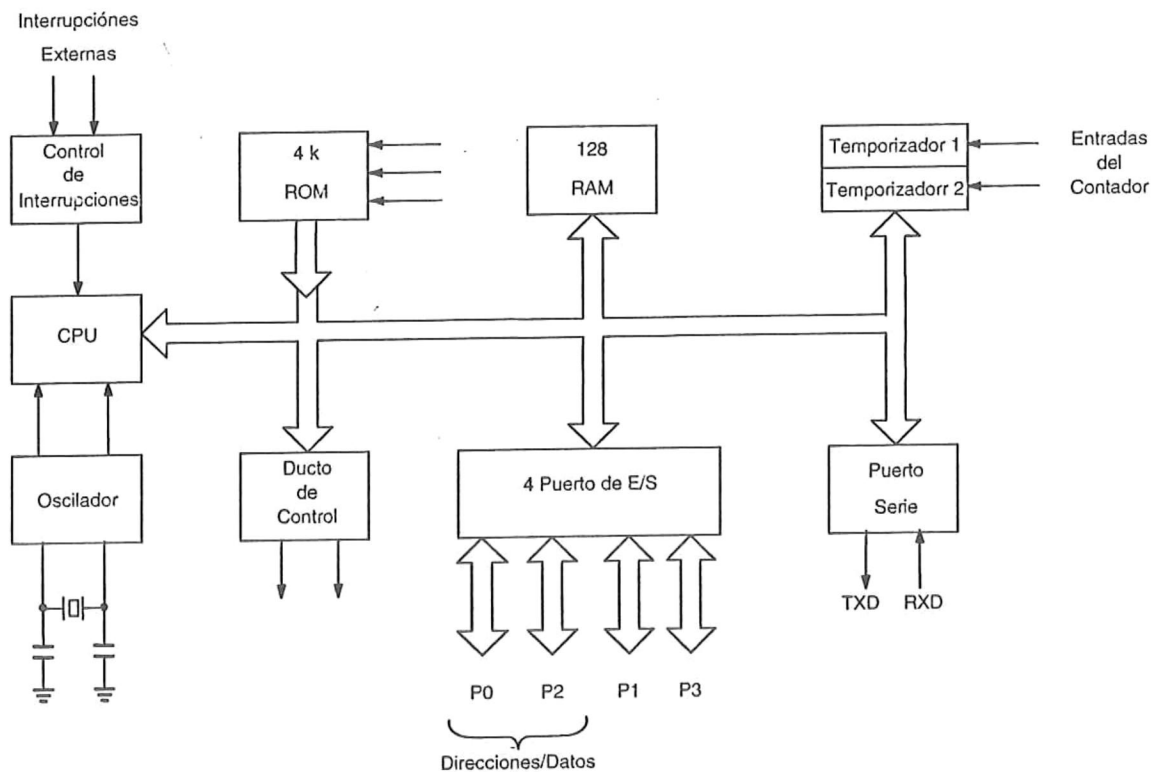


FIGURA No. 2 DIAGRAMA A BLOQUES DEL 80C51

La mayoría de las versiones de los microcontroladores basados en el 80C51 tienen una ROM interna de 4k octetos, el microcontrolador utilizado en el módulo carece de esta, y puesto que en el sistema no se utiliza, ya que el total del programa esta contenido en la memoria externa, el 80C31 nos permite realizar las mismas operaciones que el 80C51.

La memoria de datos (RAM) ocupa diferentes espacios de direcciones programables, en el microcontrolador los 128 octetos de datos significativamente más bajos se encuentran contenidos en el mismo integrado. 64k octetos de RAM externa pueden ser direccionados en los espacios de memoria externa para datos. El CPU genera las señales de lectura y escritura, RD y WR, según las necesite durante el acceso externo a la memoria de datos.

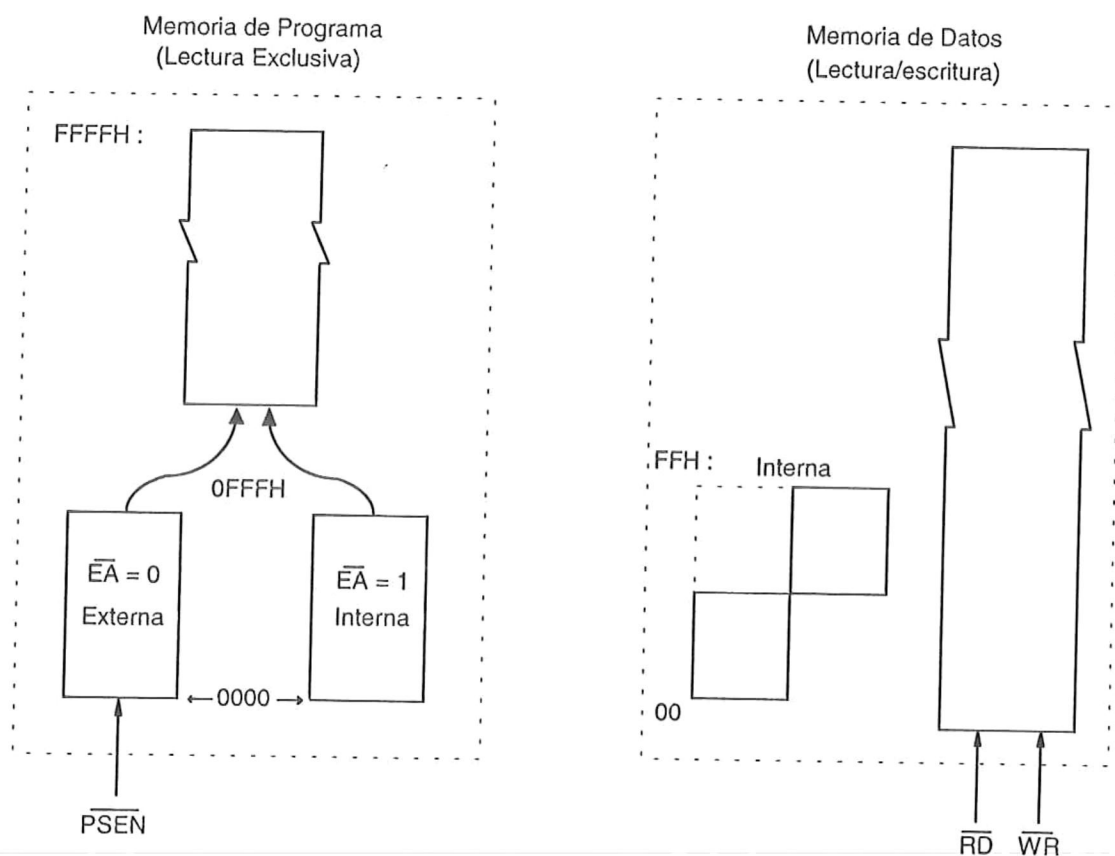


FIGURA No. 3 DISTRIBUCIÓN DE LA MEMORIA DEL MICROCONTROLADOR.

## 7.2 LAS MEMORIAS RAM Y EPROM

La memoria de acceso aleatorio que se utilizó para guardar los datos es de 8k octetos y esta es suficiente para almacenar los datos obtenidos hasta por un periodo de 31 días realizando un promedio de 18 muestras diarias (una muestra equivale a las lecturas de la fecha, hora y el nivel de gasolina en ese instante). La memoria borrable de lectura exclusiva tiene capacidad para almacenar 8k octetos de información y es donde se encuentra el programa con el cual trabaja el microcontrolador.

En esta tabla se muestra el mapa de memorias.

**TABLA IV .- MAPA DE MEMORIA**

MEMORIA	DIRECCIÓN DE INICIO	DIRECCIÓN DE FIN
LECTURA EXCLUSIVA	0000H	1FFFH
LECTURA / ESCRITURA	2000H	3FFFH

## 7.3 FUENTE DE ALIMENTACIÓN

El módulo se alimenta de la batería de 12 volts del automóvil, y cuenta con una batería de respaldo de 9 volts para cuando la del automóvil se encuentre desconectada, la fuente de alimentación esta compuesta por dos reguladores de voltaje, uno de +5 volts que nos proporciona alimentación para todo el sistema (VCC), y el otro regulador nos proporciona -5 volts (-V), éste se conecta en serie con el MAX666 y nos proporciona exclusivamente la alimentación para el circuito de interfaz serie el cual utiliza de ambos voltajes -5 y +5 volts para su funcionamiento. El regulador de +5 volts soporta hasta +18 volts de entrada, y en el módulo es alimentado por 9 o 12 volts según sea el caso, el regulador de -5 volts es alimentado a su vez con el voltaje de salida de el regulador de +5 volts.

## 7.4 LA COMUNICACIÓN CON EL USUARIO

Esta se efectúa por medio del circuito de interfase serie, un manejador RS232, el cual es un circuito CMOS por lo tanto es de bajo consumo y compatible con el resto de la circuitería, este se conecta a una computadora personal por medio de su puerto serie, el microcontrolador tiene integradas las líneas TX y RX (transmisión y recepción respectivamente) para realizar la comunicación serie con las mínimas adaptaciones.

El consumo de corriente del circuito de interfase utilizado es 600  $\mu$ A aproximadamente a diferencia de otros manejadores RS232 como por ejemplo el MAX232 de MAXIM, consume de 5 mA aproximadamente (no se realiza con tecnología CMOS), razón por la cual se prefirió el MC145406 de MOTOROLA.

## 7.5 CIRCUITOS MISCELÁNEOS

El circuito de acondicionamiento de señal utilizado es un cerrojo octal triestado de tecnología CMOS para garantizar su bajo consumo, este nos separa la direcciones de los datos y es coordinado a través de la línea de control ALE.

El decodificador/demultiplexor de 3 a 8 es utilizado para seleccionar el circuito integrado con el que el microcontrolador se comunica, y la selección se realiza mediante las direcciones A13, A14 y A15, los integrados que son activados por este circuito son la RAM, el convertidor analógico-digital y el reloj de tiempo real.

TABLA V .- MAPA DE DIRECCIONES.

DIRECCIONES	A15	A14	A13	DISPOSITIVO
0000 - 1FFF	0	0	0	EPROM
2000 - 3FFF	0	0	1	RAM
4000 - 5FFF	0	1	0	ADC0803
6000 - 7FFF	0	1	1	7242

La configuración de estos circuitos puede ser observada en la figura 8.

## 8 ADQUISICIÓN DE DATOS

Para realizar las lecturas del nivel de gasolina, se conecta la señal de salida de la resistencia (que se encuentra conectada al flotador del tanque de la gasolina), a VI+, señal que ésta en el convertidor analógico - digital del módulo, con esto podemos observar el nivel del tanque de combustible (En la parte B de este trabajo se explica más detalladamente este proceso).

Para la lectura del encendido y apagado del automóvil, se utilizó un multivibrador (doble) monoestable de disparo múltiple conectado a una compuerta AND.

Observando la tabla de funciones de este integrado (figura 4), y la configuración de este en el diagrama esquemático (figura 8), podemos ver que al recibir la señal de encendido, tenemos una transición positiva en el multivibrador A, con ello nos entrega un pulso negativo en su salida Q negada, lo cual es suficiente para que la compuerta AND cambie de estado a un nivel bajo y active la interrupción INT0, de manera semejante al desconectar la señal de encendido sucede una transición negativa a la entrada del multivibrador B, generando en su salida Q negada un pulso bajo, cambiando de estado la compuerta y activando la interrupción INT0.

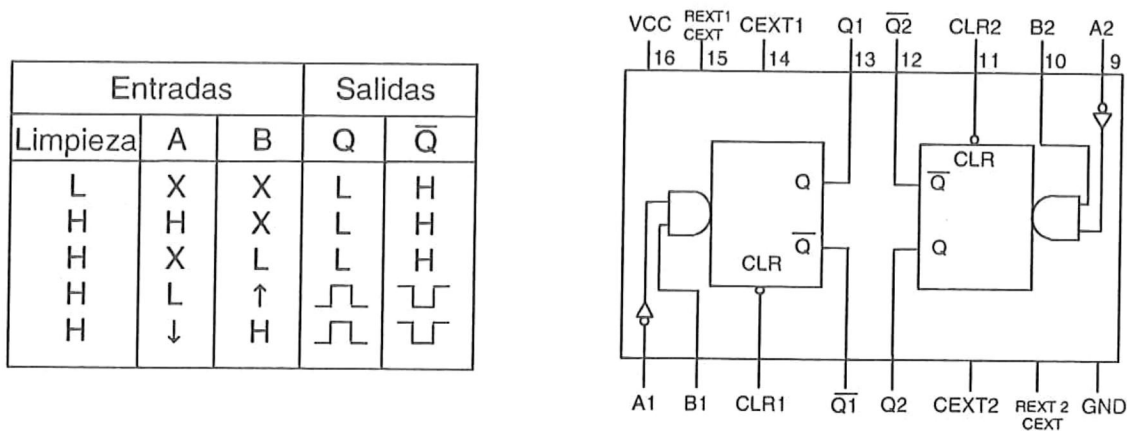


FIGURA No. 4 DIAGRAMA DE CONEXIONES Y TABLA DE FUNCIONES DEL 74HC123.

De esta manera cada que la interrupción es activada el microcontrolador lo registra y realiza la lectura de nivel de gasolina, fecha y hora, almacenando todos los datos en la RAM.

## 9 LA PROGRAMACIÓN

La programación se divide en dos partes, la destinada al manejo del módulo que es la realizada en lenguaje ensamblador del microcontrolador y la de lenguaje de alto nivel .

La primera es explicada en la parte B de esta memoria de Servicio Social. La segunda se realizó en lenguaje Turbo Pascal ya que este lenguaje es fácil de manejar y se tiene conocimiento de el, así como también es posible realizar procedimientos para lograr la comunicación serie con relativa facilidad.

El programa utilizado para la interpretación de los datos cuenta con 5 opciones con las cuales podemos manejar la información y generar los reportes correspondientes de las muestras previamente adquiridas por módulo. Las opciones con las que cuenta el programa son: adquirir la información contenida en el módulo (GET\_INFO), observar la hora del reloj de tiempo real del módulo y modificarla si lo deseamos (TIME), salvar los datos obtenidos en un archivo (SAVE\_AS), desplegar los datos en pantalla (DISPLAY\_INFO), así como generar un reporte de estos directo a la impresora (PRINT).

En esta sección sólo se analizaran los diagramas de flujo. El listado completo del programa se incluye en el anexo B.

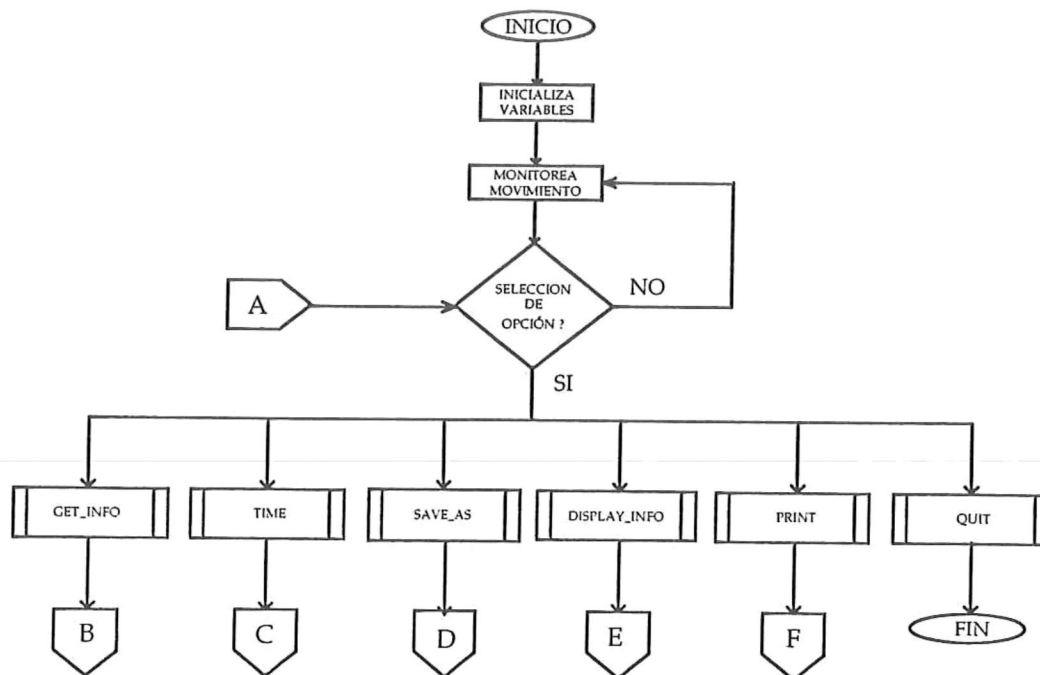


FIGURA No. 5 DIAGRAMA DE FLUJO DEL PROGRAMA EN "TURBO PASCAL".

La opción GET\_INFO (figura 6a) nos sirve para la realizar la lectura de la información contenida en el módulo o en un archivo grabado con anterioridad; la adquisición de datos del módulo se realiza mediante comunicación serie a través de un puerto RS232, obteniendo de el todos los datos que se encuentran en la RAM al momento de la lectura.

Con la opción TIME (figura 6b) podemos observar la hora que el módulo tiene registrada en el circuito generador de tiempo real y si ésta no es correcta podemos actualizarla.

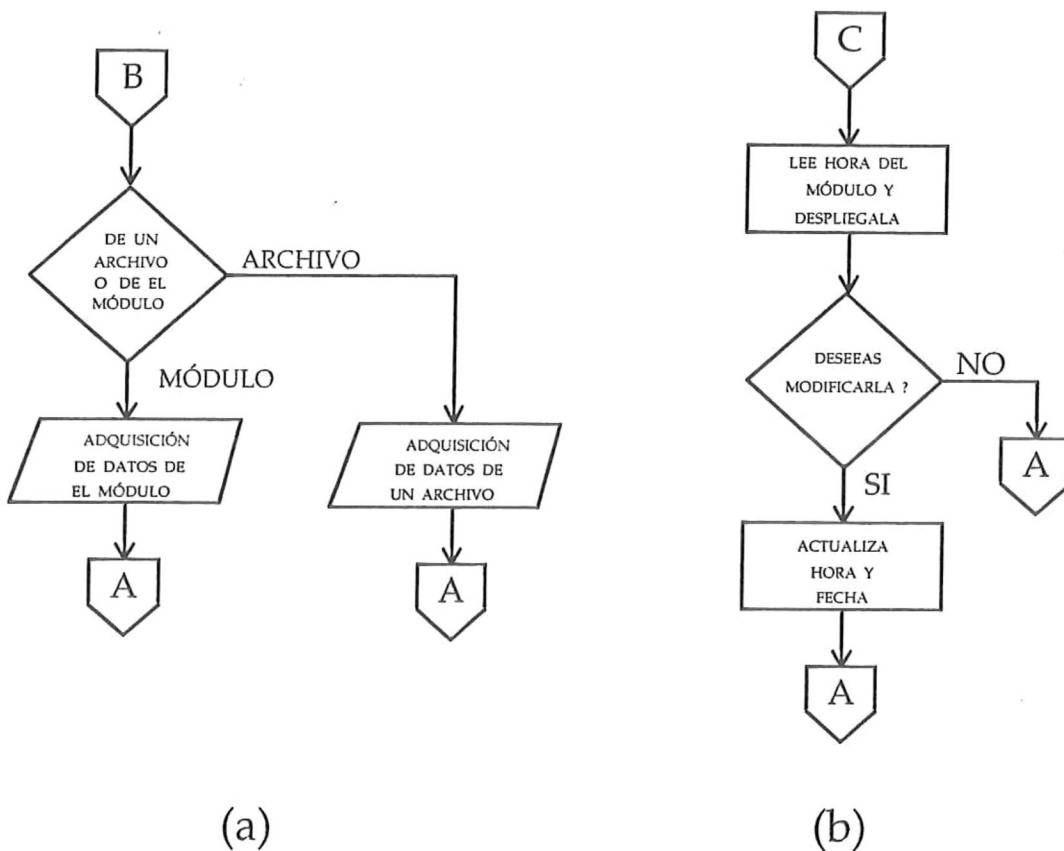


FIGURA No. 6 SUBROUTINAS DEL PROGRAMA EN "TURBO PASCAL".

La opción SAVE\_AS (figura 7a) nos permite salvar la información obtenida en un archivo, para posteriores consulta de ella.

DISPLAY\_INFO (figura 7b), esta opción nos despliega la información, previamente obtenida, en pantalla únicamente.

Y la ultima opción es PRINT (figura 7c) y con ella podemos imprimir un reporte de la información obtenida, el reporte generado es semejante al que se muestra en la tabla III.

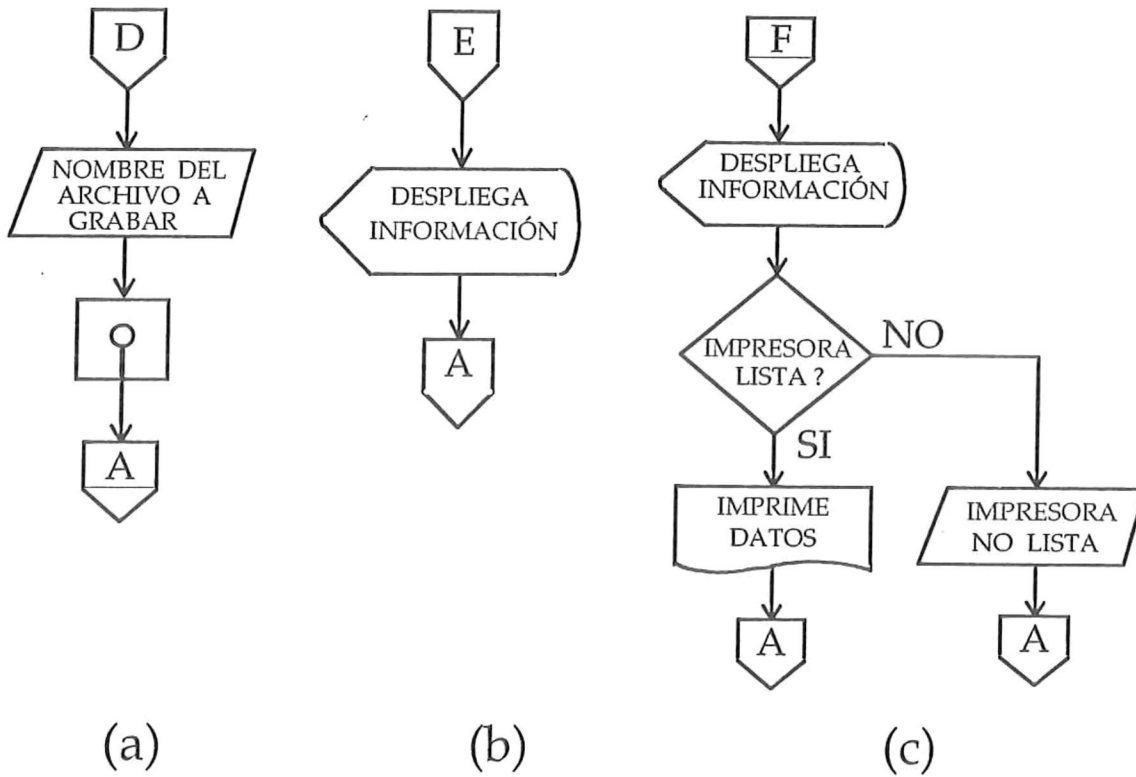


FIGURA No. 7 SUBRUTINAS DEL PROGRAMA EN "TURBO PASCAL".

## 10 EL CIRCUITO IMPRESO

El circuito impreso fue realizado con el programa OrCAD PCB, se diseñó un circuito de doble cara, en una tarjeta de 6 x 6 pulgadas, en él se encuentran los 12 circuitos que conforman todo el módulo, así como el resto de la circuitería necesaria.

La tarjeta del circuito se desarrollo en el laboratorio para impresos de esta escuela, la cual se elaboró mediante el método de exposición de placa fotosensible, las mascarillas utilizadas en su elaboración se muestran por partes en las figuras 11 y 12 las cuales corresponden a ambas caras del impreso. Las figuras 11 y 12 nos muestran la distribución de las pistas en el impreso, así mismo las figuras 9 y 10 nos dan la distribución de los componentes en el moduló.

## 11 DISCUSIONES

Los principales problemas presentados durante la realización del Servicio Social fueron :

1. La falta de apoyo económico para realizar el servicio social.
2. Necesidad de un área de trabajo definida para los prestadores de Servicio Social.
3. No existen antecedentes de esta opción de titulación en la Escuela de Ingeniería, por lo que fue difícil realizar los trámites de aprobación del Servicio Social como opción de titulación.

Los problemas que se presentaron durante la realización de la memoria fueron :

Falta de un formato definido de para su realización, ya que hasta el momento en que se escribió esta memoria no existía ningún trabajo de este tipo en la biblioteca de esta unidad.

No existe en la biblioteca de esta unidad universitaria ninguna tesis, memoria o trabajo escrito de titulación de la carrera de ingeniería electrónica, por lo cual hubo la necesidad de consultar otro tipo de tesis y trabajos con enfoques totalmente distintos a nuestra área.

## 12 CONCLUSIONES

De las características y especificaciones que se obtuvieron para el módulo de medición de parámetros de automóvil realizado como servicio social en la Escuela de Ingeniería, se puede apreciar que se cumplieron los objetivos iniciales del presente trabajo.

El diseño del módulo se realizó con las consideraciones necesarias para que éste funcionara correctamente, y aunque se podría optimizar aún más si el módulo se diseñase para que trabaje en modo de bajo consumo cuando éste se alimente por la batería de 9 volts.

Además de los resultados obtenidos de la prueba realizada a el módulo de medición de parámetros de automóvil, se concluye que el funcionamiento del mismo es bastante satisfactorio y podría funcionar perfectamente como un apoyo administrativo para cualquier empresa o institución pública.

Sin embargo en el módulo de medición de parámetros de automóvil existen algunos aspectos que pueden mejorarse :

⇒ Generalizar las lecturas de los niveles de gasolina, esto es, que el moduló funcionase en cualquier vehículo, ya que este diseño quedo hecho específicamente para automóviles DODGE ya que de esta marca era el automóvil para el que cual se realizó.

⇒ Aumentar la capacidad de memoria hasta 32k octetos, ya que con esta memoria se daría margen para que el módulo permaneciera recopilando información hasta por períodos de 4 meses aproximadamente.

⇒ Diseñar y construir un envase de plástico práctico y adecuado, así como reducir las dimensiones del impreso para su fácil instalación dentro de cualquier automóvil.

⇒ Agregar la lectura de otros parámetros, tales como velocidades máximas alcanzadas durante sus recorridos, e incluir el control de algunas variables, como por ejemplo el poder programarle al vehículo una velocidad tope, así como también un horario exclusivo de uso.

⇒ Modificar el programa y hacerlo más eficiente, para que pueda ser utilizado por cualquier persona.

### 13 BIBLIOGRAFÍA

✎ Eduardo Alcalde & Miguel García, 1987, *METODOLOGÍA DE LA PROGRAMACIÓN*, "Diagramas de flujo", Mc Graw Hill, primera edición, ESPAÑA, p 11-28

✎ Juan A. Navarro & Eduardo López, 1993, *CURSO TUTORIAL "MICROCONTROLADORES"*, III CIECE PUEBLA '93, MÉXICO, 650 pp

✎ Ken Freud and John H. Haynes, 1989, *AUTOMOTIVE ELECTRICAL MANUAL*, "Indicator lights and gauges", Haynes North America, first edition, USA, p 171-194

✎ MAXIM, 1994, *NEW RELEASE DATA BOOK Volume II*, "Power supply circuits", USA, p 4-1-4-251

✎ MOTOROLA, 1993, *COMMUNICATIONS DEVICE DATA*, "Data sheets", rev 3, USA, 2-706-2-713 pp

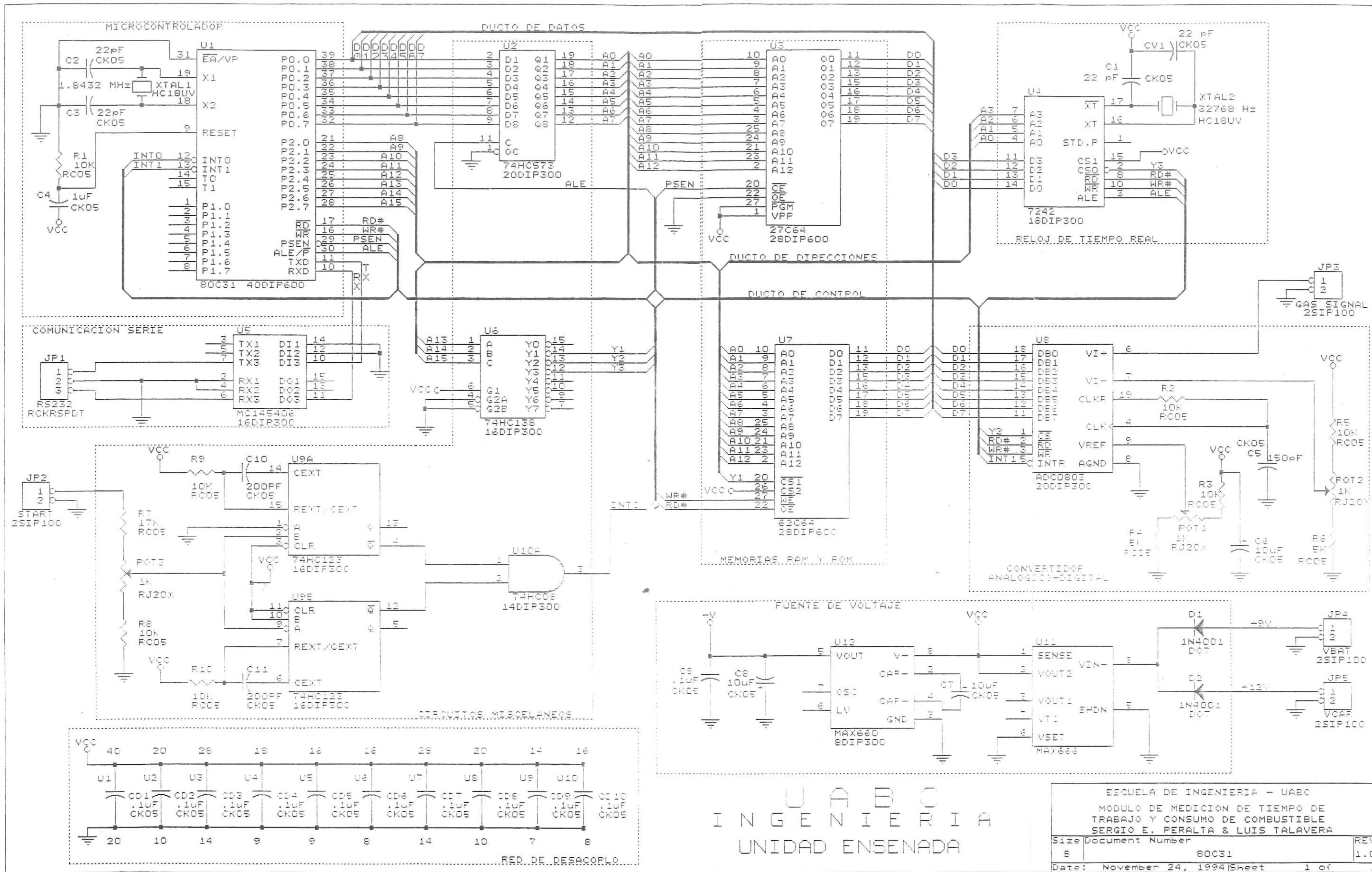
✎ MOTOROLA, 1989, *HIGH-SPEED CMOS LOGIC DATA*, "Data sheets", rev 4, USA, p 5-1-5-552

✎ National semiconductors, 1993, *DATA ACQUISITION DATA BOOK*, "Analog to digital converts", USA, p 2-1-2-547

✎ OrCAD, 1993, *SCHEMATIC DESIGN TOOLS*, sixth edition, USA, 712 pp

- ✎ OrCAD, 1993, *PCBOARD LAYOUT TOOLS 368 PLUS*, first edition, USA, 359 pp
  
- ✎ Philips, 1993, *80C51 BASED 8 BIT MICROCONTROLLERS*, USA, p 3-135
  
- ✎ Robert Jordain and the Peter Norton computing group, 1992, *PROGRAMMER'S PROBLEMS SOLVER*, Brady Publishing, second edition, USA, 596 pp
  
- ✎ Stephen K. O'Brien, 1989, *TURBO PASCAL 5.5: THE COMPLETE REFERENCE*, Borland Osborne/Mc Graw Hill, first edition, USA, 917 pp
  
- ✎ William David Cooper, 1982, *INSTRUMENTACIÓN ELECTRÓNICA Y MEDICIONES*, "Instrumentos indicadores de corriente directa", Prentice Hall Hispanoamericana, segunda edición, MÉXICO, p 51-69

## 14 ANEXO A



UABCA  
INGENIERIA  
UNIDAD ENSENADA

ESCUOLA DE INGENIERIA - UABC  
MÓDULO DE MEDICIÓN DE TIEMPO DE TRABAJO Y CONSUMO DE COMBUSTIBLE  
SERGIO E. PERALTA & LUIS TALAVERA

Size	Document Number	REV
E	80C31	1.0
Date:	November 24, 1994	Sheet 1 of 1

FIGURA No. 8 DIAGRAMA ESQUEMÁTICO DEL MÓDULO.

DISTRIBUCIÓN DE LOS COMPONENTES

LADO 1

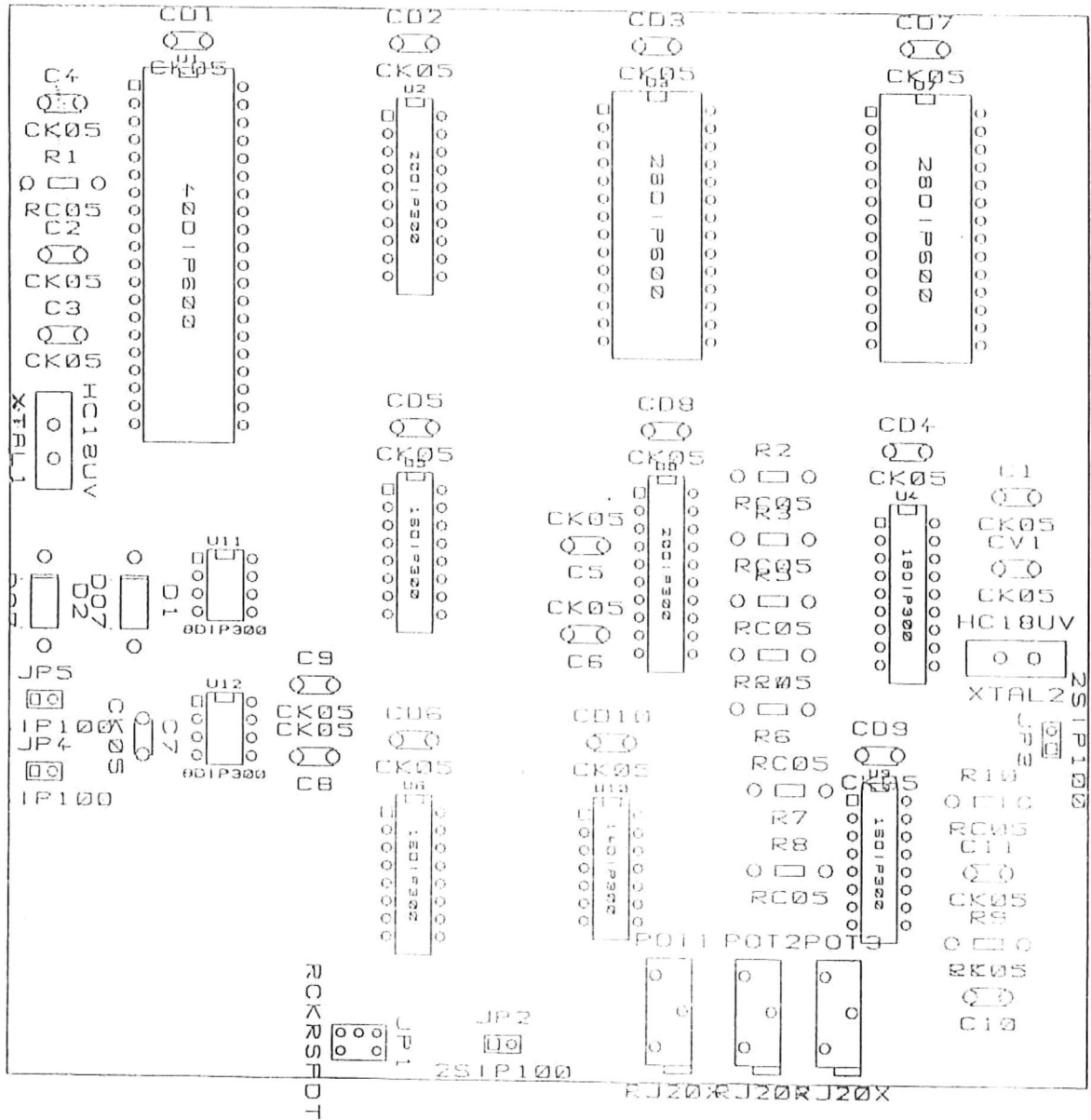


FIGURA No. 9

DISTRIBUCIÓN DE LOS COMPONENTES  
LADO 2

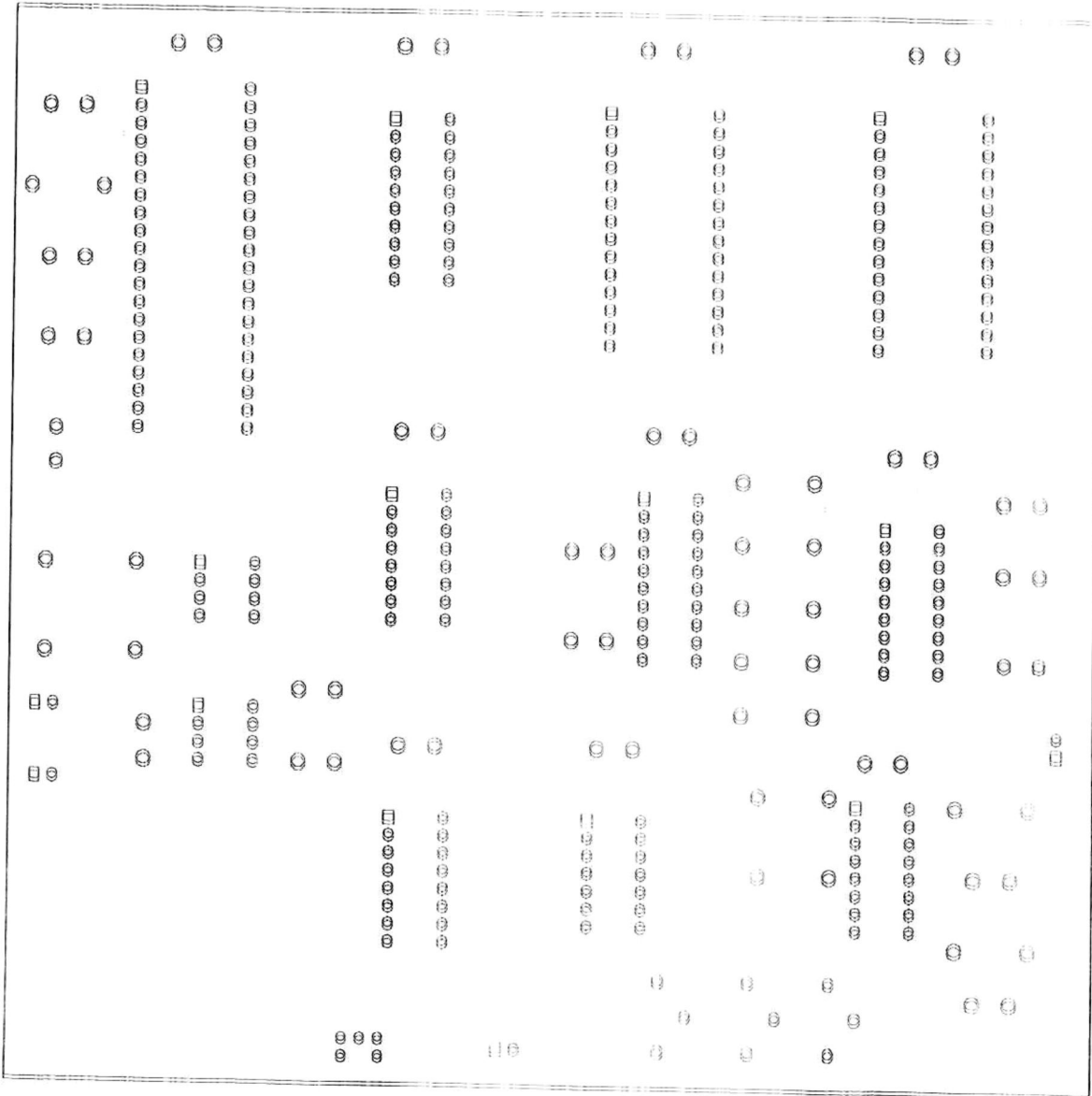


FIGURA No. 10

DISTRIBUCIÓN DE LAS PISTAS  
LADO 1

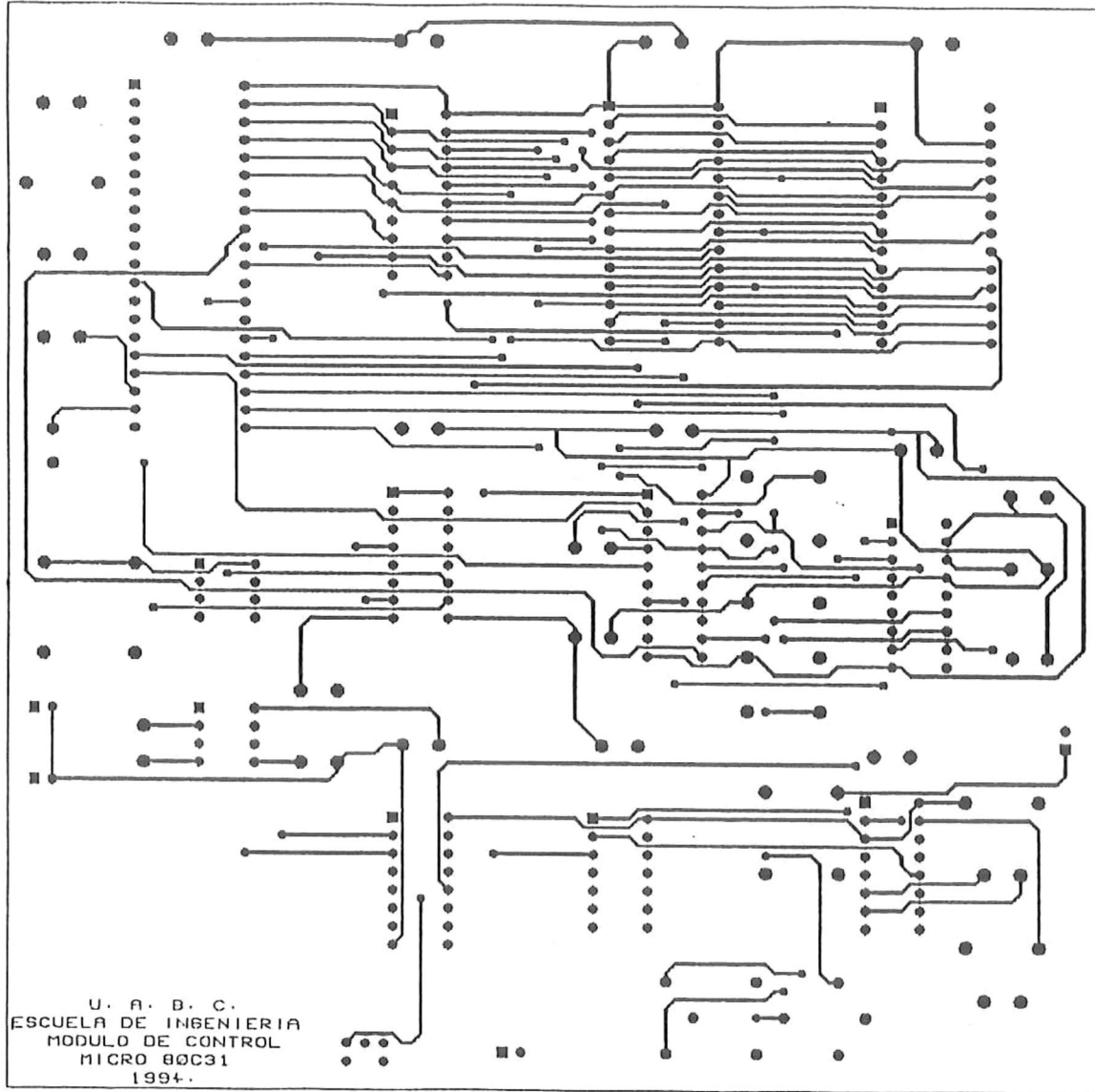


FIGURA No. 11

DISTRIBUCIÓN DE LAS PISTAS

LADO 2

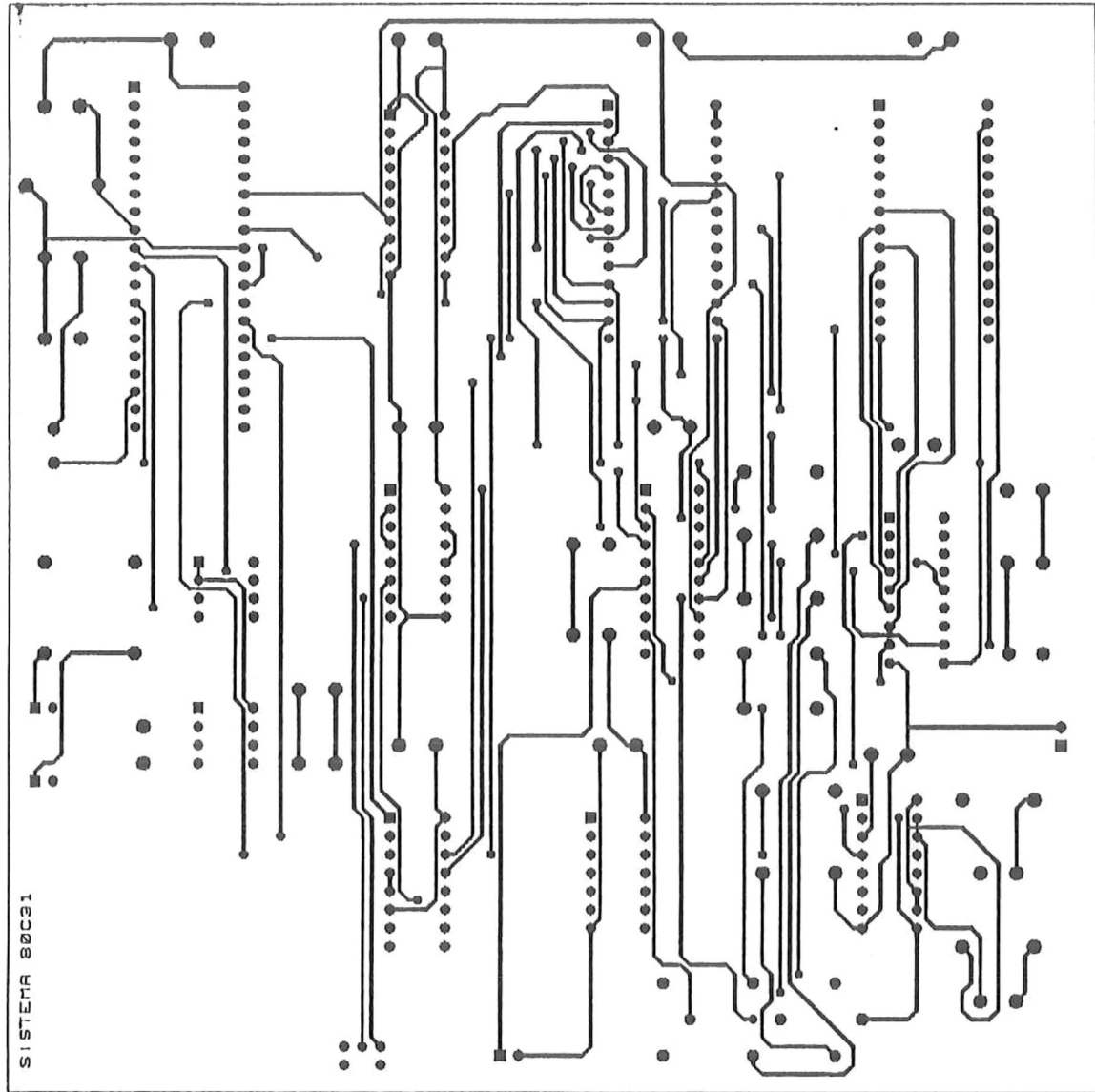


FIGURA No. 12

**15 ANEXO B**

---

## PROGRAMA EN TURBO PASCAL

Listado del programa en turbo pascal que se utilizo para la interpretación de los datos :

**PROGRAM MODULO;**

**USES**

CRT,DOS,PRINTER;

**TYPE**

DATOS\_RECIBIDOS = RECORD  
    SEGUNDOS                  : BYTE;  
    MINUTOS                   : BYTE;  
    HORAS                     : BYTE;  
    DIAS                      : BYTE;  
    MES                       : STRING[12];  
    YEAR                      : BYTE;  
    DIA\_DE\_LA\_SEMANA          : STRING[9];  
    DATO\_DEL\_SENSOR           : REAL;  
    GASTO\_EN\_LITROS           : REAL;  
    GASTO\_EN\_EFECTIVO         : REAL;  
    GASOLINA                  : STRING[10];

END;

NOMBRE\_OF\_FILE = FILE OF DATOS\_RECIBIDOS;

**VAR**

REGS                          : REGISTERS;  
RENOMBRA                      : NOMBRE\_OF\_FILE;  
CONTROLA                      : DATOS\_RECIBIDOS;  
CONDICION,TEMP,TEMP2,TEMP3   : INTEGER;  
PUERTO,PUERTOS,CUENTA,CAMBIA : INTEGER;  
A,B,C,D,E,F,G,H,I,J,K,L      : INTEGER;  
ACUMULADOR                    : INTEGER;  
RESULTADO                     : INTEGER;  
SELECCION                     : BOOLEAN;  
MONITOR                       : WORD;  
BASE                           : WORD;  
ARREGLO                       : ARRAY[0..8177] OF BYTE;  
DESPLIEGA                     : ARRAY[0..5] OF DATOS\_RECIBIDOS;

---

```
FUNCTION DIA_ACTUAL (A : INTEGER) : STRING;
```

```
BEGIN
  CASE A OF
    0 : DIA_ACTUAL := 'Domingo';
    1 : DIA_ACTUAL := 'Lunes  ';
    2 : DIA_ACTUAL := 'Martes  ';
    3 : DIA_ACTUAL := 'Miercoles';
    4 : DIA_ACTUAL := 'Jueves  ';
    5 : DIA_ACTUAL := 'Viernes  ';
    6 : DIA_ACTUAL := 'Sabado  ';
  END;
END;
```

```
FUNCTION MES_ACTUAL (A : INTEGER) : STRING;
```

```
BEGIN
  CASE A OF
    1 : MES_ACTUAL := 'Enero  ';
    2 : MES_ACTUAL := 'Febrero ';
    3 : MES_ACTUAL := 'Marzo  ';
    4 : MES_ACTUAL := 'Abril  ';
    5 : MES_ACTUAL := 'Mayo  ';
    6 : MES_ACTUAL := 'Junio  ';
    7 : MES_ACTUAL := 'Julio  ';
    8 : MES_ACTUAL := 'Agosto ';
    9 : MES_ACTUAL := 'Septiembre';
    10 : MES_ACTUAL := 'Octubre ';
    11 : MES_ACTUAL := 'Noviembre';
    12 : MES_ACTUAL := 'Diciembre';
  END;
END;
```

```
FUNCTION LITROS(A : INTEGER) : REAL;
```

```
VAR
  CAPACIDAD_TANQUE : REAL;

BEGIN
  CAPACIDAD_TANQUE := 80;
  LITROS := (A * CAPACIDAD_TANQUE) / 255;
END;
```

---

```
PROCEDURE INICIALIZA_RATA;
```

```
BEGIN
```

```
    REGS.AX := $0000;    {ACTIVA E INICIALIZA EL MOUSE}
    INTR($33,REGS);
    REGS.AX := $0001;
    INTR($33,REGS);
```

```
    REGS.AX := $0007;    {DELIMITA LA PANTALLA PARA EL MOUSE}
    REGS.CX := $0000;
    REGS.DX := 636;
    INTR($33,REGS);
    REGS.AX := $0008;
    REGS.CX := $0000;
    REGS.DX := 190;
    INTR($33,REGS);
```

```
    TEXTCOLOR(10);      {DA LA FORMA DEL CURSOR DEL MOUSE PARA}
    TEXTBACKGROUND(10); {EL MODO DE TEXTOS}
    REGS.AX := $000A;
    REGS.BX := $0000;
    REGS.CX := $FF00;
    REGS.DX := $00DB;
    INTR($33,REGS);
```

```
END;
```

```
PROCEDURE GET_RATA_POSICION;
```

```
BEGIN
```

```
    REGS.AX := 3;      {NOS ENTREGA LA POSICION DEL MOUSE}
    INTR ($33,REGS);  {CUANDO SUCEDE UN CLICK}
    IF REGS.BX = 1 THEN
        BEGIN
            IF ((REGS.CX>= 24 )AND(REGS.CX<=112))AND(REGS.DX=0) THEN CUENTA := 1;
            IF ((REGS.CX>=128)AND(REGS.CX<=216))AND(REGS.DX=0) THEN CUENTA := 2;
            IF ((REGS.CX>=232)AND(REGS.CX<=320))AND(REGS.DX=0) THEN CUENTA := 3;
            IF ((REGS.CX>=336)AND(REGS.CX<=424))AND(REGS.DX=0) THEN CUENTA := 4;
            IF ((REGS.CX>=448)AND(REGS.CX<=528))AND(REGS.DX=0) THEN CUENTA := 5;
            IF ((REGS.CX>=544)AND(REGS.CX<=632))AND(REGS.DX=0) THEN CUENTA := 6;
            IF ((REGS.CX>=252)AND(REGS.CX<=296))AND(REGS.DX=144) THEN PUERTOS:= 1;
            IF ((REGS.CX>=336)AND(REGS.CX<=380))AND(REGS.DX=144) THEN PUERTOS:= 2;
```

```
        END;
```

```
END;
```

---

PROCEDURE VALORES\_INICIALES;

```
BEGIN
  A := 14; B := 15;
  C := 14; D := 15;
  E := 14; F := 15;
  G := 14; H := 15;
  I := 14; J := 15;
  K := 14; L := 15;
END;
```

PROCEDURE INICIALIZA;

```
BEGIN
  CLRSCR;
  TEMP := 0;
  TEMP2 := 0;
  TEMP3 := 0;
  CUENTA := 1;
  CAMBIA := 0;
  PUERTO := 1;
  PUERTOS := 0;
  CONDICION:=0;
  SELECCION := FALSE;
  VALORES_INICIALES;
  INICIALIZA_RATA;
END;
```

PROCEDURE MONITOREA;

```
BEGIN

  REGS.AH :=3;
  REGS.DX :=1;
  INTR($14,REGS);
  MONITOR := (REGS.AH);
END;
```

PROCEDURE RECIBE (CUANTOS : INTEGER);

```
VAR
  DATOS,COMPARA : BYTE;
```

```
BEGIN
  COMPARA := 15;
  BASE := $2F8;
  ACUMULADOR := 0;
  REPEAT
    MONITOREA;
```

```
IF (MONITOR AND 97) =97 THEN
  BEGIN
    DATOS := PORT[BASE];
    DATOS := (DATOS)AND(COMPARA);
    ARREGLO[ACUMULADOR] := DATOS;
    WRITELN('R ',DATOS,',',ACUMULADOR);
    ACUMULADOR := ACUMULADOR + 1;
  END;
UNTIL CUANTOS = ACUMULADOR;
END;

PROCEDURE TRANSMITE(CUANTOS : INTEGER);

  BEGIN
    BASE:= $2F8;
    ACUMULADOR := 0;
    REPEAT
      MONITOREA;
      IF (MONITOR AND 96) =96 THEN
        BEGIN
          PORT[BASE]:=ARREGLO[A];
          ACUMULADOR := ACUMULADOR + 1;
        END;
      UNTIL CUANTOS = ACUMULADOR;
    END;

PROCEDURE DISPLAY OPCIONES;

  BEGIN
    TEXTBACKGROUND(TEMP);
    CLRSCR;
    TEXTCOLOR(A);
    TEXTBACKGROUND(B);
    GOTOXY(4,1);
    WRITE(' GET INFO ');
    TEXTCOLOR(C);
    TEXTBACKGROUND(D);
    GOTOXY(17,1);
    WRITE(' TIME ');
    TEXTCOLOR(E);
    TEXTBACKGROUND(F);
    GOTOXY(30,1);
    WRITE(' SAVE AS ');
    TEXTCOLOR(G);
    TEXTBACKGROUND(H);
    GOTOXY(43,1);
    WRITE('DISPLAY INFO');
    GOTOXY(56,1);
    TEXTCOLOR(I);
    TEXTBACKGROUND(J);
    WRITE(' PRINT ');
```

```
    TEXTCOLOR(K);
    TEXTBACKGROUND(L);
    GOTOXY(69,1);
    WRITE(' QUIT ');
END;
```

```
PROCEDURE DISPLAY OPCIONES_TO_GET_INFO;
```

```
BEGIN
    CONDICION := 1;
    TEXTCOLOR(10);
    TEXTBACKGROUND(14);
    GOTOXY(17,11);
    WRITE('SELECCIONA PUERTO PARA LEER INFORMACION DEL MODULO');
    GOTOXY(39,13);
    WRITE(' O ');
    GOTOXY(20,15);
    WRITE('ARCHIVO PARA LEER INFORMACION ANTES GRABADA');
    TEXTCOLOR(A);
    TEXTBACKGROUND(B);
    GOTOXY(33,19);
    WRITE('PUERTO');
    TEXTCOLOR(C);
    TEXTBACKGROUND(D);
    GOTOXY(43,19);
    WRITE('ARCHIVO');
    CAMBIA := PUERTOS;
    SELECCION := TRUE;
END;
```

```
PROCEDURE DISPLAY_PUERTOS;
```

```
BEGIN
    TEXTCOLOR(10);
    TEXTBACKGROUND(14);
    GOTOXY(25,17);
    WRITE('SELECCIONA EL PUERTO A UTILIZAR');
    TEXTCOLOR(A);
    TEXTBACKGROUND(B);
    GOTOXY(35,19);
    WRITE('COM1');
    TEXTCOLOR(C);
    TEXTBACKGROUND(D);
    GOTOXY(43,19);
    WRITE('COM2');
END;
```

---

```
PROCEDURE INICIALIZA_PUERTO;
```

```
BEGIN
  DISPLAY OPCIONES;
  CONDICION := 2;
  SELECCION := TRUE;
  VALORES_INICIALES;
  DISPLAY_PUERTOS;
  REGS.AH :=0;      {SELECCIONA EL PUERTO A UTILIZAR PARA}
  REGS.AL :=$F3;   {REALIZAR COMUNICACION SERIE COM1 o COM2}
  REGS.DX :=PUERTO;
  INTR($14,REGS);
END;
```

```
PROCEDURE ACTUALIZA;
```

```
BEGIN
  IF SELECCION <> TRUE THEN
    BEGIN
      IF TEMP2 <> CUENTA THEN
        BEGIN
          CASE CUENTA OF
            1 :
              BEGIN
                VALORES_INICIALES;
                A := 15; B := 10;
              END;
            2 :
              BEGIN
                VALORES_INICIALES;
                C := 15; D := 10;
              END;
            3 :
              BEGIN
                VALORES_INICIALES;
                E := 15; F := 10;
              END;
            4 :
              BEGIN
                VALORES_INICIALES;
                G := 15; H := 10;
              END;
            5 :
              BEGIN
                VALORES_INICIALES;
                I := 15; J := 10;
              END;
```

```
        6 :
          BEGIN
            VALORES_INICIALES;
            K := 15; L := 10;
          END;
        ELSE
          END;
        TEMP2 := CUENTA;
        DISPLAY OPCIONES;
      END;
    END;
  IF SELECCION = TRUE THEN
    BEGIN
      IF PUERTOS <> TEMP3 THEN
        BEGIN
          CASE PUERTOS OF
            1 :
              BEGIN
                VALORES_INICIALES;
                A := 15; B := 10;
              END;
            2 :
              BEGIN
                VALORES_INICIALES;
                C := 15; D := 10;
              END;
            ELSE
              END;
          IF CONDICION = 1 THEN DISPLAY OPCIONES_TO_GET_INFO;
          IF CONDICION = 2 THEN DISPLAY PUERTOS;
          TEMP3 := PUERTOS;
        END;
        PUERTO := PUERTOS - 1;
      END;
    END;
  END;
```

PROCEDURE CURSOR\_DERECHA;

```
  BEGIN
    IF SELECCION <> TRUE THEN      {MUEVE EL CURSOR A LA DERECHA}
      BEGIN
        CUENTA := CUENTA + 1;
        IF CUENTA > 6 THEN CUENTA := 1;
      END;
    IF SELECCION = TRUE THEN
      BEGIN
        PUERTOS := PUERTOS + 1;
        IF PUERTOS > 2 THEN PUERTOS := 1;
        TEMP2 := CUENTA;
        CUENTA := 7;
      END;
    ACTUALIZA;
  END;
```

```
PROCEDURE CURSOR_IZQUIERDA;

BEGIN
  IF SELECCION <> TRUE THEN    {MUEVE EL CURSOR A LA IZQUIERDA}
  BEGIN
    CUENTA := CUENTA - 1;
    IF CUENTA < 1 THEN CUENTA := 6;
  END;
  IF SELECCION = TRUE THEN
  BEGIN
    PUERTOS := PUERTOS - 1;
    IF PUERTOS < 1 THEN PUERTOS := 2;
    TEMP2 := CUENTA;
    CUENTA := 7;
  END;
  ACTUALIZA;
END;

PROCEDURE MANDA_BYTE(DATO : INTEGER);

BEGIN
  ARREGLO[0]:=DATO;
  TRANSMITE(1);
END;

PROCEDURE CAMBIA_WORD(A1,A2:INTEGER; NUMER:WORD);

BEGIN
  IF NUMER<10 THEN
  BEGIN
    ARREGLO[A1]:=NUMER;
    ARREGLO[A2]:=0;
  END;
  IF (NUMER<20)AND(NUMER>10) THEN
  BEGIN
    NUMER:=NUMER-10;
    ARREGLO[A1]:=NUMER;
    ARREGLO[A2]:=1;
  END;
  IF (NUMER<30)AND(NUMER>20) THEN
  BEGIN
    NUMER:=NUMER-20;
    ARREGLO[A1]:=NUMER;
    ARREGLO[A2]:=2;
  END;
  IF (NUMER<40) AND(NUMER>30)THEN
  BEGIN
    NUMER:=NUMER-30;
    ARREGLO[A1]:=NUMER;
    ARREGLO[A2]:=3;
  END;
END;
```

```

IF (NUMER<50)AND(NUMER>40) THEN
  BEGIN
    NUMER:=NUMER-40;
    ARREGLO[A1]:=NUMER;
    ARREGLO[A2]:=4;
  END;
IF (NUMER<60)AND(NUMER>50) THEN
  BEGIN
    NUMER:=NUMER-50;
    ARREGLO[A1]:=NUMER;
    ARREGLO[A2]:=5;
  END;

END;

PROCEDURE CARGA_LOS_DATOS;

VAR
  INDICE,INDICE_1   : INTEGER;
  COSTO_DEL_LITRO   : REAL;
  IN_OUT,ENTRE      : REAL;

BEGIN
  INDICE_1 :=0;
  INDICE :=0;
  COSTO_DEL_LITRO:=1032;
  REPEAT
    DESPLIEGA[INDICE].SEGUNDOS := ARREGLO[INDICE_1]
    + (10 * ARREGLO[INDICE_1 + 1]);
    INDICE_1 := INDICE_1 + 2;
    DESPLIEGA[INDICE].MINUTOS := ARREGLO[INDICE_1]
    + (10 * ARREGLO[INDICE_1 + 1]);
    INDICE_1 := INDICE_1 + 2;
    DESPLIEGA[INDICE].HORAS := ARREGLO[INDICE_1]
    + (10 * ARREGLO[INDICE_1 + 1]);
    INDICE_1 := INDICE_1 + 2;
    DESPLIEGA[INDICE].DIAS := ARREGLO[INDICE_1]
    + (10 * ARREGLO[INDICE_1 + 1]);
    INDICE_1 := INDICE_1 + 2;
    DESPLIEGA[INDICE].MES := MES_ACTUAL( ARREGLO[INDICE_1]
    + (10 * ARREGLO[INDICE_1 + 1]));
    INDICE_1 := INDICE_1 + 2;
    DESPLIEGA[INDICE].YEAR := ARREGLO[INDICE_1]
    + (10 * ARREGLO[INDICE_1 + 1]);
    INDICE_1 := INDICE_1 + 2;
    DESPLIEGA[INDICE].DIA_DE_LA_SEMANA := DIA_ACTUAL(ARREGLO[INDICE_1]);
    INDICE_1 := INDICE_1 + 1;
    DESPLIEGA[INDICE].DATO_DEL_SENSOR := LITROS(ARREGLO[INDICE_1]);
    IF INDICE > 0 THEN
      BEGIN
        DESPLIEGA[INDICE].GASTO_EN_LITROS :=
          ABS(DESPLIEGA[INDICE].DATO_DEL_SENSOR;
          - DESPLIEGA[INDICE-1].DATO_DEL_SENSOR);
      END;
    END REPEAT;

```

```

    DESPLIEGA[INDICE].GASTO_EN_EFECTIVO
:=DESPLIEGA[INDICE].GASTO_EN_LITROS*COSTO_DEL_LITRO;
    ENTRE := DESPLIEGA[INDICE].DATO_DEL_SENSOR
            -DESPLIEGA[INDICE-1].DATO_DEL_SENSOR;
    IN_OUT:= DESPLIEGA[INDICE].GASTO_EN_LITROS/ENTRE;
    IF IN_OUT = 1 THEN DESPLIEGA[INDICE].GASOLINA := 'INC.';
END;
    INDICE_1 := INDICE_1 + 1;
    INDICE := INDICE + 1;
UNTIL RESULTADO = INDICE_1;
END;

```

```
PROCEDURE GRABA_LOS_DATOS;
```

```
VAR
```

```

    INDICE,INDICE_1      : INTEGER;
    COSTO_DEL_LITRO     : REAL;
    IN_OUT,ENTRE        : REAL;
    TEMPO,TEMPO_1       : REAL;

```

```
BEGIN
```

```

    INDICE_1 := 0;
    INDICE := 0;
    COSTO_DEL_LITRO := 1.32;
    ASSIGN(RENOMBRA,'A:VARCHIVO.DAT');
    REWRITE(RENOMBRA);
    REPEAT
        DESPLIEGA[INDICE].SEGUNDOS := ARREGLO[INDICE_1]
            + (10 * ARREGLO[INDICE_1 + 1]);
        INDICE_1 := INDICE_1 + 2;
        DESPLIEGA[INDICE].MINUTOS := ARREGLO[INDICE_1]
            + (10 * ARREGLO[INDICE_1 + 1]);
        INDICE_1 := INDICE_1 + 2;
        DESPLIEGA[INDICE].HORAS := ARREGLO[INDICE_1]
            + (10 * ARREGLO[INDICE_1 + 1]);
        INDICE_1 := INDICE_1 + 2;
        DESPLIEGA[i].DIAS := ARREGLO[INDICE_1]
            + (10 * ARREGLO[INDICE_1 + 1]);
        INDICE_1 := INDICE_1 + 2;
        DESPLIEGA[INDICE].MES := MES_ACTUAL( ARREGLO[INDICE_1]
            + (10 * ARREGLO[INDICE_1 + 1]));
        INDICE_1 := INDICE_1 + 2;
        DESPLIEGA[INDICE].YEAR := ARREGLO[INDICE_1]
            + (10 * ARREGLO[INDICE_1 + 1]);
        INDICE_1 := INDICE_1 + 2;
        DESPLIEGA[INDICE].DIA_DE_LA_SEMANA := DIA_ACTUAL(ARREGLO[INDICE_1]);
        INDICE_1 := INDICE_1 + 1;
        DESPLIEGA[INDICE].DATO_DEL_SENSOR := LITROS (ARREGLO[INDICE_1]);
        IF INDICE > 0 THEN
            BEGIN
                DESPLIEGA[INDICE].GASTO_EN_LITROS :=
                    ABS(DESPLIEGA[INDICE].DATO_DEL_SENSOR - TEMPO);
            END;
        END;
    UNTIL RESULTADO = INDICE_1;
END;

```

```

DESPLIEGA[INDICE].GASTO_EN_EFECTIVO :=
  DESPLIEGA[INDICE].GASTO_EN_LITROS*COSTO_DEL_LITRO;
ENTRE := DESPLIEGA[INDICE].DATO_DEL_SENSOR - TEMPO;
DESPLIEGA[INDICE].GASOLINA := ' ';
IF ENTRE <> 0 THEN
  BEGIN
    IN_OUT := DESPLIEGA[INDICE].GASTO_EN_LITROS/ENTRE;
    IF IN_OUT = 1 THEN
      DESPLIEGA[INDICE].GASOLINA := 'I';
    END;
  END;
INDICE_1:=INDICE_1+1;
TEMPO:=DESPLIEGA[INDICE].DATO_DEL_SENSOR;
WITH CONTROLA DO
  BEGIN
    DIA_DE_LA_SEMANA := DESPLIEGA[INDICE].DIA_DE_LA_SEMANA;
    DIAS := DESPLIEGA[INDICE].DIAS;
    MES := DESPLIEGA[INDICE].MES;
    YEAR := DESPLIEGA[INDICE].YEAR;
    HORAS := DESPLIEGA[INDICE].HORAS;
    MINUTOS := DESPLIEGA[INDICE].MINUTOS;
    SEGUNDOS := DESPLIEGA[INDICE].SEGUNDOS;
    DATO_DEL_SENSOR := DESPLIEGA[INDICE].DATO_DEL_SENSOR;
    GASTO_EN_LITROS := DESPLIEGA[INDICE].GASTO_EN_LITROS;
    GASTO_EN_EFECTIVO := DESPLIEGA[INDICE].GASTO_EN_EFECTIVO;
    GASOLINA:=DESPLIEGA[INDICE].GASOLINA;
  END;
  WRITE(RENOMBRA, CONTROLA);
  INDICE := 1;
UNTIL RESULTADO = INDICE_1;
CLOSE(RENOMBRA);
END;

PROCEDURE LEE_DATOS_GRABADOS;

VAR

  INDICE,INDICE_1 : INTEGER;
  SUMA      : INTEGER;
  SUMA1     : STRING[25];
  CHECA    : CHAR;

BEGIN

  INDICE :=0;
  INDICE_1:=0;
  ASSIGN(RENOMBRA,'A:\ARCHIVO.DAT');
  RESET(RENOMBRA);
  WHILE NOT EOF (RENOMBRA) DO
    BEGIN
      READ(RENOMBRA,CONTROLA);

```

```

WITH CONTROLA DO
BEGIN
  INDICE := INDICE + 1;
  INDICE_1:=INDICE_1+1;
  IF ((DIA_DE_LA_SEMANA + MES) <> SUMA1)
  AND (SUMA <> (DIAS + YEAR)) THEN
  BEGIN
    WRITELN(' ');
    WRITE(' ',DIA_DE_LA_SEMANA);
    WRITE(' ',DIAS:2);
    WRITE(' DE ',MES);
    WRITELN(' de 19',YEAR:2);
  END;
  SUMA := DIAS + YEAR;
  SUMA1 := DIA_DE_LA_SEMANA + MES;
  WRITE(INDICE,' ',HORAS:2);
  WRITE(':',MINUTOS:2);
  WRITE(':',SEGUNDOS:2);
  WRITE(' ',DATO_DEL_SENSOR:2:2);
  WRITE(' ',GASTO_EN_LITROS:2:2);
  WRITE(' ',GASTO_EN_EFECTIVO :5:2);
  WRITELN(' ',GASOLINA);
  IF INDICE_1 > 10 THEN
  REPEAT
    CHECA := READKEY;
    INDICE_1 := 0;
    IF CHECA = CHR(13) THEN CLRSCR;
  UNTIL CHECA = CHR(13);
  END;
END;
CLOSE(RENOMBRA)
END;

PROCEDURE MODIFICA_TIME;

VAR
  HORAS,MINUTOS,SEGUNDOS,MICRAS   : WORD;
  YEAR,MES,DIA                    : WORD;
  SEGURIDAD                       : CHAR;

BEGIN
  DISPLAY OPCIONES;
  MICRAS := 0;
  TEXTCOLOR(10);
  TEXTBACKGROUND(14);
  GOTOXY(25,7);
  WRITE(' LA NUEVA HORA ES: ');
  GOTOXY(30,10);
  WRITE(' HORAS = ');
  READ(HORAS);
  GOTOXY(30,12);
  WRITE(' MINUTOS = ');
  READ(MINUTOS);
  GOTOXY(30,14);

```

```
WRITE(' SEGUNDOS= ');
READ(SEGUNDOS);

DISPLAY OPCIONES;
TEXTCOLOR(10);
TEXTBACKGROUND(14);
GOTOXY(25,7);
WRITE(' LA NUEVA FECHA ES: ');
GOTOXY(30,10);
WRITE(' AÑO = ');
READ(YEAR);
GOTOXY(30,12);
WRITE(' MES = ');
READ(MES);
GOTOXY(30,14);
WRITE(' DIA = ');
READ(DIA);

MANDA_BYTE(2);

CAMBIA_WORD(0,1, SEGUNDOS);
CAMBIA_WORD(2,3, MINUTOS);
CAMBIA_WORD(4,5, HORAS);
CAMBIA_WORD(6,7,DIA);
CAMBIA_WORD(8,9,MES);
CAMBIA_WORD(10,11, YEAR);
TRANSMITE(13);
END;

PROCEDURE GET_ARCHIVO;

VAR
  ARCHIVO      :: STRING[8];
  FILES        :: FILE;

BEGIN
  TEXTCOLOR(10);
  TEXTBACKGROUND(14);
  GOTOXY(15,19);
  WRITE('NOMBRE DEL ARCHIVO A LEER ');
  READLN(ARCHIVO);
  ASSIGN(FILES,ARCHIVO);
  REWRITE(FILES);
  {SUBROUTINA PARA OBTENER INFORMACION}
  CLOSE(FILES);
END;
```

```
PROCEDURE GET_DATOS_PUERTO;
```

```
VAR  
  INDICE,INDICE_1,DIRECCION_MAYOR : INTEGER;
```

```
BEGIN  
  DIRECCION_MAYOR := 0;  
  MANDA_BYTE(4);  
  RECIBE(2);  
  
  INDICE := ARREGLO[0];  
  INDICE_1 := (ARREGLO[1] - 32) * 256;  
  DIRECCION_MAYOR := INDICE + INDICE_1;  
  RECIBE(DIRECCION_MAYOR);  
END;
```

```
PROCEDURE GET_INFO;
```

```
BEGIN  
  SELECCION := TRUE;  
  CONDICION := 1;  
  VALORES_INICIALES;  
  DISPLAY OPCIONES_TO_GET_INFO;  
END;
```

```
PROCEDURE TIME;
```

```
VAR  
  SEGURIDAD : CHAR;
```

```
BEGIN  
  DISPLAY OPCIONES;  
  MANDA_BYTE(1);  
  RECIBE(13);  
  
  TEXTCOLOR(10);  
  TEXTBACKGROUND(14);  
  GOTOXY(16,7);  
  WRITE('LA HORA ACTUAL ES (hh:mm:ss) ',ARREGLO[5],ARREGLO[4]);  
  WRITE(':',ARREGLO[3],ARREGLO[2],':',ARREGLO[1],ARREGLO[0]);  
  GOTOXY(16,10);  
  WRITE('LA FECHA ACTUAL ES (dia,mes,año) ',ARREGLO[7],ARREGLO[6] );  
  WRITE(' ',ARREGLO[9],ARREGLO[8],', ',ARREGLO[11],ARREGLO[10]);  
  
  GOTOXY(16,15);  
  WRITE('DESEAS MODIFICAR LA HORA Y FECHA (S/N) ');  
  READ(SEGURIDAD);  
  IF (SEGURIDAD = 's') OR (SEGURIDAD = 'S') THEN MODIFICA_TIME;  
  DISPLAY OPCIONES;  
END;
```

---

```
PROCEDURE SAVE_AS;
```

```
VAR
  ARCHIVO      : STRING[8];
  FILES        : FILE;

BEGIN
  TEXTCOLOR(10);
  TEXTBACKGROUND(14);
  GOTOXY(15,19);
  WRITE('NOMBRE DEL ARCHIVO A GUARDAR ');
  READLN(ARCHIVO);
  ASSIGN(FILES,ARCHIVO);
  REWRITE(FILES);
  {SUBROUTINA PARA VACIAR INFORMACION}
  CLOSE(FILES);
END;
```

```
PROCEDURE DISPLAY_INFO;
```

```
VAR

  INDICE,INDICE_1 : INTEGER;

BEGIN
  INDICE_1 := 0;
  INDICE := 0;
  REPEAT
    WRITE(DESPLIEGA[INDICE].DIA_DE_LA_SEMANA);
    WRITE(' ',DESPLIEGA[INDICE].DIAS : 2);
    WRITE(' de ',DESPLIEGA[INDICE].MES);
    WRITE(' de 19',DESPLIEGA[INDICE].YEAR : 2);
    WRITE(' ',DESPLIEGA[INDICE].HORAS : 2);
    WRITE(':',DESPLIEGA[INDICE].MINUTOS : 2);
    WRITE(':',DESPLIEGA[INDICE].SEGUNDOS : 2);
    WRITE(' ',DESPLIEGA[INDICE].DATO_DEL_SENSOR : 2 : 2);
    WRITE(' ',DESPLIEGA[INDICE].GASTO_EN_LITROS : 2 : 2);
    WRITE(' ',DESPLIEGA[INDICE].GASTO_EN_EFECTIVO : 5 : 2);
    WRITELN(' ',DESPLIEGA[INDICE].GASOLINA);
    INDICE_1 := INDICE_1 + 14;
    INDICE := INDICE + 1;
  UNTIL RESULTADO = INDICE_1;
END;
```

---

```
PROCEDURE CHECA_ESTADO_DE_LA_IMPRESORA;
```

```
VAR
```

```
  REGS      : REGISTERS;  
  CONDICION : BOOLEAN;
```

```
BEGIN
```

```
  REGS.AH := 1;    {INICIALIZANDO PUERTO PARA IMPRESORA}  
  REGS.DX := 0;  
  INTR($17,REGS);
```

```
  CONDICION := FALSE;
```

```
  REPEAT
```

```
    REGS.AH := 2;  
    REGS.DX := 0;  
    INTR($17,REGS);
```

```
  CLRSCR;
```

```
  TEXTCOLOR(10);
```

```
  TEXTBACKGROUND(14);
```

```
  GOTOXY(25,19);
```

```
  IF REGS.AH = 8 THEN WRITELN('FUERA DE LINEA');
```

```
  IF REGS.AH = 176 THEN WRITELN('NECESITA PAPEL');
```

```
  IF REGS.AH = 200 THEN WRITELN('IMPRESORA APAGADA');
```

```
  IF REGS.AH = 144 THEN CONDICION := TRUE;
```

```
  UNTIL CONDICION = TRUE;
```

```
  DISPLAY OPCIONES;
```

```
END;
```

```
PROCEDURE PRINT;
```

```
VAR
```

```
  INDICE : INTEGER;  
  SUMA1  : STRING[25];  
  SUMA   : INTEGER;
```

```
BEGIN
```

```
  INDICE := 0;
```

```
  ASSIGN(RENOMBRA,'A:ARCHIVO.DAT');
```

```
  RESET(RENOMBRA);
```

```
  WHILE NOT EOF (RENOMBRA) DO
```

```
    BEGIN
```

```
      READ(RENOMBRA, CONTROLA);
```

```
      WITH CONTROLA DO
```

```
        BEGIN
```

```
          INDICE := INDICE + 1;
```

```
          CHECA_ESTADO_DE_LA_IMPRESORA;
```

```
          IF ((DIA_DE_LA_SEMANA + MES) <> SUMA1)
```

```
            AND(SUMA <> (DIAS + YEAR)) THEN
```

```
            BEGIN
```

```
              WRITELN(LST,' ');
```

```
              WRITE(LST,',',DIA_DE_LA_SEMANA);
```

```
        WRITE(LST,' ',DIAS:2);
        WRITE(LST,' de ',MES);
        WRITELN(LST,' de 19',YEAR:2);
    END;
    SUMA := DIAS + YEAR;
    SUMA1 := DIA_DE_LA_SEMANA + MES;
    CHECA_ESTADO_DE_LA_IMPRESORA;
    WRITE(LST,' ',DIA_DE_LA_SEMANA);
    WRITE(LST,' ',DIAS:2);
    WRITE(LST,' de ',MES);
    WRITE(LST,' de 19',YEAR:2);
    WRITE(LST,' ',INDICE,'-',HORAS:2);
    WRITE(LST,':',MINUTOS:2);
    WRITE(LST,':',SEGUNDOS:2);
    WRITE(LST,' ',DATO_DEL_SENSOR:2:2);
    WRITE(LST,' ',GASTO_EN_LITROS:2:2);
    WRITE(LST,' ',GASTO_EN_EFECTIVO :5:2);
    WRITELN(LST,' ',GASOLINA);
END;
END;
CLOSE(RENOMBRA)
END;

PROCEDURE QUIT;

VAR
    SEGURIDAD : CHAR;

BEGIN
    TEXTCOLOR(10);
    TEXTBACKGROUND(14);
    GOTOXY(15,19);
    WRITE('ESTAS SEGURO DE QUERER ABANDONAR EL PROGRAMA (S/N)');
    SEGURIDAD := READKEY;
    IF(SEGURIDAD = 's') OR (SEGURIDAD = 'S') THEN HALT;
    DISPLAY OPCIONES;
END;

PROCEDURE ENTER;

BEGIN
    CASE CUENTA OF

        1 : GET_INFO;
        2 : TIME;
        3 : SAVE_AS;
        4 : DISPLAY_INFO;
        5 : PRINT;
        6 : QUIT;
```

```
7:
  BEGIN
    CUENTA := TEMP2;
    VALORES_INICIALES;
    DISPLAY_OPCIONES;
    IF CAMBIA = 1 THEN INICIALIZA_PUERTO;
    IF CAMBIA = 2 THEN GET_ARCHIVO;
    CAMBIA := 0;
    SELECCION := FALSE;
  END;
ELSE
  END;
END;
```

**PROCEDURE ESCAPE;**

```
BEGIN
  VALORES_INICIALES;
  DISPLAY_OPCIONES;
  SELECCION := FALSE;
  CUENTA := 0;
END;
```

**PROCEDURE MOVIMIENTOS;**

```
VAR
  KEYCODE, SCANCODE      : BYTE;
  ALPHA, EXTENDED, CONTROL : BOOLEAN;
  REGS                    : REGISTERS;
```

```
LABEL
  NEXTINPUT, EXTENDED_CODE, CONTROL_CODE;
```

```
BEGIN
  ALPHA := TRUE;
  EXTENDED := TRUE;
  CONTROL := TRUE;
```

```
NEXTINPUT :
  KEYCODE := $FF;
  IF KEYPRESSED = TRUE THEN
    BEGIN
      REGS.AH := 0;
      INTR($16, REGS);
      KEYCODE := REGS.AL;
      SCANCODE := REGS.AH;
    END
  ELSE
    BEGIN
      GET_RATA_POSICION;
      ACTUALIZA;
    END;
```

```
ACTUALIZA;
IF KEYCODE = $FF THEN GOTO NEXTINPUT;
IF (KEYCODE = 0) AND EXTENDED THEN GOTO EXTENDEDCODE;
IF (KEYCODE < 32) AND CONTROL THEN GOTO CONTROLCODE;
IF (KEYCODE > 122) THEN GOTO NEXTINPUT;
IF NOT ALPHA THEN GOTO NEXTINPUT;
{ALPHABETIC DATA}
GOTO NEXTINPUT;
```

EXTENDED CODE :

```
IF (SCANCODE = 15) OR (SCANCODE = 75) THEN
  BEGIN
    CURSOR_IZQUIERDA;
    GOTO NEXTINPUT;
  END;
IF SCANCODE = 77 THEN
  BEGIN
    CURSOR_DERECHA;
    GOTO NEXTINPUT;
  END;
```

CONTROL CODE :

```
IF KEYCODE = 9 THEN IF SCANCODE = 15 THEN
  BEGIN
    CURSOR_DERECHA;
    GOTO NEXTINPUT;
  END;
IF KEYCODE = 13 THEN IF SCANCODE = 28 THEN
  BEGIN
    ENTER;
    GOTO NEXTINPUT;
  END;
IF KEYCODE = 27 THEN IF SCANCODE = 1 THEN
  BEGIN
    ESCAPE;
    GOTO NEXTINPUT;
  END;
IF KEYCODE = 26 THEN EXIT;
```

END;

*{ PROGRAMA PRINCIPAL }*

```
BEGIN
  INICIALIZA;
  MOVIMIENTOS;
END.
```