

43
2ej.



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

**DISEÑO E INTEGRACION DE UN SISTEMA DE
ADQUISICION DE DATOS PARA EL VEHICULO
ELECTRICO DEL INSTITUTO DE
INGENIERIA DE LA UNAM**

**TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
(AREA ELECTRICA Y ELECTRONICA)**

**P R E S E N T A N
HERNANDEZ HERNANDEZ HANIEL
VALDES CASILLAS JUAN MANUEL**

DIRECTOR DE TESIS: M. en I. LAURO SANTIAGO CRUZ



CIUDAD UNIVERSITARIA

MEXICO, D.F., 1999.

**TESIS CON
FALLA DE ORIGEN**

273798



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

PASIMACION

DISCONTINUA.

Agradecemos:

A la Universidad Nacional Autónoma de México
por habernos abierto sus puertas y darnos la
oportunidad de superarnos.

A la Facultad de Ingeniería, a sus maestros y nuestros
compañeros por la formación que en ella recibimos,
ya que esa es la principal arma para el desarrollo.

Al Instituto de Ingeniería por la experiencia y los
conocimientos ahí adquiridos.

Al M.I. Lauro Santiago Cruz por la confianza que
en nosotros depositó al apoyar el proyecto y por
su asesoría en el desarrollo del mismo.

Atentamente
Haniel Hernández Hernández
Juan Manuel Valdés Casillas

Esta tesis está dedicada a:

A mis padres Lucy y Oscar por darme siempre su ejemplo, apoyo, confianza y sobre todo la mejor herencia: el conocimiento de Dios.

A mis hermanos Alba y Oscar por que muchos de los mejores momentos de mi vida los he pasado a su lado.

A mis abuelos y tíos por sus consejos y alientos.

A mi tío Roberto y su familia por el apoyo que de ellos he recibido.

A mis amigos por estar siempre a mi lado, de los que no tengo que dar nombre ni apellido, porque ellos mismos se saben aludidos.

A Manuel por tenerme la confianza de compartir con él este trabajo y por lo aprendido de él.

A Raymundo, Guillermo, Luis y demás compañeros de carrera por el apoyo y por su amistad que es invaluable.

A Lauro por su amistad y apoyo para la realización de este trabajo.

Y por sobre todas las cosas a ti Señor Jesús, por lo que haces, has hecho y harás en mi vida.

Haniel H.H.

Dedicado:

A mi mamá

A mi papá

A mamá Margarita

A mis hermanos Oscar y Diego

A la gente que ha confiado en mí y que me ha apoyado

Agradezco a Dios por permitirme haber llegado hasta aquí.

Juan Manuel

INTRODUCCION	2
CAPITULO I ANTECEDENTES	
I.1 Desarrollo de los vehículos eléctricos	5
I.1.1 El vehículo eléctrico(VE)	6
I.1.2. Ventajas y desventajas del VE	6
I.1.3. Factibilidad de los vehículos eléctricos	7
I.1.4. Elementos básicos	9
I.1.5 Historia de los vehículos eléctricos	19
I.1.6. Los vehículos eléctricos en México	25
I.2 El minibús eléctrico del Instituto de Ingeniería	26
CAPITULO II CONCEPTOS BASICOS	
II.1 Sistemas de adquisición de datos	30
II.1.1 Convertidores analógico-digital	31
II.1.2 Características del convertidor analógico-digital	32
II.2 Microcontroladores y Memorias	36
II.2.1 Microcontroladores	36
II.2.2 Memorias	46
II.3 Comunicación RS-232	52
CAPITULO III DESCRIPCION GENERAL DEL SISTEMA	
III.1 Planteamiento del problema y propuesta de solución	57
III.2 Tarjeta de control miniatura programable (<i>Little-G</i>)	61
III.2.1 Descripción de sus <i>interfaces</i>	64
III.2.2 Comunicación serial	65
III.2.3 El <i>PLCBus</i>	67
III.2.4 Como utilizar la tarjeta	70
III.3 Tarjeta de conversión analógico-digital (TCAD)	73

III.3.1 Como usar la tarjeta	77
III.3.2 Modos de conversión de la TCAD	78
III.3.3 <i>Software</i> para la TCAD	79
III.4 Módulo de acondicionamiento	82
III.5 Microcontrolador <i>COP8SAC</i>	84
III.5.1 Características principales	84
III.5.2 Arquitectura <i>COP8SAC</i>	86
III.6 <i>Thin Card Random Acces Memory (TCRAM)</i>	92

CAPITULO IV DISEÑO DEL PROTOTIPO DEL SISTEMA

IV.1 Especificaciones para el diseño del sistema	98
IV.2 Conexión de las baterías del minibús eléctrico	99
IV.3 Tarjeta del módulo de acondicionamiento (TMA)	103
IV.4 Tarjeta del módulo de conversión analógico-digital (TMCAD)	111
IV.5 Tarjeta del módulo de control (<i>Little-G</i>)	113
IV.6 Diseño de la tarjeta del módulo externo de almacenamiento (TME)	115
IV.7 Conexiones	122
IV.8 <i>Software</i> del sistema	125
IV.8.1 Programa del módulo de control (PRODEC)	126
IV.8.2 Programa del módulo de almacenamiento externo (PROMOX)	136
IV.8.3 Programa de usuario Sistema26	139
IV.8.4 Programa de conversión de archivo (PROCAR)	142
IV.9 Instalación del SADME	144

CAPITULO V EVALUACION DEL SISTEMA

V.1 Evaluaciones en el laboratorio	147
V.2 Pruebas del sistema en el banco de baterías del minibús	159
V.3 Pruebas realizadas al <i>software</i> de usuario Sistema26	174

CAPITULO VI RESULTADOS Y CONCLUSIONES

VI.1 Resultados	176
VI.2 Conclusiones	178

BIBLIOGRAFIA	182
---------------------	-----

APENDICE A HOJAS DE ESPECIFICACION

APENDICE B LISTADO DE LOS PROGRAMAS DESARROLLADOS

APENDICE C DIAGRAMAS ELECTRONICOS

APENDICE D MANUAL DE OPERACION

APENDICE E TABLAS DE CALIBRACION

Introducción

Uno de los principales avances de la tecnología ha sido en el campo de la industria automotriz. Pues de transportes tirados por animales, se pasó al desarrollo de otros medios que fueran autónomos y que permitiesen el uso de algún tipo de combustible como lo fue el vapor, y posteriormente la gasolina, la electricidad, etcétera.

En un principio, la preocupación del desarrollo sólo se orientaba al apoyo de las tecnologías que presentaban mayores avances sin importar el deterioro del medio ambiente, por esta causa la electricidad como fuente de energía en vehículos autónomos no fue totalmente apoyada en sus inicios, aunque fue pionera en los intentos del desarrollo de los automóviles.

No obstante, cada vez es más la preocupación por el cuidado del medio ambiente, lo que da lugar a que se vuelve la vista, de nueva cuenta, a los vehículos eléctricos. Esto lleva a la creación de diferentes tipos de transportes eléctricos como son: trenes, automóviles, camiones, etcétera.

Por ello cada vez existen en todo el mundo investigaciones y trabajos relacionados con este tema. La Universidad Nacional Autónoma de México por medio de sus instituciones y en especial el Instituto de Ingeniería ha desarrollado un vehículo eléctrico (minibús) el cual pueda acoplarse a las condiciones de la Ciudad de México y al mismo tiempo contribuir al mejoramiento del medio ambiente.

Como parte del seguimiento de este vehículo se ha implementado un sistema de adquisición de datos con la finalidad de observar el comportamiento de cada uno de los parámetros más importantes que intervienen en el funcionamiento del mismo y que es el tema desarrollado en esta tesis.

En este trabajo presentamos una breve reseña del avance de los vehículos eléctricos a través del tiempo, esto es en el primer capítulo.

Los conceptos básicos relacionados con el desarrollo del sistema se presentan en el segundo capítulo.

El planteamiento del problema, que en este caso es la medición de diferentes variables que se involucran directamente con el desempeño del minibús, y su solución es presentada en el tercer capítulo. Así como los componentes seleccionados y proporcionados para la elaboración del proyecto.

En el capítulo número cuatro se presenta la integración del sistema, tanto a nivel de *hardware*, como de *software*.

Las pruebas realizadas en el laboratorio, así como las hechas con el sistema ya instalado en el minibús se presentan en el capítulo número cinco.

El sexto capítulo muestra los resultados y conclusiones. Además en el apéndice D se ha desarrollado un manual de usuario para el *software de interface* entre el usuario y el sistema.

El sistema en el momento actual se encuentra físicamente instalado en el minibús, y funcionando, pero como todo sistema es susceptible a mejoras y adecuaciones.

==== CAPITULO I ====

Antecedentes

En este capítulo presentamos un panorama general de los vehículos eléctricos, cuales son sus características principales, ventajas y desventajas en los tiempos actuales. Una breve reseña de su evolución en el mundo, para concluir con una descripción general del Minibús Eléctrico desarrollado en el Instituto de Ingeniería de la UNAM.

I.1 Desarrollo de los vehículos eléctricos

Se ha comprobado¹ que los automóviles, vehículos de combustión interna (VCI), son los que emiten la mitad de la contaminación urbana, y una cuarta parte de los gases que provocan el efecto invernadero, en ciertas épocas del año. Por esto se tienen que tomar medidas que tiendan a frenar este problema.

Las alternativas que se plantean son:

- Limitar el uso del automóvil.
- Mejorar la eficiencia de los motores y utilizar combustibles menos dañinos, que reduzcan la emisión de contaminantes.
- Buscar sistemas alternativos que contaminen menos.

En el caso particular de la Zona Metropolitana del Valle de México (ZMVM), las primeras dos opciones ya se han establecido, pero los resultados han sido poco satisfactorios y el malestar de la gente es evidente. El uso de sistemas alternativos que contaminen menos es por mucho la opción más recomendable, y en particular el uso de la energía eléctrica.

Una de las opciones más viables son los vehículos eléctricos; éste es un ejemplo de cómo ahorrar energía primaria (hidrocarburos), sustituyéndola por una forma de energía más eficiente (energía eléctrica), por lo cual algunos países han establecido políticas relacionadas con la disminución de la dependencia del petróleo, y la protección del medio ambiente, de lo cual se ha derivado en un interés común en los vehículos eléctricos (VE).

¹Sperling, Daniel. La cuestión de los coches eléctricos. Investigación y Ciencia. (Barcelona), enero, 1997. p. 62-67

Un vehículo eléctrico es un automóvil que funciona con energía eléctrica en lugar de algún derivado del petróleo.

I.1.1 El vehículo eléctrico (VE)

Actualmente este tipo de vehículo cuenta con una autonomía que varía entre los 115 km y 134 km y pueden alcanzar velocidades de hasta 100 km/h. Se alimentan de baterías que se pueden recargar en la red eléctrica. Son clasificados como vehículos con cero emisiones de contaminantes.

Ya en muchos países se considera que el automóvil urbano del futuro tendrá un motor eléctrico o híbrido, ya que un motor eléctrico alcanza un rendimiento energético del 90%, en comparación con el 20% que alcanza un motor a gasolina, asimismo el escaso roce de las piezas permite una vida más larga.

Un factor más en apoyo del VE, es que en las zona urbanas los individuos efectúan trayectos diarios menores a los 50 km, lo que nos hace pensar que el VE está equipado perfectamente para afrontar esta situación. Además posee las cualidades de realizar un funcionamiento silencioso, extremadamente limpio, también cuenta con las siguientes características:

- No usa derivados del petróleo.
- No tiene embrague.
- No tiene radiador, bujías, ni pistones.
- No emite humo.
- Funciona con un motor de Corriente Alterna o de Corriente Directa.
- En 18 segundos alcanza una velocidad de 80 km/h.

I.1.2 Ventajas y desventajas del VE

Los factores que favorecen el uso del VE son su bajo nivel de ruido si se les compara con los VCI, así como su confiabilidad y menor mantenimiento. Los VE's resultan muy

sencillos en concepto, pues básicamente constan de un banco de baterías que alimenta a un motor eléctrico a través de un control de estado sólido.

Pero en realidad, el VE presenta actualmente las siguientes limitaciones: la autonomía es del orden del 30 al 50% en comparación con el VCI; lleva mucho más tiempo la recarga de baterías que el llenado de un tanque de gasolina; su costo está en función de construir un vehículo de alta tecnología, suma eficiencia, sin reducción en sus características de resistencia, seguro contra impactos en accidentes y si se agrega el uso de electrónica especializada además de los bajos niveles de producción iniciales, el VE indudablemente resultará más costoso sobre todo en las primeras etapas de producción comercial.

A continuación se presenta un cuadro comparativo que nos ayuda a visualizar estos puntos:

Ventajas	Desventajas
No contaminan No hacen ruido Fácil manejo No necesitan cambio de aceite No requiere afinamiento El costo de mantenimiento es bajo	Autonomía de 130 km máximo Su recarga tarda 8 horas Costo de inversión inicial elevado

Tabla I.1 Ventajas y Desventajas.

1.1.3 Factibilidad de los vehículos eléctricos

Los VE's ya existen en diversas partes del mundo, muchos de ellos en demostración y ya algunos en comercialización. Estos VE's ya están diseñados para operar según las condiciones muy particulares de cada ciudad, como son las distancias, elevación del terreno y tráfico intenso. Estos vehículos realizan trayectos de prueba bajo condiciones reales de tráfico urbano en las ciudades con problemas de circulación. El objetivo de los recorridos es efectuar

un monitoreo continuo de un VE o de flotillas. El monitoreo se realiza sobre aquellas variables de suma importancia como son: estado de las baterías, velocidad promedio, distancia recorrida, gasto energético, desempeño en pendientes, etc. Estos datos son recopilados para su posterior análisis por medio de modelos matemáticos. Los resultados obtenidos en el estudio de los VE's, sirven para verificar el comportamiento general y realizar las adaptaciones pertinentes a las condiciones de cada ciudad, tratando de llegar a la mejor operación posible y con el mínimo de problemas en los VE's.

La gran mayoría de la población no está enterada de los beneficios que se pueden obtener del uso de los VE's, tanto en el ámbito social como en el económico. Por esto es importante dar a conocer los resultados de todo el trabajo que se viene desarrollando, así como los panoramas en el ahorro energético y económico, disminución de contaminantes, la posibilidad de su industrialización y la creación de empleos. El cambio propuesto al introducir los VE's proviene del estudio e investigación de muchos años por parte de científicos y de gente preocupada por las condiciones que el planeta vive.

Para que los vehículos eléctricos puedan rivalizar con los de gasolina deberán ofrecer las mismas características de autonomía, potencia, velocidad, aceleración, tiempo de recarga y costos, entre otros aspectos, que les permitan ser competitivos.

Sin duda, el principal parámetro técnico del VE es la eficiencia, y como las mejoras que se advierten en el horizonte previsible en el campo de las baterías son limitadas, los esfuerzos se están encaminando para aumentar la eficiencia hacia otros aspectos de la tecnología de VE, orientados principalmente hacia los siguientes aspectos:

- Reducción del peso muerto del vehículo eléctrico

Con respecto al peso de las baterías poco se puede hacer, que forman en gran medida este peso muerto, por lo que la reducción del mismo va encaminado hacia el uso de materiales más ligeros en su carrocería, que de alguna forma ayuda a reducir el consumo de energía. Entre los materiales que se pueden utilizar cabe señalar el aluminio y a los plásticos reforzados con diversos tipos de fibras.

- Reducción de la resistencia al aire

Destaca la reducción de la resistencia al aire, coeficiente aerodinámico (ca) adimensional, que se incrementa al cuadrado de la velocidad del vehículo. Para reducir este coeficiente, los diseñadores están limitando: electrónicamente la velocidad máxima, el área de sección transversal y la optimización del diseño de la forma de la carrocería del VE. Los automóviles modernos presentan valores de ca de 0.3 a 0.4; en comparación, los automóviles diseñados para ser VE alcanzan valores de ca de 0.2 y 0.19 [IEEE, Spectrum, Noviembre 1992].

- Reducción de la resistencia al rodamiento

Otro aspecto importante es la resistencia al rodamiento, la reducción de este coeficiente mejora la utilización de la energía disponible en los VE's y se obtiene incrementando la presión de inflado de las llantas.

- Mejoramiento de la eficiencia del motor eléctrico actual y su control

Una de las grandes ventajas de los vehículos eléctricos es el uso de motores eléctricos, ya que éstos desperdician mucho menos energía que los motores de combustión interna; los motores eléctricos tienen eficiencias que exceden el 90%, en comparación con los motores de combustión interna que tienen eficiencias máximas del orden de 20%.

1.1.4. Elementos básicos

Los elementos básicos que pueden mencionarse en cuanto al aspecto eléctrico en el VE son los siguientes:

- Sistema eléctrico del vehículo

- Motor

El motor es el encargado de transformar la energía eléctrica que lo alimenta en energía mecánica; su unión a las ruedas puede ser directa o a través de un sistema mecánico.

- Banco de baterías

Es una de las partes fundamentales del VE debido a que de aquí se obtiene la energía necesaria para el funcionamiento del motor eléctrico.

- Sistemas de recarga de las baterías

Los sistemas de recarga tienen como objetivo el suministrar la energía necesaria a las baterías cuando éstas se encuentran por debajo de un nivel de voltaje determinado. Por lo general estos sistemas obtienen la energía de recarga a partir de la red eléctrica, aunque puede haber otras fuentes como los paneles solares.

- Sistemas de frenado

Aunque los sistemas de frenado se encuentran más contenidos en sistemas hidráulicos, es importante señalarlos ya que se puede realizar la transformación de la energía cinética que posee el VE en movimiento en energía eléctrica; la principal técnica conocida es el frenado regenerativo, a través del cual, al dejar de alimentar de energía eléctrica al motor, éste se convierte en generador, con lo que el VE se frena y al mismo tiempo genera energía para cargar las baterías. Con una buena práctica operativa, utilizando al máximo el frenado regenerativo, no sólo se recupera el máximo de energía sino que también se alarga la vida del sistema de frenado mecánico.

De lo anteriormente escrito abundaremos específicamente sobre tres aspectos generales, que son los motores utilizados en diversos vehículos eléctricos, baterías y sistemas de recarga.

- ♦ **Motores utilizados en los vehículos eléctricos**

Una de las partes fundamentales del VE son los motores, existen diferentes tipos de motores que podrían adaptarse a las necesidades de cada tipo de VE. Para esto daremos una breve explicación de su estructura, sus ventajas y desventajas para esta aplicación en específico.

Los motores constan de dos partes principales, una fija llamada estator, y otra que gira con respecto al anterior conocido como rotor. Son construidas estas partes con material

ferromagnético, poseen una serie de ranuras en donde se albergan los hilos conductores de cobre que forman los devanados. Estos devanados los podemos clasificar como:

- Devanado inductor

Es el que origina el campo magnético básico para inducir los voltajes en el otro devanado. Al devanado inductor también se le conoce como devanado de excitación o de campo.

- Dévanado inducido

En él aparecen las corrientes eléctricas inducidas que producen el par de funcionamiento deseado del motor.

Estos motores o máquinas eléctricas se pueden clasificar en dos grupos: máquinas de corriente directa y máquinas de corriente alterna, estas últimas se subdividen en máquinas síncronas, en las que el campo magnético inductor se obtiene al hacer pasar corriente directa por su devanado; y máquinas asíncronas, en las que se obtienen al hacer pasar una corriente alterna por el devanado.

En la práctica, en los VE's se han utilizado motores de corriente directa, debido a que funcionan con la corriente de la batería sin tener elementos de control complejos. Sin embargo se han conseguido motores de corriente alterna más eficientes y prácticos para su empleo en el VE, estos motores pueden suministrar alta potencia durante un rango de velocidad mayor que los de corriente directa. Además se les puede emplear, sacándoles así más provecho, en sistemas de frenado regenerativo. Aunque una desventaja es la de convertir corriente directa a alterna por medio de la electrónica de potencia.

En los motores de corriente directa se hace circular corriente directa por el estator y también por el rotor a través de las escobillas. Los motores de corriente directa con excitación serie fueron de los primeros empleados debido a su característica par-velocidad, de forma hiperbólica, los cuales entregan un par de arranque elevado que se va debilitando a medida que aumenta la velocidad. En general son buenos motores para cuando se necesitan pares altos, y

su sistema de control son relativamente simple. Por el contrario las desventajas que presentan al usar escobillas y colectores es su mantenimiento, el límite de velocidad de rotación máxima está restringido por el rozamiento entre los mismos, además de presentar pérdidas elevadas en el rotor.

En los motores síncronos el estator se alimenta por una fuente trifásica, y el rotor por corriente directa, cuando el campo magnético coincide en velocidad con el del rotor, aparece un par motor sobre el rotor, obteniéndose así la energía mecánica necesaria en el eje.

El motor síncrono de rotor embobinado presenta características electromecánicas idénticas a las de un motor de corriente directa y la misma forma de desempeño, par constante hasta una velocidad base. Sin embargo existe otra opción de este tipo de motores síncronos, los de imanes permanentes. En ellos los imanes están contenidos en el rotor, mientras que el estator está formado por un devanado trifásico conectado en estrella.

La ventaja de este tipo de motores es la ausencia de contactos móviles, y del embobinado de excitación, haciéndolos más compactos que los de rotor embobinado y ofrecen un rendimiento mayor. Pueden girar a mayor velocidad que los de corriente directa, como el embobinado inducido se encuentra en el estator se facilita la disipación de calor originado por las pérdidas (efecto Joule), por lo que en comparación con los motores de corriente directa, para la misma potencia, su tamaño es un 25 % menor.

Algunos de los inconvenientes que presentan están relacionado con los imanes que, además de caros, pierden fuerza magnética a medida que aumenta la temperatura, por lo que se necesita un sistema de refrigeración y los controladores para este tipo de motor tampoco son baratos.

Para el caso de los motores asíncronos, el voltaje en el rotor es inducido por efecto transformador, produciéndose la corriente y el campo magnético del rotor. El estator es alimentado por un voltaje alterno, se necesita un inversor trifásico *PWM* (Modulador por

Ancho de Pulso), de esta forma por sus devanados circulan las corrientes que producen un campo magnético que gira con una velocidad de rotación llamada velocidad de sincronismo.

El movimiento relativo del rotor con respecto al campo magnético del estator es el que produce el voltaje inducido en las barras del rotor. Un motor asíncrono puede girar a velocidades cercanas a la de sincronismo, pero nunca la podrá alcanzar exactamente. Este tipo de motores es relativamente barato, de construcción robusta y apenas necesitan mantenimiento, la ausencia de rozamiento de las escobillas permite alcanzar velocidades máximas de rotación, permitiendo obtener un par mayor. Por lo que se podría conseguir un motor más compacto que uno de corriente directa.

Las desventajas están encaminadas al control de velocidad que resulta bastante complejo, las variaciones de velocidad se consiguen cambiando la frecuencia del voltaje de alimentación por medio un control electrónico.

◆ **Baterías**

Los principales parámetros técnicos de las baterías, desde el punto de vista de su aplicación en el VE, son el voltaje, la energía específica, la potencia específica, el ciclo de vida y, por supuesto, el costo. El significado de dichos parámetros es el siguiente:

• **Voltaje**

Cada batería tiene un voltaje a circuito abierto, que es mayor que el voltaje nominal. Cuando a la batería se le demanda una corriente este voltaje tiende al valor nominal. Los valores máximos y mínimos de operación de la batería son especificados por el fabricante.

• **Energía Específica**

Es la cantidad de Watts-hora de electricidad que la batería puede almacenar por kilogramo de masa para un índice específico de descarga.

- Potencia Específica

Es la cantidad de Watts por kilogramo que la batería puede entregar a un estado de carga definido, generalmente de 80%.

- El Ciclo de Vida

Es el número de veces que la batería puede ser recargada, antes de descender por debajo de un valor predeterminado.

- Costo

El costo elevado de las baterías es propiciado por cada elemento que la componen, por lo que se buscan nuevas tecnologías con mejores eficiencias y menor precio.

Las baterías juegan un papel muy importante en la eficiencia de los VE's, ya que para proporcionar una rápida aceleración y viajes de larga distancia, las baterías han de tener una gran potencia y energía específica.

Las baterías más utilizadas en los VE's son las de plomo-ácido, que continúa siendo la de mayor disponibilidad en el mercado; aunque existen otras tecnologías de construcción de baterías como son de níquel-cadmio, sodio-sulfuro, níquel-hidruros metálicos, litio-ión, etc.

La necesidad de desarrollar una batería con más potencia específica, más energía específica y mayor ciclo de vida impulsó la creación del *Advanced Lead-Acid Battery Consortium (ALABAC)* y el *United States Advanced Battery Consortium (USABAC)* que son organismos que regulan los estándares y pruebas que se realizan a las baterías .

En la tabla 1.2 se muestra la comparación entre los diversos tipos de baterías en el mercado.

BATERIA	ENERGÍA ESPECÍFICA (Wh/ kg)	POTENCIA ESPECÍFICA (Wh/ kg)	CICLO DE VIDA	AUTONOMÍA (km)	TIEMPO DE RECARGA EN HORAS
Plomo-ácido	33	130	400	96.56	8 a 17
Níquel-Cadmio	57	200	2000	193.121	8
Níquel-Acero	55	100	2000	177.028	4 a 8
Níquel-Metal Hidrido	70	250	600	402.336	<6
Litio-Ion.	100	300	1200	313.82	<3

Tabla I.2 Comparación de las especificaciones de diferentes baterías.

◆ Sistemas de carga

La tecnología de carga incluye sistemas de carga conductivos (contacto metal-a-metal) e inductivos (magnéticamente enlazados).

• Sistemas Conductivos

Los sistemas conductivos utilizan un enchufe y cable que puede variar según el tipo de conector usado, el nivel de voltaje y la corriente. Puede estar localizado dentro del vehículo o fuera de él. La figura I.1 muestra un sistema conductivo.



Figura 1.1 Conector para el sistema de carga conductivo.

- Sistemas inductivos

Los sistemas de carga inductivos utilizan un cable y un conector en forma de pala que la transfiere la energía del sistema eléctrico exterior al VE por medio de la inducción magnética, la figura 1.2 nos muestra uno de ellos.

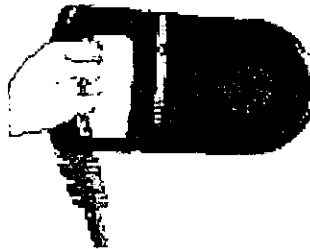


Figura 1.2 Conector para el sistema de carga inductivo.

Para los vehículos eléctricos se buscará que cada usuario tenga en su casa un sistema de carga de baterías; que sea tan sencillo como conectarse al tomacorriente. En el caso de estar lejos de casa y no tener carga suficiente para realizar la carga, se tiene pensado tener centros de carga públicos. La figura 1.3 muestra un sistema de carga típico.

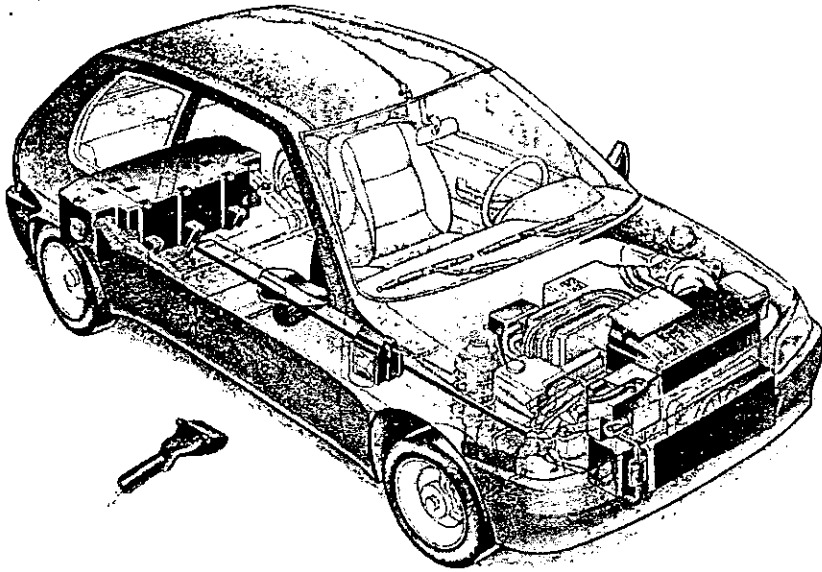


Figura I.3 Sistema de carga de un vehículo eléctrico.

En Estados Unidos de América, en particular en las ciudades de Los Angeles, California y Tucson Arizona, los departamentos encargados de regular las normas de distribución de energía eléctrica, se han asociado con *General Motors*, *Saturn* y *Edison EV* con el propósito de construir una infraestructura de estaciones de recarga pública para vehículos eléctricos. En la figura I.4 se puede apreciar una estación de carga existente en el estado de California.

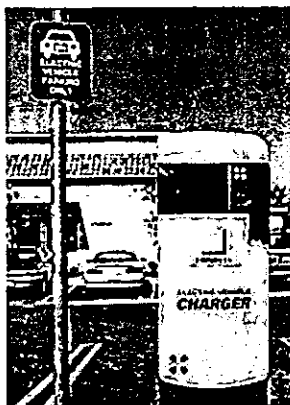


Figura I.4 Estación de carga ubicada en el estado de California.

Hasta la fecha, se han instalado 70 estaciones de carga pública en California y más de 30 en Arizona.

♦ Niveles estándar de carga de los VE's

El *National Electric Vehicle Infrastructure Working Council*, anunció en el año de 1994, la estandarización de los niveles de carga de los VE's, los cuales son:

- Nivel 1. La carga puede ser desde una conexión estándar de 120 V, con una salida de 3 A pues existen en cualquier casa.
- Nivel 2. Cargar a 240 V, con una estación de 40 A.
- Nivel 3. Tecnología de alto nivel, actualmente en desarrollo, que proporcionará una carga en 5-10 minutos, haciéndola similar al llenado de gasolina de los vehículos convencionales.

La electricidad requerida para el funcionamiento y carga de vehículos eléctricos puede ser suministrada por una amplia combinación de energías primarias: hidroeléctrica, nuclear y fósil, entre otras. En un futuro cercano no se requerirían nuevas centrales aun cuando se adopte masivamente el uso del vehículo eléctrico, si se considera que la recarga puede hacerse

fuera de las horas pico, periodos en los que la capacidad de generación está prácticamente ociosa.

Los vehículos eléctricos representan un nuevo mercado potencial para las compañías de generación del orden de cientos de millones de kWh. Esta idea resulta aún más atractiva si la mayor parte de la electricidad para la recarga se genera durante la noche, mejorando, al mismo tiempo, la carga del sistema y el factor de utilización de la central.

Ya hemos visto a grandes rasgos que es un VE, el porque de ellos y la tecnología que gira a su alrededor, con lo que podremos ahora revisar un poco de la historia que ha hecho este tipo de vehículos.

1.1.5 Historia de los vehículos eléctricos

Como parte de la necesidad del hombre de trasladarse para realizar diferentes actividades como el comercio con individuos de otros núcleos poblacionales, el transporte de material bélico en las guerras, o simplemente el placer de viajar, ha hecho que se busquen y desarrollen diferentes medios de transporte.

Los medios de transporte creados por el hombre necesitan una fuente de energía que a través de diferentes mecanismos pueda realizar la conversión a energía mecánica para impulsar ruedas, hélices, bandas, etcétera.

Durante el siglo XIX se desarrollaron en diversas partes del mundo algunos modelos de vehículos eléctricos. Se tiene conocimiento de que en 1834 el primer VE construido en América fue desarrollado en Estados Unidos, por Thomas Davenport, el cual utilizaba baterías no recargables.

En 1839 se construyó en Escocia el primer VE en Europa. El motor estaba construido con ocho bobinas, y se alimentaba con baterías primitivas. En el año de 1870, se desarrolló en el Reino Unido, un VE con un motor más evolucionado y ligero que permitía alcanzar

velocidades de 13 km/h; el peso de sus baterías fue causa de que la autonomía de este vehículo fuera inferior en comparación con los vehículos a vapor.

En 1896 el gobierno de España proyectó un VE, "el Julián", el cual se basaba en un rudimentario chasis y una carrocería de un constructor de carretas, *Trupp-Maberly*. Este modelo no tuvo continuidad. En 1897 la compañía *London Electric CA* pone en operación vehículos eléctricos donde se utilizaban baterías de plomo-ácido de 40 celdas y el motor tenía una capacidad de 2.24 kW con una autonomía de 80 km.

En 1899 se crea en Estados Unidos de América la *Electric Company*, que es la primera industria que fabrica vehículos eléctricos en gran escala, llegando a construir, en 1904, 2 000 unidades de taxis eléctricos que se emplearon en Nueva York, Chicago y Boston.

En 1900, nuevamente en España, se presentó un autobús de motor eléctrico, el cual tenía 20 plazas, medía 5 metros y conseguía con dos motores eléctricos 18 km/h subiendo pendientes de hasta 12%, en teoría. Cuando fue presentado falló en las primeras subidas suaves. En 1903 se desarrolló un vehículo híbrido; que se muestra en la figura I.5, éste consistía en un arreglo de un banco de baterías complementado por un motor a gasolina.

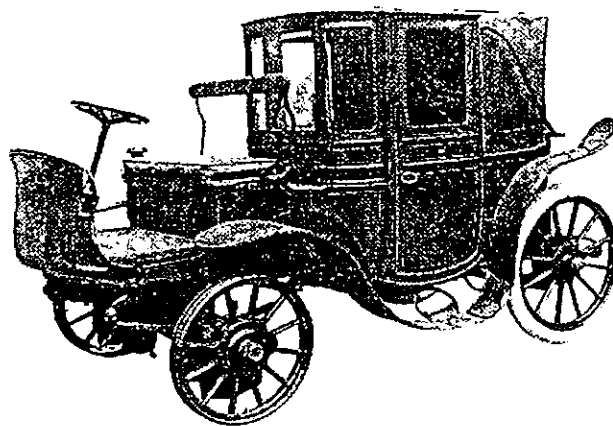


Figura I.5 Vehículo híbrido creado en 1903.

En la Exposición Internacional de Berlín en 1904, se presentaron producciones de todo el mundo, pero destacaron tres compañías de coches eléctricos: la *ABAM* de Berlín, la *Gottfr.Hagen* de Colonia y *Krieger* de París. Estos coches utilizaban el sistema *Krieger* con sus motores herméticamente cerrados para protegerlos del polvo y la humedad. El *ABAM* llevaba dos motores con 8 velocidades, que resistía doble sobrecarga. *Gottfr.Hagen* presentó una batería que consistía en 40 celdas y que decían que alcanzaba una velocidad de 30 km/h y una autonomía de 100 km, su peso era de 320 kg.

En 1905 la empresa Siemens empezó la fabricación de vehículos con motores eléctricos, sustituyendo el motor de explosión por acumuladores, delante del asiento del conductor, lo cual facilitaba su reemplazo. Los construyeron con diversas carrocerías pero la producción masiva no se logró por las limitaciones que presentaban las baterías en esos tiempos. En 1906 la compañía *Buffalo Electric Carriage* llegó a vender 34 000 vehículos de un modelo de dos plazas, que llegó a alcanzar velocidades de hasta 48 km/h.

Todavía en los años 1911 a 1920 en E.U. se ofrecían todavía grandes y lujosas limosinas con motor eléctrico, el *Broc* y el *Borland* son un ejemplo. También las marcas *Detroit Electric*, *Baker* y *Fritchle*, ofrecían pequeños VE urbanos, y vendieron unos miles entre 1912 y 1920.

En particular, los vehículos eléctricos no fueron rentables por el alto costo que su construcción y mantenimiento implicaban, aunque eran considerados más confiables y seguros, aunado al desarrollo de los motores a gasolina, que logró una mayor autonomía en los automóviles basados en este combustible en comparación con los eléctricos, motivo a la gente para cambiar de los vehículos híbridos o eléctricos a los de combustión interna alrededor de 1915.

En el año de 1920 las cualidades de VCI superan notablemente al VE. Y es el año en que Henry Ford fabricó vehículos a gasolina con mejores posibilidad de desarrollo y con reducidos costos. A partir de donde caen un poco en el olvido los vehículos eléctricos.

En Suiza se fabricaron furgones y furgonetas de la empresa *Tribelhorn*, con tres ruedas, que eran utilizados en hoteles y por el servicio de correo. En Alemania destacaron los triciclos *Geba* y los furgones *Bergmann*, un cliente importante para ellos fue *Bolle* de Berlín con un parque que sobrevivió a la segunda Guerra Mundial.

Acabada la II Guerra Mundial, con la falta de materia primas, prepararon un micro-coche para una persona y un ligero motor eléctrico que ofrecía no más de medio caballo de fuerza. Este modelo fue mejorado continuamente y más tarde ofrecía dos caballos de fuerza, y tomando en cuenta su escaso peso 180 kg y una transmisión de 1:20, se decía que tenía un buen arranque hasta los 20 km/h.

Ya en los años sesenta se empiezan a presentar diversos programas de desarrollo para emprender una investigación orientada al desarrollo de la tecnología de la propulsión eléctrica, motivados por la protección del medio ambiente. Para los años setenta surgió otra motivación que es la conservación de las fuentes de energía, el primer periodo de actividad fue dedicado al sistema de propulsión basado en tecnologías que reunieran los requerimientos de fácil manejo y ahorro de energía, incluyendo el freno regenerativo.

Es precisamente en 1976 que en los Estados Unidos de América se presenta la primera Ley Pública para el desarrollo y demostración del vehículo eléctrico e híbrido.

El almacenamiento de energía fue al mismo tiempo desarrollado en términos del óptimo uso de las baterías, incluyendo un manejo térmico y características de seguridad. La atención fue dedicada a los vehículos comerciales y de servicio público. Durante los Juegos Olímpicos de Munich se utilizó un VE fabricado por *BMW* que acompañó a los corredores del maratón. Posteriormente, en 1990, la misma empresa presentó un modelo cuyas baterías eran a base de sulfuro de sodio (NaS), con una capacidad total del banco instalado es de 19 kW/h y con aceleración de 0 a 50 km en 9 segundos.

En esas fechas la *FIAT* presentó un VE comercial llamado *Fiat Panda Elettra* con baterías de 6 Volts formando un banco de 12 baterías que dan una suma de 72 Volts, y un sistema auxiliar de 12 Volts, éste podía desarrollar una velocidad máxima de 70 km/h.

La compañía *General Motors* ha desarrollado actualmente un VE, el cual recibe el nombre de *GM EV1*. Este automóvil contiene un motor enfriado por agua, tiene la capacidad de desarrollar hasta 137 caballos de fuerza y va desde 0 hasta 13000 r.p.m., sin cambiar marcha. Puede ir de 0 a 96.56 km por hora en menos de 9 segundos, su máxima velocidad es de 128 km por hora, además de contar con frenado regenerativo.

En la figura 1.6 podemos apreciar el vehículo eléctrico de la *General Motors*.



Figura 1.6 EV1 de la *General Motors*.

En general todas las compañías automotrices han desarrollado algún tipo de vehículo eléctrico o híbrido, tanto en América como en Europa. Dejando de manifiesto la importancia que han cobrado en los últimos años.

Aunque todos los fabricantes tienen la idea de abarcar el enorme mercado del auto familiar, las flotillas urbanas de reparto de mercancías, alimentos y servicio de correo parecen ser el lugar indicado para iniciar la penetración comercial de los vehículos eléctricos.

Las tendencias de las características de los VE's actualmente en desarrollo en el mundo se muestran en la tabla 1.3.

Tipo de Vehículos	Capacidad de Transporte	Peso Vehicular (kg)	Tipo de Baterías	Tipo de Motor	Autonomía (km)	Velocidad Máxima (km/h)
Vehículos grandes de uso personal	3-5 pasajeros o hasta 850 kg.	850 a 1 630	plomo-ácido, níquel-cadmio, níquel-hidruros, metálicos	Continúa siendo de diversas características, no hay un patrón determinado,	75 hasta 322 depende del tipo de baterías	80 a 161
Medianos de uso personal	4 pasajeros o hasta 363 kg.	700 a 1000	níquel-cadmio, níquel-	los hay de corriente alterna o directa con o sin escobillas, de imanes permanentes, etc.	48 hasta 390 depende del tipo de baterías	72 a 140
Chicos de uso personal	2 pasajeros o hasta 250 kg.	645 a 817	hidruros, metálicos, litio-ión		40 hasta 110 dependiendo del tipo de baterías	48 a 97
Pequeños de uso personal	2 pasajeros	377a 580	níquel-cadmio, plomo-ácido		hasta 60	29 a 40

Tabla 1.3 Características de diferentes vehículos eléctricos.

A continuación presentamos una breve semblanza de los factores que han propiciado el desarrollo de este tipo de vehículos en la República Mexicana.

I.1.6 Los vehículos eléctricos en México

En cuanto a nuestro país no estamos aislados de la tendencia que han tenido los VE's, sino que también existe preocupación por parte de la comunidad científica y política por proveer de un sistema de transporte limpio y eficiente.

Como todos sabemos, el problema de la contaminación ambiental en México es cada día más grave; presentándose, durante la mayor parte del año, elevados índices de contaminación atmosférica, particularmente en la ZMVM. La fuente principal de contaminación la representan los vehículos de combustión interna, de las cuales el sector del transporte aportó en 1994 el 75% de las emisiones.

Dentro del Plan Nacional de Desarrollo 1995-2000, existe un programa encabezado por el Instituto de Ecología de la SEMANARP para mejorar la calidad del aire en la ZMVM, que proyecta la industrialización de los VE's, que quizá tendrá un precio muy alto, pero a largo plazo no lo será tanto en comparación con los gastos que se hacen actualmente con la aplicación de los programas de "Verificación" y "Hoy no circula".

Uno de los programas incorporados a la estrategia del Proaire es el de fomentar la introducción de la fabricación masiva de vehículos eléctricos de carga y pasajeros, así como extender su uso en aquellas áreas urbanas que por su trazo y condiciones de tránsito hagan más rentable la medida.

El Valle de México cuenta con zonas de gran densidad de población, esto ha motivado que casi el 50% de las rutas y ramales de microbuses tengan recorridos menores a 10 km. La mayor parte de la orografía del Valle de México corresponde a zonas planas sin pendiente en donde se facilita la utilización de vehículos eléctricos.

En México, la energía eléctrica se genera principalmente en plantas termoeléctricas e implica cierta emisión de gases contaminantes. Sin embargo, tomando en cuenta este hecho, el efecto neto de la sustitución de vehículos de combustión interna por vehículos eléctricos

representa una reducción drástica de emisiones totales. Cabe señalar, además, que las emisiones de las plantas termoeléctricas ocurren lejos de las concentraciones urbanas.

Existen estudios que comparan el rendimiento de los vehículos con motor de combustión interna con el de los vehículos eléctricos autónomos, con base en la energía primaria consumida por kilómetro recorrido y suponiendo el uso de energía eléctrica proveniente de plantas operadas con combustibles derivados del petróleo. En dichos estudios se toman en cuenta las pérdidas asociadas con la refinación y transporte del crudo, las eficiencias de conversión termoeléctrica y distribución de la energía eléctrica, etc. Los resultados favorecen a los vehículos eléctricos, cuyo rendimiento resulta ser del orden de un 60% mayor que el de los vehículos con motor de combustión interna.

I.2 El minibús eléctrico del Instituto de Ingeniería

El vehículo eléctrico UNAM, desarrollado en el Instituto de Ingeniería, es un minibús cuya fuente de energía es un banco de 52 baterías. Tiene una autonomía de 60 km, y está diseñado para proporcionar un servicio de transporte, urbano cómodo, seguro, no contaminante y muy silencioso.

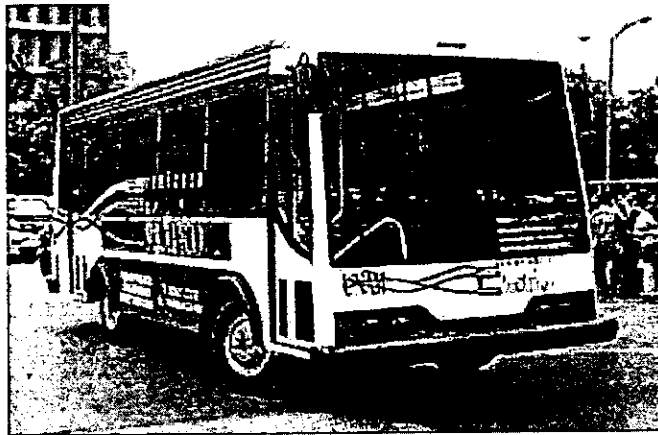


Figura 1.7 Minibús eléctrico desarrollado en el Instituto de Ingeniería.

Gracias a la participación del Centro de Investigaciones en Diseño Industrial de la Facultad de Arquitectura, se resaltaron en el diseño los aspectos ergonómico y estético, en la figura I.7 podemos apreciar la estética del diseño.

Entre las características más importantes se pueden señalar las siguientes:

Capacidad	30 pasajeros
Velocidad de crucero	40 km/h
Velocidad máxima	60 km/h
Aceleración inicial	0.9 m/s ²
Tiempo en alcanzar de 0-40 km./h	14 s
Habilidad de pendiente (10%)	15 km/h
Radio de viraje	7.5 m
Autonomía	60 km
Banco de baterías	52 baterías selladas de plomo-ácido que proporcionan 160 Ah. 312 Volts Corriente Directa
Potencia del motor de tracción	50 kW
Peso total con carga	6 Toneladas

Un sistema de transporte de pasajeros muy común en el Valle de México corresponde a vehículos automotores de media capacidad (microbuses), que podrían ser sustituidos por vehículos eléctricos de capacidad similar, como el mostrado en la figura I.8.



Figura I.8 Vista lateral del minibus eléctrico.

El Vehículo Eléctrico de la UNAM representa un esfuerzo en el que han participado investigadores y técnicos de diversas disciplinas, así como estudiantes. Es una muestra del trabajo en equipo que requieren los desarrollos tecnológicos modernos que pretendan competir en un ámbito internacional.

Después de ver los antecedentes, procederemos a dar una revisión de los conceptos básicos que necesitaremos para el desarrollo del presente trabajo.

==== CAPITULO II ====

Conceptos básicos

En este capítulo abarcaremos los conceptos básicos necesarios para entender en conjunto las características del sistema a desarrollar. Entre los conceptos a mencionar están, por ejemplo: los convertidores analógicos-digital, multiplexaje, memorias, características de algunos microcontroladores y el estándar RS-232.

II.1 Sistemas de adquisición de datos

Las técnicas digitales utilizadas en la obtención, transmisión, procesamiento y evaluación de información han tenido en los últimos años un rápido desarrollo. Sus ventajas consisten en que las señales discretas, por ser menos sensibles a las perturbaciones, se transmiten y procesan con menos errores, y por otra parte, en que dichas señales son más fáciles de almacenar. Los circuitos digitales operan, casi siempre, con señales binarias, de modo que sólo se distinguen dos estados lógicos. La designación de los estados lógicos con H y L, representan respectivamente los potenciales alto (*high*) y bajo (*low*).

En un sistema digital que opera en conjunto con otros dispositivos o procesos, puede ser necesario incluir un convertidor analógico-digital (CAD) o digital-analógico (CDA). En algunos procesos las señales recibidas por el sistema pueden incluir voltajes de corriente directa los cuales son de forma analógica. Dado que el sistema digital sólo opera con números binarios, es necesario convertir estos datos analógicos en digitales. La conversión de analógico a digital se realiza por una unidad CAD. Las ventajas de representar las señales digitalmente, en lugar de analógicamente, por ejemplo en el área de medición, son el evitar errores de lectura, la mayor precisión, la mayor seguridad en el funcionamiento y la sencillez con que se almacenan y procesan en una computadora.

En las aplicaciones con microcontroladores, los datos de entrada que han de ser manejados son magnitudes de voltajes analógicos, ya que la mayoría de los parámetros físicos, susceptibles de ser controlados, son analógicos, por ejemplo: temperatura, humedad, velocidad, presión, etc. Lógicamente, los transductores que convierten estas magnitudes a voltaje entregan niveles de voltaje analógicos. De la misma manera, el microprocesador entrega datos digitales que es posible que se necesiten nuevamente convertir a analógicos. Para todo esto es necesario disponer de una circuitería adicional de interconexiones entre la

unidad de E/S del microprocesador y los dispositivos periféricos, para permitir la conversión de señales analógicas a digitales y viceversa.

II.1.1 Convertidores analógico-digital

Los distintos tipos de convertidores analógico-digital se obtienen de las formas diferentes de procesar las cantidades con mayor o menor rapidez de cambio de valores continuos, esto es para corriente y voltaje. Hay al menos tres métodos distintos de CAD que utilizan principios independientes entre sí, además de algunos arreglos que abarcan combinaciones de los fundamentos de cada uno de éstos. Estos son:

- Los convertidores que trabajan con el método de integración, o sistema contador de incrementos. Entre los más conocidos están el de integración (o rampa) simple y el de integración (o pendiente) doble; siendo este último una clase de convertidor de voltaje a frecuencia. También están el tipo potenciométrico, el de balance de carga, el convertidor de voltaje a frecuencia, etc.
- Los llamados servosistemas digitales (en analogía con métodos de control por retroalimentación y búsqueda de error cero de servomotores), donde se consideran tres tipos sobresalientes que hacen uso de CDA en la trayectoria de retroalimentación. Es decir, el CAD de escalones o rampa discreta escalonada (también conocido como CAD con comparación por rampa de precisión o rampa contadora), el de rastreo de señal por rampa contadora continua (a veces llamado CAD de balance continuo) y, uno de los más importantes dentro de esta clasificación, el tipo de aproximaciones sucesivas.
- El basado en comparación directa de niveles, que frente a los métodos anteriores resulta incuestionablemente más rápido, pero más elaborado y costoso si ha de competir en varios aspectos.

Existen circuitos comerciales de convertidores CAD de 8, 10, 12 bits, realizados mediante la técnica de aproximaciones sucesivas; por ejemplo, el ADC0801 de National. De esta forma, se simplifica notablemente la programación determinada del microprocesador, ya que el CI realiza por sí solo todo el proceso de conversión. Pero también la conversión por

software constituye una alternativa económica al conversor por *hardware*, aunque acostaba de emplear el microcontrolador en el proceso de conversión, implementándole la correspondiente rutina de conversión.

II.1.2 Características del convertidor analógico-digital

La salida digital de un convertidor analógico-digital ideal de n-bits se grafica en función del voltaje de entrada analógica. Para un convertidor ideal de 4-bits la siguiente gráfica, figura II.1 , nos muestra su característica de entrada-salida.

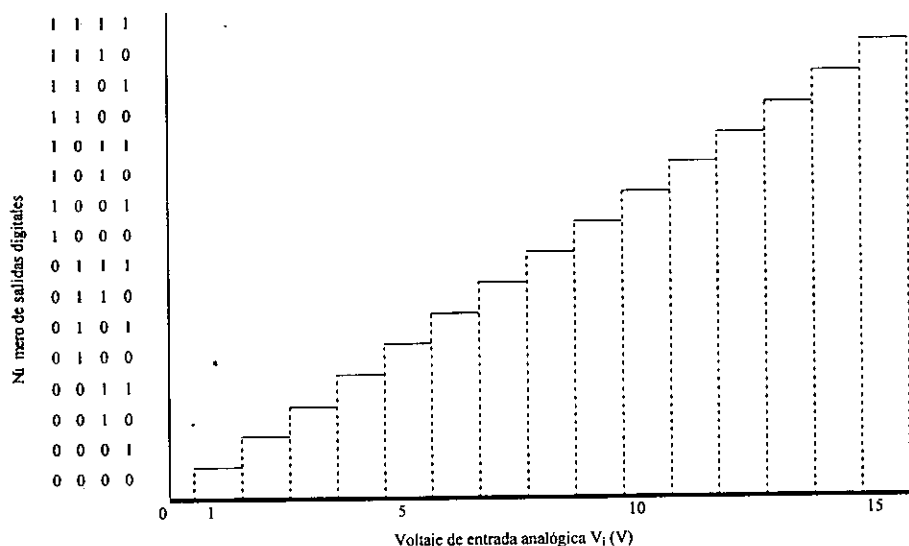


Figura II.1 Característica de entrada-salida de un CAD.

Esta gráfica nos muestra que para determinadas rangos de voltajes de entradas analógicas (V_i), existe una salida digital que está representada por un código binario de salida.

♦ Tiempos de adquisición y apertura

Cuando se desea digitalizar una señal que cambia rápidamente de valor puede ocurrir un cambio significativo en el valor de dicha señal antes de que termine la conversión. Esta anomalía en sí es una fuente de error que además puede generar otros errores.

Para reducir los efectos de ésta anomalía se emplea un circuito denominado muestra y retención (*sample and hold*). Se trata de un interruptor electrónico de alta velocidad usado para tomar muestras de la señal y un circuito capacitivo seguido de un amplificador de ganancia unitaria, usado para almacenar el valor de muestreado de la señal.

Dos parámetros de este circuito son muy importantes: el tiempo de adquisición y el tiempo de apertura. El primero se refiere al tiempo necesario para cerrar el interruptor y cargar el capacitor al voltaje de la señal de entrada. El tiempo de apertura es el tiempo que requiere el interruptor para abrir completamente. Idealmente estos tiempos deben tender a cero.

♦ Precisión y resolución

Dos factores importantes de la conversión son la precisión y la resolución. Mientras que la precisión indica que tan cercano es el valor medido respecto al valor real, la resolución indica que tanto es posible distinguir entre dos voltajes. Una precisión del uno por ciento indica que 100 volts pueden ser en realidad de 99 a 101 volts, y no se puede distinguir con más detalle. Una resolución de 100 mV significa que no se puede distinguir entre voltajes sucesivos menores de 100mV (0.1V).

Frecuentemente no resulta muy claro que una operación se pueda hacer con una mayor resolución que la precisión obtenible. Puede ser posible resolver entre 10.1 y 10.2 volts y el voltaje real puede haber sido 10.5V. Aquí la precisión fue más pobre que la resolución.

Tendría más sentido obtener una resolución tan buena como la precisión de valores significativos. En cambio, una alta precisión y baja resolución, también es pobre, ya que se puede obtener algo muy cercano al voltaje real, pero no es posible apreciar durante las operaciones las diferencias más pequeñas.

La resolución de un convertidor analógico-digital se define de dos maneras. Primero, es el número máximo de códigos de salida digital, la cual se define por:

$$\text{Resolución} = 2^n$$

La resolución también se define como la razón de cambio del valor en el voltaje de entrada, V_i , que se necesita para cambiar en 1 *LSB* la salida digital. Si se conoce el valor de voltaje de entrada a escala completa, V_{iFS} , que se requiere para producir una salida digital de todos los unos, es posible calcular la resolución mediante:

$$\text{Resolución} = \frac{V_{iFS}}{2^n - 1}$$

♦ **Monotonicidad**

Un convertidor analógico-digital ó digital-analógico es monotónico si su salida se incrementa o permanece constante cuando la entrada se incrementa para todo el rango de la escala.

♦ **Error de cuantificación**

Existe una incertidumbre entre el valor exacto de V_i en una determinada salida. En la figura II.1 muestra que la salida binaria es 0101 para todos los valores de V_i entre 4.5 y 5.5 V. Existe una incertidumbre inevitable respecto al valor exacto de V_i cuando la salida es 0101. Dicha incertidumbre se especifica como error de cuantificación, su valor es $\pm \frac{1}{2}$ *LSB*. Al incrementar la cantidad de bits se logra una mejor resolución y el error de cuantificación es menor.

♦ **Linealidad**

Para un convertidor analógico-digital la linealidad se conoce con el nombre de precisión relativa y se define por: la desviación del código digital de salida del A/D con respecto a una línea recta trazada del valor 0 al valor de escala completa de entrada.

♦ **Tiempo de conversión**

El tiempo de conversión es el tiempo requerido para que la circuitería de conversión analógico-digital convierta una señal analógica a un conteo digital equivalente. En cualquier sistema esto depende tanto de la sincronía como del máximo número de conteos.

Para un convertidor de aproximaciones sucesivas de 12-bits, sólo se requieren 12 pasos de sincronía, sin importar el valor del voltaje de entrada. Esto es, el convertidor por aproximaciones sucesivas sólo requiere un número de etapas de sincronía equivalente al

tamaño de bit de conversión, y está fijado por esta medida y la frecuencia de la señal de reloj usada, esto es:

$$\text{Tiempo de conversión} = n \times \frac{1}{f}$$

donde n es el número de bits y f la frecuencia de la señal de reloj.

♦ Multiplexaje

Frecuentemente una sola unidad de CAD es usada para manejar varias entradas analógicas diferentes. En estos casos, se completa la conversión para una entrada, se reajusta el circuito, y se aplica una nueva entrada al convertidor. El método de conmutación de una señal de entrada a otra para permitir que una sola unidad de pueda manipular muchos canales de información se llama multiplexaje. No es necesario abordar un estudio de muestreo en sistemas de datos para esto, sino considerar básicamente la operación de muestreo de señales variantes, así como la relación de muestra en la señal de salida en el sistema usado. El multiplexaje, en sí mismo, es el método de conmutar muchas entradas en un canal, donde el multiplexor es la unidad muestreadora. Un sistema típico de multiplexaje es mostrado en la figura II.2.

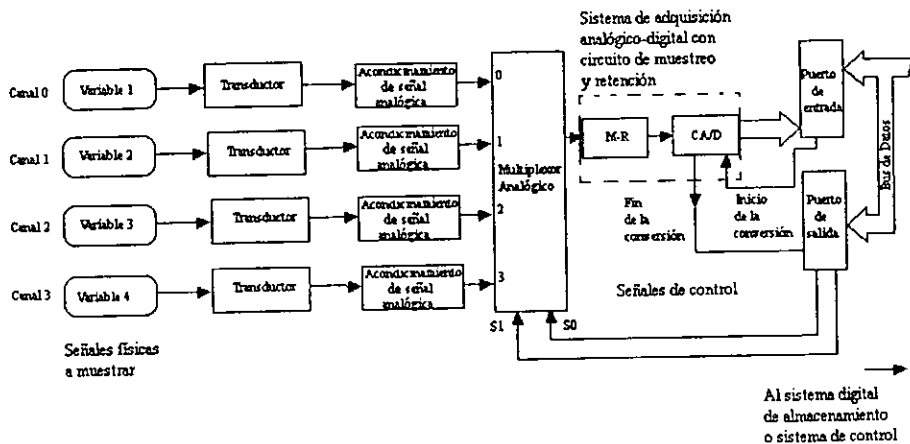


Figura II.2 Sistema de multiplexaje.

El sistema de multiplexaje nos muestra la adquisición de cuatro variables físicas que ocupan un solo canal del convertidor. Cada señal es convertida en señal eléctrica por medio de un transductor, el cual lo entrega a un circuito de acondicionamiento. La unidad mutiplexora selecciona una a una cada señal de los acondicionadores, las cuales son recibidas por la unidad de conversión analógico a digital, en donde por medio del circuito de muestreo y retención, es capaz de congelar la señal a adquirir. Una vez digitalizada, cada señal es entregada al sistema digital de almacenamiento o en su defecto a un sistema de control, el cual usará esa información para un fin predeterminado.

Existen diferentes métodos para multiplexar señales, uno de ellos es la Multiplexación por División de Tiempo (TMD). Esta técnica de multiplexaje consiste en aprovechar los tiempos entre los distintos muestreos para así poder hacer un cambio en la entrada hacia las señales y así poder hacer su adquisición. La consideración en este caso es que se debe conocer el orden en que son accedadas dichas señales, para poder así llevar a cabo su seguimiento.

La otra técnica es llamada Multiplexación por División de Frecuencia (MDF), y consiste en aprovechar el ancho de banda del canal del multiplexor. Aquí todas las señales son procesadas al mismo tiempo, con la diferencia que a cada una le corresponde una frecuencia distinta.

II.2 Microcontroladores y memorias

II.2.1 Microcontroladores

Un microcontrolador es un circuito integrado (CI) de una gran escala de integración (LSI) extremadamente complejo, que tiene la capacidad de ser programable. Está constituido básicamente de los siguientes bloques:

- Unidad Aritmética Lógica.
- Registros.
- Memoria.
- Puertos de comunicación.

- Temporización.
- Control.
- Convertidores analógico-digital.

Con la interacción de estos bloques, además de la existencia de un programa, el microcontrolador es capaz de realizar operaciones lógicas, aritméticas, controles de memoria, así como el control de dispositivos periféricos de entrada y salida. Para realizar esta interacción el microcontrolador necesita un conjunto de instrucciones, del cual toma la función del programa. Los microcontroladores fueron creados específicamente para controlar máquinas. Su parte principal es un microprocesador, al cual se le agregan dispositivos, tales como, contadores programables, puertos seriales de comunicación (UART's), puertos paralelos de entrada/salida (I/O), convertidores analógico-digital (CAD), etc.

El microcontrolador (MCU) está empaquetado en un CI que puede ser programado por el usuario con una serie de instrucciones cargadas en memoria. Antes de los MCUs, los circuitos de control eran muy complejos en su diseño, en su alambrado, y su operación era determinada por los circuitos y conexiones contenidas en ellos.

La operación de un control basado en un MCU está determinada primeramente por el programa en lugar de los componentes y alambres. Cualquier función que puede ser implementada usando circuitos integrados (CIs) y complejos alambrados, también puede ser implementada y ejecutada por un MCU.

Un diagrama de una unidad de microcontrolador es mostrado en la figura II.3, y ahí se pueden observar los diferentes componentes que en general integran un MCU, esta figura nos ayuda a comprender mejor como está constituido.

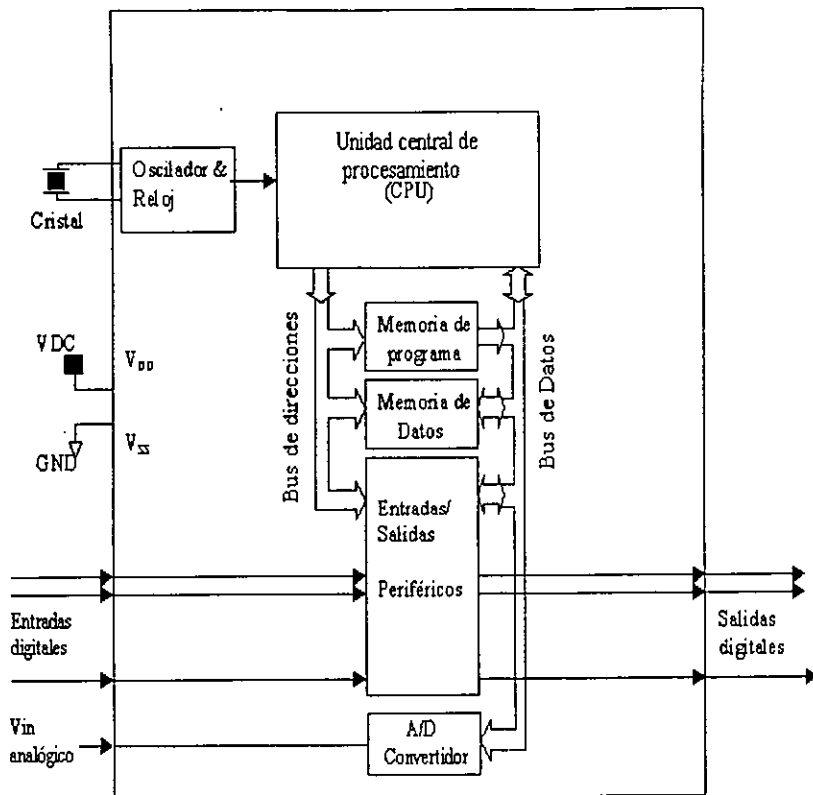


Figura II.3 Diagrama de la unidad de microcontrolador.

Como el tamaño y la complejidad de los dispositivos se incrementa, los MCU se hacen más atractivos por dos razones:

- El control diseñado con compuertas y alambres necesita adicionar más CI's, si desea un funcionamiento más complejo, un MCU requiere una nueva programación.
- Los microcontroladores son más versátiles. Cualquier cambio en un sistema en base a compuertas usualmente requiere reemplazar CIs y redefinir trayectorias. En la mayoría de las modificaciones en un sistema MCU simplemente hay que hacer cambios en el programa.

Los MCU son muy útiles cuando se requiere tomar muchas decisiones o hacer muchos cálculos. Esto es realmente fácil para un sistema computacional en comparación de un sistema de lógica discreta.

Ahora también son muy usados para reemplazar los diseños existentes porque ellos son mucho más simple de usar que los circuitos lógicos convencionales. Gracias a su característica de ser programables hacen posible no adicionar costos al sistema.

♦ Aplicaciones de los microcontroladores

El desarrollo de una nueva aplicación para el MCU está limitado únicamente por la habilidad e imaginación, ya que los elementos un microcontrolador son fácilmente ensamblados. Sus aplicaciones generalmente permiten implementar nuevas funciones que van desde muy simples hasta procesos de control, a un costo reducido.

Para algunos casos se requieren entradas y salidas analógicas. El resultado es equivalente a los controladores analógicos con un número de lazos de control. Los lazos de control regulan una salida como función de una o más entradas.

Algunas aplicaciones están constituidas por sensores y mecanismos de control. El costo de los sensores requeridos para las entradas y de los dispositivos de control para las salidas es usualmente mucho mayor que el costo de un MCU estándar.

La ventaja de un sistema MCU es que se usa *software* para reemplazar complejos y costosos diseños de *hardware* anteriormente requeridos.

♦ Operación de los microcontroladores

En la gran mayoría de las aplicaciones de control, la MCU puede satisfacer todos los requerimientos del sistema sin circuitos integrados adicionales.

Los sistemas se vuelven de bajo costo y de una muy alta flexibilidad, estos nuevos y poderosos dispositivos se pueden encontrar dentro de muchas aplicaciones que anteriormente se hacían con lógica combinacional o por otros mecanismos.

De los microcontroladores que se encuentran en el mercado, existen varias familias que manejan palabras de ocho bits, como son las de *National*, *Motorola*, *Intel*, *Microchip*, *AVR*, *Toshiba* etc., cabe mencionar que también existen los que manejan palabras de mayor tamaño.

Dentro de estas familias de microcontroladores, de las que más información e infraestructura tenemos son: *National*, *Motorola*, *Intel*, *Microchip* y *AVR*, pero con los que tenemos más experiencia son los siguientes:

♦ **Familia COP de National**

El microcontrolador de *National* (COP) tiene una arquitectura *Harvard* modificada con un mapa de memoria de entrada/salida. La CPU contiene un registro acumulador de 8 bits (A), un contador de programa (CP), dos apuntadores de 8 bits (B y X), un *stack pointer* de 8 bits (SP), un procesador de estado de palabra (PSW), y un registro de control de 8 bits (CNTRL). La memoria de datos incluye un banco de 16 registros (incluyendo los tres apuntadores) que tienen atributos especiales. Toda la RAM, los puertos de entrada/salida, y registros (excepto A y PC) están mapeados dentro del espacio de datos de la memoria. La sección de temporizadores incluye un timer de 16-bit, y un registro asociado de autorecarga.

La sección genérica de entradas/salidas del COP8 incluye dos puertos de entrada/salida de 8 bits, cada uno tiene asociado un registro de configuración y un registro de datos de 8 bits cada uno.

La sección de entrada/salida también tiene un puerto dedicado a salida, uno dedicado a entrada y un puerto de propósito especial. Los puertos de entrada/salida son programables por *software*.

Esta familia está construida con tecnología NMOS, tiene como ventajas principales bajo costo en los dispositivos, lógica de interrupción muy versátil, sistema completo de sincronización, memoria ROM, memoria RAM y puertos de entrada y salida.

Algunos de los miembros de esta familia son los siguientes: *COP620C*, *COP640C*, *COP820C*, *COP888CL/CP*, *COP8SAA*, *COP8SAB*, *COP8SAC*.

Las características de esta familia incluyen arquitectura de 8 bits en el *bus* de datos y direcciones, *interfaces* seriales de entrada-salida, contador, reloj de 16 bits con registros de salida y niveles de interrupción, *set* de instrucciones versátil y compatible con cada elemento de la familia, cada puerto de entrada-salida tiene un *software* con opciones que permite la adecuación específica, además esta familia opera con un rango de voltajes de 2.6 a 6 V.

Algunas otras características que ofrece esta familia son:

- Bajo costo, en casi todos sus microcontroladores de 8 bits.
- Frecuencia de operación de 10 MHz (ciclo de instrucción de 1 microsegundo).
- 1 Kbytes de memoria ROM y 64 bytes de memoria RAM en los MCU's *COP820C*; 2 Kbytes de memoria ROM y 128 bytes de memoria RAM en los MCU's *COP840C*.
- Reloj de 16 bits de lectura-escritura que opera en tres modos: reloj con registro de autocarga de 16 bits, contador de eventos externo de 16 bits, reloj con registro de captura de 16 bits.
- Fuente de interrupción cuyos modos son: interrupción activada por un *reset* de limpieza, modo externo de interrupción con flanco de interrupción programable, reloj de interrupciones o captura de interrupciones y modo de interrupción de *software*.
- *Watchdog*.
- Registro *stack-pointer* de 8 bits.
- Cuenta con un bit de seguridad que permite proteger contra la lectura de sus códigos programados.
- *Set* de instrucciones para aritmética BCD.
- *Microware*.

- *Interface* serial de entrada y salida.
- Modo de bajo consumo de energía.
- Opciones de uso del *software* para los modos de entrada-salida (tres estados, *Push-Pull*, etc.).
- Entradas tipo *Schmitt-Trigger* en el puerto G.
- Rangos de temperatura de -40°C a +85°C y -55°C a +125°C.

◆ **Familia MC68HC11 de MOTOROLA**

Esta familia está formada por un grupo sofisticado de microcontroladores de 8 bits, con tecnología HCMOS, con velocidad de *bus* nominal de 2 MHz. Las características generales de la familia son:

- Un sistema de memoria de 8 kbytes de memoria ROM, 512 bytes de memoria EEPROM y 256 bytes de memoria RAM.
- Un puerto de 8 líneas multiplexadas a un convertidor analógico-digital con una resolución de 8 bits.
- Sistema de monitoreo tipo *watchdog* de la circuitería interna del circuito integrado, para prevenir fallas.
- Una *interface* para comunicación serial asíncrona SCI y una *interface* para periféricos seriales síncronos SPI.
- El sistema principal de reloj tiene tres líneas de captura de entrada, 5 líneas de salida de comparación y una función real para interrupción.

- Subsistema acumulador de pulsos de 8 bits, capaz de contar y/o convertir eventos externos con ó sin periodicidad, entre otras aplicaciones.

También por medio de *software*, *WAIT* y *STOP*, es posible controlar la potencia requerida por el CI, logrando como consecuencia conservar energía adicional. Estos dos modos hacen que la familia MC698HC11 sea de gran interés para aplicaciones automotrices ó donde se requiere un buen aprovechamiento de una batería. Es muy importante aclarar que el conjunto de instrucciones es completamente compatible con toda la familia. Así mismo algunas instrucciones son tan versátiles que permiten que varias operaciones sean ejecutadas por una sola instrucción.

A continuación se muestra en la tabla II.1, las características de los principales miembros de la familia de microcontroladores de *MOTOROLA* 68HC11.

	68HC11	68HC11E	68HC11E	68HC11F	68HC11A	68HC11D3	68HC11D3	68HC11E9
	A8	9	2	1	8			
ROM	8 k	12 K	0	0	8 K EEPROM	4 K EEPROM	4 K	12 K EEPROM
RAM	256	512	256	1 K	256	192	192	512
EEPROM	512	512	2 K	512	512	0	0	512
RELOJ	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT
SPI	SI	SI	SI	SI	SI	SI	SI	SI
SCI	SI	SI	SI	SI	SI	SI	SI	SI
A/D	SI	SI	SI	SI	SI	NO	NO	SI
INT.T. REAL	SI	SI	SI	SI	SI	SI	SI	SI
WATCH DOG	SI	SI	SI	SI	SI	SI	SI	SI
ENCAP- SULADO	PLCC 52 DIP 48	PLCC 52	PLCC 52	PLCC 68	PLCC 52	PLCC 44 DIP 40	PLCC 44 DIP 40	PLCC 52
I/O	38	38	38	30	38	32	32	38
VEL.DEL BUSMHz	0-2.1	0-2.1	0-2.1	0-4.2	0-2.1	0-2.1	0-2.1	0-2.1

Tabla II.1 Características de la familia MC68HC11.

◆ **Familia 80C51 de Intel**

El 80C51 tiene una arquitectura *Harvard* modificada con los puertos de entrada/salida mapeados en memoria. Los registros de la CPU incluyen un acumulador de 8-bits (A), un registro auxiliar de 8-bits (B) para multiplicación y división, un contador de programa de 16-bits (PC). Una palabra de 8-bits de control (PSW), un *stack pointer* de 8-bits (SP), y un apuntador de datos de 16-bits (DPTR). Su banco de registros consiste en ocho registros especiales de trabajo que van desde R0 a R7. Los registros R0 y R1 pueden usarse como apuntadores de memoria. La sección de *timers* incluyen dos contadores de 16-bits, un registro de 8-bits para el modo de operación (TMOD), y un registro de control de 8-bits (TCON). La sección de entradas/salidas incluye cuatro puertos de 8-bits, cada uno asociado a un registro de control de 8-bits. Esta familia tiene una amplia variedad de microcontroladores para las aplicaciones en tiempo real, tales como instrumentación médica, control industrial y aplicaciones de control automatiz. Estos microcontroladores son de 8 bits y están disponibles en versiones que cuentan internamente con ROM, EEPROM, o solamente el CPU. Además pueden direccionar externamente una memoria de 64 kbytes para datos y 64 kbytes de memoria de programa.

Algunas características del 80C51 son:

- CPU de 8 bits optimizado para aplicaciones de control.
- Capacidad de procesamiento Booleano para lógica de un solo bit.
- RAM interna de 128 bytes para datos.
- 2 relojes/contadores de 16 bits.
- 32 líneas bidireccionales I/O y programables.
- UART (Transmisor Receptor Universal Asíncrono) *full duplex*.
- Estructura de interrupción de 5 fuentes con dos niveles de prioridad.
- Memoria interna de programa de 4 kbytes.
- Circuitería interna en el CI de reloj oscilador.
- Direccionamiento de 64 kbytes de memoria de datos.
- Direccionamiento de 64 kbytes para memoria de programa.
- Su presentación es en encapsulado DIP 40 y PLCC de 44 terminales.

A continuación se presenta una tabla comparativa de diversas microcontroladores de esta familia , tabla II.2.

Nombre	Version sin rom	Version eeprom	A/d	Puertos	PWM	Reloj	Puerto Serial	Rom Bytes	Ram Bytes	Tipo de Circuito
80C51	80C31	87C51	-	4	-	2 st.	UART	4 K	128	HMOS
80CL51	-	-	-	4	-	2 st.	UART	4 K	128	COMS
80C52	80C32	87C52	-	4	-	2 st.	UART	8 K	256	COMS
83C053	-	87C054	-	3 1/2	9	2 st.	-	8 K	192	COMS
83CL410	80C410	-	-	4	-	2 st.	I ² C	4 K	128	COMS
83C451	80C451	87C451	-	7	-	2 st.	UART	4 K	128	COMS
83C528	80C528	87C528	-	4	-	2 Reloj. 2 st.	UART I ² C	32 K	512	COMS
83C550	80C550	87C550	8c/8b	4	-	2 st. Watchdog	UART	4 K	128	COMS
83C552	80C552	87C552	8c/10b	6	2	2 st. 2 Reloj. Watchdog	UART I ² C	8 K	256	COMS
83C562	80C562	-	8c/8b	6	2	2 st. 2 Reloj. Watchdog	UART	8 K	256	COMS
83C575	80C575	87C575	4 CO MP	4	5	2 st. 2 Reloj. Watchdog PCA	UART	8 K	256	COMS
87C592	-	87C592	8c/10b	6	2	2 st. 2 Reloj. Watchdog	UART CAN	16 K	512	COMS
83C652	80C652	87C652	-	4	-	2 st.	UART I ² C	8 K	256	COMS

Tabla II.2 Características de la familia 80C51.

II.2.2 Memorias

Una memoria es un dispositivo o medio en el cual se puede almacenar la información y después recuperar. Las memorias se utilizan para almacenar la información temporalmente y en grandes períodos de tiempo en sistemas como computadoras, en calculadoras, y en controladores industriales. También se utilizan las memorias en los instrumentos electrónicos tales como osciloscopios y multímetros digitales.

Una memoria de semiconductor es un dispositivo MSI ó LSI de transistores bipolares o transistores MOS. Las memorias varían su capacidad de almacenar información, puede haber sólo unos cuantos bits o puede haber miles de bits almacenados en un solo CI.

La capacidad de una memoria se define por la cantidad de bits que puede almacenar, esta capacidad en memoria electrónica de los sistemas digitales varía desde poco menos de 100 bits, para una simple calculadora de bolsillo de 4 operaciones, a entre 10^4 y 10^6 bits para computadoras personales y hasta 10^9 bits o más para las grandes computadoras comerciales.

Los diseñadores de circuitos normalmente hablan de capacidades de memoria en términos de bits; por otro lado, los diseñadores de sistemas suelen hablar de capacidades de memoria en términos de bytes (8 bits), donde cada byte representa un dato almacenado.

En un sistema microcontrolador, los datos e instrucciones del programa son almacenados en la memoria. Cada "celda" de la memoria puede almacenar un bit, estando las memorias constituidas por varios miles de estas celdas. El conjunto de celdas en las que almacena una palabra se llama "posición de memoria". Para poder introducir datos en la memoria, y para poder sacarlos de ella, cada posición de memoria viene dada por su correspondiente "dirección". La dirección es una palabra binaria que define la posición, es importante distinguir entre lo que es una dirección de una posición de memoria, y el dato que puede ser almacenado en esa dirección.

Las líneas de dirección proporcionan la selección de la localidad de memoria para leer o escribir datos sobre ella. Las líneas de datos manejan entonces los bits que van a ser leídos de, o grabados a la localidad de memoria seleccionada. Aunque tanto la dirección como el

dato son palabras binarias, la primera es siempre fija, pero la segunda puede cambiar sin más que alterar el dato. El número de posiciones de memoria que pueden existir estará determinado la capacidad de la memoria. Debido a que el direccionamiento de las memorias está basado en códigos binarios, las capacidades más comunes son aquellas que son potencias de 2. Donde por convenio tenemos que, según la tabla II.3:

1024 byte	1Kbyte
2048 byte	2Kbyte
4096 byte	4Kbyte
32768 byte	32Kbyte
65536 byte	64Kbyte
1049000 byte	1Mbyte

Tabla II.3 Convención de nomenclatura de tamaños de memorias.

◆ Clasificación de las memorias

Las memorias las clasificaremos principalmente en dos aspectos:

Volátiles: Para que el contenido permanezca, es necesario una fuente de alimentación. Al desconectarla se pierde la información almacenada. Las Memorias de Acceso Aleatorio (*Random Access Memory*, RAM) pertenecen esta categoría.

No volátiles: La información persiste aún desconectando la fuente de alimentación de la unidad de memoria, esto es, el contenido es respaldado sin consumo de energía. Las Memorias de Sólo Lectura (*Read Only Memory*, ROM) se encuentran en esta categoría.

• Memorias RAM

Se trata de una memoria de lectura-escritura, gracias a que admite la posibilidad de efectuar ambas operaciones, estando clasificadas en dos grandes categorías: estáticas y dinámicas.

Las RAM estáticas utilizan como elementos de memoria biestables tales como los Flip-Flop, y almacenan la información hasta que se desconecta la alimentación. Las RAM dinámicas almacenan los datos por carga de capacitores, normalmente, las capacidades de puerta de un MOSFET. Debido a la tendencia de este tipo de carga a desaparecer, después de unos pocos milisegundos, las RAM dinámicas deben ser refrescadas continuamente, para mantener su carga y evitar que los datos se pierdan.

Una memoria RAM está constituida por el bloque de almacenamiento, capaz de almacenar "n" bytes (palabras de 8 bits); las líneas de entrada de direcciones; las líneas de entrada y salida de datos, y finalmente, las líneas de control, el siguiente diagrama nos ilustra su constitución, figura II.4.

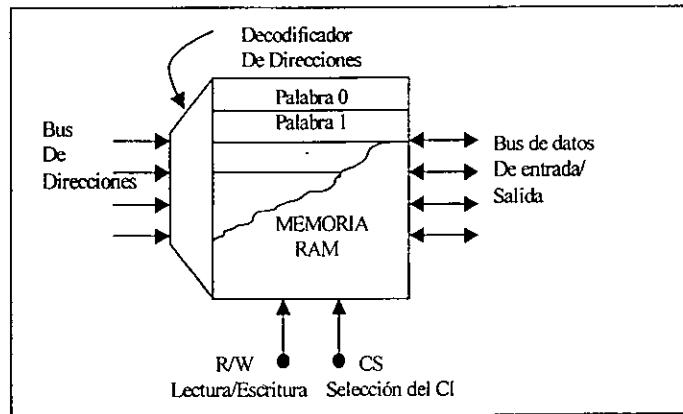


Figura II.4 Estructura de una Memoria de Acceso Aleatorio.

El *bus* de datos es bidireccional, la entrada y salida de datos se efectúan sobre las mismas líneas. El *bus* de direcciones está formado por las líneas que nos indican la posición en la memoria.

Las líneas de control están formadas, por lo general, de las siguiente señales:

R/W : El que los datos sean leídos o escritos, dependerá del estado de esta señal de control:

R/W = 1 se permitirá la lectura.

R/W = 0 se permitirá la escritura.

CS : Esta línea de control, selección de CI, permite mediante la aplicación de un "0" lógico, la conexión efectiva de la unidad de memoria a los *buses* de datos y direcciones, y por otro lado, si CS = 1, las líneas de conexión de la memoria a los *buses* entrarán en estado de alta impedancia y se impedirá su funcionamiento.

Los bits en las líneas de datos, en resumen, se pueden escribir en una palabra seleccionada en la memoria, controlada por la dirección y las señales de lectura/escritura.

En un sistema basado en microcontrolador, la memoria RAM se utiliza para las aplicaciones en las que la información debe ser continuamente almacenada y extraída, como lo es un banco de datos temporal.

La lectura de una memoria RAM se realiza de acuerdo con la siguiente secuencia:

- 1) Habilitar la unidad de memoria. CS = 0.
- 2) Llevar la dirección de la palabra de memoria, cuyo contenido se desea extraer.
- 3) Enviar la orden de lectura.

Lo que permitirá que el contenido de la posición de memoria direccionada pase al registro de información, y de éste al *bus* de datos.

El ciclo de lectura de una memoria es mostrado en la figura II.5.

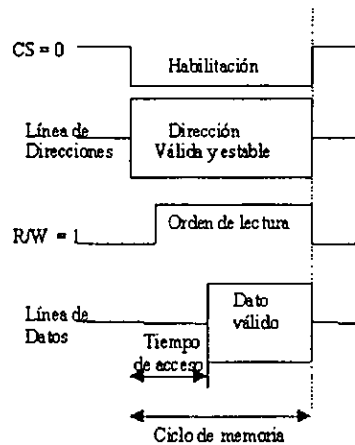


Figura II.5 Diagrama de tiempo del proceso de lectura de una memoria.

En caso de escribir, el procedimiento es como sigue:

- 1) Poner sobre el *bus* de direcciones la configuración binaria que identifica a la posición de memoria donde deseamos almacenar el dato.
- 2) Llevar al *bus* de datos la información a escribir en la célula de memoria.
- 3) Autorizar la actuación de la unidad de memoria mediante $CS = 0$.
- 4) Mandar la orden de escritura $R/W = 0$.

Hecho lo anterior, la información presente en el *bus* de datos pasará al interior de la posición de memoria direccionada.

• Memorias ROM

Las memorias ROM son aquellas que permiten únicamente la operación de lectura. La información registrada en una ROM es permanente e inamovible. Su contenido se establece durante la fabricación del circuito integrado y no puede ser alterado después de dicha fabricación. Las unidades ROM contienen, por ejemplo: constantes fijas, instrucciones fijas de programas, o datos de conversión. Se emplean para almacenar programas de aplicación

específica, que permitirán ejecutar una tarea especializada. La figura II.6 muestra la estructura de una Memoria ROM.

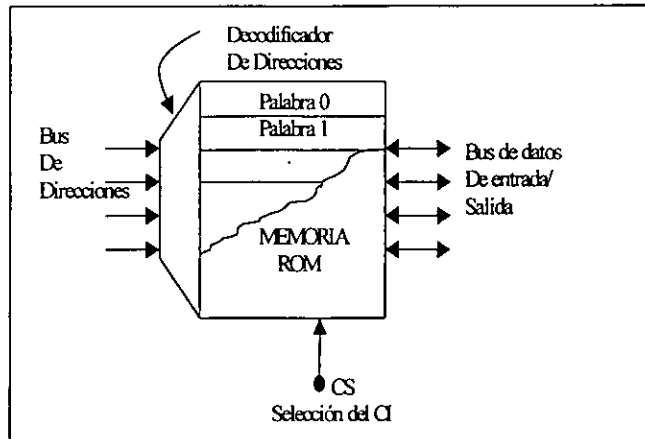


Figura II.6 Estructura de una Memoria de Sólo Lectura.

El alimentar datos a una ROM se llama programación de la ROM. Una vez que una ROM está programada, los datos pueden ser leídos en cualquier tiempo que sea necesario.

Los enlaces que conectan una ROM se programan por el fabricante para que se adapten a una aplicación en particular. Tales ROM se llaman programadas con máscara o ROM programadas por el cliente debido a la máscara fotográfica especial que se prepara por el fabricante para producir la ROM. El costo de tal máscara es relativamente elevado y sólo se justifica cuando se requiere una gran cantidad de ROM idénticas.

Al contrario que las RAM, las memorias de sólo lectura no necesitan la línea de control R/W, ya que la operación a realizar es única, y por lo tanto, basta con tener la entrada de selección del CI, CS.

Los pasos a realizar para efectuar una operación de lectura en una ROM serían los siguientes:

- 1) Direcccionar la posición de memoria cuyo contenido se va a leer.
- 2) Autorizar la habilitación de la ROM, CS = 0.

Una vez hecho estos, el dato pasará al registro de salida, y de éste al *bus* de datos. Para el caso que se tuviera la necesidad de modificar la información almacenada en este tipo de memorias, se han desarrollado otros tipos de ROM que prevén la posibilidad de reprogramación. Dentro de esta categoría se encuentran las llamadas memorias EPROM y EEPROM, las cuales pueden almacenar datos y mantenerlos aunque se corte la energía, el método para limpiarla es exponiéndola a la luz ultravioleta, con lo cual, después se puede reprogramar.

Otro tipo de memorias ROM son las PROM, o memorias programables de sólo lectura. Por medio del equipo de programación correspondiente, el usuario puede escribir en la PROM la información que desee; pero hay que tener en cuenta que este proceso es irreversible por que sólo permite una única grabación.

II.3 Comunicación RS-232

La *interface* RS-232 es un estándar desarrollado por Electronic Industries Association (EIA) en 1969, en colaboración de *Bell System*. La norma RS-232C define las señales y voltajes usados cuando existe comunicación entre una terminal de computadora, Equipos Terminales de Datos (ETD), y un módem, Equipos de Comunicación de Datos (ETCD). Cabe aclarar que aunque se hizo pensando en una *interface* que conectase terminales y módems, hoy en día se utiliza para unir dos dispositivos cualesquiera que requieran ser conectados.

El estándar RS-232 define las características eléctricas, la descripción funcional de los circuitos de intercambio, y una lista de aplicaciones estándar. La *interface* RS-232 está diseñada para velocidades hasta de 20 kbits/seg y distancias de cable de no más de 15m.

Distancias más largas son posibles si se usa cable tipo par-trenzado y una capacitancia de alrededor de 2500pF. Esta *interface* es una de las más usadas en la comunicación en equipo de cómputo. La *interface* RS-232 maneja niveles de voltaje de +3 a +15 VDC para un "0"

lógico y -3 a -15 VDC para un "1" lógico, con 25 V máximo, a circuito abierto. El estándar RS-232 puede acomodarse a diferentes tipos de circuitos de operación de datos, usando diferentes combinaciones de líneas. Por ejemplo: punto-a-punto, en un sistema dedicado requiere un número mínimo de líneas de control en la *interface*.

El tipo del conector físico no es especificado, pero por lo general, en la práctica se usa un conector DB25 (de 25 terminales, los cuales están definidos en la tabla II.4), aunque para conectar una PC a un módem se requieren normalmente un número menor. Las comunicaciones asíncronas requieren como máximo 9 o 12 conductores, y las comunicaciones síncronas requieren como máximo 12 o 16 conductores. Esta diferencia del número de conductores necesario es debida a las distintas características de operación del módem.

Cuando es usado un conector DB-25, por convención cada terminal tiene una asignación, la cual es mostrada en un conector en la figura II.7.

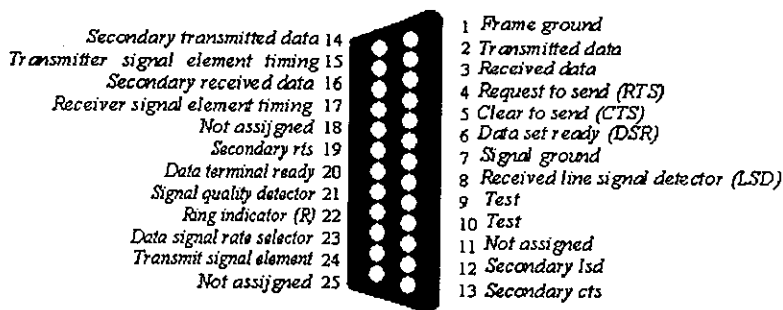


Figura II.7 Asignación de terminales para el conector DB-25.

TERMINAL	ASIGNACION	DESIGNACION EIA
01	<i>Frame ground</i>	AA
02	<i>Transmitted data</i>	BA
03	<i>Received data</i>	BB
04	<i>Request to send (RTS)</i>	CA
05	<i>Clear to send (CTS)</i>	CB
06	<i>Data set ready (DSR)</i>	CC
07	<i>Signal ground</i>	AB
08	<i>Received line signal detector (LSD)</i>	CF
09	<i>Test</i>	
10	<i>Test</i>	
11	<i>No asignado</i>	
12	<i>Secondary lsd</i>	SCF
13	<i>Secondary cts</i>	SCB
14	<i>Secondary transmitted data</i>	SCA
15	<i>Transmitter signal element timing (modem o terminal)</i>	DB
16	<i>Secondary received data</i>	SBB
17	<i>Receiver signal element timing</i>	DD
18	<i>No asignando</i>	
19	<i>Secondary rts</i>	SCA
20	<i>Data terminal ready</i>	CD
21	<i>Signal quality detector</i>	CG
22	<i>Ring indicator (R)</i>	CE
23	<i>Data signal rate selector</i>	CA/CI
24	<i>Transmit signal element (terminal to modem)</i>	DA
25	<i>No asignado</i>	

Tabla II.4 Designación de terminales para el estándar RS-232.

En el flujo de datos entre la PC (ETD) y el módem (ETCD) se utilizan tres terminales principales. La terminal 2, que es por donde circulan los datos del ordenador al módem (transmisión); la terminal 3, que es por donde circulan los datos del módem a la PC (recepción); y la terminal 7, que es la señal de tierra a la que están referidas las tensiones de las terminales anteriores.

Con tres terminales se puede implementar la comunicación RS-232 entre PC's o algún dispositivo serial. La figura II.8 muestra la conexión física de estas tres terminales.

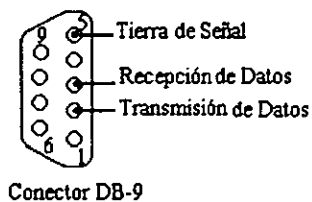


Figura II.8 Mínimo de terminales para poder establecer comunicación RS-232.

Después de haber visto los conceptos más importantes para el desarrollo teórico de este sistema, pasaremos a ver las características más importantes de los elementos que conforman el sistema en el siguiente capítulo.

CAPITULO III

Descripción
general
del
sistema

Para tener una visión del problema al cual nos enfrentamos, en este capítulo, damos un planteamiento del mismo y su posible solución, así como las características de cada una de los elementos que se buscan integrar para este sistema.

III.1 Planteamiento del problema y propuesta de solución

Como parte del desarrollo del vehículo eléctrico del Instituto de Ingeniería de la UNAM, se hace necesario conocer los parámetros que son más representativos en el funcionamiento del mismo, con el fin de observar el comportamiento de cada uno de ellos.

Para esto los parámetros que resultan de mayor importancia son los siguientes:

- Voltaje individual de cada una de las baterías que conforman el banco de alimentación del microbús eléctrico. En total está constituido por 52 baterías conectadas en forma paralelo serie.

- Voltaje total del banco.

- Corriente que consume el vehículo durante su funcionamiento.

- Velocidad.

- Temperatura del banco de baterías.

Todo esto con el objeto de evaluar rendimiento, desempeño y tener un criterio para establecer si se deben de realizar algunas modificaciones a ciertas partes del vehículo, además de operarlo dentro de un cierto límite en el que no sufra daños parciales o permanentes en partes tan importantes del sistema como son las baterías con que se alimenta.

El problema podría resolverse utilizando algunos dispositivos de medición tradicional como galvanómetros, adecuándolos de tal forma que en sus carátulas se desplegaran los valores en todo instante y sólo cuando hubiera la necesidad de hacer un muestreo alguna de las

variables en especial, se realizarían las lecturas y las anotaciones correspondientes. Esto puede resultar fácil si sólo se requiere de leer una carátula, pero si se tratan de cuatro o cinco el problema se dificulta, pues debe prestarse la atención a varias carátulas y anotar los valores medidos, con la posibilidad de realizar una lectura errónea o equivocarnos al escribir el valor.

Además, si se requiere hacer el muestreo en intervalos regulares de tiempo, esta actividad se tendría que realizar por dos o más personas. Otra opción es integrar un sistema que sea lo más autónomo y automático posible, con el fin de evitar los inconvenientes antes mencionados, en el cual la única interacción del usuario sea a través de un *software* que despliegue los datos procesados y las condiciones de operación del vehículo.

Con base en lo descrito anteriormente, se propone implementar un sistema que requiera del mínimo de intervención humana, que cumpla con las características de confiabilidad de los datos al momento del muestreo, que pueda almacenarlos, y transmitirlos a una computadora, para su posterior procesamiento.

Para llevar al cabo el sistema descrito anteriormente se propone:

- Localizar de que partes del microbús eléctrico de donde van a ser obtenidas las señales y las fuentes alimentación para el sistema a implementar.
- El utilizar un convertidor analógico digital para las señales provenientes de los acondicionadores y transductores.
- Multiplexar las diferentes señales, provenientes del acondicionamiento, almacenarlas dentro una memoria con una capacidad predeterminada y por un tiempo seleccionado. Adecuar los tiempos de adquisición dentro de las necesidades requeridas.
- Manejar todos estos parámetros a través de un módulo de control. Este módulo deberá tener periféricos que permitan la interconexión con los otros módulos del sistema.

- Transmitir los datos almacenados en cada uno de los muestreos a una computadora personal.
- Procesar los datos sabiendo el escalamiento y acondicionamiento que se hizo de las señales, empleando las ecuaciones necesarias para desplegar los valores reales que se midieron.
- Presentar tablas y gráficas correspondientes al análisis de la información.

Se muestra parte del banco de baterías en la figura III.1.

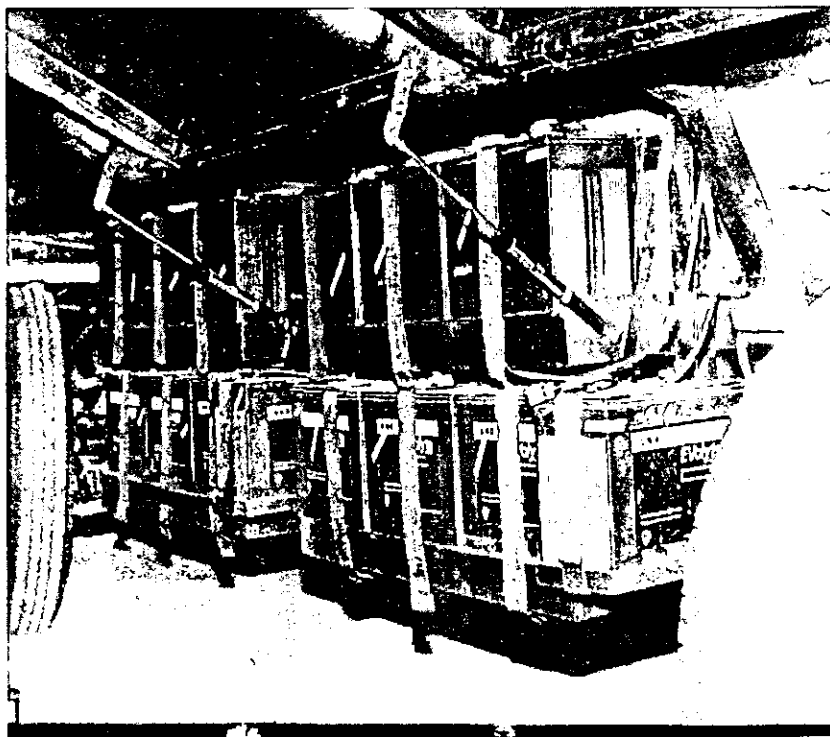


Figura III.1 Vista lateral izquierdo del banco de baterías.

Un diagrama a bloques de un sistema que pueda cumplir con las características antes mencionadas puede ser el presentado en la figura III.2.

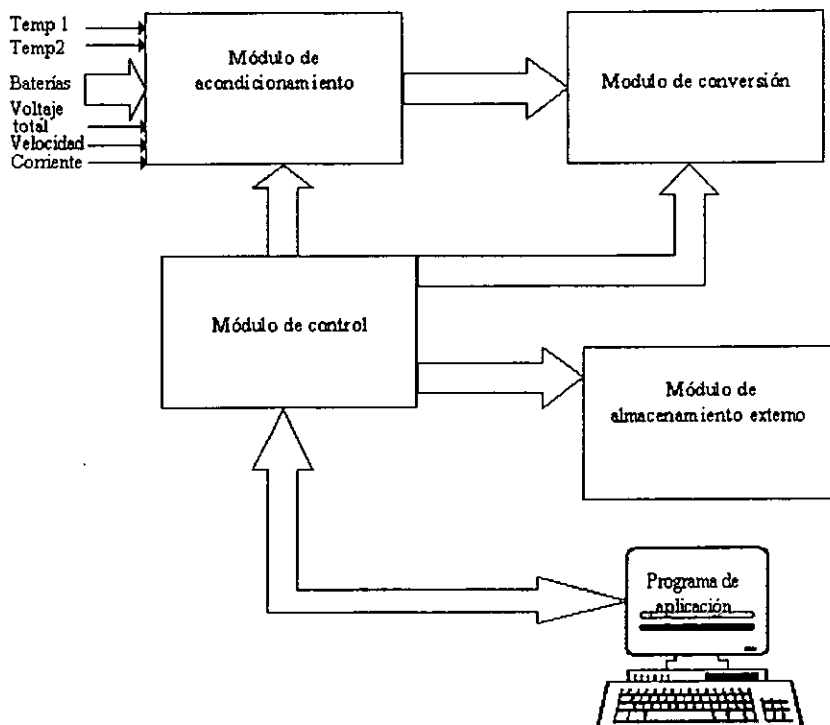


Figura III.2 Diagrama a bloques de un sistema de monitoreo.

Como podemos observar se plantea la organización del sistema en cuatro bloques:

- Módulo de control.
- Módulo de conversión analógico-digital
- Módulo de acondicionamiento.
- Módulo de almacenamiento externo.

◆ **Módulo de control**

Tendrá como objetivo organizar y controlar las acciones de los demás módulos. Aquí se llevará a cabo la inicialización y establecimiento de los parámetros de funcionamiento, permitirá la comunicación entre tarjetas, establecerá el manejo de datos y su almacenamiento. Este módulo es el corazón del sistema.

◆ **Módulo de conversión analógico-digital**

Este módulo se encargará de la conversión de las señales analógicas para su posterior manejo y almacenamiento. Aquí se lleva a cabo el multiplexaje y la selección de ganancia para la conversión.

◆ **Módulo de acondicionamiento**

En esta parte del sistema se llevará a cabo los ajustes y adaptaciones necesarias a las señales eléctricas provenientes de los transductores para que puedan ser utilizadas por los circuitos que alimentan.

◆ **Módulo de almacenamiento externo**

En este módulo tendremos la opción de acrecentar la capacidad de almacenamiento del sistema, así como otra opción para poder recuperar los datos almacenados.

Estos son a grandes rasgos los módulos en los que se ha dividido el sistema, a continuación describiremos las características de los componentes que se eligieron para integrar éstos módulos.

III.2 Tarjeta de control miniatura programable (*Little-G*)

La tarjeta de control miniatura programable *Little-G* es la empleada para este sistema. Por lo general esta tarjeta es usada en la industria como tarjeta de control o adquisidora de datos.

La tarjeta presenta las siguientes características:

- Tamaño 115 X 106 mm.
- 12 entradas digitales.
- 14 salidas digitales.
- Comunicación serial RS-485 y RS232.
- Reloj a 9.216 MHz.
- Puerto de expansión (*PLCBus*).

En la figura III.3 se muestra un diagrama físico de la tarjeta.

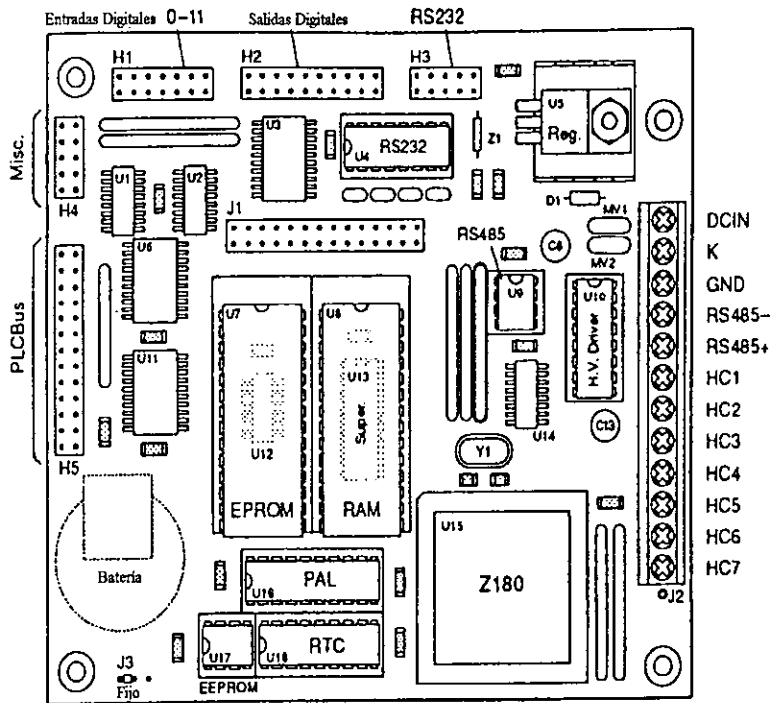


Figura III.3 Tarjeta *Little-G*.

En la figura III.4 se muestra en forma de bloques cómo está constituida la tarjeta *Little-G*.

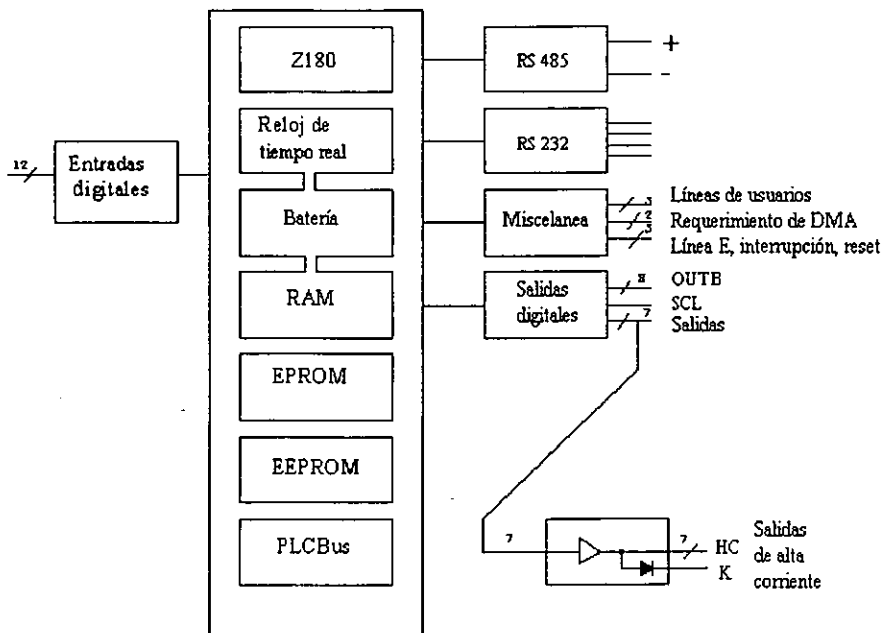


Figura III.4 Diagrama a bloques de la tarjeta *Little-G*.

En el diagrama de bloques se puede observar que la tarjeta cuenta con una memoria RAM, cuya capacidad puede llegar a ser de hasta 512 Kbytes, además contiene memoria EPROM o *flash* EPROM de hasta 512 Kbytes, *timers* programables, DMA, EEPROM, *watchdog*, un reloj de tiempo real, un sistema de interrupción por fallo de energía y una batería que respalda la memoria RAM y al reloj de tiempo real.

La potencia máxima de disipación, a 50 °C, es de 5W con el disipador estándar. La máxima temperatura de operación es de 70 °C, pero no se puede exponer la tarjeta a operar por grandes períodos a esta temperatura.

III.2.1 Descripción de sus interfaces

La *Little-G* contiene entradas digitales, salidas digitales, comunicación serial y salidas digitales de alta corriente, que están distribuidas como lo muestra la tabla III.1 (ver figura III.3).

Conector	Descripción
H1	12 entradas digitales
H2	14 salidas digitales (EN485, SCL)
H3	Gomunicación RS-232
H4	Conectores varios
H5	Puerto de expansión <i>PLCBus</i>
J2	Alimentación, RS-485 y 7 canales de salida de alta corriente.

Tabla III.1 Distribución de las distintas entradas y salidas de la tarjeta.

- **H1**, entradas digitales.

El conector H1 acepta 12 entradas digitales, IN00-IN11. Las 12 entradas digitales aceptan niveles de voltaje *CMOS* (de 2.5 V a 5 V). Estas se dividen en dos grupos: 1-8 y 9-12. Cada entrada tiene una resistencia de *pull-up* de 4.7k.

- **H2**, salidas digitales.

El conector H2 provee dos señales diferentes (EN485 y SCL) y dos grupos de salidas digitales, *OUT1-OUT6* y *OUTB1-OUTB8*. La terminal EN485 puede ser utilizada también como una salida digital *OUT7* si no se usa para la transmisión RS485. La línea SCL puede ser usada como una 8ª salida si no hay una EEPROM en la tarjeta.

Los grupos de salidas están divididas de la siguiente manera:

<i>OUTB1-OUTB8</i>	Puerto paralelo de 8 bits
<i>OUT1-OUT6</i> , EN485, y SCL	8 líneas seleccionadas individualmente

Siete de las salidas digitales, *OUT1-OUT7*, manejan alta corriente a través de (HC1-HC7), que pueden usarse para manejar relés, solenoides, lámparas, etc.

- **H3, comunicación.**

El conector H3 provee un canal de 5 líneas o dos canales de 3 líneas para la comunicación RS-232, dependiendo de la configuración de las terminales 18, 20, 22 y 23 al 26, utilizando *jumpers* del conector J1. Cuando el conector H3 está siendo usado como puerto de programación, éste no puede ser usado como puerto de comunicación por el programa de aplicación.

- **H4, conectores varios.**

El conector H4 provee de tres líneas de usuarios, dos líneas de requerimiento del DMA, y tres señales diversas. Estas señales son para desarrollo de tarjetas de expansión diseñadas por los usuarios. Complementadas con las líneas A0-A3, estas tres líneas determinan si se desea leer o escribir hasta 48 dispositivos en el sistema. Las líneas son excluyentes, ya que sólo puede ser usada una a la vez. Cada línea identifica un grupo de registros.

- **J2, alimentación, RS485, y 7 canales de salida de alta corriente.**

La configuración de las terminales es la siguiente, dos de ellas son dedicadas para la alimentación de la tarjeta. Siete para las salidas digitales de alta corriente (HC1-HC7), cada salida de alta corriente incluye un diodo de protección para evitar regresos de corriente que dañen la fuente. Los diodos usan una terminal común llamada K. Además de dos terminales para la comunicación RS485, dependiendo de la configuración de las terminales 23 y 26 en el conector J1.

El conector H5 (*PLC Bus*) lo describiremos en un apartado especial.

III.2.2 Comunicación Serial

La tarjeta contiene dos puertos seriales que soportan la comunicación en rangos de baudaje desde 300 a 115,200 bits/seg. Los manejadores pueden configurarse como dos puertos de 3 líneas o uno de 5 líneas (con RTS y CTS) y un puerto RS485 con comunicación *half-duplex* con par de cable trenzado hasta 3 km; según se muestran en las figuras III.5 y III.6 respectivamente.

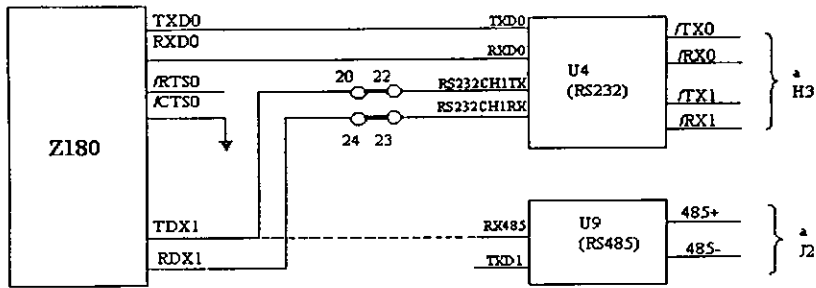


Figura III.5 Configuración para dos canales RS232 con tres hilos.

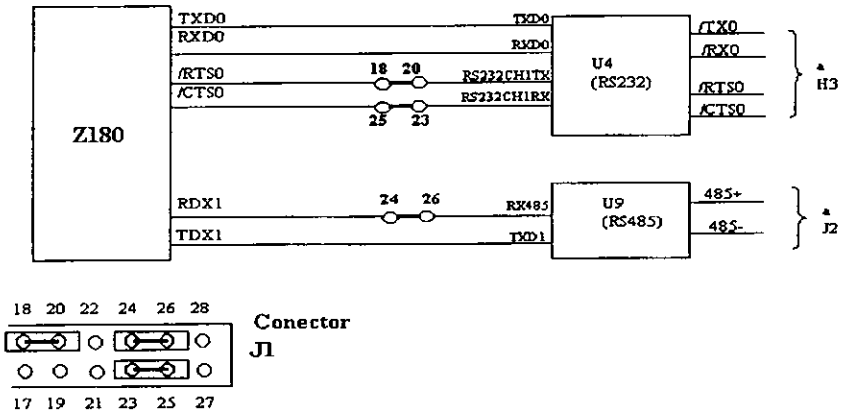


Figura III.6 Configuración de la comunicación RS-232 con 5 hilos y un canal RS-485 half-duplex.

La tarjeta tiene librerías de comunicación serial RS232 para los puertos 0 y 1, y para la tarjeta RS232 que actúa a través del *PLCBus*.

Estas librerías incluyen:

- Inicialización de los puertos seriales.
- Monitoreo y lectura para un *buffer* de recepción circular.
- Monitoreo y escritura para un *buffer* de transmisión circular.
- Una opción *echo*.
- Líneas de control CTS y RTS.
- XMODEM, protocolo para comunicación vía módem.
- Una opción de módem.

La comunicación serial es hecha fácilmente con el respaldo de una rutina de interrupción que actualiza la recepción y transmisión del *buffer*.

Para la recepción cada vez que el *buffer* recibe un carácter, la rutina de interrupción lo coloca en el *buffer* de recepción. Si el puerto serial no está transmitiendo en ese momento, la función de escritura inicializa automáticamente la transmisión.

En el caso de que sea transmitido el último carácter del *buffer*, la transmisión es terminada. El programa puede leer un dato cada vez o una cadena de caracteres terminada con un carácter especial.

Si la opción *echo* es activada durante el inicio del puerto serial (con *Dinit_z0*, *Dinit_z1*, o *Dinit_uart*), cualquier carácter recibido es automáticamente retransmitido. Esta característica es ideal para usar una terminal *dumb* y también para chequeo de los caracteres recibidos. El puerto 0 del Z180 está constituido por *hardware* para tener la CTS poleada en bajo por el dispositivo RS232 con que está comunicado. Una tarjeta de expansión RS232, casi siempre, puede activar o desactivar el efecto de la línea CTS. El puerto 1 no soporta las líneas CTS/RTS. Las líneas CTS está puesta a tierra cuando no se usa.

III.2.3 EL *PLCBus*

El conector *H5* es conocido como *PLCBus*. Es un puerto de expansión para el sistema de la tarjeta *Little-G*, con la finalidad de comunicarse con otras tarjetas que manejan el mismo

bus. La estructura del *PLCBus* se presenta en la figura III.7, posee un conector con cable plano de 26 hilos con la siguiente asignación:

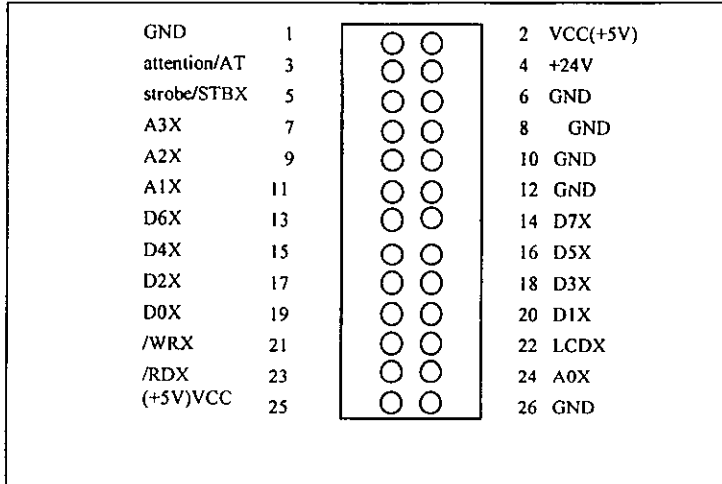


Figura III.7 Distribución de terminales del *PLCBus*.

No todos los terminales son usados por el *PLCBus*. El *PLCBus* consiste básicamente de 13 terminales: /AT, /STB, A1X, A2X, A3X, DOX-D7X, cuya función es especificada en la tabla III.2.

Línea	Descripción
/AT	Línea de atención
/STBX	<i>Strobe</i> negativo
A1X, A2X, A3X	Líneas de selección para la operación del <i>bus</i>
DOX-D7X	Línea de datos bidireccional. D0X-D3X son usadas para las operaciones de 4 bits

Tabla III.2 Asignación de terminales del *PLCBus*.

El *PLCBus* tiene diferentes formas de uso. Se puede usar como un *bus* de 4-bits (usando D0X-D3X) o un *bus* de 8-bits (usando D0X-D7X). El uso del *bus* depende de la codificación de la dirección que se ponga en el mismo. Algunas tarjetas lo requieren de 4-bits otras de 8-bits.

Un dispositivo en el *PLCBus* generalmente tiene uno o más registros de control y uno o más registro de datos. Todas las direcciones en el *bus* son únicas. Cada dispositivo reconoce su propia dirección (por la codificación de la PAL, los *jumpers* y otras formas lógicas y responden cuando su dirección es requerida). Puede tener un registro de expansión de 4-bits para su dirección.

Con varios modos de direccionamiento y registros de expansión, en teoría, se puede construir un *bus* con millones de dispositivos o periféricos con millones de direcciones. Los dispositivos de la empresa *Z-World* típicamente tienen direcciones particulares. Para trabajar con el *bus*, obviamente, todos los componentes son absolutamente distinguibles.

La forma de seleccionar cada tarjeta accesoria es la siguiente: primero elegimos un dispositivo, para poner su dirección en el *bus*. Después, se manda un dato al dispositivo o se lee un dato desde el dispositivo. Si este tiene un registro de expansión, se necesita colocar el valor.

Cuando se usa el *bus* de 4-bits, una dirección consiste de tres *nibbles* de 4-bits cada uno, y son puestos en sucesión en el *bus*, teniendo así una dirección de 12-bits. Cuando se usa el *bus* de 8-bits, una dirección consiste de tres palabras de 8-bits en sucesión, teniendo así una dirección de 24-bits.

Para poner una dirección en el *PLCBus*, escribir o leer de dato a un dispositivo, se lee o escribe en un registro de *I/O* de un mapa de memoria, o se llama una función que lo haga. Estos son los ocho registros correspondientes a los modos determinados por las líneas del *bus* A1X, A2X y A3X y en la tabla III.3 se muestra sus distintas codificaciones, así como las funciones que realizan.

Registro	Dirección	A3	A2	A1	Significado
BUSRD0	C0	0	0	0	Lee un dato, 1° nibble
BUSRD1	C2	0	0	1	Lee un dato, 2° nibble
BUSRD2	C4	0	1	0	Lee un dato, 3° nibble
BUSRESET	C6	0	1	1	Lee el registro de reset del PLCBus
BUSADR0	C8	1	0	0	1° nibble de dirección o byte
BUSADR1	CA	1	0	1	2° nibble de dirección o byte
BUSADR2	CC	1	1	0	3° nibble de dirección o byte
BUSWR	CE	1	1	1	Escribe un dato

Tabla III.3 Codificación de las líneas del PLCBus.

Escribiendo o leyendo uno de estos registros se activa la línea de *strobe* del PLCBus.

III.2.4 Como utilizar la tarjeta

Para empezar a trabajar con la tarjeta *Little-G* hay que seguir los siguientes pasos:

- 1) Colocar las terminales de configuración o *jumpers* en una posición válida.
- 2) Conectar los cables de comunicación.
- 3) Conectar la alimentación.
- 4) Ejecutar el *software* de desarrollo *Dynamic C*.

La tarjeta se puede programar conectándose al puerto serial de una computadora de las siguientes formas:

- Conectando la tarjeta *Z-World's Z485* en la PC; esta tarjeta es del tipo ISA y evita el trabajo de velocidad de compilación y carga. La Z485 tiene un puerto de 9 terminales y uno de 25 con conectores tipo DB. La tarjeta *Little-G* se conecta a cualquiera de esos puertos.
- Conectando la *Little-G* a la tarjeta de expansión RS-232, la cual se conecta por medio del puerto de comunicaciones PLCBus. La tarjeta de expansión RS-232 puede conectarse

directamente a la PC o a la tarjeta Z485. Llevando a cabo una conexión directa a algún puerto serial de la computadora.

- Por medio del conector H3, usando un cable de nueve terminales, y conectandoló directamente al puerto serial de la PC.

La velocidad de la comunicación serial para programación de la tarjeta es normalmente de 19,200 baud, pero puede llegar hasta los 57,600 bauds. Mientras un programa está en desarrollo, la tarjeta normalmente permanece conectado a la computadora personal. Una vez terminado el desarrollo, el programa recién escrito puede residir en los siguientes lugares:

- En la RAM respaldado por la batería.
- Escrito en una ROM independiente y sustituyendo a la ROM estándar de la tarjeta.

Al proporcionarle energía a la *Little-G*, ésta consulta los *jumpers* (J1) y los contenidos de la EPROM para determinar su modo de operación. Los modos de operación están expuestos en la tabla III.4.

- Correr el programa en EPROM. Si éste es un programa válido, éste corre sin tomar en cuenta los *jumpers*. Mientras se está desarrollando el *software*, la tarjeta puede usar normalmente el monitor de la EPROM.
- Correr el programa guardado en la memoria RAM.

Conectando las terminales 1-2 y 3-4 del *jumper* J1 son seleccionados los modos de operación. Ver tabla III.4.

Terminales 1-2	Terminales 3-4	Significado
No conectado	No conectado	Corre desde la RAM
Conectado	No conectado	Programando a 9,600 baud
No conectado	Conectado	En este modo verifica el valor almacenado en la localidad 1 de la EPROM
Conectado	Conectado	Programando a 19,200 baud (default)

Tabla III.4 Configuración de operación de la tarjeta.

Además se puede instalar memoria SRAM y EPROM desde 32K hasta 512K, y memorias *flash* EPROM desde 64K hasta 256K. Estas pueden tener entre 28 o 32 terminales. Ajustando las terminales 5-16 del *jumper* J1 se determina el tamaño de la memoria instalada. Las terminales 12, 14 y 16 especifican el tamaño de la memoria RAM (figura III.8).

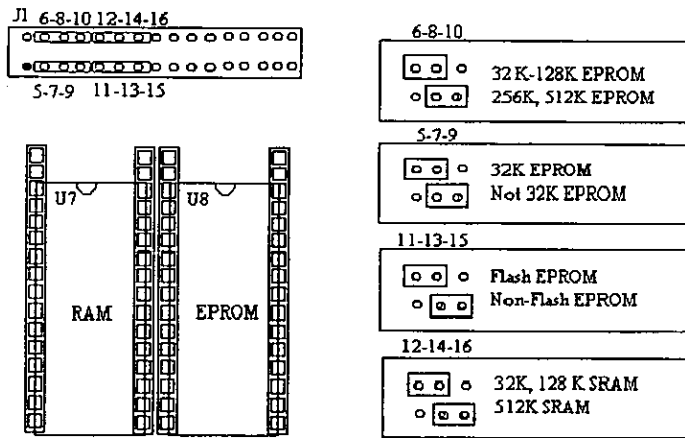


Figura III.8 Disposición de las terminales de configuración para seleccionar la memoria.

De fábrica la tarjeta contiene 32K de SRAM y 32K de EPROM.

La EEPROM maneja constantes que se aplican en la tarjeta *Little-G* estándar, las cuales se muestran en la tabla III.5.

DIRECCION	DEFINICION
0	Modo de arranque. Si 1, entra en el modo de programación; si 8, ejecuta el programa que está cargado.
1	Programando a una razón de 19200 bauds. Los valores de fábrica están puestos a 48 que significa una razón de 57.6K bauds. Esta localización se aplica únicamente si el <i>jumper J1</i> no tiene conectado las terminales 1-2, y conectadas las terminales 3-4.
0x100	Esta constante refiere información de formato BCD para la hora y fecha con la siguiente distribución: Segundos, minutos, horas, día, mes, año.
0x108	Proporciona la velocidad del microprocesador en unidades de 1200 Hz (16-bits). Para un cristal de 9.216 MHz este valor es 7680.
0x16C	Este coeficiente nos determina la relación que existe entre el valor del cristal del microprocesador, la velocidad del microprocesador, para establecer la velocidad del reloj de tiempo real. El valor nominal es 107,374,182 cuando está a 1/40 de segundo el reloj del microprocesador sobre la escala donde 2^{32} está 1 segundo. Este requiere 4 bytes de la EEPROM, guardando primero el byte menos significativo.

Tabla III.5 Constantes almacenadas en la memoria EPROM.

III.3 Tarjeta de conversión analógico-digital (TCAD)

La tarjeta de conversión analógico-digital (TCAD) de 12-bits (EXP-A/D-12), de ganancia programable, puede ser usada en conjunto con alguna tarjeta de control miniatura de la empresa fabricante. Esta puede conectarse directamente al puerto *PLCBus* de una tarjeta de control *Little-G*.

La TCAD, mostrada en la figura III.9, tiene ocho canales de entrada. Estos canales pueden ser leídos como ocho señales diferenciales con una ganancia seleccionada por *software*, o como dieciséis señales simples con ganancia unitaria. Las ganancias de 2, 6, 22, 42, 102, y 202x, así como la ganancia unitaria, pueden ser seleccionadas. Cada tarjeta TCAD es calibrada de fábrica. Los coeficientes de compensación para todos los canales del TCAD son almacenadas en una tabla en la memoria EEPROM. La EEPROM puede ser leída y escrita por un programa de aplicación. Normalmente esto no se hace dado que la EEPROM tiene grabadas constantes para la conversión. Se puede, sin embargo, volver a calibrar la tarjeta TCAD en un momento dado.

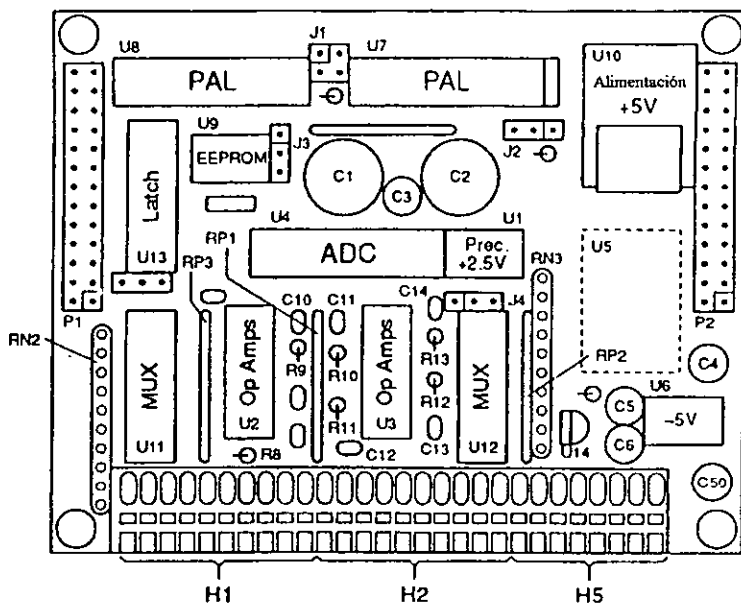


Figura III.9 Diagrama físico de la TCAD.

La tarjeta TCAD (ver figura III.9) cuenta con una fuente de alimentación principal LM340T5 (U10) que le provee +5V a 1A. Los CI's 7662D (U6) y 74AX (U14) proveen un voltaje negativo V_- , regulado a -5V hasta 500mA. El LT1019 (U1) provee una referencia de precisión de 2.5 V (V_{ref+}) hasta 10 mA.

Las señales analógicas entran a la tarjeta por terminales en dos conectores tipo *Wago*, H1 y H2. Un tercer conector, H5, provee voltajes de excitación para *strain gauges* y otras entradas analógicas para sensores. Note que H1, H2 y H5 están dispuestos en línea horizontal como si fueran un solo conector. Las señales de entrada de la tarjeta (AN_x+ y AN_- , $0 \leq x \leq 7$) son enviadas al CI LTC1294 (U4), convertidor analógico-digital (A/D), por dos multiplexores DG509ACJ (U11 y U12) y diversos amplificadores diferenciales LT1014 (U2 y U3).

La tarjeta usa cuatro *jumper*s, J1 hasta J4. Cuando se coloca J1, se acopla con el código de la PAL CI PALCE16V8 (U7), determinando la dirección de la tarjeta en el *PLCBus*. El *jumper* J2 especifica la fuente de alimentación para la circuitería analógica.

Cuando usamos la tarjeta *Little-G*, la alimentación principal la obtiene del U10, tomando 24V+ del *PLCBus* y convierte éste en 5V+. Conectando las terminales 2-3 en el J2 envía 5V+ desde U10 a los circuitos analógicos.

El *jumper* J3 es un interruptor de protección contra escritura para la EEPROM 24C04A (U9). Si se está usando las características de fábrica con la EEPROM 24C01A de 128-byte, la cual no tiene protección contra escritura, J3 no tiene efecto. Si se está usando la 24C04A de 512-byte (con una opción especial instalada de fabrica) conectando las terminales 1-2 de J3 protegemos la mitad alta de la EEPROM. Sin embargo, como opción de fabrica, la tarjeta trae ya conectados las terminales 2-3.

El *jumper* J4 proporciona un voltaje de referencia al CI A/D U4. De fábrica se conectan las terminales 2-3, enviando de U1 una referencia de precisión de 2.5V de salida al chip A/D. Los ocho canales de entrada de la tarjeta (ANx+ y ANx-) son enviados al A/D CI U4 por dos multiplexores (MUX) y diferentes amplificadores diferenciales.

Ganancias de 2, 6, 22, 42, 102 y 202X, como también la ganancia unitaria, pueden ser seleccionados por *software*. Esto se puede comprender mejor en la figura III.10.

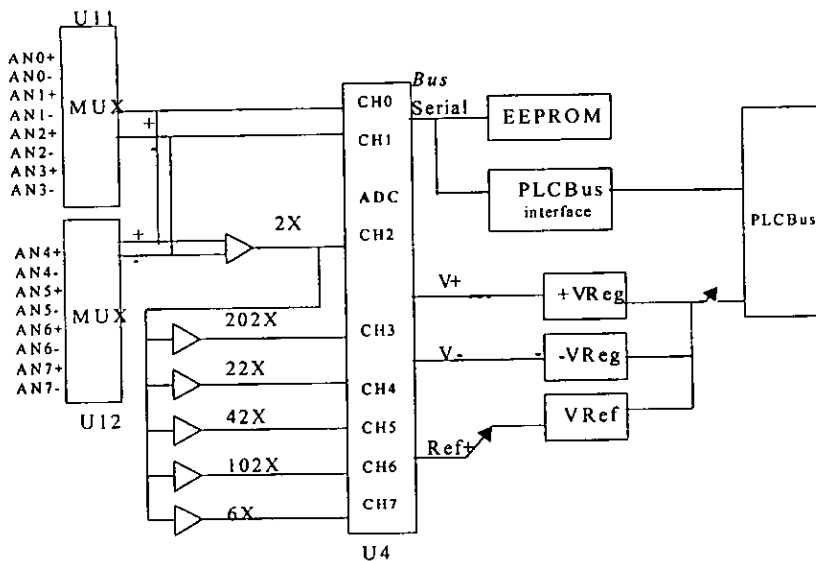


Figura III.10 Diagrama a bloques del circuito analógico.

El circuito multiplexor de la tarjeta abarca dos CI's U11 y U12. Únicamente uno de los dos puede ser usado a la vez. Las líneas de salida de los MUXs, IN+ e IN-, envían las líneas de señal de entrada seleccionadas directamente a las entradas CH0 y CH1 del A/D, proviendo ganancia unitaria. Ellos también pasan la señal a la entrada del amplificador diferencial de ganancia variable.

Las salidas desde el amplificador de varios estados (líneas G1-G6) son dirigidas a las entradas CH2-CH7 del CI A/D. La ganancia en cada línea es mostrada en los diagramas. El CI A/D puede ser programado para convertir una señal puesta en cualquiera de los ocho canales de entrada de dos modos distintos: unipolar o bipolar. Los rangos de los valores de entrada y salida son permisibles en los siguientes casos, tabla III.6.

MODO	RANGO DE ENTRADA	RANGO DE SALIDA
Bipolar	-2.5 V a +2.5V	-2048 a +2047
Unipolar	0 a +2.5V	0 a 4095

Tabla III.6 Modos de operación de entrada.

Si fuera necesario, se puede reducir un voltaje de entrada o el rango requerido con una cadena de resistencias. De otra manera, cuando un voltaje de entrada pasa el rango requerido, el CI reporta si esta en un máximo o un valor mínimo.

III.3.1 Como usar la tarjeta

Generalmente, así es como se usa una la TCAD:

- 1)Mandar un comando de *reset* al *PLCBus*.
- 2)Poner la dirección de la tarjeta en el *PLCBus*.
- 3)Leer uno de los canales de entrada, permitido por los tiempos del MUX.
- 4)Calcular el valor real, como temperatura o desplazamiento, desde los datos obtenidos.
- 5)Usar los datos para control de relevadores, interruptores, u otros dispositivos.

Una vez que fue leído un canal, se puede seguir leyendo repetidamente, tan rápido como 165 μ sec. por lectura (en un microcontrolador a 9 MHz). La dirección particular de 12 bits de la tarjeta está determinada por:

- La codificación de la PAL U7 instalada en la tarjeta.
- Por la terminal de configuración J1.

Dieciséis diferentes configuraciones de PAL's son disponibles y J1 puede ser colocado de cuatro diferentes formas, consiguiendo 65 direcciones únicas:

000p 10px ppRy, donde:

R = 0, para un registro S3A, 1 para un registro S3B

$y = 1$, donde las terminales 3-4 de J1 no son conectadas

$x = 1$, donde las terminales 1-2 de J1 no son conectadas

III.3.2 Modos de conversión de la TCAD

La TCAD lee la ganancia y la polaridad (unipolar o bipolar) deseada, seleccionando un "modo de conversión A/D" particular. Existen 18 modos diferentes en la tarjeta. En la siguiente tabla, tabla III.7, ANx es una de las ocho entradas multiplexadas, donde x va del rango de 0 a 7.

Modo	Ganancia	La TCAD lee...
0	1X	ANx+, modo bipolar
1	1X	ANx-, modo bipolar
2	2X	La diferencia entre ANx+ y ANx-, bipolar
3	202X	La diferencia entre ANx+ y ANx-, bipolar
4	22X	La diferencia entre ANx+ y ANx-, bipolar
5	42X	La diferencia entre ANx+ y ANx-, bipolar
6	102X	La diferencia entre ANx+ y ANx-, bipolar
7	6X	La diferencia entre ANx+ y ANx-, bipolar
8	1X	ANx+, modo unipolar
9	1X	ANx-, modo unipolar
10	2X	La diferencia entre ANx+ y ANx-, unipolar
11	202X	La diferencia entre ANx+ y ANx-, unipolar
12	22X	La diferencia entre ANx+ y ANx-, unipolar
13	42X	La diferencia entre ANx+ y ANx-, unipolar
14	102X	La diferencia entre ANx+ y ANx-, unipolar
15	6X	La diferencia entre ANx+ y ANx-, unipolar
16	1X	La diferencia entre IN+ y IN-, unipolar
17	1X	La diferencia entre IN+ y IN-, unipolar

Tabla III.7 Modos de conversión de la TCAD.

Cuando se lee una señal de entrada diferencial, se conecta el sensor a las terminales ANx+ y ANx-. En caso que sea una señal simple en los modos 2-7 o 10-17 de la TCAD, se conecta la fuente de voltaje a ANx+ y se conecta ANx- a tierra. Cuando se llega a leer una

señal simple en los modos 0, 1, 8 o 9 la tierra no es necesaria. Simplemente se conecta la fuente de voltaje a la terminal deseada ANx+ o ANx-.

III.3.3 Software para la TCAD

Cuando usamos la TCAD con la *Little-G*, podemos usar las funciones estándar del *PLCBus* que se encuentran en la librería *DRIVERS.LIB* y las funciones especiales en la librería *PLC_EXP.LIB*.

Al usar algunas de las funciones en *PLC_EXP.LIB*, se puede crear una "tabla de señal". Para crear una tabla de señal se declara un arreglo como sigue:

```
struct signal_rec adtab[size];
```

El nombre *adtab* debe ser global. La constante *size* representa el número de señales en la tabla, hay que poner en ella el valor que necesite cada aplicación.

Estas son algunas funciones específicas que se pueden emplear en el manejo de la tarjeta:

```
int add_sig_table (int entry, unit addr, byte in_ch, byte mode, byte comp)
```

<i>entry</i>	selecciona la entrada de la tabla
<i>addr</i>	dirección de 12-bits de la tarjeta en el <i>PLCBus</i>
<i>in_chi</i>	canal de entrada a leer, 0-7
<i>mode</i>	modo de conversión (ganancia y polaridad), 0-17
<i>comp</i>	1 = usa los coeficientes de compensación de la EEPROM 0 = sin compensación

La función regresa un código que indica si el proceso se realizó con éxito o si existió alguna falla:

- | | |
|----|---|
| 0 | La señal de fue ingresada satisfactoriamente a la tabla |
| -1 | Error de <i>hardware</i> de la EPROM |
| -2 | Error de protección contra escritura de la EEPROM |
| -3 | Parámetro invalido fue entregado |

int plcad_addr(int board)

Regresa 12-bits (*nibble-swapeados*) de la dirección para identificar a la tarjeta en el *bus* a cambio de una dirección lógica (0-63).

float ad_conv(int entry, int value)

Regresa el voltaje representado por el valor de 12-bits obtenido en la entrada *value*, asumiendo una escala de 2.5V. El término *entry* identifica la señal en la tabla.

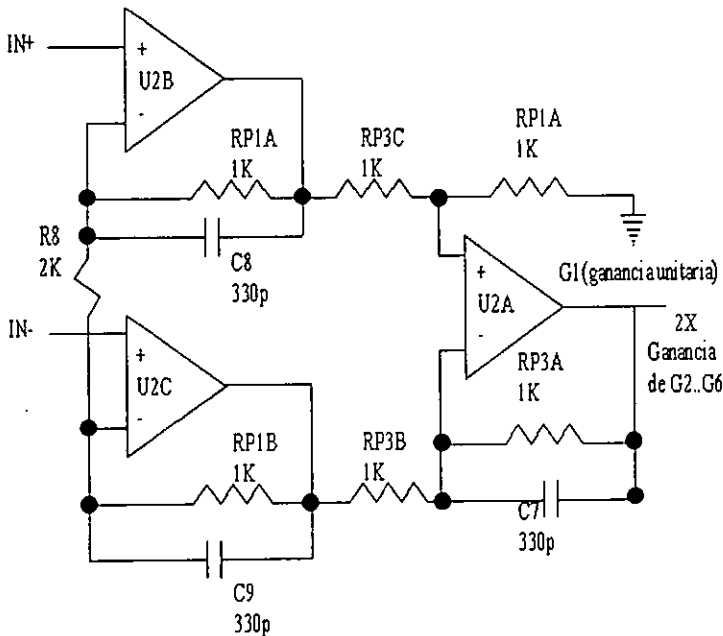
Int plad_rd12(int mode)

Regresa un valor de 12-bits leído por el CI A/D. El término *mode* especifica uno de los modos de conversión a usar.

Una forma de verificar el desempeño del convertidor es poner las señales de entrada a tierra para calcular la desviación estándar de los canales del TCAD. La prueba consiste de 1000 muestras por canal para apreciar el desempeño de las tarjetas típicas en modo unipolar y bipolar. Bajo estas condiciones, la desviación estándar de la baja ganancia del canal (ganancias de 1, 6, 22 y 42) fueron típicamente 0.1 – 0.4 (fuera de 2047) en modo bipolar. Para la alta ganancia de los canales (ganancias de 102 y 202), la desviación estándar fue típicamente de 0.6-0.8. Todos los canales fueron consistentemente debajo de 1.0, aunque la desviación estándar para el canal X202 que ocasionalmente su rango fue mayor que 0.95.

Predeciblemente, los números en modo unipolar fueron alrededor del doble que los adquiridos en modo bipolar. Las ganancias de bajo nivel fueron consistentemente debajo de 1.0, típicamente 0.2-0.9. La desviación estándar para las ganancias de alto nivel estuvieron en el rango de 1.2 a 1.9.

Para determinar la ganancia del convertidor la tarjeta cuenta con dos paquetes de cuatro amplificadores operacionales, U2 y U3, cada uno contiene cuatro amplificadores operacionales de precisión. Una alta impedancia de entrada, y ganancia ajustable a una entrada diferencial son obtenidas con tres de estos amplificadores, U2A, U2B y U2C. El diagrama del circuito analógico para la selección de la ganancia se muestra en la figura III.11.



$$\begin{aligned} \text{Ganancia} &= 1 + (RP1A + RP1B) / R8 \\ &= 1 + 2K / 2K = 2X \end{aligned}$$

Figura III.11 Circuito analógico de selección de ganancia.

III.4 Módulo de acondicionamiento

Para realizar las lecturas necesarias se utiliza un módulo de acondicionamiento de señales constituido por una serie de tarjetas desarrolladas en el Instituto de Ingeniería, las cuales se describen a continuación.

Este módulo se compone dos tipos de tarjetas: la tarjeta principal y las tarjetas adicionales. Estas tarjetas no son independientes, y su funcionamiento está ligado. En la figura III.12 se muestra un diagrama a bloques de cómo interactúan estas tarjetas.

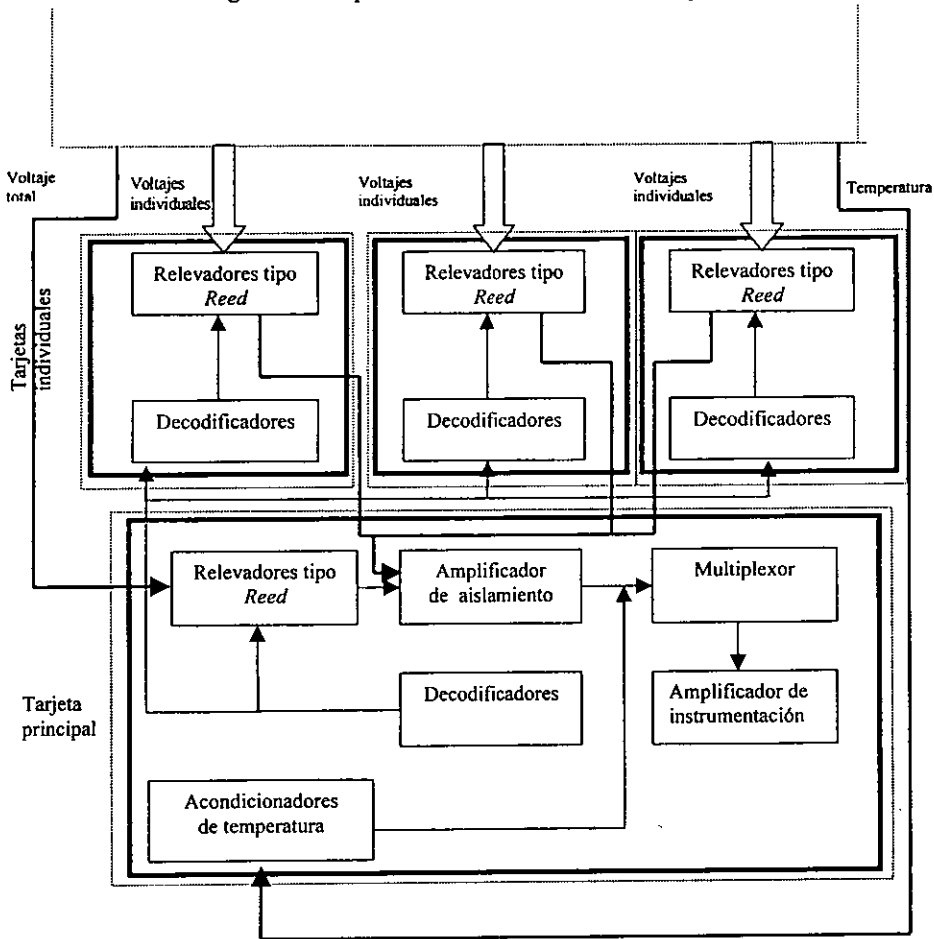


Figura III.12 Diagrama a bloques de la tarjeta del módulo de acondicionamiento.

La tarjeta principal cuenta con:

- Relevadores tipo *Reed*.

Estos relevadores son del tipo de estado sólido, son utilizados para tomar muestras del voltaje total del banco de baterías permitiendo tener un aislamiento eléctrico del sistema, cuando no se esté midiendo el voltaje total.

- Amplificador de aislamiento.

Este tipo de amplificadores permite el tener un aislamiento adicional al obtenido con los relevadores, además de presentar una gran flexibilidad de entrada, pues pueden implementarse diferentes filtros, sumar voltajes, manejar rangos de alto voltaje y corriente.

- Multiplexor.

Su entrada maneja voltajes diferenciales, una alta inmunidad al ruido y salida en forma diferencial.

- Amplificador de instrumentación.

Permite el manejo de señales diferenciales y por lo tanto presenta una alta inmunidad al ruido, pueden seleccionarse varias configuraciones de ganancia que van desde 1 hasta 1000 veces el voltaje de entrada. Cuenta con una entrada de offset que se suma a la señal de entrada.

Las tarjetas adicionales son idénticas y las constituyen los siguientes elementos:

- Relevadores del tipo *Reed*.

Como se mencionó anteriormente, este tipo de relevadores permiten un aislamiento eléctrico y en este caso son utilizados para tomar muestras de cada una de las baterías. Cada tarjeta cuenta con 17 relevadores.

- Decodificador.

Permite habilitar cada uno de los relevadores.

El módulo de acondicionamiento tiene la capacidad de adquirir muestras del voltaje total del banco de baterías, así como de cada una de las baterías individualmente, además de contar con dos sensores de temperatura para cada uno de los bancos.

III.5 Microcontrolador *COP8SAC*

El *COPSAC* OTP es un microcontrolador de la familia *COP8*, que usa un chip con arquitectura de 8 bits. Estos dispositivos son fabricados por *National Semiconductors* con EPROM en procesos de alta densidad, y ofrecen una gran cantidad de presentaciones, rangos de voltaje y de temperatura, para satisfacer la mayoría de las aplicaciones.

Sus características incluyen una memoria de 8 bits mapeada, un *timer/counter* de 16 bits con dos registros asociados que soportan tres modos de operación (*Processor Independent PWM generation*, *External Event Counter*, e *Input Capture capabilities*), dos modos de respaldo de energía HALT/IDLE con una multifuente *wakeup/interrupt* capacidad, un circuito oscilador RC, salidas de alta corriente, opción de seleccionar el *WATCHDOG*, configuraciones para el oscilador y un circuito de *reset*.

III.5.1 Características principales

Entre sus características principales tenemos:

- Bajo costo.
- Espacio para un programa con protección contra lectura/escritura.
- *Quiet Desing* (baja emisión de radiaciones).
- Entradas *wakeup* con opción a atender interrupciones (de 4 a 8).
- Oscilador seleccionable.
 - a) Cristal/Resonante con o sin circuito de polarización con resistencia.
 - b) Oscilador externo.
 - c) Oscilador interno con circuito RC.
- *Reset* interno seleccionable.
- *WATCHDOG* y Monitoreo Lógico del Reloj, seleccionable.

- Más de 12 terminales de salida de alta corriente.
- *Set* de instrucciones fácil de usar.
- 1 μ s en cada ciclo de instrucción.
- Ocho vectores para el Servicio de interrupción.
 - a) Interrupciones externas.
 - b) *Idle timer* T0.
 - c) *Timer* uno (con dos interrupciones).
 - d) Interface serial *MICROWIRE/PLUS*.
 - e) Multi entradas *wake up*.
 - f) Enmascaradas por *software*.
 - g) *VIS*, vector de interrupciones por *default*.
- *Stack Pointer* de 8 bit (RAM).
- Dos registros de 8 bits para apuntar indirectamente datos de la memoria.
- Manipulación directa de bits.
- Memoria mapeada I/O.
- Instrucciones aritméticas en BCD.
- Un *timer* de 16 bits con dos registros de 16 bits asociados.
 - a) Modo para Procesador independiente de PWM.
 - b) Modo de contador de eventos externos.
 - c) Modo de captura.
 - d) *Idle timer*.
 - e) Interface serial *MICROWIRE/PLUS* (compatible con SPI).
- Entrada/Salida seleccionables por *software* para cada terminal.
 - a) Salida tres estados.
 - b) Salida de *Push-pull*.
 - c) Entrada con *Weak-pull*.
 - d) Entradas de alta impedancia.
- Entradas *Schmitt trigger* en los puertos G y L.
- Eficiente selección del número de terminales.
- Baja corriente de drenaje (típicamente $< 4\mu$ A).
- Operación con una fuente simple (2.7 V a 5.5V).

- Rango de operación de temperatura 0°C a +70°C, -40°C a +85°C, y -40°C a 125°C.
- Distintos tipos de presentaciones DIP y PLCC
- Simulación en tiempo real y depuración de programas ofrecidos por el Sistema de Desarrollo *MetaLink*.

III.5.2 Arquitectura COP8SAC

La familia *COP* de *National*, cuenta con las características siguientes:

- Memoria RAM de 128 bytes para datos.
- Dos relojes-contadores de 16 bits y 12 bits, además de un *WATCHDOG*.
- Cuatro puertos bidireccionales de 8 bits de I/O y un puerto de salida de 8 bits.
- *Interface Serial MICROWIRE/PLUS* (SPI Compatibles).
- Multi-entradas *Wake-up* con opción de interrupciones.
- Cuenta hasta con 12 salidas de alta corriente.
- Baja emisión de radiación.
- Opciones programables de entrada de reloj.

Un diagrama a bloques de la arquitectura del *COP8SAC* es mostrado en la figura III.13.

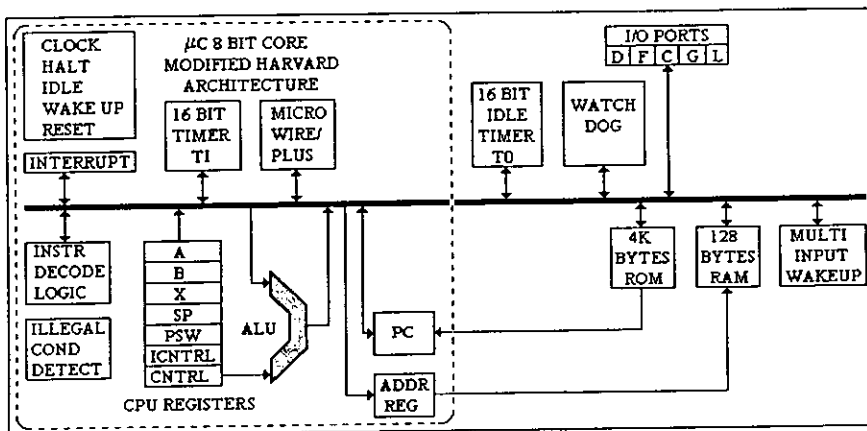


Figura III.13 Arquitectura del *COP8SAC*.

Este microcontrolador divide la memoria en espacio de datos y espacio de programa, esta separación lógica de memoria permite el acceso a la memoria de datos por direccionamiento de 8 bits, el cual nos facilita el almacenamiento y la manipulación en un CPU de 8 bits. La memoria de programa EPROM solamente se puede leer y no escribir, esta puede ser de hasta 4Kbytes.

La operación del microcontrolador requiere 5V de alimentación, y una señal de reloj la cual puede ser por un oscilador R/C interno, externo ó por un oscilador de cristal de cuarzo.

Cuenta con terminales bidireccionales las cuales pueden ser configuradas como línea de salida y como entrada con alta impedancia. También cuenta con salidas de alta corriente.

Los cuatro puertos bidireccionales de 8 bits de entrada y salida son los puertos C, G, L, y F, donde cada bit individual puede ser configurado independientemente como entrada (entradas tipo *Schmitt trigger* en los puertos L y G), o salidas tres estados mediante un control por *software* (tabla III.8). Cada puerto de I/O tiene asociado tres registros en el mapa de memoria de 8 bits, el registro de configuración, el registro de datos de entrada y el registro de dato de salida. La dirección en el mapa de memoria es también reservada para las líneas de entrada de cada puerto de I/O.

REGISTRO DE CONFIGURACION	REGISTRO DE DATO	PUERTO CONFIGURADO
0	0	Alta impedancia de entrada ó Salida de tres estados
0	1	Entrada <i>weak Pull-up</i>
1	0	Cero a la salida <i>Push-Pull</i>
1	1	Uno a la salida <i>Push-Pull</i>

Tabla III. 8 Registros de configuración.

El puerto G es de 8 bits. Las terminales G0, G2-G5 son líneas bidireccionales de I/O. La terminal G6 es siempre una entrada de alta impedancia de propósito general. Todas las terminales tienen entradas tipo *Schmitt Triggers*.

La terminal G1 está dedicada para la salida *WDOUT* del *WATCHDOG*, esta función especial tiene que ser activada por el registro *ECON*. Cuando no se habilita el *WATCHDOG* esta terminal es de propósito general y se utiliza como *I/O*.

La terminal G7 es puesto en uno o en cero dependiendo del tipo de oscilador que se seleccione. Si se selecciona el oscilador con cristal, la terminal G7 sirve de salida para el reloj *CKO*. Con la opción de oscilador *R/C* interno o externo, la terminal G7 es una entrada de alta impedancia de propósito general. Haciendo uso de sus registros asociados, cada una de las 5 terminales de *I/O* (G0, G2-G5), las cuales pueden ser configuradas individualmente mediante un control por *software*.

El puerto F es de 8 bits de *I/O*, mediante su registro de configuración se pueden declarar las terminales como entradas o como salidas. El puerto D es de 8 bits, sus salidas se establecen en alto cuando el *reset* se va a un bajo, este puerto es exclusivamente de salida y es de alta corriente.

◆ Registros de la CPU

La CPU puede hacer sumas, restas y operaciones lógicas de 8 bits en una sola instrucción. Hay seis registros en la CPU, y son los siguientes:

- El registro A es un acumulador de 8 bits.
- El registro PC es el contador de programa de 15 bits.
 - PU es la parte alta del contador de programa y es de 7 bits.
 - PL es la parte baja del contador de programa y es de 8 bits.
- El registro B es el apuntador de direcciones de la RAM, el cual puede incrementarse o decrementarse automáticamente.
- El registro X se puede alternar con el apuntador de direcciones de la RAM, el cual puede incrementarse o decrementarse automáticamente.
- El *stack pointer* (SP) es de 8 bits, el cual cada que ocurre un salto directo a una subrutina interrumpe en la memoria RAM y guarda la última dirección del programa, a la cual se debe de regresar una vez que haya terminado la subrutina. Con un *reset* el SP puede ser inicializado en la dirección 02F ó 06F de la memoria RAM.

Todos los registro de la CPU están en el mapa de memoria con excepción del acumulador A y el contador de programa PC.

◆ Memoria de Programa

La memoria de programa consiste en 1024, 2048 o 4096 bytes de EPROM. Estos bytes pueden tener instrucciones de programa o datos de algunas constantes. La memoria de programa es direccionada por medio del contador de programa (PC). La tabla III.9 muestra los diferentes tamaños de memoria disponibles de acuerdo a la serie del dispositivo.

DISPOSITIVO	MEMORIA DE PROGRMA(BYTES)	MEMORIA DE DATOS (BYTES)	MEMORIA PARA EL USUARIO (BYTES)
COP8SAA7	1024	64	8
COP8SAB7	2048	128	8
COP8SAC7	4096	128	8

Tabla III.9 Muestra diferentes tamaños de memoria.

◆ Memoria de Datos

El espacio de direcciones de la memoria de datos incluida en la RAM, contiene registros para datos, registros de I/O, registro de control, registro del *MICROWIRE/PLUS*, registro de corrimiento, y de varios registros, y de registros asociados con los temporizadores (con excepción del temporizador IDLE). La memoria de datos es direccionada directamente por alguna instrucción o indirectamente por los registros B y X, y por medio del apuntador de programa (SP).

◆ Memoria de usuario

Son 8 bytes de espacio en la EPROM, los cuales no se protegen contra lectura cuando se activa el bit de seguridad en el registro ECON. Estos 8 bytes están fuera del rango normal de direccionamiento de la memoria EPROM, y no pueden ser accedados por *software*. Son usados típicamente para definir números de series, códigos, información de derechos de autor, versión de *software* y otros.

◆ Circuito oscilador

Existen cuatro opciones disponibles de osciladores: Cristal oscilador con o sin resistencia en el circuito integrado, arreglo R/C de resistencia y capacitor externo, y un oscilador externo. Estas distintas opciones pueden ser habilitadas mediante el registro ECON, como se muestra claramente en la tabla III.10.

ECON4	ECON3	OPCIONES DEL OSCILADOR
0	0	Oscilador Externo
1	0	Cristal Oscilador sin resistencia
0	1	R/C Oscilador
1	1	Cristal Oscilador con resistencia

Tabla III.10 Opciones del Oscilador.

• Cristal Oscilador

La opción del cristal oscilador, con o sin resistencia, puede ser habilitada si en el bit 4 del registro ECON se coloca un 1. La terminal del microcontrolador que se utiliza como entrada del reloj es la CKI, mientras que la terminal G7/CKO es la salida que se utiliza como alimentación del cristal. La resistencia en el circuito integrado puede ser habilitada si en el bit 3 del registro ECON se coloca un 1, con la opción del cristal oscilador habilitado. El valor de la resistencia R2 está en un rango de entre 0.5 MΩ y 2 MΩ (típicamente de 1 MΩ). La resistencia R1 únicamente se utiliza cuando la resistencia del circuito integrado es deshabilitada por el usuario tabla III.11 y figura III.14.

R1 (KΩ)	R2 (MΩ)	C1 (pF)	C2 (pF)	CKI FREQ. (MHZ)
0	1	100	30	15
0	1	45	32	10
0	1	32	30 - 36	4
5.6	1	30	100 - 156	0.455

Tabla III.11 Configuración del Cristal Oscilador.

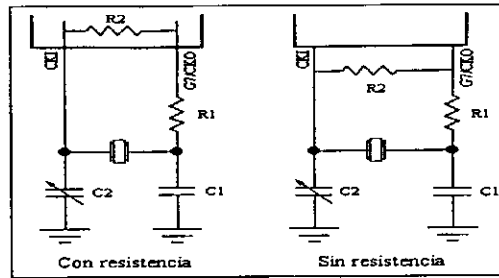


Figura III.14 Oscilador de Cristal.

• Oscilador Externo

El modo del oscilador externo puede ser habilitado si el bit 3 y el bit 4 tienen un cero en el registro ECON. Por la terminal CKI se puede introducir una señal de reloj, con una frecuencia y un nivel determinado. Cuando se utiliza la opción de oscilador externo, la terminal G7/CKO está disponible para un propósito general. En la figura III.15 se muestra este oscilador externo.

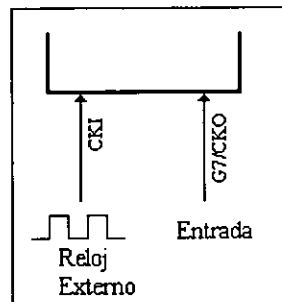


Figura III.15 Oscilador Externo.

• Oscilador R/C

El modo del oscilador R/C puede ser habilitado si el bit 3 tiene un 1 y el bit 4 tiene un 0 del registro ECON. En el modo de oscilador R/C, la terminal CKI está desconectada, mientras que la terminal G7/CKO es una entrada de propósito general o para el control del HALT.

La máxima frecuencia del oscilador R/C es de 5 MHz \pm 35 % para una alimentación que puede estar entre 4.5 V y 5.5 V, con un rango de temperatura entre - 40 °C hasta + 85 °C. Para frecuencias mayores la terminal CKI debe desconectarse. Para frecuencias menores debe conectarse un capacitor externo entre la terminal CKI y tierra (ver tabla III.12 y figura III.16).

CAPACITOR EXTERNO (pF)	FREQ DE OSCILACIÓN R/C (MHZ)	CICLO DE INSTRUCCIÓN (μ S)
0	5	2.0
9	4	2.5
52	2	5.0
150	1	10
TBD	32 kHz	312.5

Tabla III.12 Configuración del oscilador R/C.

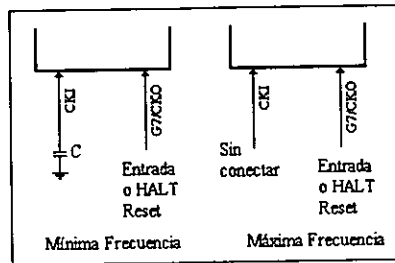


Figura III.16 Oscilador R/C.

III.6 Thin Card Random Acces Memory (TCRAM)

En este tipo de memorias se requiere de menos tiempo para ser escritas o borradas en comparación con las memorias EPROM (*Erasable Programable ROM*). Basada en el mismo proceso que las memorias EPROM, la tecnología de las memorias *Thin Card* es muy parecida a aquellas en su confiabilidad, su no-volatilidad (esto es debido a una batería interna que mantiene actualizada la información) y su densidad.

Los registros de comandos de las memorias *Thin Card* permiten que sean compatibles con los microprocesadores y de esta forma se les pueda borrar, programar o realizar en ellas otras operaciones, dentro de un sistema digital. El diseñar sistemas con memorias *Thin Card* es lo mismo o muy semejante a hacerlo con cualquier otra tecnología.

Se requiere asegurar la alimentación permanente de este tipo de memorias ya que sólo así se garantiza que los datos se mantengan en ella. Las tarjetas *Thin Card* están diseñadas bajo las normas de la *PCMCIA* (*Personal Computer Memory Card International Association*, Asociación Internacional de Tarjetas de Memoria para Computadoras Personales").

La norma especifica el aspecto físico, el eléctrico, la información sobre su estructura y las características sobre el formato de los datos. La superficie de este tipo de tarjetas es de 85.6 mm por 54.0 mm con una altura de 3.3 mm.

Estas tarjetas reciben el nombre de *Thin Card* (TC), Tarjeta Delgada, porque son similares en área a una tarjeta telefónica.

El número de terminales es de 68 e incluye 26 líneas de direcciones usadas para acceder directamente 64 Mbytes. Las tarjetas TC están disponibles con capacidades de 1 a 32 Mbytes con expectativas de crecimiento, haciéndose cada vez más competitivas con los manejadores de disco (*drives*).

Las tarjetas TC tienen la ventaja de poder manejarse tanto en diferentes sistemas con microprocesadores como en una PC, en esta última utilizando un *drive* especial para estas tarjetas. Al conectar el *drive* a la PC se tienen dos opciones, la primera es darle formato a la tarjeta como si fuera un disco y usarla como tal. La segunda opción es el hacer uso de una programación (*software*) exclusiva para las tarjetas. Ambas opciones permiten que la tarjeta sea escrita y borrada por medio de la PC. El *software* con que cuenta nos permite manipular la información contenida en ella. Este *software* se llama *CardTalk* e incluye comandos que nos permiten dar formato a la tarjeta *tcformat*, transferir su información a un archivo de texto *tcread*, borrar información *tceraser*, etc.

El hecho de que estas tarjetas sean portátiles permite que se les den diversas aplicaciones, pues se puede manejar la misma información en un sistema independiente con microprocesador e interactuar con la PC, sin que estén necesariamente conectados el sistema y la PC.

Para que la información contenida en la tarjeta TC no sea alterada, es necesario mantener en forma constante la alimentación de la misma. El diseño físico de las tarjetas permite contener una batería intercambiable dentro de ellas, con el objeto de poderlas transportar y mantener su información.

En la tabla III.13 se muestran las características eléctricas de una tarjeta *Thin Card*.

<i>Características Eléctricas</i>	
Tipo de dispositivo: <i>CMOS-SRAM</i>	
Voltaje de polarización: 3 a 5 V	
Densidad: 1 megabyte x 8 bits	
Tiempo de acceso: 200 ns	
Consumo de potencia	
Activo:	120 mA
En espera:	1.5 mA
Temperatura de operación: de 0 a 60 C	
Temperatura de almacenamiento: de -20 a 60 C	

Tabla III.13 Características eléctricas de la tarjeta de memoria.

La tabla III.14 nos muestra la distribución y función de los terminales de control de la tarjeta *Thin Card*.

Terminales	Función
9	HABILITACION DE SALIDA(L)
15	HABILITACION DE ESCRITURA (L)
16	HABILITACION DE ESCRITURA (L)
7	HABILITACION DE ENTRADAS (D0-D7)
42	HABILITACION DE ENTRADAS (D8-D15)
68	TIERRA (GND)
33	PROTECCION CONTRA ESCRITURA (L)

Tabla III.14 Señales de control de la memoria.

La tabla III.15 nos muestra la distribución de las terminales de la tarjeta *Thin Card* y su designación.

Terminales		Terminales		Terminales		Terminales	
Número	Nombre	Número	Nombre	Número	Nombre	Número	Nombre
1	GND	18	Vprom	35	GND	52	Vprom
2	D3	19	A16	36	CD1	53	A22
3	D4	20	A15	37	D11	54	A23
4	D5	21	A12	38	D12	55	A24
5	D6	22	A7	39	D13	56	A25
6	D7	23	A6	40	D14	57	RFU
7	CE1	24	A5	41	D15	58	RFU
8	A10	25	A4	42	CE2	59	RFU
9	OE	26	A3	43	RFSH	60	RFU
10	A11	27	A2	44	RFU	61	REG
11	A9	28	A1	45	RFU	62	BUD2
12	A8	29	AO	46	A17	63	BUD1
13	A13	30	DO	47	A18	64	D8
14	A14	31	D1	48	A19	65	D9
15	WE/PGM	32	D2	49	A20	66	D10
16	RDY/BS	33	WP	50	A21	67	CD2
17	VCC	34	GND	51	VCC	68	GND

Tabla III.15 Asignación de terminales de la memoria.

Con esto concluimos la descripción general del sistema destacando las principales características de cada uno de los elementos que componen al sistema.

En el siguiente capítulo mencionaremos las formas en que interactúan cada uno de los módulos mencionados y la forma en que se integraron.

CAPITULO IV

Diseño
del
prototipo
del
sistema

En este capítulo presentamos como se realizó la integración de todos los módulos del Sistema de Adquisición de Datos del Minibús Eléctrico (SADME), las especificaciones y restricciones que se presentaron. Además del desarrollo de los programas para las interacciones entre cada uno de estos módulos, así como con el usuario final.

IV.1 Especificaciones para el diseño del sistema

Los requerimientos que demanda el sistema ya descrito son básicamente:

- Un sistema de monitoreo y adquisición de variables para un vehículo eléctrico.
- Capacidad adquirir 32 variables por muestreo distribuidas de la siguiente manera:
 - Voltaje individual de 26 baterías.
 - Voltaje total del banco.
 - Temperaturas de los bancos de baterías (2).
 - Velocidad del Minibús Eléctrico.
 - Corrientes (2).
- Capacidad de comunicación vía RS-232 con una computadora personal.
- Capacidad de almacenamiento del sistema.

Para llevar a cabo lo descrito anteriormente, el sistema se conformó de cuatro partes fundamentales:

- Tarjeta del módulo de acondicionamiento (TMA).
- Tarjeta del módulo de conversión analógico-digital (TMCAD).
- Tarjeta del módulo de control (*Little-G*).
- Tarjeta del módulo externo de almacenamiento(TME).

De cada uno de los módulos mencionados se expondrá sus características de funcionamiento, la manera en que se acoplaron entre ellos y que limitantes presentan.

Es importante explicar el cableado en el banco de baterías para realizar el muestreo de cada una de ellas, así como lo referente a los sensores de temperatura, y la comunicación entre los módulos. Para entender la integración del sistema se presenta en el siguiente apartado una descripción de la conformación del banco de baterías del Minibús Eléctrico.

IV.2 Conexión de las baterías del minibús eléctrico

El banco de baterías que proporcionan energía al motor está constituido de 52 baterías; este banco a su vez se agrupa en banco derecho e izquierdo, cada uno con 26 baterías y a su vez se subdividen en parte superior e inferior para ambos agrupamientos. Las baterías de ambos agrupamientos se encuentran conectados en arreglo paralelo-serie, por lo que finalmente se tienen que realizar la medición de 26 voltajes, es decir, trece voltajes por agrupación. En la figura IV.1 se muestra el diagrama de conexión de las baterías.

El voltaje individual de cada batería es de 12 V, por lo que en teoría se tendrá un voltaje total del arreglo de 312 V. En el diagrama se puede observar que las baterías marcadas con el número 26 representan el nivel más negativo en el banco, y las etiquetadas con el número 1 representan la parte más positiva.

Además se observan en la figura IV.1 las abreviaciones Cn, indican el lugar en donde fueron instaladas las terminales, con el propósito de tender un cableado que permita realizar los muestreos.

Para realizar el cableado se utilizaron zapatas redondas de 6.35 mm, como terminales directos a los bornes de las baterías y cable del número 20. Que se soldaron a conectores hembras del tipo CPC de 16 canales para cada subdivisión del banco. En la parte central de cada agrupamiento del banco, se encuentra la contra del CPC de 16 canales, en el que se tiene la conexión de otro cable que llega a la parte trasera el minibús, que es el lugar en donde se encuentra el sistema de adquisición (ver figura IV.1).

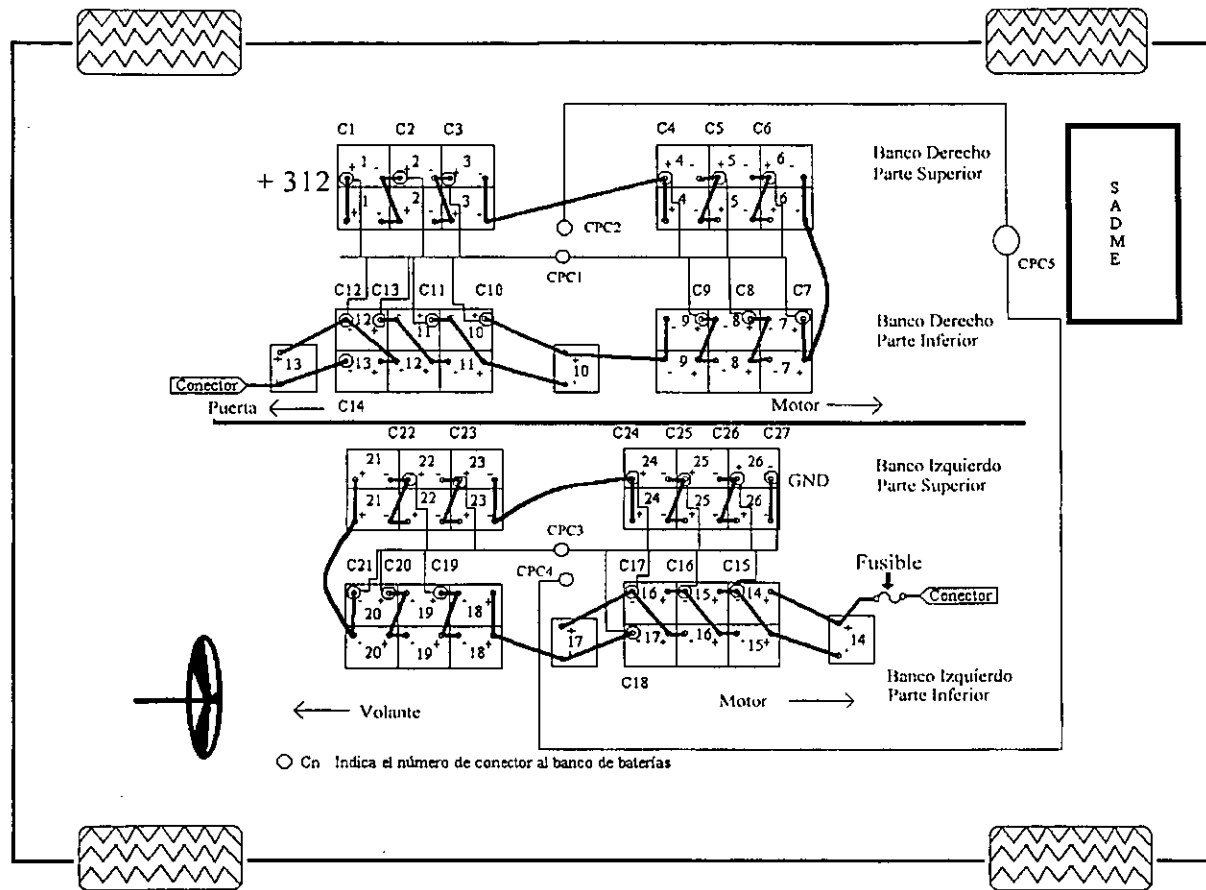


Figura IV.1 Conexión del banco de baterías del minibús eléctrico.

En la figura, las siglas CPC1 y CPC3 indican los conectores hembra de 16 terminales, las siglas CPC2 y CPC4 indican los conectores macho de 16 terminales. Las siglas CPC5 indica el conector de 37 terminales, de las cuales sólo se utilizan 27.

Los bancos derecho e izquierdo se interconectan a través de un fusible por razones de seguridad, ver figura IV.1.

El voltaje total del banco de baterías es tomado de los cables 1 y 26 que son, como se mencionó, los extremos del banco. Por lo que se aprovecha el mismo cableado para realizar la medición del voltaje total. En cada agrupación se encuentra un sensor de temperatura, cada sensor presenta un cableado similar desde la parte central del banco hasta la parte posterior del minibús.

En la figura IV.2, se presenta la distribución de las terminales en los conectores de 16 terminales CPC1 y CPC3.

En las tablas IV.1 y IV.2 se presentan la conexión de cada una de las terminales con los conectores CPC1, CPC2, CPC3, CPC4, CPC5. Cabe señalar que los números de las segundas columnas corresponden a los conectores del CPC de 37 terminales.

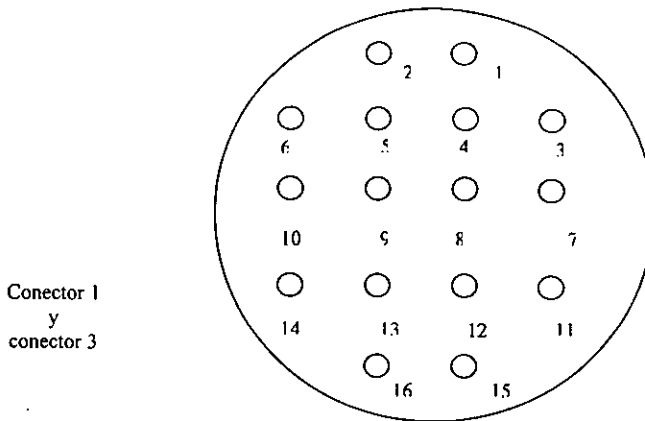


Figura IV.2 Conector para cada uno de las subdivisiones.

Terminales en el banco de baterías	Terminales de los conectores CPC1 y CPC2	Terminales del conector CPC5	Código de colores entre (CPC2) y (CPC5)
1	1	1	Blanco
2	2	2	Gris-blanco
3	3	3	Rojo-blanco
4	4	4	Gris
5	5	5	Violeta-blanco
6	6	6	Azul-Blanco
7	7	7	Negro
8	8	8	Naranja
9	9	9	Café
10	10	10	Verde-blanco
11	11	11	Rojo
12	12	12	Verde
13	13	13	Amarillo-blanco
14	14	14	Morado
-	15	Sin conexión	Sin conexión
-	16	Sin conexión	Sin conexión

Tabla IV.1 Correspondencia de terminal conector para la parte derecha del banco.

Terminales en el banco de baterías	Terminales de los conectores CPC3 y CPC4	Terminales del conector CPC5	Código de colores entre (CPC4) y (CPC5)
15	1	15	Blanco
16	2	16	Amarillo
17	3	17	Rojo-blanco
18	4	18	Blanco-café
19	5	19	Negro
20	6	20	Azul
21	7	21	Verde-negro
22	8	22	Verde
23	9	23	Verde-blanco
24	10	24	Blanco-verde
25	11	25	Rojo-negro
26	12	26	Azul-blanco
27	13	27	Negro-amarillo
-	14	Sin conexión	Sin conexión
-	15	Sin conexión	Sin conexión
-	16	Sin conexión	Sin conexión

Tabla IV.2 Correspondencia de terminal conector para la parte izquierda del banco.

IV.3 Tarjeta del módulo de acondicionamiento (TMA)

El módulo de acondicionamiento, como lo mencionamos en el capítulo anterior, consta de cinco tarjetas. Cuatro (denominadas tarjetas adicionales) están constituidas por relevadores del tipo *Reed*, los que se encuentran distribuidos en todas las tarjetas, siendo 17 para cada tarjeta. Teniendo como propósito la medición individual del voltaje de cada batería.

Estos relevadores son accionados por un decodificador, 74HC154 de 4 líneas a 16 salidas. Para la medición del voltaje de cada batería se activan 2 relevadores a la vez (ver figura IV.3), lo que constituye un primer paso, y por la manera en que se encuentran conectados los relevadores al amplificador de aislamiento (AD202) se obtiene una lectura de voltaje positivo. Segundo paso, se apaga el primer relevador que se encendió quedando el segundo en funcionamiento además de accionar un tercero, teniendo un voltaje negativo.

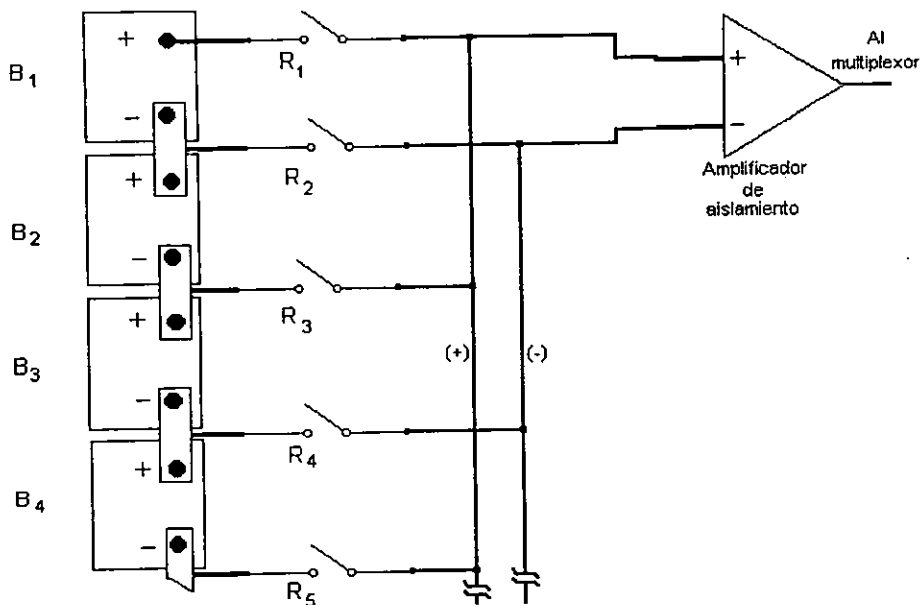


Figura IV.3 Secuencia de encendido de los relevadores.

El tercer relevador permanece en su estado anterior y el cuarto es habilitado lo que lleva a una lectura de voltaje positivo, considerándose el tercer paso. La figura anterior ejemplifica de forma esquemática dicho procedimiento. Después se continúa de forma secuencial hasta concluir el muestreo de las 26 pares de baterías, de donde se observa que se tienen 13 voltajes positivos y 13 voltajes negativos.

Las cuatro tarjetas adicionales tienen capacidad para 16 baterías por tarjeta, pero en el momento actual sólo se utilizan una y media tarjeta de las cuatro. Debido a que sólo se hacen muestreos de 26 voltajes de baterías.

En la tabla siguiente, IV.3, se muestra la secuencia que debe seguir el decodificador para habilitar cada par de relevadores de cada una de las tarjetas adicionales.

D3	D2	D1	D0	Habilita
0	0	0	0	Relevadores 1 y 2
0	0	0	1	Relevadores 2 y 3
0	0	1	0	Relevadores 3 y 4
0	0	1	1	Relevadores 4 y 5
0	1	0	0	Relevadores 5 y 6
0	1	0	1	Relevadores 6 y 7
0	1	1	0	Relevadores 7 y 8
0	1	1	1	Relevadores 8 y 9
1	0	0	0	Relevadores 9 y 10
1	0	0	1	Relevadores 10 y 11
1	0	1	0	Relevadores 11 y 12
1	0	1	1	Relevadores 12 y 13
1	1	0	0	Relevadores 13 y 14
1	1	0	1	Relevadores 14 y 15
1	1	1	0	Relevadores 15 y 16
1	1	1	1	Relevadores 16 y 17

Tabla IV.3 Codificación para la habilitación de los relevadores de una tarjeta.

De donde D3, D2, D1, D0 son los bits que habilitan a cada uno de los relevadores, esta secuencia es similar para cada una de las cuatro tarjetas. En la figura IV.4 se aprecia un esquema de la interacción relevadores decodificadores para cualquier tarjeta adicional.

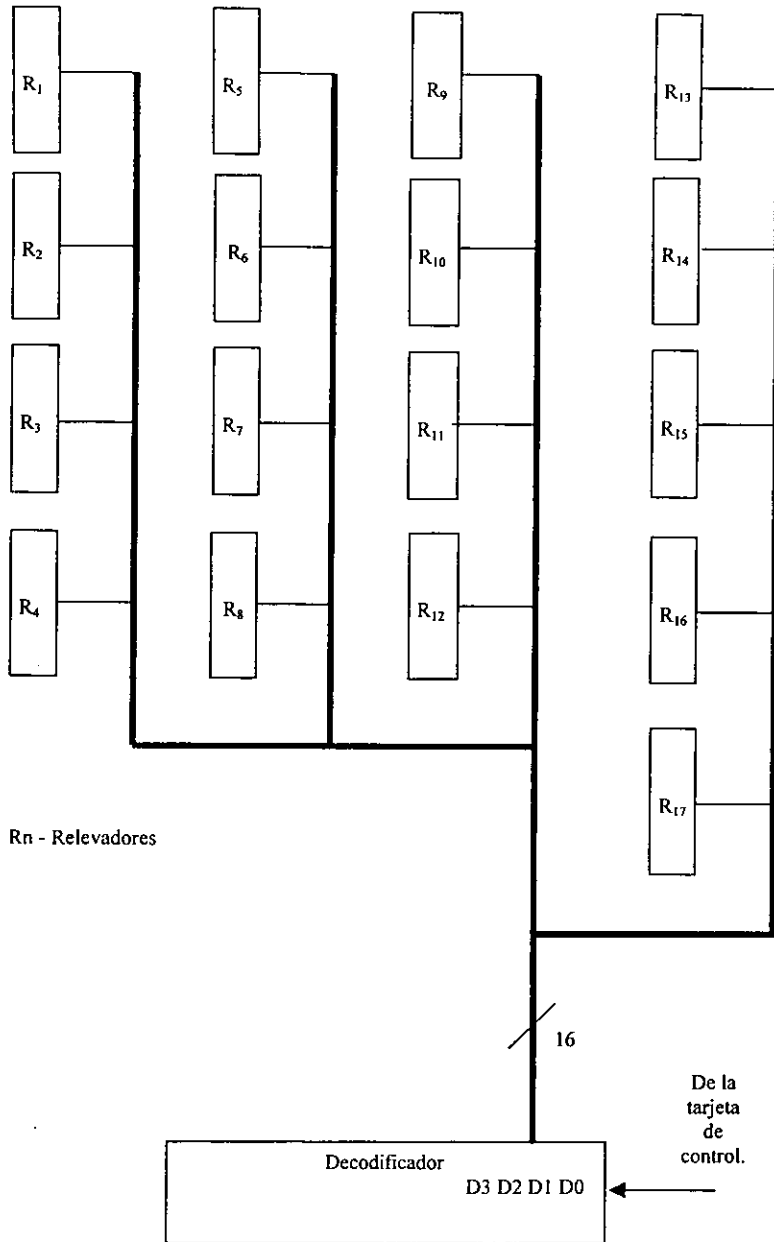


Figura IV.4 Control de los relevadores.

El voltaje de las baterías es escalado, utilizando resistencias, que se encuentran antes de los relevadores, es decir, primero pasa el voltaje por las resistencias de escalamiento y después por los relevadores. Lo que permite manejar niveles de voltaje de +2.5 V a -2.5 V.

En la tarjeta principal del módulo se encuentra otros dos pares de relevadores del tipo *Reed* los cuales son utilizados de la siguiente forma:

Un par de ellos sirve para dar paso a la medición del banco total de baterías, y el otro par se utiliza para direccionar los voltajes provenientes de las tarjetas adicionales. Posteriormente, ya sea el voltaje total, o los voltajes individuales, pasan a través del amplificador de aislamiento, con el fin de evitar ruido y de aislar eléctricamente todo el sistema restante.

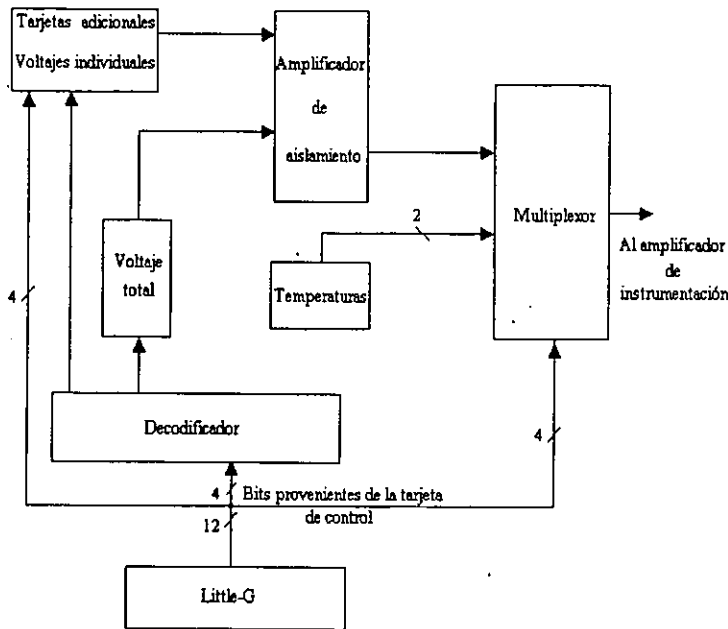
Para realizar el control de lo señalado anteriormente, se utiliza un decodificador de 3x8 74LS138, el que permite seleccionar la tarjeta a utilizar y que par de relevadores de la tarjeta principal accionar.

La tabla IV.4 muestra la codificación que se debe seguir para habilitar las tarjetas.

D6	D5	D4	EN2	Habilita
0	0	0	1	Tarjeta 1
0	0	1	1	Tarjeta 2
0	1	0	1	Tarjeta 3
0	1	1	1	Tarjeta 4
1	0	0	1	Voltaje Total

Tabla IV.4 Codificación para elegir las tarjetas.

Enseguida se encuentra un circuito multiplexor (ADG527A), el cual es utilizado en la conmutación entre las lecturas de voltajes de las baterías y los provenientes de cada sensor de temperatura (LM35). En la figura IV.5 se señala de forma esquemáticamente la operación de esta parte del módulo. La tabla IV.5 muestra la codificación que se debe de seguir para multiplexar cada uno de los canales de interés.



Bits provenientes de la tarjeta de control

Figura IV.5 Control de la tarjeta principal y adicionales.

AD2	AD1	AD0	ENI	Habilita
0	1	0	1	Voltajes
0	1	1	1	Temperatura
1	0	0	1	Temperatura

Tabla IV.5 Codificación de los canales del multiplexor.

Finalmente la salida del multiplexor, que está configurado en forma diferencial, es entregada a un amplificador de instrumentación, para tener una mayor precisión en las lecturas de voltaje escalado; el voltaje de salida es enviado a la tarjeta de conversión analógico-digital.

El voltaje de salida del amplificador de instrumentación (AD624) se encuentra en el rango de +2.5 V a -2.5V. Una fotografía de este módulo se puede observar en la figura IV.6.

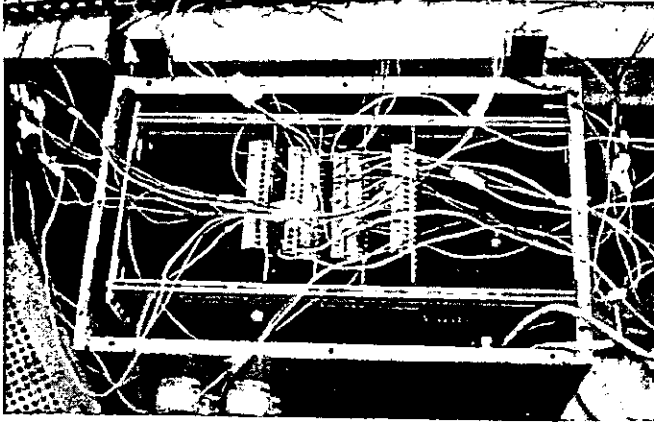


Figura IV.6 Módulo de acondicionamiento.

En el diagrama a bloques, figura IV.7, se muestra la manera en que funciona dicho módulo, así como la interacción con los demás elementos, como se explicó en los párrafos anteriores.

Este módulo se conecta a la *Little-G* por medio de un conector DB 25, cuyas características se explicaran más adelante.

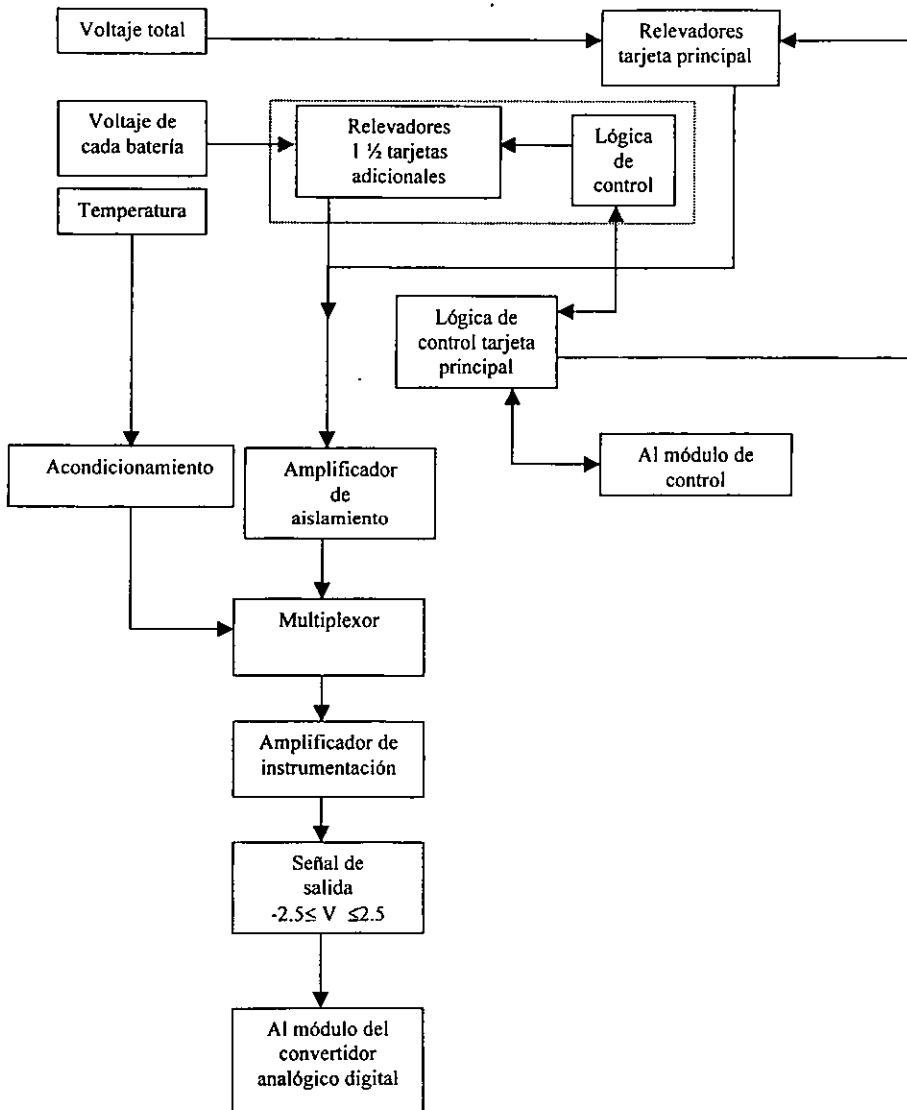


Figura IV.7 Diagrama a bloques que describe el funcionamiento del módulo de acondicionamiento.

IV.4 Tarjeta del módulo de conversión analógico-digital (TMCAD)

El módulo de conversión está formado por la tarjeta de conversión analógico-digital, la cual contiene 16 canales en modo unipolar y 8 en modo bipolar, los cuales se pueden elegir por medio de programación.

Para el sistema en particular, se utilizaron 3 canales en modo unipolar con ganancia unitaria, uno para la señal de velocidad, otro para la corriente que consume el motor, y se dejó como opción un canal adicional para la señal de corriente proveniente de algún sistema de regeneración. En estos casos los voltajes de entrada deben variar de 0 a +2.5 V.

También se utilizó un canal en modo bipolar con ganancia unitaria, en el que se toman las lecturas del voltaje total del banco, el voltaje individual de cada una de las baterías, y las señales provenientes de los sensores de temperatura. Cabe señalar que para este último modo de operación los voltajes de entrada al convertidor varían entre +2.5 V a -2.5V.

Para indicarle a la tarjeta de conversión en que momento actuar y que canal habilitar, ésta se programó a través del *software* de desarrollo de la tarjeta *Little-G*.

En el diagrama de flujo figura IV.8 se muestran los pasos para programar la tarjeta. Se puede observar que primero se tiene que declarar el modo de conversión de la tarjeta, es decir, si se va a leer el voltaje de entrada en forma unipolar o bipolar.

Posteriormente se debe indicar la ganancia que va tener la lectura, ya que ésta puede variar entre 1 hasta 202. Después se declara el número de canal a leer, proporcionándole un tiempo de retardo para asegurar que el canal esté listo y se lleve a cabo la lectura correcta. Finalmente por medio de la instrucción *float ad_conv(int entry, int value)* se regresa el voltaje representado por el valor de 12-bits obtenido en la entrada *value*, asumiendo una escala de 2.5V.

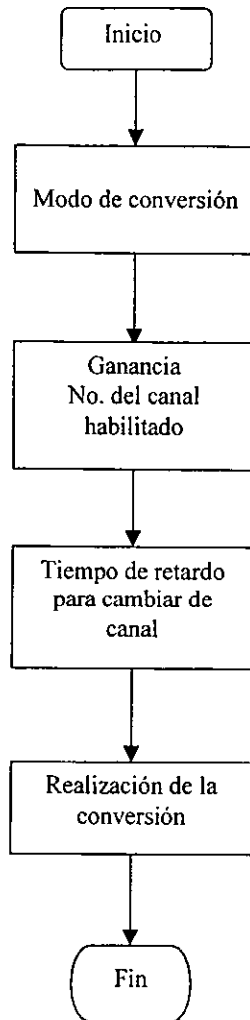


Figura IV.8 Diagrama de flujo de la programación del modulo de conversión analógico-digital.

La tarjeta del módulo de conversión analógico-digital se conecta a través de un cable plano de 20 líneas a la tarjeta del módulo de control *Little-G*, por el puerto *PLCBus*.

IV.5 Tarjeta del módulo de control (*Little-G*)

Para la coordinación de todos los demás módulos, se hace uso del módulo de control integrado por la tarjeta *Little-G*, de la cual se aprovechan las salidas digitales que presenta, la capacidad de memoria instalada, las ventajas de realizar la programación en lenguaje C y la *interface* serial RS-232.

Para el manejo de la tarjeta del módulo de acondicionamiento (TMA) se controla por medio de circuitos decodificadores TTL (apartado IV.2). Además de que se deben controlar los canales del multiplexor. Para realizar tal tarea, ésta se efectúa por medio de las salidas digitales que contiene la *Little-G*, se envían los bits que habilitan y direccionan tanto a los decodificadores como al multiplexor.

Se utilizan 12 de las 14 salidas digitales de la tarjeta, las que primero habilitan al circuito multiplexor del TMA, para direccionar el canal número 2; (del multiplexor) los dos primeros canales no son utilizados por el módulo, ya que están conectados para una aplicación diferente.

Después se habilita el decodificador 74LS138 con el propósito de tomar una lectura del voltaje total, esta lectura se envía al TMCAD, y posteriormente es almacenada en la *Little-G*. Siguiendo con la secuencia, son habilitadas cada una de las tarjetas que toman muestras de cada una de las baterías, de una manera similar ocurre para la conversión y el almacenamiento. Los siguientes canales que se habilitan del multiplexor son el número 3 y 4, para tener las lecturas de temperatura, enviándose al TMCAD y posteriormente almacenándolos.

La secuencia de operación que sigue se basa, principalmente, en direccionar la variable y una vez que se encuentra lista dicha variable se realiza la conversión analógica-digital. En el diagrama de flujo de la figura IV.9 se puede observar la secuencia que se sigue para llevar a cabo el control de la tarjeta.

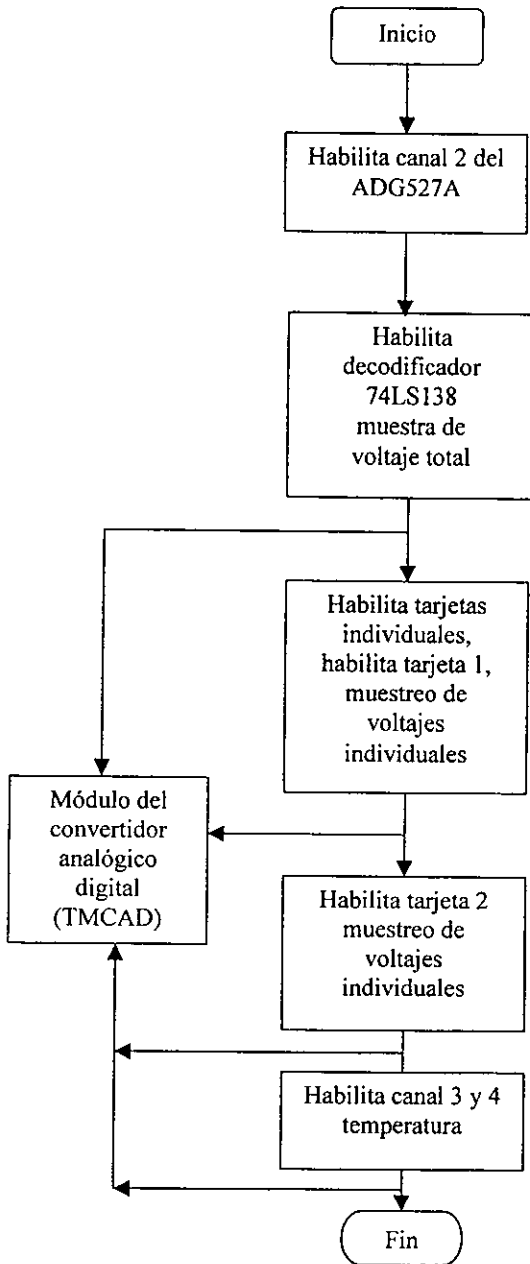


Figura IV.9 Diagrama de flujo de control del módulo de acondicionamiento.

IV.6 Diseño de la tarjeta del módulo externo de almacenamiento (TME)

El módulo de adquisición y almacenamiento de datos cuenta con una capacidad de hasta 450 Kbytes de almacenamiento.

Sin embargo una de las desventajas que se presenta en el almacenamiento de datos, es que cuando ocurra alguna situación, en la que no se pudiera contar con una PC, o que se perdiera el suministro de energía en el módulo, la transmisión de datos no podría realizarse. Es en este momento cuando se buscan alternativas para poder solucionar la contingencia. Se puede pensar en algún medio por el cual tuviera que prescindirse, en algún momento, de una PC para poder realizar el acopio de información para su consecuente proceso.

Una de las opciones que se consideran es el uso de las tarjetas de memoria *Thin Card Random Acces Memory (TCRAM)*, las cuales se mencionaron en el capítulo III. Estas tarjetas son del tamaño de una tarjeta de crédito, y pueden removerse del módulo sin necesidad de que el mismo esté alimentado.

La tarjeta *TCRAM* presenta la ventaja de un dispositivo RAM capaz de retener información gracias a que cuenta con una batería de respaldo, aunado a que la presentación de esta memoria (muy parecida a un disco flexible) nos permite transportarla muy fácilmente al lugar donde se terminara el proceso de los datos; en este caso en vez de llevar una PC al módulo para respaldo, llevamos los datos a una PC para su proceso.

Otra de las ventajas de la *TCRAM* sobre la memoria del módulo, es su capacidad, ya que se cuentan, en el mercado, con tarjetas de hasta 32 Mbytes, con lo cual se incrementa el alcance de respaldo. El problema principal de poder usar esta herramienta en un sistema diseñado, como lo es la *Little-G*, son las características que se requieren para su funcionamiento.

La tarjeta *Little-G* trabaja con un *bus* muy particular (*PLCBus* apartado III.2.3) y la forma en que accesa a sus dispositivos periféricos lo es también. Esto hace necesario la

construcción de algún medio que permita la interacción de ambos elementos. Este dispositivo, llamado *interface*, deberá contar con la capacidad tanto de adecuarse a la forma de comunicación de la tarjeta *Little-G* con sus módulos, como a la forma de trabajar de la *TCRAM*.

Entre las necesidades se tiene que considerar que la tarjeta *TCRAM* requiere:

- 24 líneas de dirección (ADR0-ADR23)
- 8 líneas de datos (D0-D7)
- al menos 4 líneas de control, (CE1, CE2, W, R)

Teniendo un requerimiento total de 36 líneas para poder manejar la *TCRAM*.

La tarjeta de control dispone de:

- un bus bidireccional para datos y direcciones de 8 líneas (D0-D7)
- 3 líneas de codificación de las instrucciones (A1-A3)
- una línea de sincronía para sus instrucciones (STR)

Con base en estas características se procede a diseñar la *interface*.

La *interface* a diseñar estará integrada alrededor de un microcontrolador. El microcontrolador que se escogió es el *COP8SAC*, que tiene la característica de contar con un número elevado de terminales, los cuales al ser independientes en su forma de selección, permiten hacer una distribución más eficiente.

El microcontrolador cuenta con 44 terminales, de las cuales se usan 4 distribuidos de la siguiente forma:

- 2 para polarización del circuito
- 1 en el reloj externo
- 1 en el *reset*

Quedando 40 líneas que se pueden utilizar para realizar el control de la *TCRAM*. La distribución del uso de estas terminales se lleva a cabo de la siguiente manera:

- 8 líneas de transmisión y recepción de datos (bus de datos bidireccional, E/S)
- 24 líneas de direccionamiento (acceso hasta 8 Mbytes de memoria, salidas)
- 3 líneas de codificación del módulo (A1-A3, entradas)
- 1 línea de sincronía (STR, entrada)
- 4 líneas de control de la tarjeta (CE1, CE2, W, R, salidas)

Al parecer el problema ya está resuelto, debido a que el dispositivo seleccionado cumple con las características, del número de terminales, pero ahora analizando las velocidades con que trabajan los dispositivos se encuentra con que son muy diferentes.

Dentro de la forma de trabajar de la *Little-G* está la transmisión de los datos. El modo de direccionamiento de los periféricos es por medio de direcciones de 12-bits, esto indica que los datos con los que trabaja el módulo son de 12-bits. Recordando que el *bus* es bidireccional.

La forma en que se lleva a cabo el intercambio de datos en el *PLCBus* es por medio de bloques de 4-bits, llamados *nibbles*. Para poder llevar a cabo este intercambio, se hace necesario presentar una forma de codificación que permita distinguir entre los bloques de información. Esta codificación se hace por medio de 3 líneas de control A1-A3, las cuales nos permite identificar el proceso.

Otra línea que se hace presente es la de sincronía con la que el módulo indica que se está llevando a cabo una operación. Y por último, las ocho líneas por donde se presenta el dato.

Los tres eventos descritos se presentan según la secuencia mostrada en la figura IV.8. El primer evento en presentarse es la línea de sincronía (*strobe*), la cual tiene una duración de 900 ns aproximadamente, e indica que se está llevando a cabo un evento. Los dos eventos restantes son simultáneos, y se presentan 450 ns después de iniciar el de sincronía., los tres

primeros bits son los de decodificación y los ocho restantes son el dato a transmitir. La duración de los dos eventos es de 450 ns, es decir al terminar el bit de sincronía pierde todo significado la información presente en el bus. Este proceso se ejemplifica en la figura IV.10 y fue obtenido utilizando un analizador lógico.

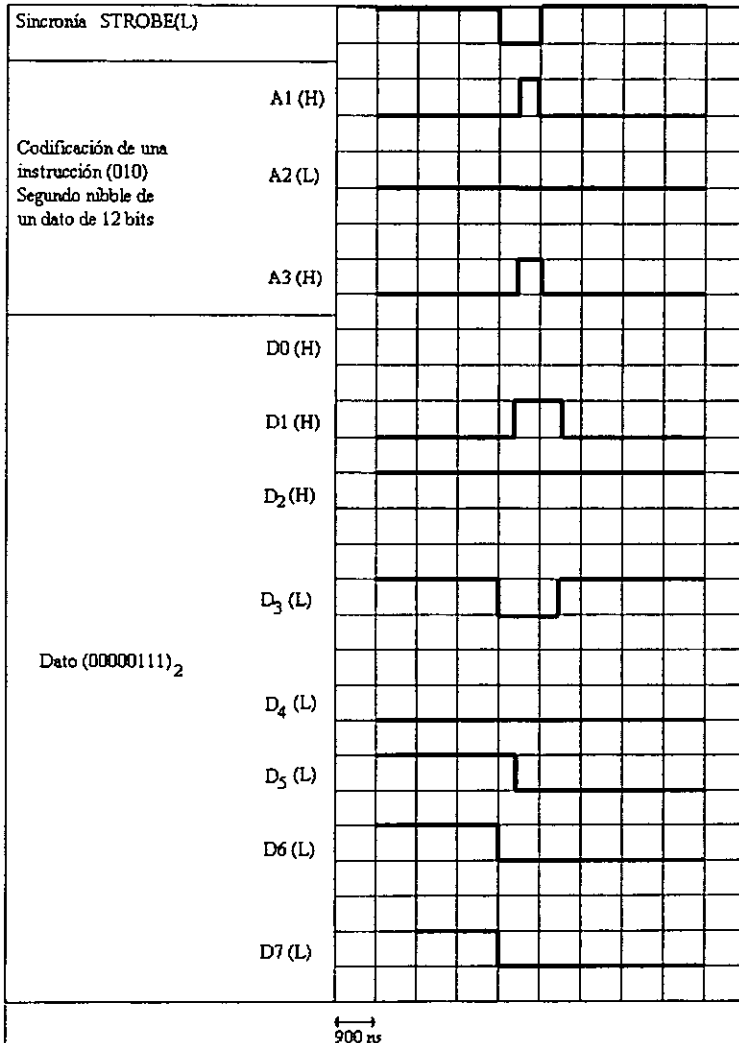


Figura IV.10 Diagrama de tiempos de un proceso de intercambio de datos en el *PLCBus*.

En la figura se observa que la acción presentada consiste en el envío del segundo *nibble* de un dato de 12-bits (B7D)_H, y el *nibble* transferido es el 7 (00000111)₂.

El COP8SAC trabaja con una cristal de cuarzo a una frecuencia de 10MHz, lo que da un ciclo de máquina de 1 μ s, esto deja ver que la instrucción que menos ciclos de máquina requiere, que es 1, tarda 1 μ s para realizarse, con lo cual ya no se cumple con la especificación para el intercambio de información.

Tomando en cuenta esto y que el *bus* debe ser bidireccional, se llega a la conclusión que necesitamos de circuitería adicional para llevar a cabo el intercambio de información.

Para lograr lo anterior proponemos el uso de un *latch* para poder disponer de los datos el tiempo necesario. Después el uso de un buffer tres estados para evitar el choque de información. Con estos dos dispositivos lo que obtenemos es un circuito transmisor-receptor (*tranceptor*) y se muestra en la figura IV.11.

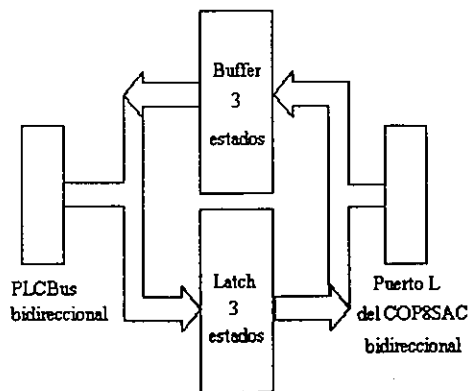


Figura IV.11 Diagrama a bloques del circuito tranceptor.

Para poder tener control sobre el circuito tranceptor que se acaba de describir, se necesita adicionalmente del uso de una lógica de control, lo que permitirá establecer el flujo de información de la manera que más convenga.

Esta lógica de control se llevó a cabo considerando nuevamente los recursos disponibles hasta este momento y tratando de evitar el uso adicional de líneas de control. Para ello tomamos las líneas de codificación del puerto del módulo y la línea de sincronía del mismo, y así se puede ver en la tabla IV.6.

W	Entradas			Salidas		
	A3	A2	A1	Enable Buffer	Output Control	Enable Latch
1	1	0	0	1	0	H
1	1	0	1	1	0	H
1	1	1	0	1	0	H
1	0	0	0	0	1	X
1	0	1	1	1	0	H
1	1	1	1	1	0	H

Tabla IV.6 Relación de entradas salidas necesarias para el diseño de la lógica de control.

Finalmente se llevó a cabo el diseño de la lógica de control que se utilizó, en la que empleamos una compuerta AND, una compuerta OR, y una compuerta NOT. En la figura IV.12 podemos observar el circuito lógico.

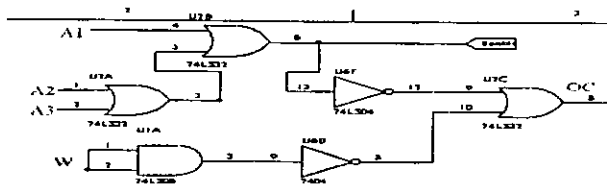


Figura IV.12 Lógica de control de la tarjeta de almacenamiento.

La integración final del módulo de almacenamiento externo cuenta entonces con los siguientes elementos: un circuito trancceptor, una lógica de control, el microcontrolador y la tarjeta *TCRAM*. Un diagrama a bloques del circuito final es mostrado en la figura IV.13.

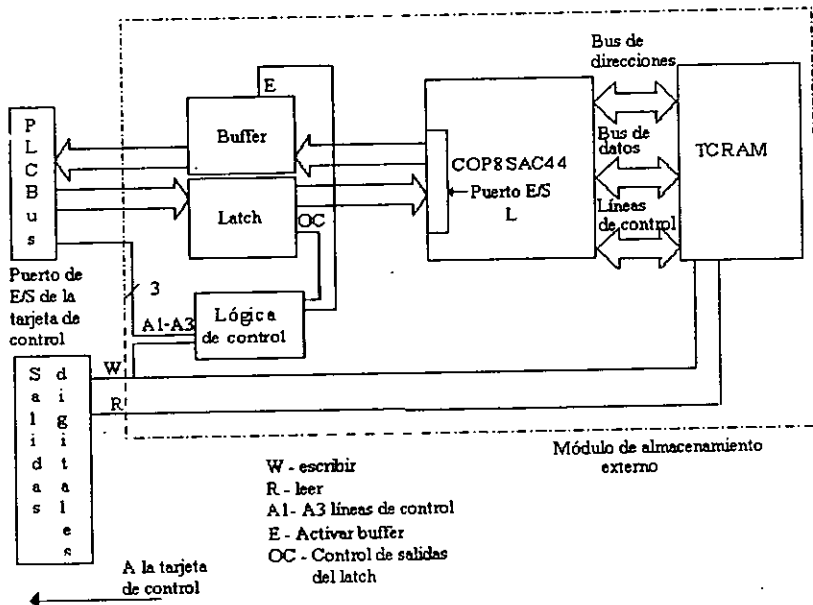


Figura IV.13 Diagrama del módulo de almacenamiento externo.

La figura IV.14 nos muestra la *interface* entre el módulo de almacenamiento y la *Little-*

G.

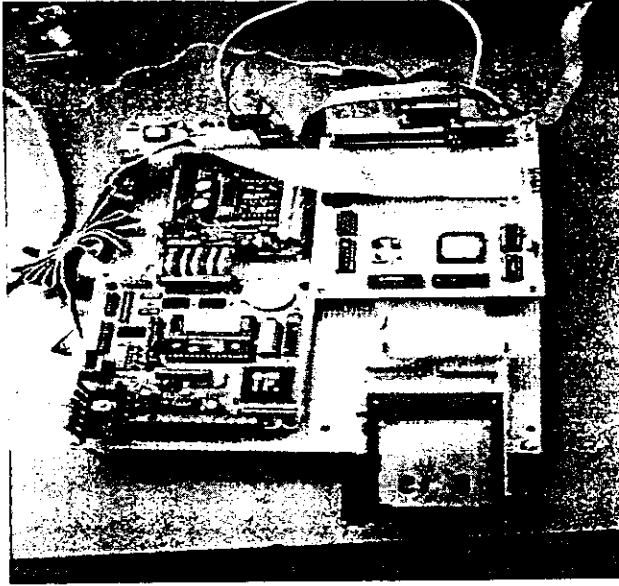


Figura IV.14 Interface entre el módulo de almacenamiento y la tarjeta de control.

IV.7 Conexiones

Como parte de la integración del SADME se necesita realizar las conexiones físicas correspondientes entre cada uno de los módulos del sistema y con el banco de baterías. Los módulos de control (*Little-G*), el TMCAD y parte del TME se conectan por medio de los puertos *PLCBus*, esto se logra utilizando conectores planos de 20 terminales.

En la figura IV.15 se observa que los números dentro de la foto indican:

- El número 1 señala el cable plano de 20 terminales correspondiente al puerto *PLCBus* entre la *Little-G* y el TMCAD.
- El número 2 indica el cable plano del *PLCBus* al TME.
- Con respecto al número 3 es un cable plano de 40 terminales del TME al *socket* donde se inserta la *TCRAM*.

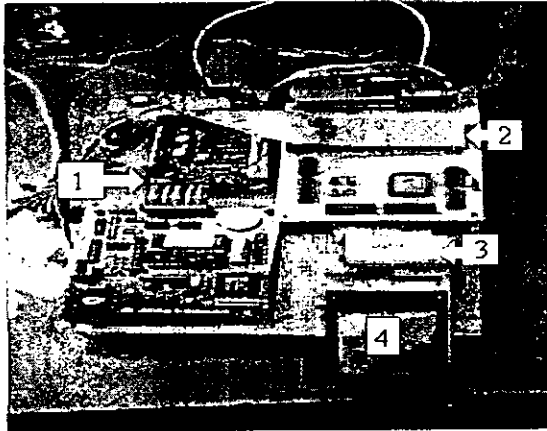


Figura IV.15 Conexiones de los módulos.

Entre el módulo TMA y la *Little-G* se utiliza un conector del tipo DB-25, en tabla IV.7 se muestra la configuración de terminales.

Terminal	Asignación	Terminal	Asignación
1	15 V	15	D2
2	-15 V	16	D3
3	+5V	17	D4
4	+5V	18	D5
5	GND	19	D6
6	GND	20	EN2
7	GND	21	GND
8	AD0	22	GND
9	AD1	23	GND
10	AD2	24	Salida
11	EN1	25	GND
12	NC	-	-
13	D0	-	-
14	D1	-	-

Tabla IV.7 Terminales del conector DB-25.

En la figura IV.16 se señala el conector DB25 con el número 1, el número 2 corresponde a un conector DB9 que sirve para realizar la comunicación serial hacia la PC.

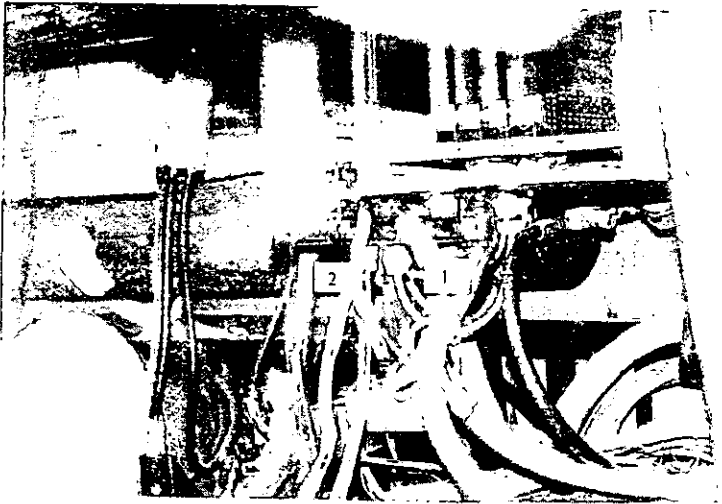


Figura IV.16 Conectores DB25 y DB9.

En la figura IV.17 los números indican que:

- 1) Agrupamiento de los módulos *Little-G*, TMCAD y TME.
- 2) Módulo de acondicionamiento TMA.
- 3) Conector CPC de 37 terminales proveniente del banco de baterías.
- 4) Conector CPC de 4 terminales proveniente de los sensores de temperatura.

Cabe aclarar que existen otros 3 conectores de 4 terminales CPC, uno es utilizado para la alimentación del SADME y otro para las señales de corriente y velocidad, el último para la alimentación. Para su asignación de terminales ver la figura IV.26.

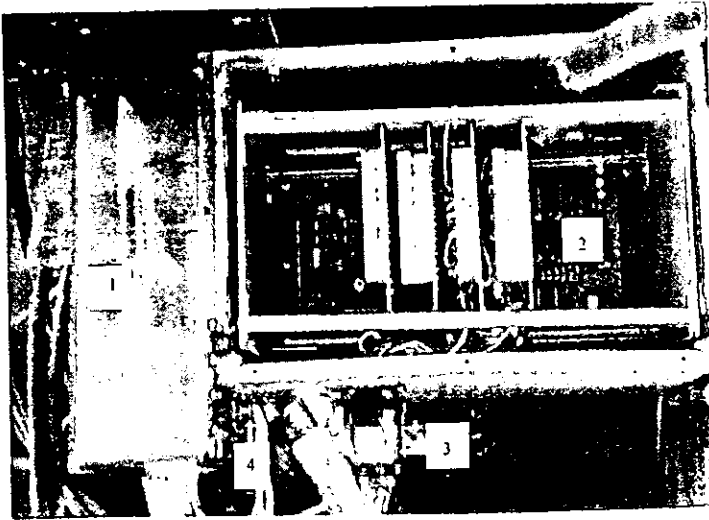


Figura IV.17 Conectores CPC y módulos.

Después de haber revisado la parte física del SADME haremos una referencia al *software* del sistema.

IV.8 *Software* del sistema

En este apartado mencionaremos los programas realizados para la interacción entre los diferentes módulos así como el *software* desarrollado para el usuario.

Estos se pueden dividir en los siguientes:

- Programa del módulo de control (PRODEC)
- Programa del módulo de almacenamiento externo (PROMOX)
- Programa de *interface* de usuario (SISTEMA26)
- Programa de conversión de archivo (PROCAR)

De los programas listados daremos una descripción de cada uno de ellos. Iniciaremos con la del PRODEC, en el apéndice B se muestran los listados de programa para una referencia más completa.

IV.8.1 Programa del módulo de control (PRODEC)

Este programa lo podemos dividir básicamente en las siguientes rutinas:

- **Ciclo principal.**
 - Comunicación serial.
 - Control de relevadores, decodificadores y multiplexor.
 - Rutina de la TMCAD.
 - Almacenamiento de datos.
 - Transmisión de datos.
 - Almacenamiento externo de datos.

- **Ciclo principal**

En este ciclo se anidan cada una de las rutinas del programa; en la figura IV.18 se observa que primero se necesita inicializar los parámetros. Estos parámetros son propios del programa ya sean variable, constantes, nombre de rutinas, llamado a librerías, etcétera.

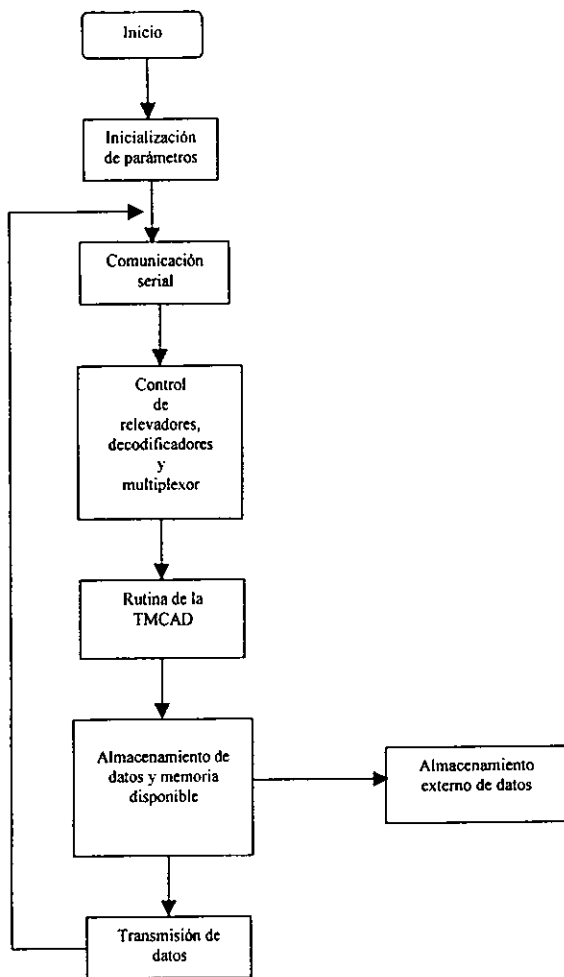


Figura IV.18 Ciclo principal del programa.

Ahora describiremos cada una las rutinas que componen al ciclo principal.

- **Comunicación serial**

Para realizar la comunicación entre el módulo de control y la computadora personal, se necesita explicar los parámetros con que se programó el módulo de control. Entre los parámetros de los cuales se hicieron uso están los que a continuación se mencionan, transmitir

8 bits sin paridad, con una velocidad de 19 200 bps, un bit de paro, se deshabilitó la conexión por módem y no se realizó eco después de la transmisión, debido a que la configuración para la comunicación que se utiliza es la mínima posible. Por otra parte la forma en que se programó la aplicación de la transmisión se aprecia en la figura IV.19, y es como sigue:

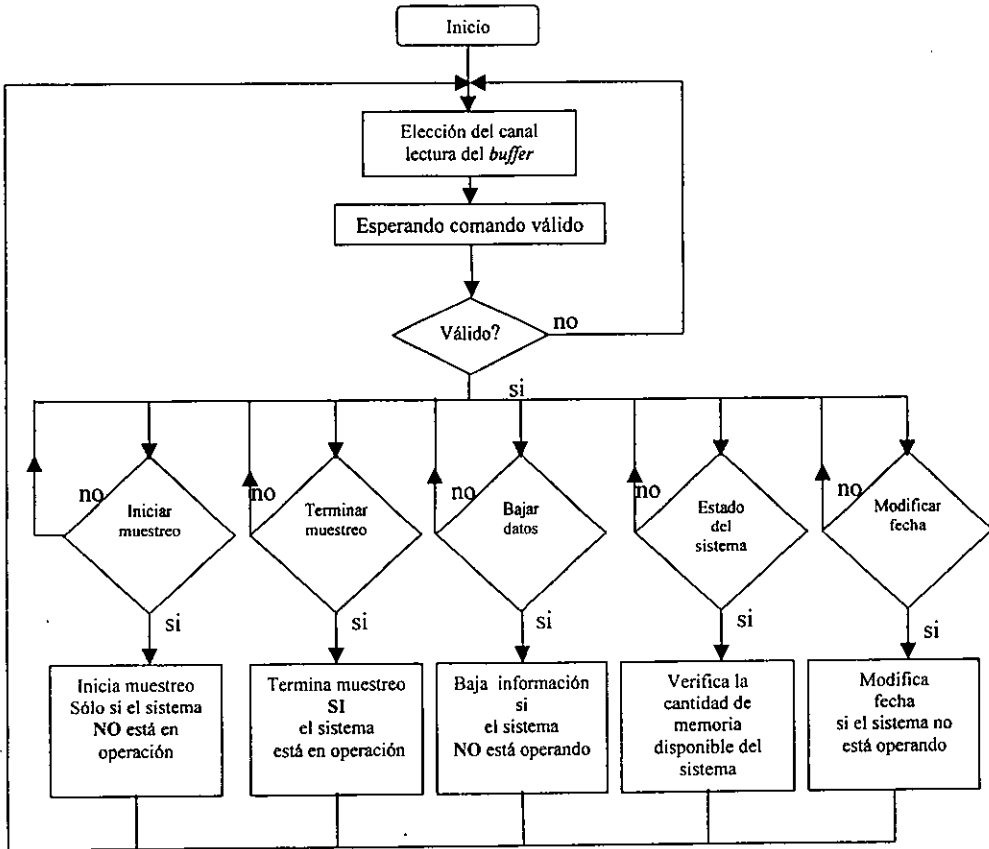


Figura IV.19 Diagrama de flujo de la comunicación de la tarjeta.

Primero se habilita algún canal de comunicación, en este caso el canal número 1, después se lee el *buffer* de recepción para esperar algún comando válido proveniente de la computadora personal, en caso de no ser correcto el sistema sigue esperando, de forma contraria, se evalúa el comando para determinar que acción es la que debe ejecutar el módulo de control. Después de lo cual el módulo sigue en espera de algún otro comando válido, estos

comandos son caracteres que se definen de antemano. Los comandos a los que se hacen referencias son los enlistados en la tabla IV.8.

Carácter	Función	Transmite
2	Inicia el muestreo	"iniciado"
3	Termina el muestreo	"Fin"
4	Acceso a los datos en memoria.	Carácter de inicio(y), hora y fecha del inicio del muestreo, bloque de 32 datos
5	Indica el estado de operación del sistema	En caso de estar en operación transmite: "A,horafecha,Nº muestra " donde la "A" indica que se encuentra en operación.
		En caso de estar en espera y con datos en memoria. "B,D,Nº muestra" donde la "B" indica que se encuentra en espera, la "D" con datos en memoria.
		En caso de estar en espera y sin datos en memoria: "B,C" donde "C" indica sin datos en memoria.
A	Envía el siguiente bloque de datos	Bloque de 32 datos
C	Cancela el envío de datos	"Findatos"
R	Restablece al sistema con sus parámetros iniciales.	
M	Comando de prueba	"Menu"
T	Transmite la fecha y hora que tiene el sistema	Hora y fecha
M	Se asume que la fecha y la hora son correctas	"M"
S	Se espera modificación de fecha a través de una cadena de caracteres.	"dia" cuando se terminó de modificar la fecha y la hora.

Tabla IV.8 Comandos del sistema.

Control de relevadores, decodificadores y multiplexor

En esta rutina se utiliza la instrucción *up_setout(canal, estado)* para designar uno de los 14 canales que tiene la tarjeta *Little-G* y determinar si la salida debe estar en alto (5 V) o bajo (0 V), esto se hace basándose en las tablas IV.3, IV.4, IV.5. Lo que nos da un total de 12 salidas que se traduce en la utilización de 12 de los 14 canales disponibles. Una vez que se ha seleccionado la variable a muestrear se pasa a la rutina del TMCAD que se describe enseguida.

- **Rutina del TMCAD**

Cuando el programa llama a esta rutina se procede a realizar la conversión de la variable se escala y se envía para su almacenamiento, como se mencionó en los apartados IV.4 y IV.5.

- **Almacenamiento de datos**

El módulo de control contiene una memoria RAM de hasta 512 Kbytes, en donde son almacenados los datos obtenidos. Sin embargo, existe la restricción del tamaño de la memoria, en el sentido de almacenar la mayor cantidad de datos posibles, por lo que se trata de optimizar la capacidad de la misma. Después que se ha realizado la conversión, el resultado es guardado en una variable del tipo flotante, la cual ocupa 4 bytes de memoria, reduciendo así la capacidad de almacenamiento. Por lo que se realiza un truncamiento al tipo *char* que sólo ocupa un byte.

De los 512 Kbytes, 62Kbytes son utilizados para el programa y 450 Kbytes como memoria de almacenamiento de los datos. Estrechamente relacionado con la capacidad de la memoria se encuentra la cantidad de muestras, por cada una se almacenan las 32 variables adquiridas lo que equivale a ocupar 32 bytes de memoria, lo que se representaría como 32 bytes/muestra, y si nuestra capacidad de almacenamiento es de 450 Kbytes entonces esto implica que el número de muestras antes de agotarse la memoria esta dada por la siguiente relación:

$$C.A. = \frac{450Kbytes}{32 \frac{bytes}{muestra}} = \frac{450 * 1024bytes}{32 \frac{bytes}{muestra}} = \frac{460800bytes}{32 \frac{bytes}{muestra}} = 14400muestras$$

Donde C.A. representa la capacidad de almacenamiento por muestra. Ahora, si cada muestra se realiza en un segundo, entonces el tiempo necesario para llenar la memoria es de 14400 segundos.

Si esto lo dividimos entre 3600 para obtener el número de horas, tenemos que:

$$C.T. = \frac{14400segundos}{3600} = 4horas$$

Donde C.T. representa la capacidad de almacenamiento por tiempo. Una vez llegado a ocupar totalmente la capacidad de la memoria, y si no se han transferidos datos a la computadora personal, el sistema empieza a sobrescribir la memoria, iniciando un nuevo muestreo.

Cabe aclarar él porque se realiza el muestreo de las 32 variables en 1 segundo, aunque cada canal es leído de manera muy rápida, se debe dar un tiempo de retardo para que cada canal este habilitado, además que se consideró sacar un promedio de muestras (35 muestras por lectura) por cada batería para minimizar la posibilidad de que ocurra una lectura errónea.

- **Transmisión de datos**

Otro aspecto importante es en el momento en que se transfieren los datos almacenados en la tarjeta a la computadora personal, pues se realiza el envío en bloques de 32 datos, esto es con el propósito de evitar algún error en la transmisión de datos, para posteriormente aplicarles los pasos correspondientes en el programa de usuario y así obtener el valor real.

- **Almacenamiento externo de datos**

Cuando esta rutina es llamada se inspeccionan que el módulo de almacenamiento externo esté presente, esto lo realiza la *Little-G* por medio de las entradas digitales, si se lee un nivel lógico bajo (0V) se considera que el módulo de almacenamiento externo está listo.

Por otro lado, cuando se revisa un apuntador de memoria que indica la cantidad de memoria ocupada, el SADME detiene el muestreo si se está realizando para transferir los datos al TME.

Esta rutina se subdivide en otras 3 subrutinas que son:

- Inicialización de la TME
- Determinación de la acción a realizar
- Escribir o leer datos desde la *TCRAM*

Inicialización de la TME

En esta subrutina el programa manda el comando de *reset* al *PLCBus* y así prepara la tarjeta para su eventual uso. Después transmite el código de reconocimiento de la TME, 12-bits en 3 *nibbles* de 4-bits. Estas funciones fueron implementadas en lenguaje ensamblador y consisten en mandar el código de direccionamiento y el dato de reconocimiento. Después se presenta un retardo para dar tiempo al TME de hacer el reconocimiento del mismo. Este proceso se repite 3 veces para poder mandar los 3 *nibbles*. La figura IV.20 lo ejemplifica.

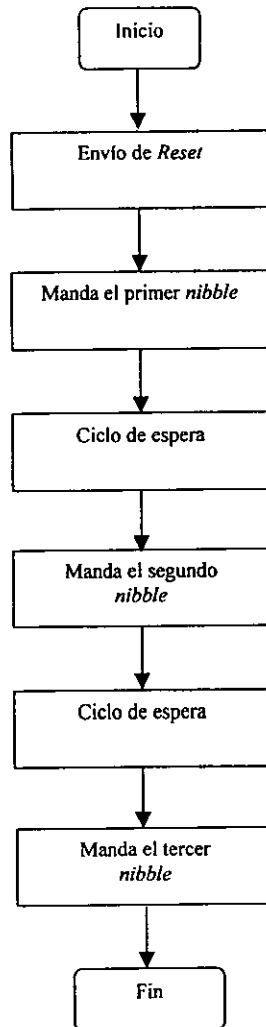


Figura IV.20 Diagrama de flujo de la rutina de reconocimiento.

Determinación de la acción a realizar

Cuando el código es recibido por la TME el programa transmite un código en el cual se indica la acción a realizar. Esta acción se realiza por medio de la instrucción de escribir al *PLCBus* y en el *bus* de datos se pone un código 0x00 para lectura y 0x01 para escritura. Esta acción se realiza para preparar la TME para cualquiera de estas acciones y el programa en la tarjeta de control ya conoce la acción de manera anticipada.

Escribir o leer datos de la TCRAM

Esta parte del programa realiza cualquiera de dos acciones: leer datos desde la *TCRAM* o escribir los datos en ella.

Cuando la tarjeta del módulo de control adquiere datos lo hace primero en su área de memoria. Cuando es alcanzada la capacidad de almacenamiento, la tarjeta de control dejará el proceso de adquisición y vaciará los datos en ella contenida a la tarjeta TME. Esta acción es realizado por medio de bloques de 64Kbytes.

La tarjeta del módulo control transmite los datos a la TME en formato de 8 bits y por medio de su instrucción de escritura al *PLCBus*. La tarjeta lleva un contador en el cual se va llevando el registro de los datos transmitidos. Al transferir un dato la tarjeta del módulo de control entra en un *loop* del cual saldrá solamente cuando reciba un código que manda la TME para avisarle que esta en disposición de recibir el datos siguiente. Cuando un bloque es enviado completamente se continua con el siguiente, cuando se termina de enviar los bloques la tarjeta de control vuelve al proceso de adquisición.

Este proceso es un ciclo que termina cuando la memoria esta completamente llena. La figura IV.21 muestra en un diagrama de flujo este proceso.

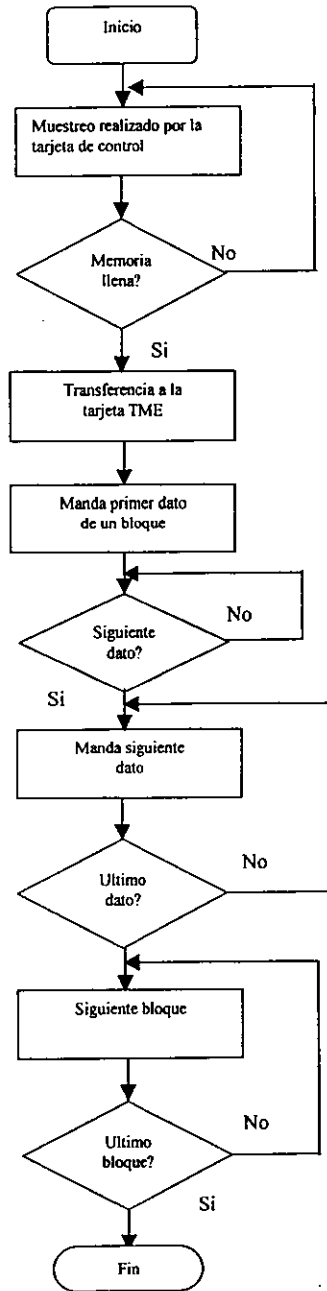


Figura IV.21 Subrutina para escribir en la TCRAM.

IV.8.2 Programa del módulo de almacenamiento externo (PROMOX)

La tarjeta del módulo externo de almacenamiento está programada en lenguaje ensamblador para el microcontrolador *COP8SAC44*. Este programa consta de tres partes en especial, la primera parte de reconocimiento de la tarjeta, la segunda de reconocimiento de la acción a realizar y la tercera de la acción específica.

La tarjeta está conectada de manera permanente en el *PLCBus* de la tarjeta del módulo de control, en paralelo con la tarjeta *TMCAD*, y cada una de ellas funciona de manera exclusiva, esto quiere decir, que cuando cualquiera de ellas está funcionando, la otra tarjeta no lo hace.

Con base en el diagrama de flujo presentado en la figura IV.22, el procedimiento de operación es el siguiente:

a) Cuando el sistema así lo requiere, establece comunicación con la tarjeta, esto lo realiza mediante un código único (ver tabla III.3) en el cual manda la dirección conformado por 3 bits que corresponde a ella.

b) Después de detectar el requerimiento, la tarjeta retorna un dato de control (0xDE), que a su vez indica que está lista a recibir la nueva orden.

c) La nueva orden recibida (0 escritura, ó 1 lectura), indicará si el proceso es de lectura o escritura a la *TCRAM*. Para el proceso de escritura se realizan los siguientes pasos:

- Como el proceso de transferencia de datos se realiza por bloques, primero se carga la última dirección en la cual se almacenaron datos, esta información se almacena en algunas localidades de la memoria.
- Después de esto la tarjeta controladora enviará el dato a guardar, con un código específico para esto (ver tabla III.3), el cual después de ser recibido por el *COP8SAC44* se almacenará en la tarjeta de memoria.

- Después de almacenar el dato se verifica si se ha alcanzado la capacidad de la tarjeta, y siendo el caso detener la acción de escritura.
- A continuación el programa revisa que el bloque haya sido enviado en su totalidad.
- Si ya se terminó la transferencia del bloque, el sistema guarda nuevamente la última dirección en la que se escribió para que una futura transferencia se continúe ahí mismo.

Para el proceso de lectura a la tarjeta se realizan los siguientes pasos:

- Se inicializa el *bus* de direcciones.
- Se lee el dato de la localidad.
- Se incrementa el apuntador.
- Se revisa que se haya terminado ya la acción.

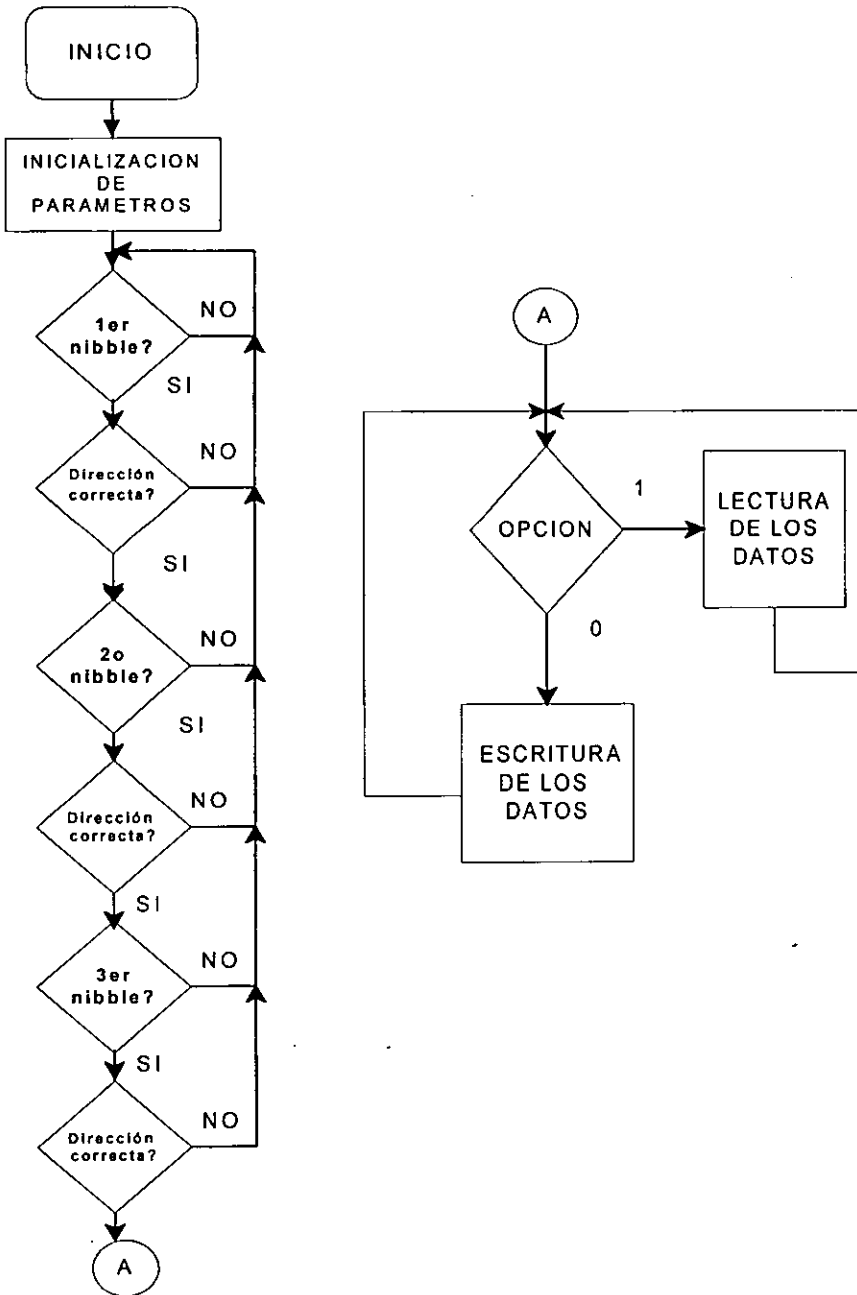


Figura IV.22 Diagrama de flujo del programa de la *interface* de la tarjeta de almacenamiento.

IV.8.3 Programa de usuario Sistema26

El programa Sistema26, es un *software* desarrollado con el propósito de ser una *interface* amigable entre el SADME y el usuario que tenga a su cargo el monitoreo de las diversas variables que tiene el Minibús Eléctrico. Este programa fue desarrollado bajo el lenguaje de programación llamado *Visual Basic 4*, siendo éste un lenguaje orientado a objetos.

Se utilizó tal lenguaje debido a que hoy día la gran mayoría de las aplicaciones (programas) que se ejecutan en las PC's actuales trabajan bajo el ambiente *Windows*, por lo que se hace necesario realizar un *software* compatible con las tendencias que en este momento se presentan, y *Visual Basic 4* nos permite desarrollar dicha labor.

La filosofía que se siguió en el desarrollo del *software*, fue el que se realizaran todas las rutinas necesarias para el completo control del SADME y que fueran lo más sencillas posibles. Para esto es necesario tomar en cuenta lo siguiente:

- Comunicación RS-232 con el SADME.
- Verificación del estado que guarda el SADME.
- Cantidad de memoria disponible del SADME.
- Controlar el inicio y fin de cada muestreo, así como el de poder restablecer el sistema y decidir en que momento transmitir los datos.
- Recibir los datos que transmite el SADME y su posterior procesamiento para obtener el valor real de las variables y su almacenamiento.
- Modificación de parámetros, como: puerto en que se encuentra disponible en la PC, modificación de hora y fecha del SADME, cancelación de la transmisión de datos.
- Poder guardar los datos en un archivo de texto con el propósito de realizar su análisis en alguna hoja de cálculo.

La secuencia básica que debe seguir la estructura del *software* se observa en el diagrama de la figura IV.23.

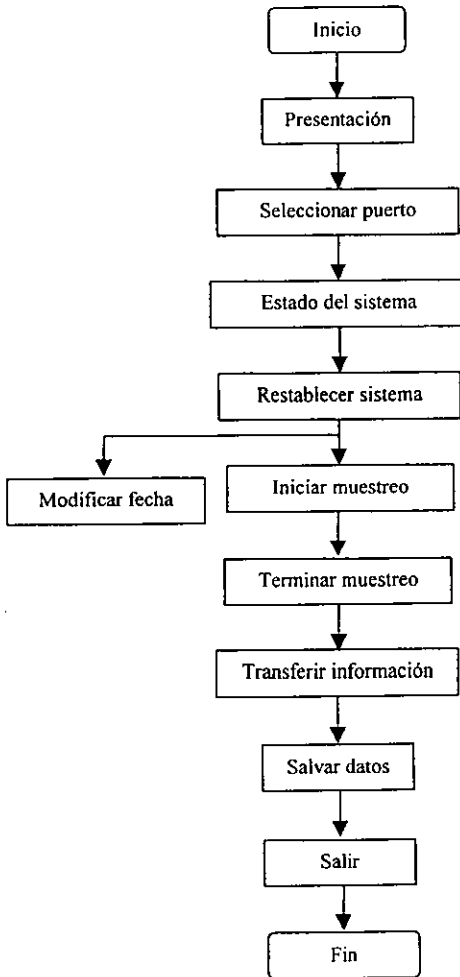


Figura IV.23 Secuencia básica del programa.

Presentación

Esta es solamente la parte introductoria para iniciar la sesión del *software*.

Seleccionar puerto

En esta rutina se eligen los parámetros para establecer la comunicación, éstos están programados por anticipado y lo que se le pide al usuario es seleccionar alguno de los puertos serie de la PC.

Entre los parámetros de los cuales se hicieron uso están los que a continuación se mencionan, transmitir 8 bits sin paridad, con una velocidad de 19 200 bps, un bit de paro, se deshabilitó la conexión por módem.

Cuando el usuario elige alguno de los puertos disponibles de su PC, esta rutina verifica que por el puerto se encuentre conectado el SADME, ya que este último envía una cadena de caracteres preestablecidos con el propósito de que se pueda verificar la comunicación.

En caso de que se reciba una cadena de caracteres (ver tabla IV.8) se verifica su validez y se prosigue su procesamiento para establecer el estado del SADME. Esto con la intención de controlar las demás rutinas y poder habilitar los botones necesarios del Sistema26. Si lo anterior no se llega a cumplir se envía un mensaje de error.

Estado del sistema

Cuando se ha verificado que el sistema está presente y por tanto se ha establecido la comunicación, esta rutina es habilitada con la intención de que el usuario pueda verificar el estado del SADME. Esto se realiza a partir de enviar una serie de caracteres preestablecidos (ver tabla IV.8) y que por tanto, el SADME los reconoce, de donde el sistema envía una nueva cadena de caracteres y estos son recibidos y procesados, desplegando enseguida el estado en que el SADME se encuentra.

Modificar fecha

Esta rutina se habilita siempre y cuando el sistema no se encuentre haciendo un muestreo o con datos en memoria. El objetivo es verificar y en su caso modificar la fecha del SADME, resulta de gran importancia ya que al iniciar algún muestreo la fecha y la hora son almacenadas en el SADME.

Restablecer sistema

Para el caso en que el usuario decida reinicializar el sistema, se ha implementado esta rutina, la cual manda al SADME a sus parámetros iniciales.

Iniciar y terminar muestreo

Cuando se ha determinado cual es el estado actual del sistema, alguno de estas dos rutinas son habilitadas, siendo de forma simple enviar un caracter de inicio y de fin (ver tabla IV.8) previamente determinado.

Transferir información

Esta rutina solamente es habilitada cuando se ha detenido el muestreo, esto es con el propósito de que se pueda realizar de manera más rápida la transferencia de la información, ya que el sistema esta solamente dedicado a bajar los datos. El usuario tiene las opciones de: detener el envio de datos o iniciar un nuevo muestreo.

Cuando los datos son totalmente transferidos a la PC éstos son almacenados en una memoria temporal para su posterior adecuación y almacenamiento.

Salvar datos

Una vez que los datos son transferidos a la PC, son procesados y posteriormente almacenados en un archivo de texto. Para que después puedan ser analizados en una hoja de cálculo.

IV.8.4 Programa de conversión de archivo (PROCAR)

Como la manera en que se almacena la información en la tarjeta *TCRAM* es igual a como se guarda en una memoria de estado sólido, es necesario darle un formato adecuado con el cual se pueda procesar en una hoja de cálculo.

La información de la *TCRAM* es transferida a un archivo de texto por medio del *software* de la misma (*icread*), y leída desde una PC que cuente con una unidad especial para ello. Este archivo de texto carece de formato alguno (figura IV.24), por lo cual se desarrolló una utilidad en *Visual Basic* para homologar el archivo de tal manera que no hubiera forma de diferenciar entre un archivo adquirido directamente por el *software* del sistema vía comunicación serial y uno bajado en la *TCRAM*, utilizando esta aplicación (PROCAR) como lo muestra la figura IV.25.

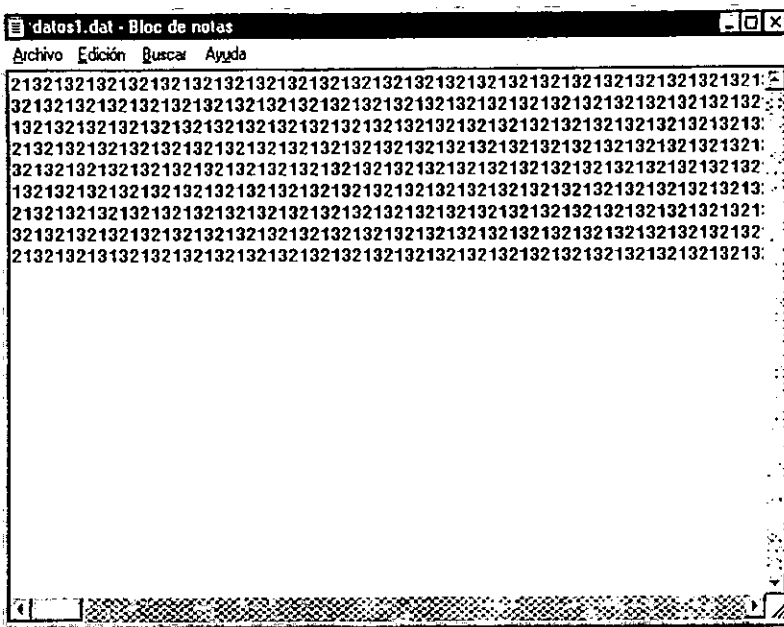


Figura IV.24 Archivo de datos almacenado en la TCRAM sin formato.

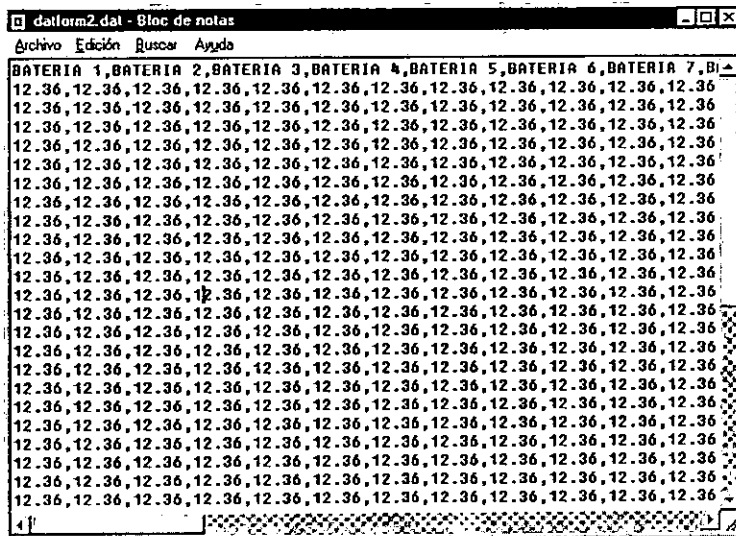


Figura IV.25 Archivo de datos después de la aplicación de la utilería PROCAR.

IV.9 Instalación del SADME

Para concluir presentamos el diagrama de instalación del sistema, (figura IV.26) para realizar la instalación, donde se indican las conexiones de las variables al SADME. Cabe aclarar que sólo se alambrió la conexión de corriente que consume el motor, por que la de regeneración aún no está disponible, sin embargo se mantiene habilitada para su uso.

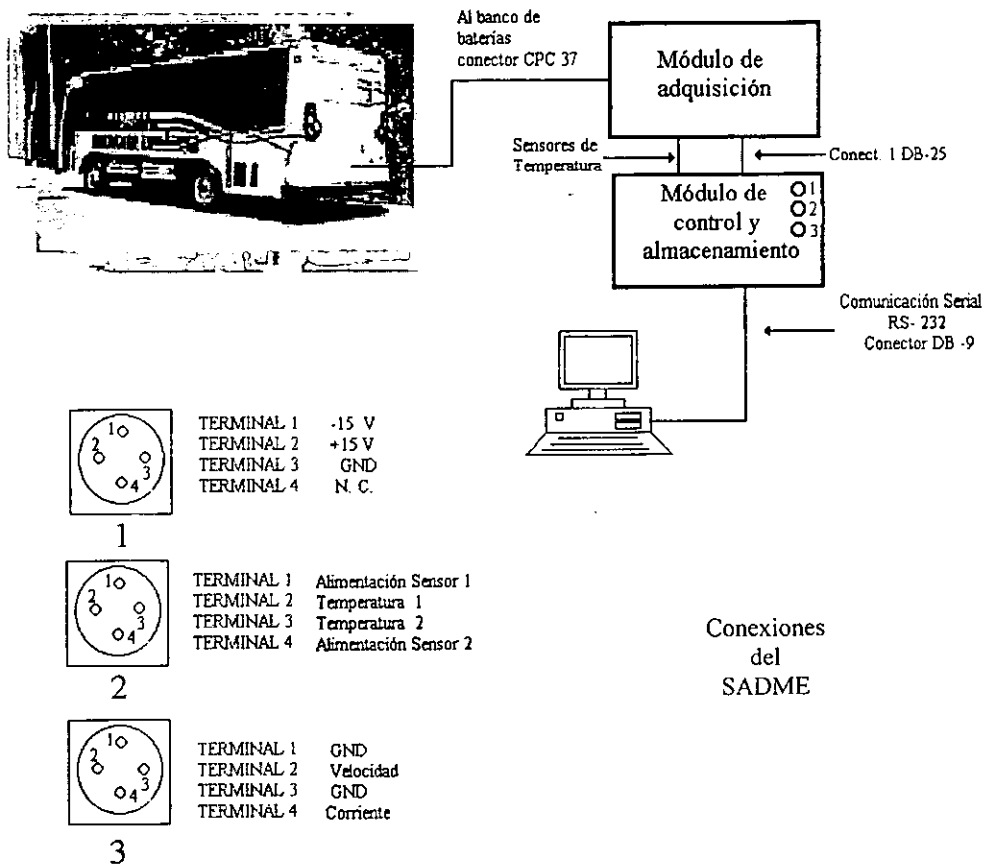


Figura IV.26 Diagrama de instalación del SADME.

Finalmente en la figura IV.27 observamos en la parte posterior del Minibús el SADME ya instalado.

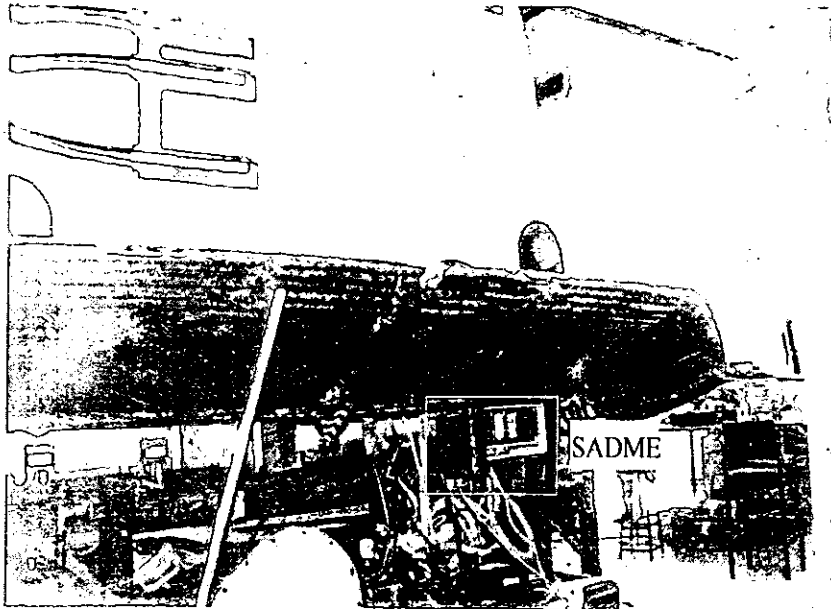


Figura IV.27 El SADME en el minibús eléctrico.

Como vimos en el capítulo, la integración de los módulos, consistió de dos partes fundamentales, lo referente al *hardware* y sus conexiones y al *software* utilizado, ya sea como parte de alguna de las tarjetas de los módulos o *interface* con el usuario.

En el siguiente capítulo veremos las pruebas, tanto de laboratorio como en el minibús, que se realizaron.

====CAPITULO IV====

Diseño del prototipo del sistema

En este capítulo presentamos como se realizó la integración de todos los módulos del Sistema de Adquisición de Datos del Minibús Eléctrico (SADME), las especificaciones y restricciones que se presentaron. Además del desarrollo de los programas para las interacciones entre cada uno de estos módulos, así como con el usuario final.

IV.1 Especificaciones para el diseño del sistema

Los requerimientos que demanda el sistema ya descrito son básicamente:

- Un sistema de monitoreo y adquisición de variables para un vehículo eléctrico.
- Capacidad adquirir 32 variables por muestreo distribuidas de la siguiente manera:
 - Voltaje individual de 26 baterías.
 - Voltaje total del banco.
 - Temperaturas de los bancos de baterías (2).
 - Velocidad del Minibús Eléctrico.
 - Corrientes (2).
- Capacidad de comunicación vía RS-232 con una computadora personal.
- Capacidad de almacenamiento del sistema.

Para llevar a cabo lo descrito anteriormente, el sistema se conformó de cuatro partes fundamentales:

- Tarjeta del módulo de acondicionamiento (TMA).
- Tarjeta del módulo de conversión analógico-digital (TMCAD).
- Tarjeta del módulo de control (*Little-G*).
- Tarjeta del módulo externo de almacenamiento(TME).

De cada uno de los módulos mencionados se expondrá sus características de funcionamiento, la manera en que se acoplaron entre ellos y que limitantes presentan.

Es importante explicar el cableado en el banco de baterías para realizar el muestreo de cada una de ellas, así como lo referente a los sensores de temperatura, y la comunicación entre los módulos. Para entender la integración del sistema se presenta en el siguiente apartado una descripción de la conformación del banco de baterías del Minibús Eléctrico.

IV .2 Conexión de las baterías del minibús eléctrico

El banco de baterías que proporcionan energía al motor está constituido de 52 baterías; este banco a su vez se agrupa en banco derecho e izquierdo, cada uno con 26 baterías y a su vez se subdividen en parte superior e inferior para ambos agrupamientos. Las baterías de ambos agrupamientos se encuentran conectados en arreglo paralelo-serie, por lo que finalmente se tienen que realizar la medición de 26 voltajes, es decir, trece voltajes por agrupación. En la figura IV.1 se muestra el diagrama de conexión de las baterías.

El voltaje individual de cada batería es de 12 V, por lo que en teoría se tendrá un voltaje total del arreglo de 312 V. En el diagrama se puede observar que las baterías marcadas con el número 26 representan el nivel más negativo en el banco, y las etiquetadas con el número 1 representan la parte más positiva.

Además se observan en la figura IV.1 las abreviaciones Cn, indican el lugar en donde fueron instaladas las terminales, con el propósito de tender un cableado que permita realizar los muestreos.

Para realizar el cableado se utilizaron zapatas redondas de 6.35 mm, como terminales directos a los bornes de las baterías y cable del número 20. Que se soldaron a conectores hembras del tipo CPC de 16 canales para cada subdivisión del banco. En la parte central de cada agrupamiento del banco, se encuentra la contra del CPC de 16 canales, en el que se tiene la conexión de otro cable que llega a la parte trasera el minibús, que es el lugar en donde se encuentra el sistema de adquisición (ver figura IV.1).

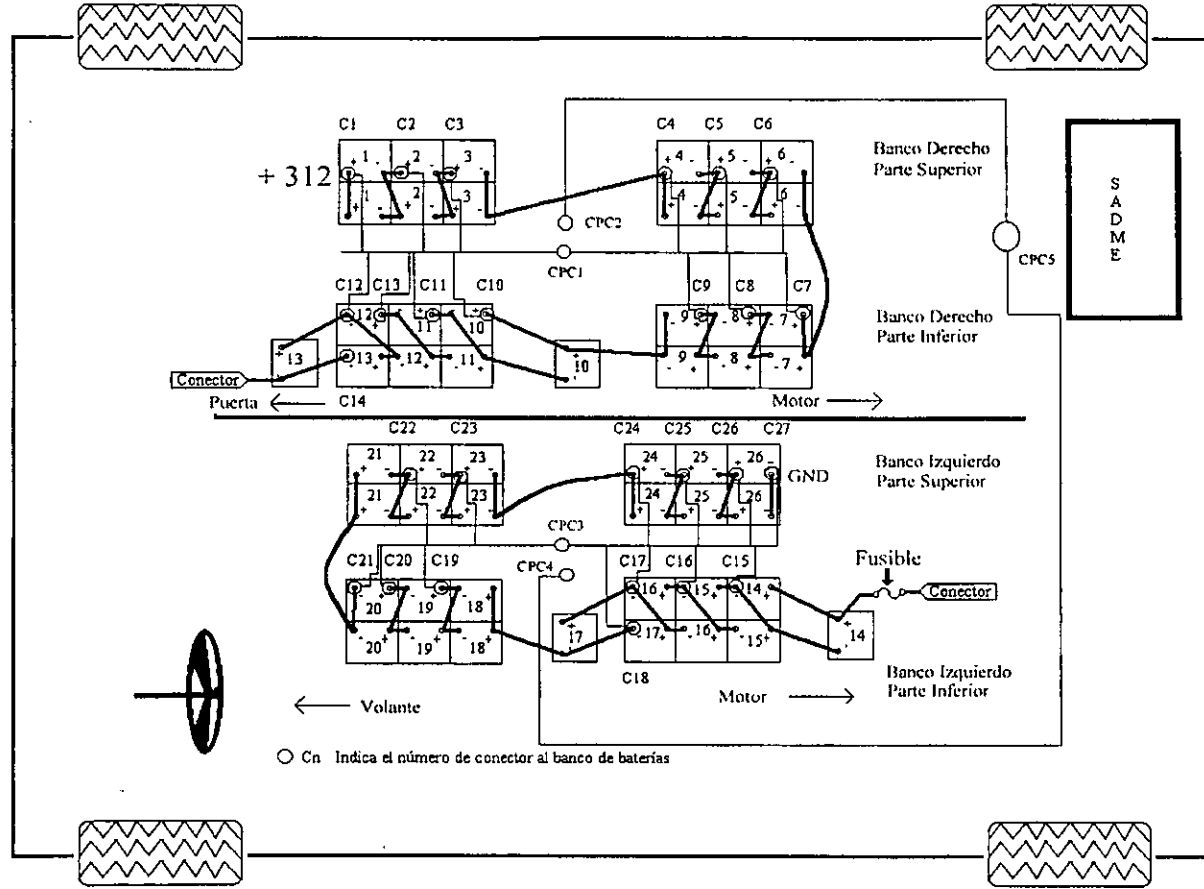


Figura IV.1. Conexión del banco de baterías del minibus eléctrico.

En la figura, las siglas CPC1 y CPC3 indican los conectores hembra de 16 terminales, las siglas CPC2 y CPC4 indican los conectores macho de 16 terminales. Las siglas CPC5 indica el conector de 37 terminales, de las cuales sólo se utilizan 27.

Los bancos derecho e izquierdo se interconectan a través de un fusible por razones de seguridad, ver figura IV.1.

El voltaje total del banco de baterías es tomado de los cables 1 y 26 que son, como se mencionó, los extremos del banco. Por lo que se aprovecha el mismo cableado para realizar la medición del voltaje total. En cada agrupación se encuentra un sensor de temperatura, cada sensor presenta un cableado similar desde la parte central del banco hasta la parte posterior del minibús.

En la figura IV.2, se presenta la distribución de las terminales en los conectores de 16 terminales CPC1 y CPC3.

En las tablas IV.1 y IV.2 se presentan la conexión de cada una de las terminales con los conectores CPC1, CPC2, CPC3, CPC4, CPC5. Cabe señalar que los números de las segundas columnas corresponden a los conectores del CPC de 37 terminales.

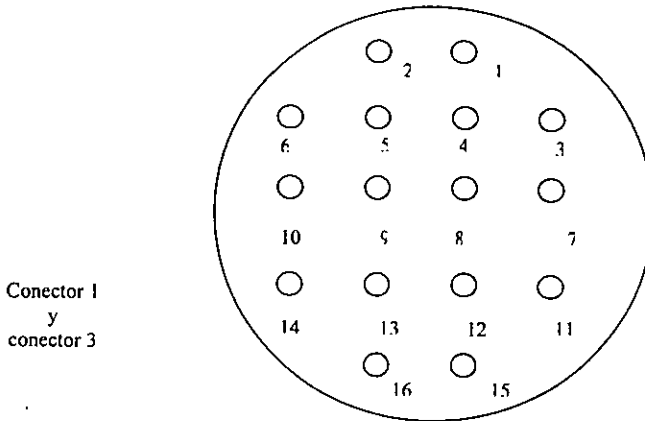


Figura IV.2 Conector para cada uno de las subdivisiones.

Terminales en el banco de baterías	Terminales de los conectores CPC1 y CPC2	Terminales del conector CPC5	Código de colores entre (CPC2) y (CPC5)
1	1	1	Blanco
2	2	2	Gris-blanco
3	3	3	Rojo-blanco
4	4	4	Gris
5	5	5	Violeta-blanco
6	6	6	Azul-Blanco
7	7	7	Negro
8	8	8	Naranja
9	9	9	Café
10	10	10	Verde-blanco
11	11	11	Rojo
12	12	12	Verde
13	13	13	Amarillo-blanco
14	14	14	Morado
-	15	Sin conexión	Sin conexión
-	16	Sin conexión	Sin conexión

Tabla IV.1 Correspondencia de terminal conector para la parte derecha del banco.

Terminales en el banco de baterías	Terminales de los conectores CPC3 y CPC4	Terminales del conector CPC5	Código de colores entre (CPC4) y (CPC5)
15	1	15	Blanco
16	2	16	Amarillo
17	3	17	Rojo-blanco
18	4	18	Blanco-café
19	5	19	Negro
20	6	20	Azul
21	7	21	Verde-negro
22	8	22	Verde
23	9	23	Verde-blanco
24	10	24	Blanco-verde
25	11	25	Rojo-negro
26	12	26	Azul-blanco
27	13	27	Negro-amarillo
-	14	Sin conexión	Sin conexión
-	15	Sin conexión	Sin conexión
-	16	Sin conexión	Sin conexión

Tabla IV.2 Correspondencia de terminal conector para la parte izquierda del banco.

IV.3 Tarjeta del módulo de acondicionamiento (TMA)

El módulo de acondicionamiento, como lo mencionamos en el capítulo anterior, consta de cinco tarjetas. Cuatro (denominadas tarjetas adicionales) están constituidas por relevadores del tipo *Reed*, los que se encuentran distribuidos en todas las tarjetas, siendo 17 para cada tarjeta. Teniendo como propósito la medición individual del voltaje de cada batería.

Estos relevadores son accionados por un decodificador, 74HC154 de 4 líneas a 16 salidas. Para la medición del voltaje de cada batería se activan 2 relevadores a la vez (ver figura IV.3), lo que constituye un primer paso, y por la manera en que se encuentran conectados los relevadores al amplificador de aislamiento (AD202) se obtiene una lectura de voltaje positivo. Segundo paso, se apaga el primer relevador que se encendió quedando el segundo en funcionamiento además de accionar un tercero, teniendo un voltaje negativo.

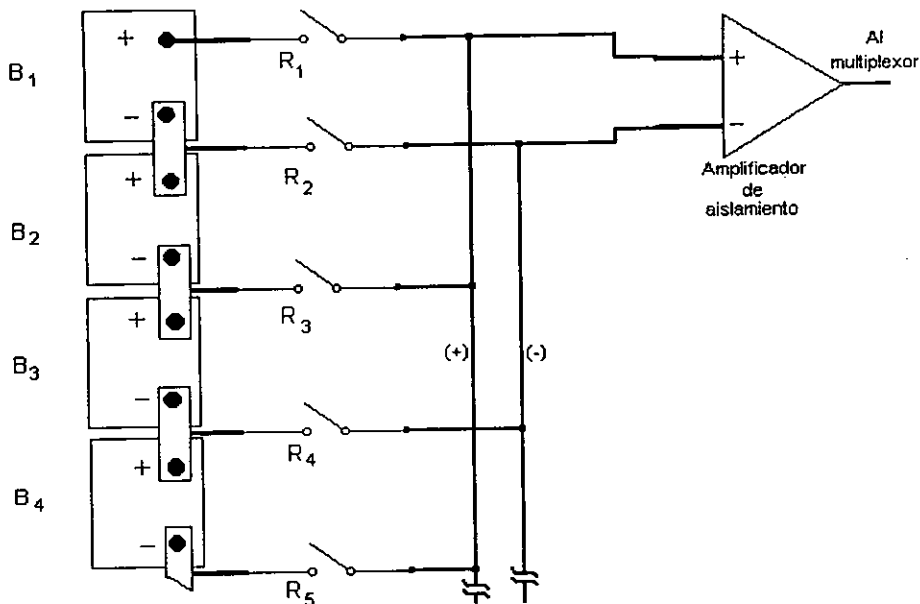


Figura IV.3 Secuencia de encendido de los relevadores.

El tercer relevador permanece en su estado anterior y el cuarto es habilitado lo que lleva a una lectura de voltaje positivo, considerándose el tercer paso. La figura anterior ejemplifica de forma esquemática dicho procedimiento. Después se continúa de forma secuencial hasta concluir el muestreo de las 26 pares de baterías, de donde se observa que se tienen 13 voltajes positivos y 13 voltajes negativos.

Las cuatro tarjetas adicionales tienen capacidad para 16 baterías por tarjeta, pero en el momento actual sólo se utilizan una y media tarjeta de las cuatro. Debido a que sólo se hacen muestreos de 26 voltajes de baterías.

En la tabla siguiente, IV.3, se muestra la secuencia que debe seguir el decodificador para habilitar cada par de relevadores de cada una de las tarjetas adicionales.

D3	D2	D1	D0	Habilita
0	0	0	0	Relevadores 1 y 2
0	0	0	1	Relevadores 2 y 3
0	0	1	0	Relevadores 3 y 4
0	0	1	1	Relevadores 4 y 5
0	1	0	0	Relevadores 5 y 6
0	1	0	1	Relevadores 6 y 7
0	1	1	0	Relevadores 7 y 8
0	1	1	1	Relevadores 8 y 9
1	0	0	0	Relevadores 9 y 10
1	0	0	1	Relevadores 10 y 11
1	0	1	0	Relevadores 11 y 12
1	0	1	1	Relevadores 12 y 13
1	1	0	0	Relevadores 13 y 14
1	1	0	1	Relevadores 14 y 15
1	1	1	0	Relevadores 15 y 16
1	1	1	1	Relevadores 16 y 17

Tabla IV.3 Codificación para la habilitación de los relevadores de una tarjeta.

De donde D3, D2, D1, D0 son los bits que habilitan a cada uno de los relevadores, esta secuencia es similar para cada una de las cuatro tarjetas. En la figura IV.4 se aprecia un esquema de la interacción relevadores decodificadores para cualquier tarjeta adicional.

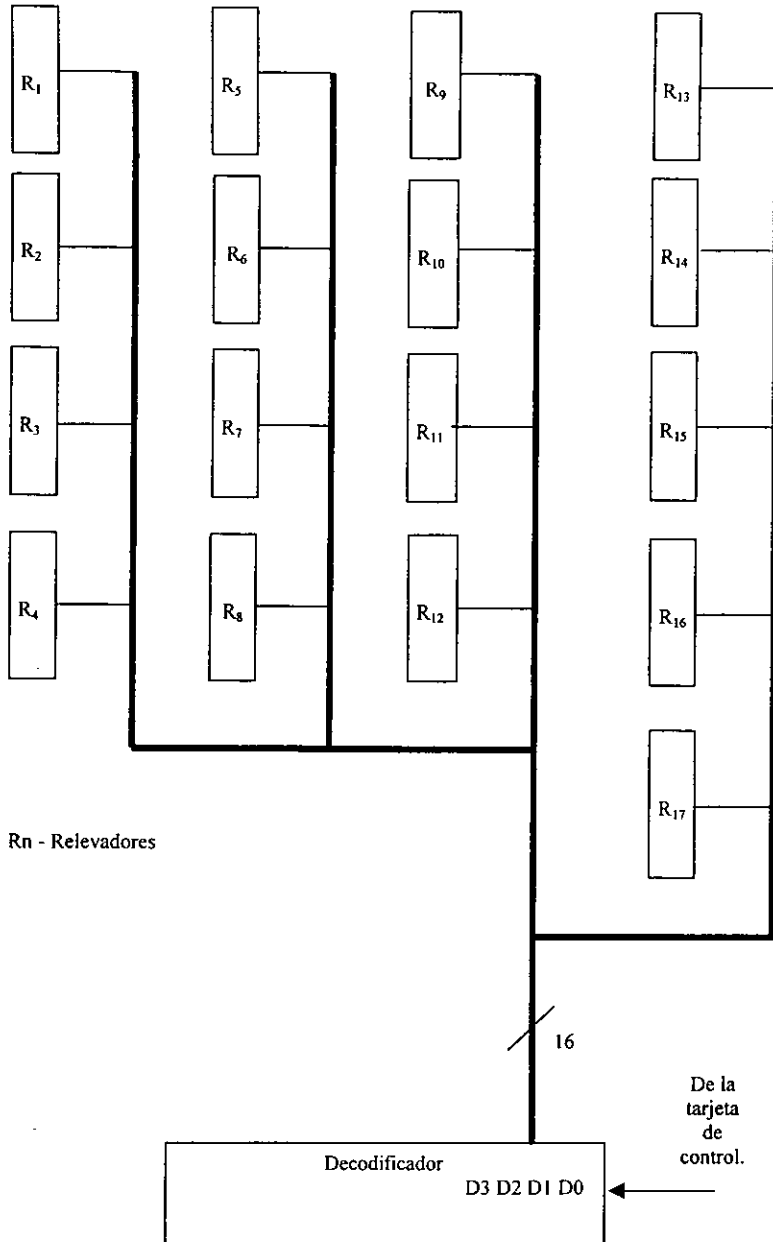


Figura IV.4 Control de los relevadores.

El voltaje de las baterías es escalado, utilizando resistencias, que se encuentran antes de los relevadores, es decir, primero pasa el voltaje por las resistencias de escalamiento y después por los relevadores. Lo que permite manejar niveles de voltaje de +2.5 V a -2.5 V.

En la tarjeta principal del módulo se encuentra otros dos pares de relevadores del tipo *Reed* los cuales son utilizados de la siguiente forma:

Un par de ellos sirve para dar paso a la medición del banco total de baterías, y el otro par se utiliza para direccionar los voltajes provenientes de las tarjetas adicionales. Posteriormente, ya sea el voltaje total, o los voltajes individuales, pasan a través del amplificador de aislamiento, con el fin de evitar ruido y de aislar eléctricamente todo el sistema restante.

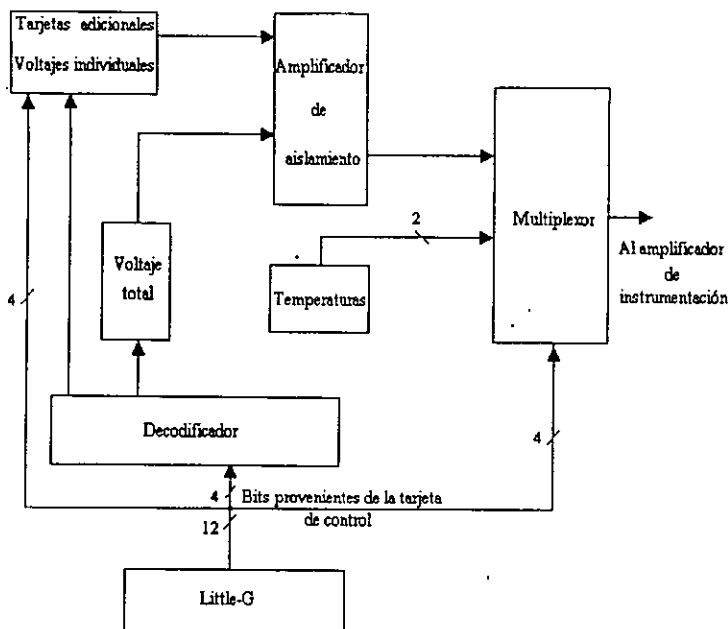
Para realizar el control de lo señalado anteriormente, se utiliza un decodificador de 3x8 74LS138, el que permite seleccionar la tarjeta a utilizar y que par de relevadores de la tarjeta principal accionar.

La tabla IV.4 muestra la codificación que se debe seguir para habilitar las tarjetas.

D6	D5	D4	EN2	Habilita
0	0	0	1	Tarjeta 1
0	0	1	1	Tarjeta 2
0	1	0	1	Tarjeta 3
0	1	1	1	Tarjeta 4
1	0	0	1	Voltaje Total

Tabla IV.4 Codificación para elegir las tarjetas.

Enseguida se encuentra un circuito multiplexor (ADG527A), el cual es utilizado en la conmutación entre las lecturas de voltajes de las baterías y los provenientes de cada sensor de temperatura (LM35). En la figura IV.5 se señala de forma esquemáticamente la operación de esta parte del módulo. La tabla IV.5 muestra la codificación que se debe de seguir para multiplexar cada uno de los canales de interés.



Bits provenientes de la tarjeta de control

Figura IV.5 Control de la tarjeta principal y adicionales.

AD2	AD1	AD0	EN1	Habilita
0	1	0	1	Voltajes
0	1	1	1	Temperatura
1	0	0	1	Temperatura

Tabla IV.5 Codificación de los canales del multiplexor.

Finalmente la salida del multiplexor, que está configurado en forma diferencial, es entregada a un amplificador de instrumentación, para tener una mayor precisión en las lecturas de voltaje escalado; el voltaje de salida es enviado a la tarjeta de conversión analógico-digital.

El voltaje de salida del amplificador de instrumentación (AD624) se encuentra en el rango de +2.5 V a -2.5V. Una fotografía de este módulo se puede observar en la figura IV.6.

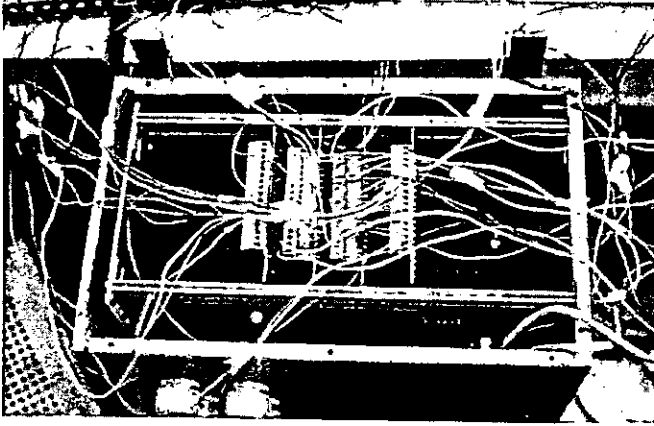


Figura IV.6 Módulo de acondicionamiento.

En el diagrama a bloques, figura IV.7, se muestra la manera en que funciona dicho módulo, así como la interacción con los demás elementos, como se explicó en los párrafos anteriores.

Este módulo se conecta a la *Little-G* por medio de un conector DB 25, cuyas características se explicaran más adelante.

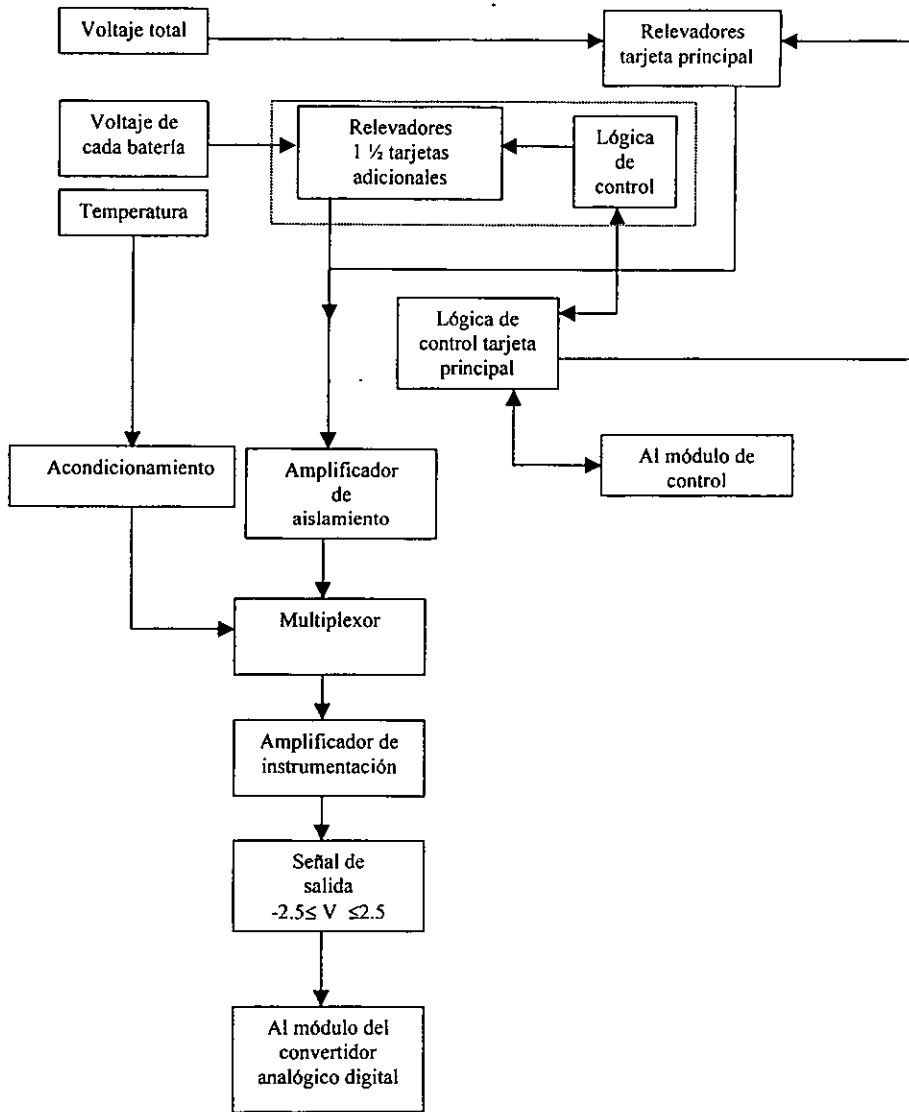


Figura IV.7 Diagrama a bloques que describe el funcionamiento del módulo de acondicionamiento.

IV.4 Tarjeta del módulo de conversión analógico-digital (TMCAD)

El módulo de conversión está formado por la tarjeta de conversión analógico-digital, la cual contiene 16 canales en modo unipolar y 8 en modo bipolar, los cuales se pueden elegir por medio de programación.

Para el sistema en particular, se utilizaron 3 canales en modo unipolar con ganancia unitaria, uno para la señal de velocidad, otro para la corriente que consume el motor, y se dejó como opción un canal adicional para la señal de corriente proveniente de algún sistema de regeneración. En estos casos los voltajes de entrada deben variar de 0 a +2.5 V.

También se utilizó un canal en modo bipolar con ganancia unitaria, en el que se toman las lecturas del voltaje total del banco, el voltaje individual de cada una de las baterías, y las señales provenientes de los sensores de temperatura. Cabe señalar que para este último modo de operación los voltajes de entrada al convertidor varían entre +2.5 V a -2.5V.

Para indicarle a la tarjeta de conversión en que momento actuar y que canal habilitar, ésta se programó a través del *software* de desarrollo de la tarjeta *Little-G*.

En el diagrama de flujo figura IV.8 se muestran los pasos para programar la tarjeta. Se puede observar que primero se tiene que declarar el modo de conversión de la tarjeta, es decir, si se va a leer el voltaje de entrada en forma unipolar o bipolar.

Posteriormente se debe indicar la ganancia que va tener la lectura, ya que ésta puede variar entre 1 hasta 202. Después se declara el número de canal a leer, proporcionándole un tiempo de retardo para asegurar que el canal esté listo y se lleve a cabo la lectura correcta. Finalmente por medio de la instrucción *float ad_conv(int entry, int value)* se regresa el voltaje representado por el valor de 12-bits obtenido en la entrada *value*, asumiendo una escala de 2.5V.

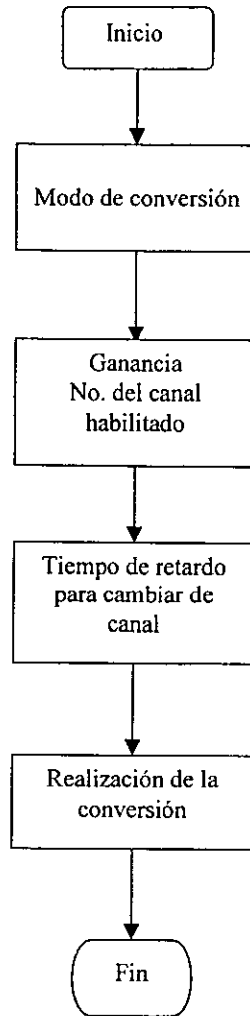


Figura IV.8 Diagrama de flujo de la programación del modulo de conversión analógico-digital.

La tarjeta del módulo de conversión analógico-digital se conecta a través de un cable plano de 20 líneas a la tarjeta del módulo de control *Little-G*, por el puerto *PLCBus*.

IV.5 Tarjeta del módulo de control (*Little-G*)

Para la coordinación de todos los demás módulos, se hace uso del módulo de control integrado por la tarjeta *Little-G*, de la cual se aprovechan las salidas digitales que presenta, la capacidad de memoria instalada, las ventajas de realizar la programación en lenguaje C y la *interface* serial RS-232.

Para el manejo de la tarjeta del módulo de acondicionamiento (TMA) se controla por medio de circuitos decodificadores TTL (apartado IV.2). Además de que se deben controlar los canales del multiplexor. Para realizar tal tarea, ésta se efectúa por medio de las salidas digitales que contiene la *Little-G*, se envían los bits que habilitan y direccionan tanto a los decodificadores como al multiplexor.

Se utilizan 12 de las 14 salidas digitales de la tarjeta, las que primero habilitan al circuito multiplexor del TMA, para direccionar el canal número 2; (del multiplexor) los dos primeros canales no son utilizados por el módulo, ya que están conectados para una aplicación diferente.

Después se habilita el decodificador 74LS138 con el propósito de tomar una lectura del voltaje total, esta lectura se envía al TMCAD, y posteriormente es almacenada en la *Little-G*. Siguiendo con la secuencia, son habilitadas cada una de las tarjetas que toman muestras de cada una de las baterías, de una manera similar ocurre para la conversión y el almacenamiento. Los siguientes canales que se habilitan del multiplexor son el número 3 y 4, para tener las lecturas de temperatura, enviándose al TMCAD y posteriormente almacenándolos.

La secuencia de operación que sigue se basa, principalmente, en direccionar la variable y una vez que se encuentra lista dicha variable se realiza la conversión analógica-digital. En el diagrama de flujo de la figura IV.9 se puede observar la secuencia que se sigue para llevar a cabo el control de la tarjeta.

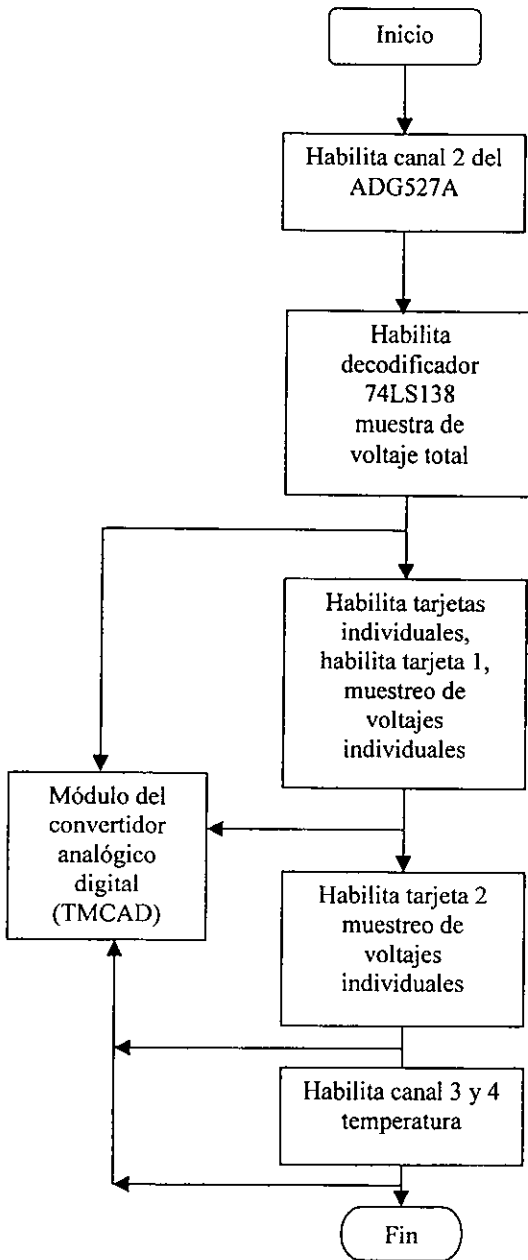


Figura IV.9 Diagrama de flujo de control del módulo de acondicionamiento.

IV.6 Diseño de la tarjeta del módulo externo de almacenamiento (TME)

El módulo de adquisición y almacenamiento de datos cuenta con una capacidad de hasta 450 Kbytes de almacenamiento.

Sin embargo una de las desventajas que se presenta en el almacenamiento de datos, es que cuando ocurra alguna situación, en la que no se pudiera contar con una PC, o que se perdiera el suministro de energía en el módulo, la transmisión de datos no podría realizarse. Es en este momento cuando se buscan alternativas para poder solucionar la contingencia. Se puede pensar en algún medio por el cual tuviera que prescindirse, en algún momento, de una PC para poder realizar el acopio de información para su consecuente proceso.

Una de las opciones que se consideran es el uso de las tarjetas de memoria *Thin Card Random Acces Memory (TCRAM)*, las cuales se mencionaron en el capítulo III. Estas tarjetas son del tamaño de una tarjeta de crédito, y pueden removerse del módulo sin necesidad de que el mismo esté alimentado.

La tarjeta *TCRAM* presenta la ventaja de un dispositivo RAM capaz de retener información gracias a que cuenta con una batería de respaldo, aunado a que la presentación de esta memoria (muy parecida a un disco flexible) nos permite transportarla muy fácilmente al lugar donde se terminara el proceso de los datos; en este caso en vez de llevar una PC al módulo para respaldo, llevamos los datos a una PC para su proceso.

Otra de las ventajas de la *TCRAM* sobre la memoria del módulo, es su capacidad, ya que se cuentan, en el mercado, con tarjetas de hasta 32 Mbytes, con lo cual se incrementa el alcance de respaldo. El problema principal de poder usar esta herramienta en un sistema diseñado, como lo es la *Little-G*, son las características que se requieren para su funcionamiento.

La tarjeta *Little-G* trabaja con un *bus* muy particular (*PLCBus* apartado III.2.3) y la forma en que accesa a sus dispositivos periféricos lo es también. Esto hace necesario la

construcción de algún medio que permita la interacción de ambos elementos. Este dispositivo, llamado *interface*, deberá contar con la capacidad tanto de adecuarse a la forma de comunicación de la tarjeta *Little-G* con sus módulos, como a la forma de trabajar de la *TCRAM*.

Entre las necesidades se tiene que considerar que la tarjeta *TCRAM* requiere:

- 24 líneas de dirección (ADR0-ADR23)
- 8 líneas de datos (D0-D7)
- al menos 4 líneas de control, (CE1, CE2, W, R)

Teniendo un requerimiento total de 36 líneas para poder manejar la *TCRAM*.

La tarjeta de control dispone de:

- un bus bidireccional para datos y direcciones de 8 líneas (D0-D7)
- 3 líneas de codificación de las instrucciones (A1-A3)
- una línea de sincronía para sus instrucciones (STR)

Con base en estas características se procede a diseñar la *interface*.

La *interface* a diseñar estará integrada alrededor de un microcontrolador. El microcontrolador que se escogió es el *COP8SAC*, que tiene la característica de contar con un número elevado de terminales, los cuales al ser independientes en su forma de selección, permiten hacer una distribución más eficiente.

El microcontrolador cuenta con 44 terminales, de las cuales se usan 4 distribuidos de la siguiente forma:

- 2 para polarización del circuito
- 1 en el reloj externo
- 1 en el *reset*

Quedando 40 líneas que se pueden utilizar para realizar el control de la *TCRAM*. La distribución del uso de estas terminales se lleva a cabo de la siguiente manera:

- 8 líneas de transmisión y recepción de datos (bus de datos bidireccional, E/S)
- 24 líneas de direccionamiento (acceso hasta 8 Mbytes de memoria, salidas)
- 3 líneas de codificación del módulo (A1-A3, entradas)
- 1 línea de sincronía (STR, entrada)
- 4 líneas de control de la tarjeta (CE1, CE2, W, R, salidas)

Al parecer el problema ya está resuelto, debido a que el dispositivo seleccionado cumple con las características, del número de terminales, pero ahora analizando las velocidades con que trabajan los dispositivos se encuentra con que son muy diferentes.

Dentro de la forma de trabajar de la *Little-G* está la transmisión de los datos. El modo de direccionamiento de los periféricos es por medio de direcciones de 12-bits, esto indica que los datos con los que trabaja el módulo son de 12-bits. Recordando que el *bus* es bidireccional.

La forma en que se lleva a cabo el intercambio de datos en el *PLCBus* es por medio de bloques de 4-bits, llamados *nibbles*. Para poder llevar a cabo este intercambio, se hace necesario presentar una forma de codificación que permita distinguir entre los bloques de información. Esta codificación se hace por medio de 3 líneas de control A1-A3, las cuales nos permite identificar el proceso.

Otra línea que se hace presente es la de sincronía con la que el módulo indica que se está llevando a cabo una operación. Y por último, las ocho líneas por donde se presenta el dato.

Los tres eventos descritos se presentan según la secuencia mostrada en la figura IV.8. El primer evento en presentarse es la línea de sincronía (*strobe*), la cual tiene una duración de 900 ns aproximadamente, e indica que se está llevando a cabo un evento. Los dos eventos restantes son simultáneos, y se presentan 450 ns después de iniciar el de sincronía., los tres

primeros bits son los de decodificación y los ocho restantes son el dato a transmitir. La duración de los dos eventos es de 450 ns, es decir al terminar el bit de sincronía pierde todo significado la información presente en el *bus*. Este proceso se ejemplifica en la figura IV.10 y fue obtenido utilizando un analizador lógico.

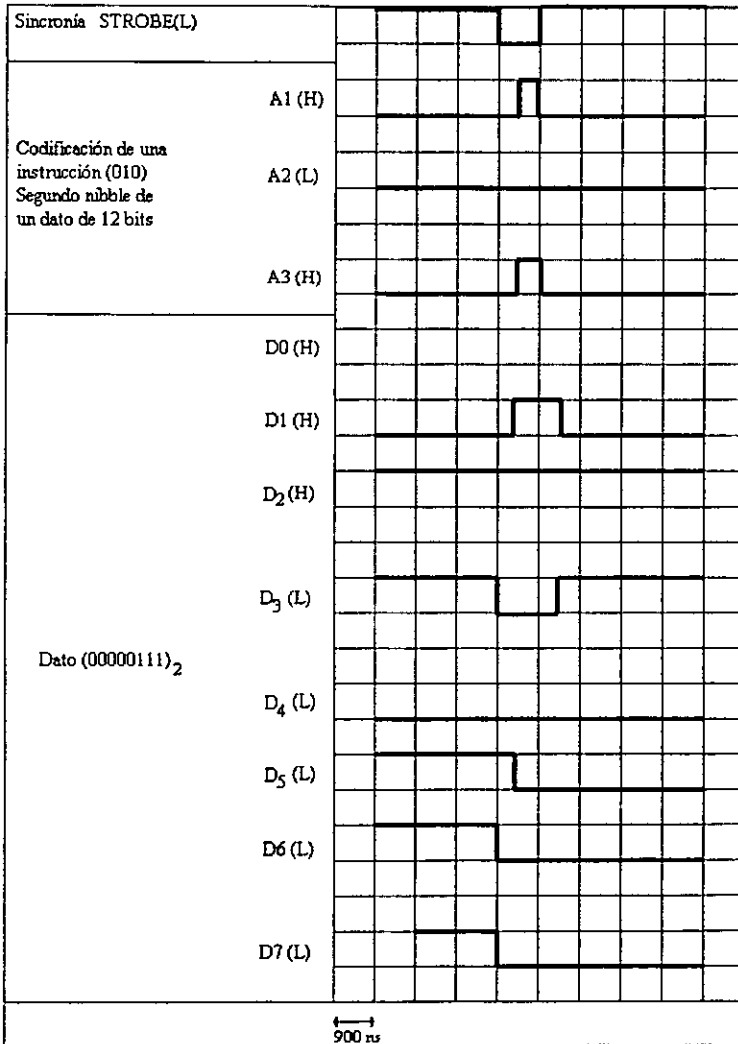


Figura IV.10 Diagrama de tiempos de un proceso de intercambio de datos en el *PLCBus*.

En la figura se observa que la acción presentada consiste en el envío del segundo *nibble* de un dato de 12-bits $(B7D)_H$, y el *nibble* transferido es el 7 $(00000111)_2$.

El *COP8SAC* trabaja con una cristal de cuarzo a una frecuencia de 10MHz, lo que da un ciclo de máquina de $1\mu s$, esto deja ver que la instrucción que menos ciclos de máquina requiere, que es 1, tarda $1\mu s$ para realizarse, con lo cual ya no se cumple con la especificación para el intercambio de información.

Tomando en cuenta esto y que el *bus* debe ser bidireccional, se llega a la conclusión que necesitamos de circuitería adicional para llevar a cabo el intercambio de información.

Para lograr lo anterior proponemos el uso de un *latch* para poder disponer de los datos el tiempo necesario. Después el uso de un buffer tres estados para evitar el choque de información. Con estos dos dispositivos lo que obtenemos es un circuito transmisor-receptor (*tranceptor*) y se muestra en la figura IV.11.

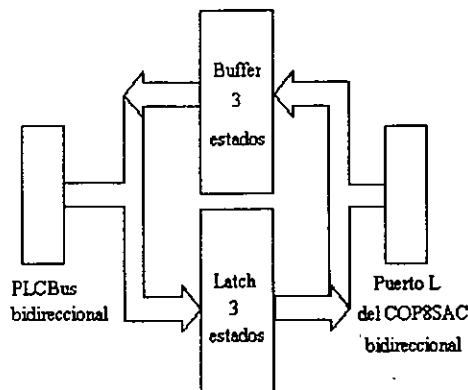


Figura IV.11 Diagrama a bloques del circuito tranceptor.

Para poder tener control sobre el circuito trancceptor que se acaba de describir, se necesita adicionalmente del uso de una lógica de control, lo que permitirá establecer el flujo de información de la manera que más convenga.

Esta lógica de control se llevó a cabo considerando nuevamente los recursos disponibles hasta este momento y tratando de evitar el uso adicional de líneas de control. Para ello tomamos las líneas de codificación del puerto del módulo y la línea de sincronía del mismo, y así se puede ver en la tabla IV.6.

W	Entradas			Salidas		
	A3	A2	A1	Enable Buffer	Output Control	Enable Latch
1	1	0	0	1	0	H
1	1	0	1	1	0	H
1	1	1	0	1	0	H
1	0	0	0	0	1	X
1	0	1	1	1	0	H
1	1	1	1	1	0	H

Tabla IV.6 Relación de entradas salidas necesarias para el diseño de la lógica de control.

Finalmente se llevó a cabo el diseño de la lógica de control que se utilizó, en la que empleamos una compuerta AND, una compuerta OR, y una compuerta NOT. En la figura IV.12 podemos observar el circuito lógico.

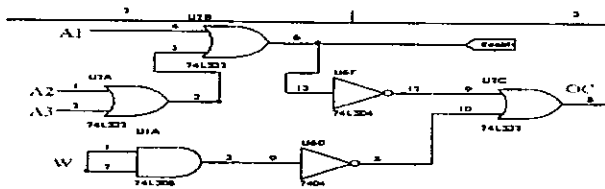


Figura IV.12 Lógica de control de la tarjeta de almacenamiento.

La integración final del módulo de almacenamiento externo cuenta entonces con los siguientes elementos: un circuito trceptor, una lógica de control, el microcontrolador y la tarjeta TCRAM. Un diagrama a bloques del circuito final es mostrado en la figura IV.13.

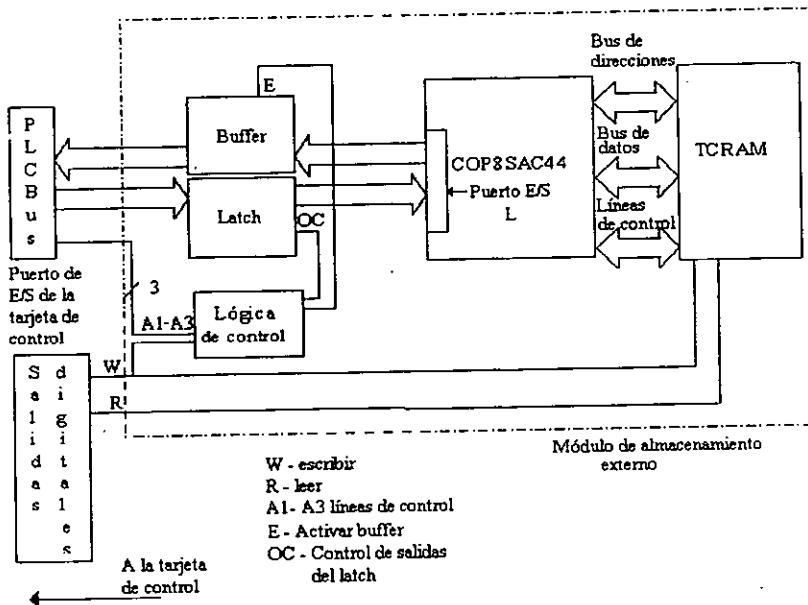


Figura IV.13 Diagrama del módulo de almacenamiento externo.

La figura IV.14 nos muestra la *interface* entre el módulo de almacenamiento y la Little-

G.

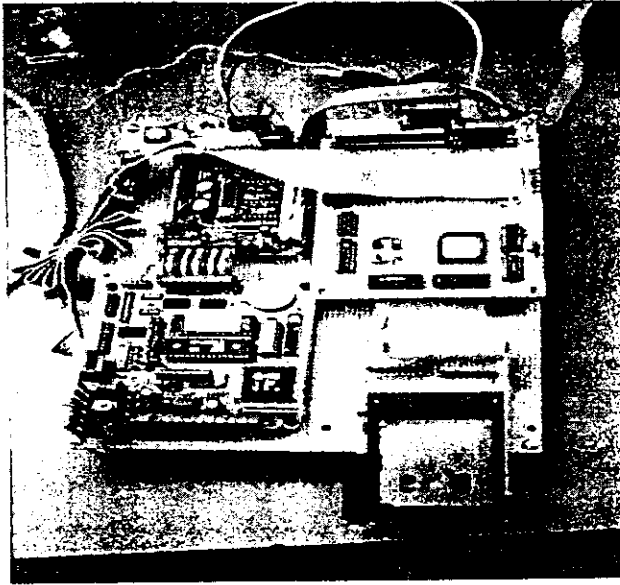


Figura IV.14 *Interface* entre el módulo de almacenamiento y la tarjeta de control.

IV.7 Conexiones

Como parte de la integración del SADME se necesita realizar las conexiones físicas correspondientes entre cada uno de los módulos del sistema y con el banco de baterías. Los módulos de control (*Little-G*), el TMCAD y parte del TME se conectan por medio de los puertos *PLCBus*, esto se logra utilizando conectores planos de 20 terminales.

En la figura IV.15 se observa que los números dentro de la foto indican:

- El número 1 señala el cable plano de 20 terminales correspondiente al puerto *PLCBus* entre la *Little-G* y el TMCAD.
- El número 2 indica el cable plano del *PLCBus* al TME.
- Con respecto al número 3 es un cable plano de 40 terminales del TME al *socket* donde se inserta la *TCRAM*.

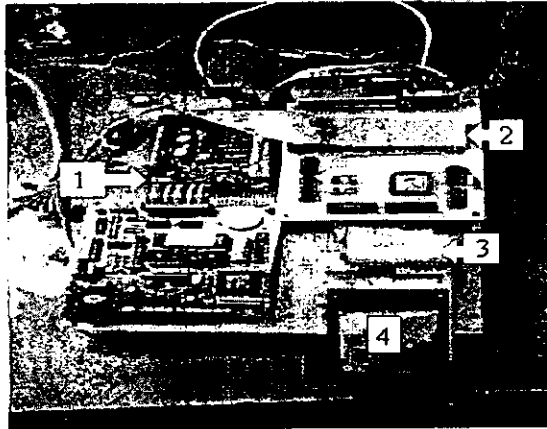


Figura IV.15 Conexiones de los módulos.

Entre el módulo TMA y la *Little-G* se utiliza un conector del tipo DB-25, en tabla IV.7 se muestra la configuración de terminales.

Terminal	Asignación	Terminal	Asignación
1	15 V	15	D2
2	-15 V	16	D3
3	+5V	17	D4
4	+5V	18	D5
5	GND	19	D6
6	GND	20	EN2
7	GND	21	GND
8	AD0	22	GND
9	AD1	23	GND
10	AD2	24	Salida
11	EN1	25	GND
12	NC	-	-
13	D0	-	-
14	DI	-	-

Tabla IV.7 Terminales del conector DB-25.

En la figura IV.16 se señala el conector DB25 con el número 1, el número 2 corresponde a un conector DB9 que sirve para realizar la comunicación serial hacia la PC.

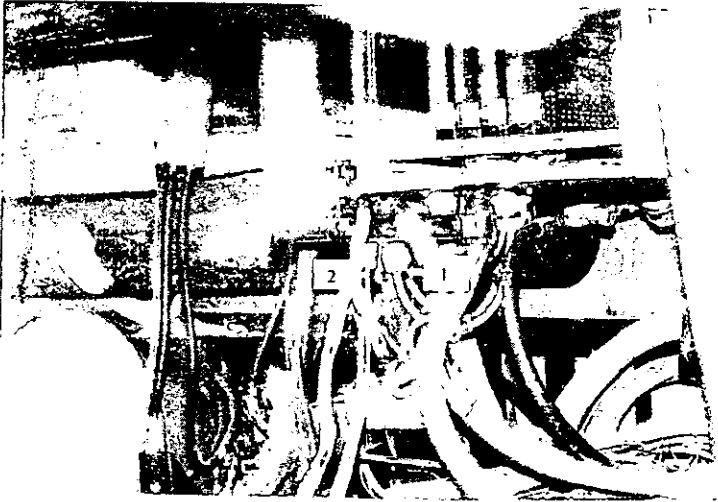


Figura IV.16 Conectores DB25 y DB9.

En la figura IV.17 los números indican que:

- 1) Agrupamiento de los módulos *Little-G*, TMCAD y TME.
- 2) Módulo de acondicionamiento TMA.
- 3) Conector CPC de 37 terminales proveniente del banco de baterías.
- 4) Conector CPC de 4 terminales proveniente de los sensores de temperatura.

Cabe aclarar que existen otros 3 conectores de 4 terminales CPC, uno es utilizado para la alimentación del SADME y otro para las señales de corriente y velocidad, el último para la alimentación. Para su asignación de terminales ver la figura IV.26.

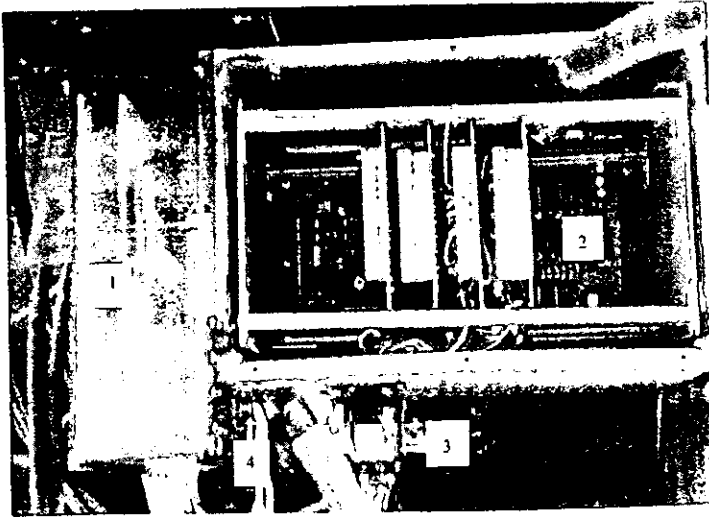


Figura IV.17 Conectores CPC y módulos.

Después de haber revisado la parte física del SADME haremos una referencia al *software* del sistema.

IV.8 *Software* del sistema

En este apartado mencionaremos los programas realizados para la interacción entre los diferentes módulos así como el *software* desarrollado para el usuario.

Estos se pueden dividir en los siguientes:

- Programa del módulo de control (PRODEC)
- Programa del módulo de almacenamiento externo (PROMOX)
- Programa de *interface* de usuario (SISTEMA26)
- Programa de conversión de archivo (PROCAR)

De los programas listados daremos una descripción de cada uno de ellos. Iniciaremos con la del PRODEC, en el apéndice B se muestran los listados de programa para una referencia más completa.

IV.8.1 Programa del módulo de control (PRODEC)

Este programa lo podemos dividir básicamente en las siguientes rutinas:

- **Ciclo principal.**
 - Comunicación serial.
 - Control de relevadores, decodificadores y multiplexor.
 - Rutina de la TMCAD.
 - Almacenamiento de datos.
 - Transmisión de datos.
 - Almacenamiento externo de datos.

- **Ciclo principal**

En este ciclo se anidan cada una de las rutinas del programa; en la figura IV.18 se observa que primero se necesita inicializar los parámetros. Estos parámetros son propios del programa ya sean variable, constantes, nombre de rutinas, llamado a librerías, etcétera.

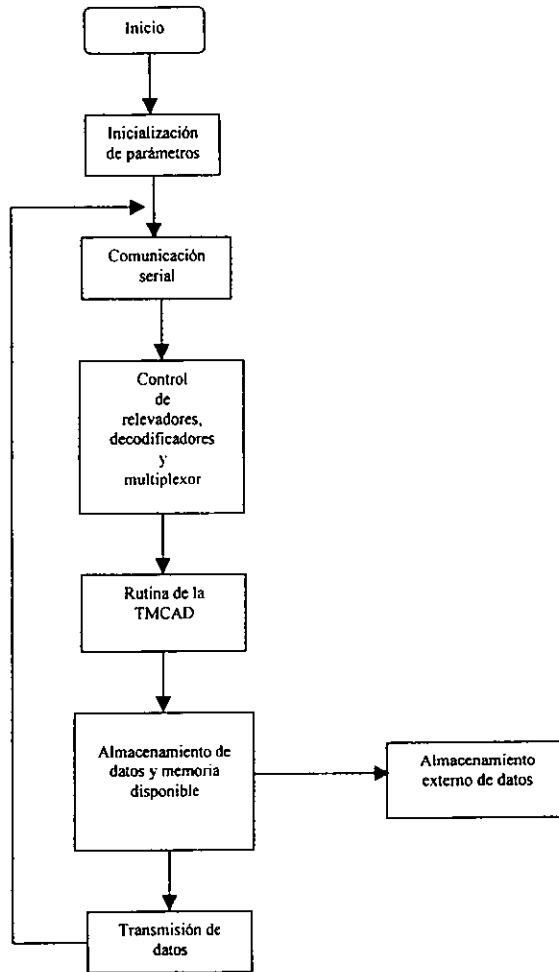


Figura IV.18 Ciclo principal del programa.

Ahora describiremos cada una las rutinas que componen al ciclo principal.

- **Comunicación serial**

Para realizar la comunicación entre el módulo de control y la computadora personal, se necesita explicar los parámetros con que se programó el módulo de control. Entre los parámetros de los cuales se hicieron uso están los que a continuación se mencionan, transmitir

8 bits sin paridad, con una velocidad de 19 200 bps, un bit de paro, se deshabilitó la conexión por módem y no se realizó eco después de la transmisión, debido a que la configuración para la comunicación que se utiliza es la mínima posible. Por otra parte la forma en que se programó la aplicación de la transmisión se aprecia en la figura IV.19, y es como sigue:

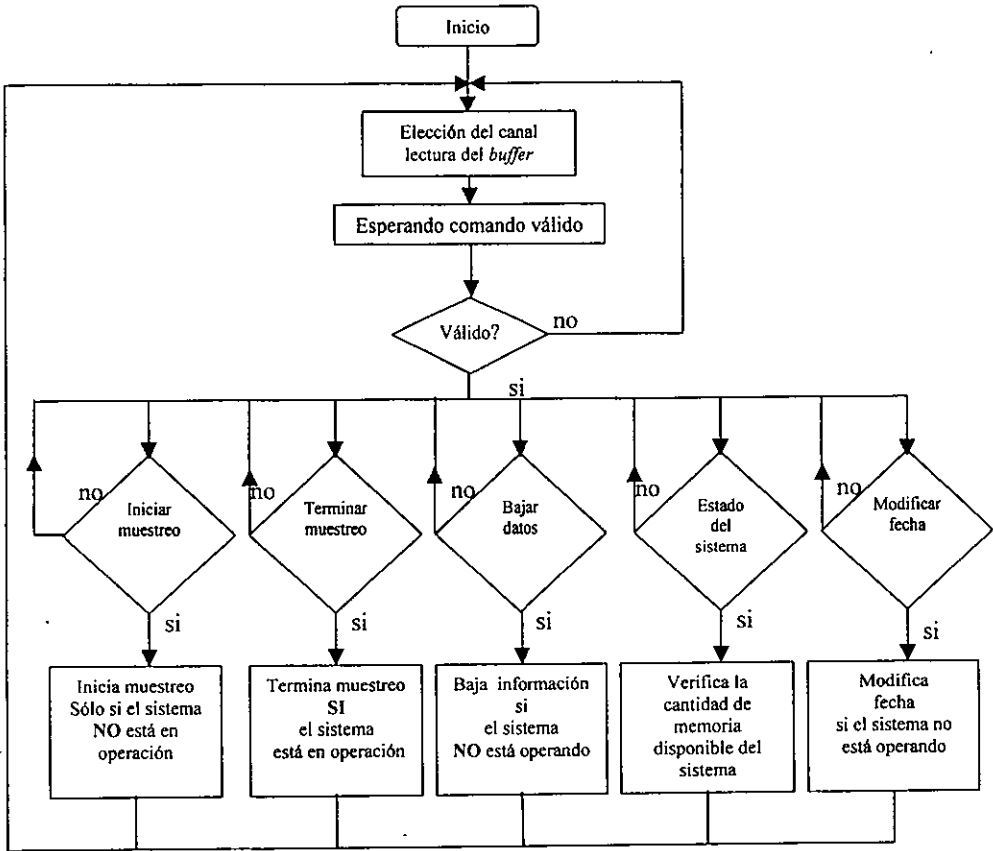


Figura IV.19 Diagrama de flujo de la comunicación de la tarjeta.

Primero se habilita algún canal de comunicación, en este caso el canal número 1, después se lee el *buffer* de recepción para esperar algún comando válido proveniente de la computadora personal, en caso de no ser correcto el sistema sigue esperando, de forma contraria, se evalúa el comando para determinar que acción es la que debe ejecutar el módulo de control. Después de lo cual el módulo sigue en espera de algún otro comando válido, estos

comandos son caracteres que se definen de antemano. Los comandos a los que se hacen referencias son los enlistados en la tabla IV.8.

Carácter	Función	Transmite
2	Inicia el muestreo	"iniciado"
3	Termina el muestreo	"Fin"
4	Acceso a los datos en memoria.	Carácter de inicio(y), hora y fecha del inicio del muestreo, bloque de 32 datos
5	Indica el estado de operación del sistema	En caso de estar en operación transmite: "A,horafecha,Nº muestra " donde la "A" indica que se encuentra en operación.
		En caso de estar en espera y con datos en memoria. "B,D,Nº muestra" donde la "B" indica que se encuentra en espera, la "D" con datos en memoria.
		En caso de estar en espera y sin datos en memoria: "B,C" donde "C" indica sin datos en memoria.
A	Envía el siguiente bloque de datos	Bloque de 32 datos
C	Cancela el envío de datos	"Findatos"
R	Restablece al sistema con sus parámetros iniciales.	
M	Comando de prueba	"Menu"
T	Transmite la fecha y hora que tiene el sistema	Hora y fecha
M	Se asume que la fecha y la hora son correctas	"M"
S	Se espera modificación de fecha a través de una cadena de caracteres.	"día" cuando se terminó de modificar la fecha y la hora.

Tabla IV.8 Comandos del sistema.

Control de relevadores, decodificadores y multiplexor

En esta rutina se utiliza la instrucción *up_setout(canal, estado)* para designar uno de los 14 canales que tiene la tarjeta *Little-G* y determinar si la salida debe estar en alto (5 V) o bajo (0 V), esto se hace basándose en las tablas IV.3, IV.4, IV.5. Lo que nos da un total de 12 salidas que se traduce en la utilización de 12 de los 14 canales disponibles. Una vez que se ha seleccionado la variable a muestrear se pasa a la rutina del TMCAD que se describe enseguida.

- **Rutina del TMCAD**

Cuando el programa llama a esta rutina se procede a realizar la conversión de la variable se escala y se envía para su almacenamiento, como se mencionó en los apartados IV.4 y IV.5.

- **Almacenamiento de datos**

El módulo de control contiene una memoria RAM de hasta 512 Kbytes, en donde son almacenados los datos obtenidos. Sin embargo, existe la restricción del tamaño de la memoria, en el sentido de almacenar la mayor cantidad de datos posibles, por lo que se trata de optimizar la capacidad de la misma. Después que se ha realizado la conversión, el resultado es guardado en una variable del tipo flotante, la cual ocupa 4 bytes de memoria, reduciendo así la capacidad de almacenamiento. Por lo que se realiza un truncamiento al tipo *char* que sólo ocupa un byte.

De los 512 Kbytes, 62Kbytes son utilizados para el programa y 450 Kbytes como memoria de almacenamiento de los datos. Estrechamente relacionado con la capacidad de la memoria se encuentra la cantidad de muestras, por cada una se almacenan las 32 variables adquiridas lo que equivale a ocupar 32 bytes de memoria, lo que se representaría como 32 bytes/muestra, y si nuestra capacidad de almacenamiento es de 450 Kbytes entonces esto implica que el número de muestras antes de agotarse la memoria esta dada por la siguiente relación:

$$C.A. = \frac{450Kbytes}{32 \frac{bytes}{muestra}} = \frac{450 * 1024bytes}{32 \frac{bytes}{muestra}} = \frac{460800bytes}{32 \frac{bytes}{muestra}} = 14400muestras$$

Donde C.A. representa la capacidad de almacenamiento por muestra. Ahora, si cada muestra se realiza en un segundo, entonces el tiempo necesario para llenar la memoria es de 14400 segundos.

Si esto lo dividimos entre 3600 para obtener el número de horas, tenemos que:

$$C.T. = \frac{14400segundos}{3600} = 4horas$$

Donde C.T. representa la capacidad de almacenamiento por tiempo. Una vez llegado a ocupar totalmente la capacidad de la memoria, y si no se han transferidos datos a la computadora personal, el sistema empieza a sobrescribir la memoria, iniciando un nuevo muestreo.

Cabe aclarar él porque se realiza el muestreo de las 32 variables en 1 segundo, aunque cada canal es leído de manera muy rápida, se debe dar un tiempo de retardo para que cada canal este habilitado, además que se consideró sacar un promedio de muestras (35 muestras por lectura) por cada batería para minimizar la posibilidad de que ocurra una lectura errónea.

- **Transmisión de datos**

Otro aspecto importante es en el momento en que se transfieren los datos almacenados en la tarjeta a la computadora personal, pues se realiza el envío en bloques de 32 datos, esto es con el propósito de evitar algún error en la transmisión de datos, para posteriormente aplicarles los pasos correspondientes en el programa de usuario y así obtener el valor real.

- **Almacenamiento externo de datos**

Cuando esta rutina es llamada se inspeccionan que el módulo de almacenamiento externo esté presente, esto lo realiza la *Little-G* por medio de las entradas digitales, si se lee un nivel lógico bajo (0V) se considera que el módulo de almacenamiento externo está listo.

Por otro lado, cuando se revisa un apuntador de memoria que indica la cantidad de memoria ocupada, el SADME detiene el muestreo si se está realizando para transferir los datos al TME.

Esta rutina se subdivide en otras 3 subrutinas que son:

- Inicialización de la TME
- Determinación de la acción a realizar
- Escribir o leer datos desde la *TCRAM*

Inicialización de la TME

En esta subrutina el programa manda el comando de *reset* al *PLCBus* y así prepara la tarjeta para su eventual uso. Después transmite el código de reconocimiento de la TME, 12-bits en 3 *nibbles* de 4-bits. Estas funciones fueron implementadas en lenguaje ensamblador y consisten en mandar el código de direccionamiento y el dato de reconocimiento. Después se presenta un retardo para dar tiempo al TME de hacer el reconocimiento del mismo. Este proceso se repite 3 veces para poder mandar los 3 *nibbles*. La figura IV.20 lo ejemplifica.

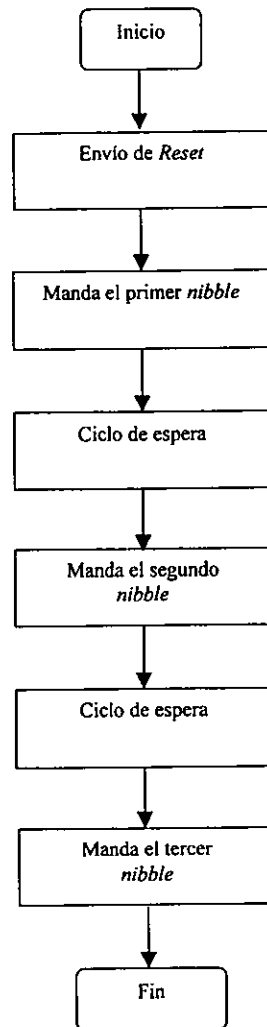


Figura IV.20 Diagrama de flujo de la rutina de reconocimiento.

Determinación de la acción a realizar

Cuando el código es recibido por la TME el programa transmite un código en el cual se indica la acción a realizar. Esta acción se realiza por medio de la instrucción de escribir al *PLC Bus* y en el *bus* de datos se pone un código 0x00 para lectura y 0x01 para escritura. Esta acción se realiza para preparar la TME para cualquiera de estas acciones y el programa en la tarjeta de control ya conoce la acción de manera anticipada.

Escribir o leer datos de la *TCRAM*

Esta parte del programa realiza cualquiera de dos acciones: leer datos desde la *TCRAM* o escribir los datos en ella.

Cuando la tarjeta del módulo de control adquiere datos lo hace primero en su área de memoria. Cuando es alcanzada la capacidad de almacenamiento, la tarjeta de control dejará el proceso de adquisición y vaciará los datos en ella contenida a la tarjeta TME. Esta acción es realizado por medio de bloques de 64Kbytes.

La tarjeta del módulo control transmite los datos a la TME en formato de 8 bits y por medio de su instrucción de escritura al *PLCBus*. La tarjeta lleva un contador en el cual se va llevando el registro de los datos transmitidos. Al transferir un dato la tarjeta del módulo de control entra en un *loop* del cual saldrá solamente cuando reciba un código que manda la TME para avisarle que esta en disposición de recibir el datos siguiente. Cuando un bloque es enviado completamente se continua con el siguiente, cuando se termina de enviar los bloques la tarjeta de control vuelve al proceso de adquisición.

Este proceso es un ciclo que termina cuando la memoria esta completamente llena. La figura IV.21 muestra en un diagrama de flujo este proceso.

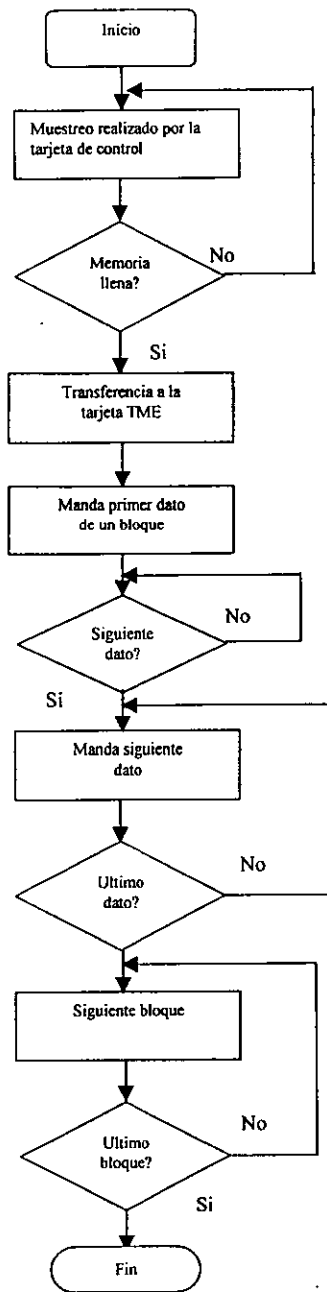


Figura IV.21 Subrutina para escribir en la TCRAM.

IV.8.2 Programa del módulo de almacenamiento externo (PROMOX)

La tarjeta del módulo externo de almacenamiento está programada en lenguaje ensamblador para el microcontrolador *COP8SAC44*. Este programa consta de tres partes en especial, la primera parte de reconocimiento de la tarjeta, la segunda de reconocimiento de la acción a realizar y la tercera de la acción específica.

La tarjeta está conectada de manera permanente en el *PLCBus* de la tarjeta del módulo de control, en paralelo con la tarjeta *TMCAD*, y cada una de ellas funciona de manera exclusiva, esto quiere decir, que cuando cualquiera de ellas está funcionando, la otra tarjeta no lo hace.

Con base en el diagrama de flujo presentado en la figura IV.22, el procedimiento de operación es el siguiente:

a) Cuando el sistema así lo requiere, establece comunicación con la tarjeta, esto lo realiza mediante un código único (ver tabla III.3) en el cual manda la dirección conformado por 3 bits que corresponde a ella.

b) Después de detectar el requerimiento, la tarjeta retorna un dato de control (0xDE), que a su vez indica que está lista a recibir la nueva orden.

c) La nueva orden recibida (0 escritura, ó 1 lectura), indicará si el proceso es de lectura o escritura a la *TCRAM*. Para el proceso de escritura se realizan los siguientes pasos:

- Como el proceso de transferencia de datos se realiza por bloques, primero se carga la última dirección en la cual se almacenaron datos, esta información se almacena en algunas localidades de la memoria.
- Después de esto la tarjeta controladora enviará el dato a guardar, con un código específico para esto (ver tabla III.3), el cual después de ser recibido por el *COP8SAC44* se almacenará en la tarjeta de memoria.

- Después de almacenar el dato se verifica si se ha alcanzado la capacidad de la tarjeta, y siendo el caso detener la acción de escritura.
- A continuación el programa revisa que el bloque haya sido enviado en su totalidad.
- Si ya se terminó la transferencia del bloque, el sistema guarda nuevamente la última dirección en la que se escribió para que una futura transferencia se continúe ahí mismo.

Para el proceso de lectura a la tarjeta se realizan los siguientes pasos:

- Se inicializa el *bus* de direcciones.
- Se lee el dato de la localidad.
- Se incrementa el apuntador.
- Se revisa que se haya terminado ya la acción.

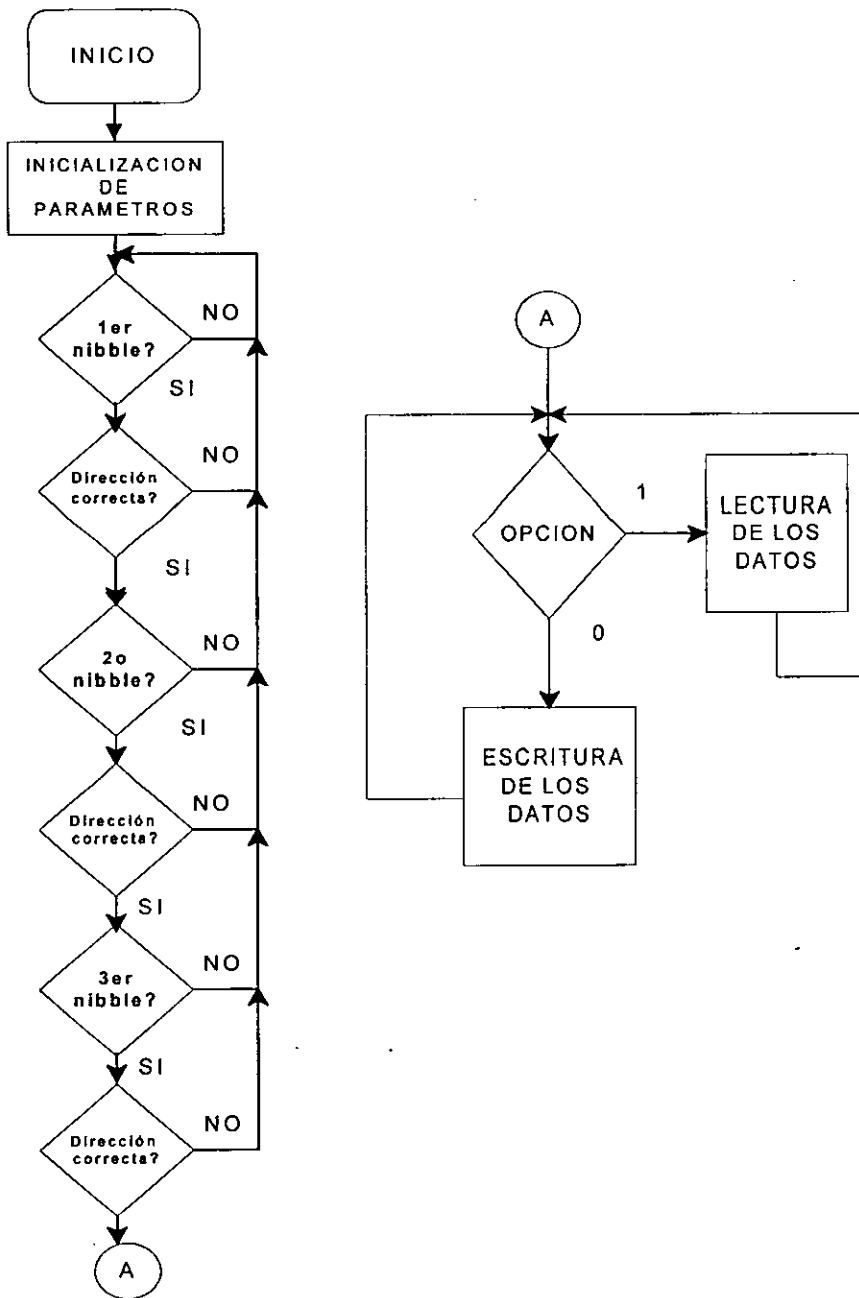


Figura IV.22 Diagrama de flujo del programa de la *interface* de la tarjeta de almacenamiento.

IV.8.3 Programa de usuario Sistema26

El programa Sistema26, es un *software* desarrollado con el propósito de ser una *interface* amigable entre el SADME y el usuario que tenga a su cargo el monitoreo de las diversas variables que tiene el Minibús Eléctrico. Este programa fue desarrollado bajo el lenguaje de programación llamado *Visual Basic 4*, siendo éste un lenguaje orientado a objetos.

Se utilizó tal lenguaje debido a que hoy día la gran mayoría de las aplicaciones (programas) que se ejecutan en las PC's actuales trabajan bajo el ambiente *Windows*, por lo que se hace necesario realizar un *software* compatible con las tendencias que en este momento se presentan, y *Visual Basic 4* nos permite desarrollar dicha labor.

La filosofía que se siguió en el desarrollo del *software*, fue el que se realizaran todas las rutinas necesarias para el completo control del SADME y que fueran lo más sencillas posibles. Para esto es necesario tomar en cuenta lo siguiente:

- Comunicación RS-232 con el SADME.
- Verificación del estado que guarda el SADME.
- Cantidad de memoria disponible del SADME.
- Controlar el inicio y fin de cada muestreo, así como el de poder restablecer el sistema y decidir en que momento transmitir los datos.
- Recibir los datos que transmite el SADME y su posterior procesamiento para obtener el valor real de las variables y su almacenamiento.
- Modificación de parámetros, como: puerto en que se encuentra disponible en la PC, modificación de hora y fecha del SADME, cancelación de la transmisión de datos.
- Poder guardar los datos en un archivo de texto con el propósito de realizar su análisis en alguna hoja de cálculo.

La secuencia básica que debe seguir la estructura del *software* se observa en el diagrama de la figura IV.23.

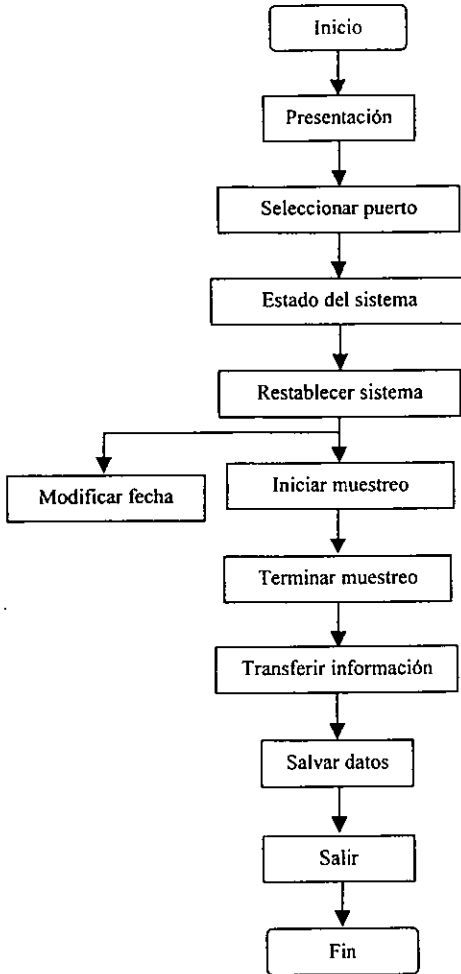


Figura IV.23 Secuencia básica del programa.

Presentación

Esta es solamente la parte introductoria para iniciar la sesión del *software*.

Seleccionar puerto

En esta rutina se eligen los parámetros para establecer la comunicación, éstos están programados por anticipado y lo que se le pide al usuario es seleccionar alguno de los puertos serie de la PC.

Entre los parámetros de los cuales se hicieron uso están los que a continuación se mencionan, transmitir 8 bits sin paridad, con una velocidad de 19 200 bps, un bit de paro, se deshabilitó la conexión por módem.

Cuando el usuario elige alguno de los puertos disponibles de su PC, esta rutina verifica que por el puerto se encuentre conectado el SADME, ya que este último envía una cadena de caracteres preestablecidos con el propósito de que se pueda verificar la comunicación.

En caso de que se reciba una cadena de caracteres (ver tabla IV.8) se verifica su validez y se prosigue su procesamiento para establecer el estado del SADME. Esto con la intención de controlar las demás rutinas y poder habilitar los botones necesarios del Sistema26. Si lo anterior no se llega a cumplir se envía un mensaje de error.

Estado del sistema

Cuando se ha verificado que el sistema está presente y por tanto se ha establecido la comunicación, esta rutina es habilitada con la intención de que el usuario pueda verificar el estado del SADME. Esto se realiza a partir de enviar una serie de caracteres preestablecidos (ver tabla IV.8) y que por tanto, el SADME los reconoce, de donde el sistema envía una nueva cadena de caracteres y estos son recibidos y procesados, desplegando enseguida el estado en que el SADME se encuentra.

Modificar fecha

Esta rutina se habilita siempre y cuando el sistema no se encuentre haciendo un muestreo o con datos en memoria. El objetivo es verificar y en su caso modificar la fecha del SADME, resulta de gran importancia ya que al iniciar algún muestreo la fecha y la hora son almacenadas en el SADME.

Restablecer sistema

Para el caso en que el usuario decida reinicializar el sistema, se ha implementado esta rutina, la cual manda al SADME a sus parámetros iniciales.

Iniciar y terminar muestreo

Cuando se ha determinado cual es el estado actual del sistema, alguno de estas dos rutinas son habilitadas, siendo de forma simple enviar un caracter de inicio y de fin (ver tabla IV.8) previamente determinado.

Transferir información

Esta rutina solamente es habilitada cuando se ha detenido el muestreo, esto es con el propósito de que se pueda realizar de manera más rápida la transferencia de la información, ya que el sistema esta solamente dedicado a bajar los datos. El usuario tiene las opciones de: detener el envio de datos o iniciar un nuevo muestreo.

Cuando los datos son totalmente transferidos a la PC éstos son almacenados en una memoria temporal para su posterior adecuación y almacenamiento.

Salvar datos

Una vez que los datos son transferidos a la PC, son procesados y posteriormente almacenados en un archivo de texto. Para que después puedan ser analizados en una hoja de cálculo.

IV.8.4 Programa de conversión de archivo (PROCAR)

Como la manera en que se almacena la información en la tarjeta *TCRAM* es igual a como se guarda en una memoria de estado sólido, es necesario darle un formato adecuado con el cual se pueda procesar en una hoja de cálculo.

La información de la *TCRAM* es transferida a un archivo de texto por medio del *software* de la misma (*tcread*), y leída desde una PC que cuente con una unidad especial para ello. Este archivo de texto carece de formato alguno (figura IV.24), por lo cual se desarrolló una utilidad en *Visual Basic* para homologar el archivo de tal manera que no hubiera forma de diferenciar entre un archivo adquirido directamente por el *software* del sistema via comunicación serial y uno bajado en la *TCRAM*, utilizando esta aplicación (PROCAR) como lo muestra la figura IV.25.

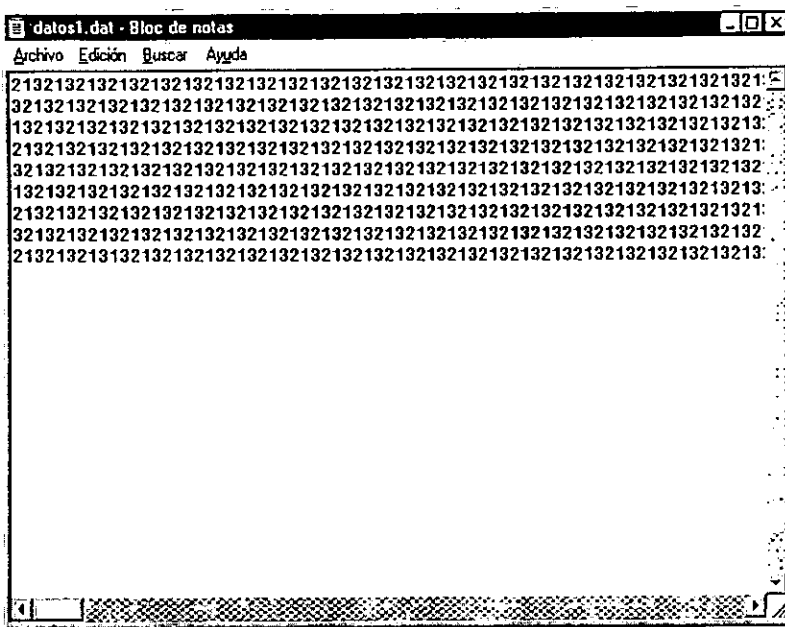


Figura IV.24 Archivo de datos almacenado en la TCRAM sin formato.

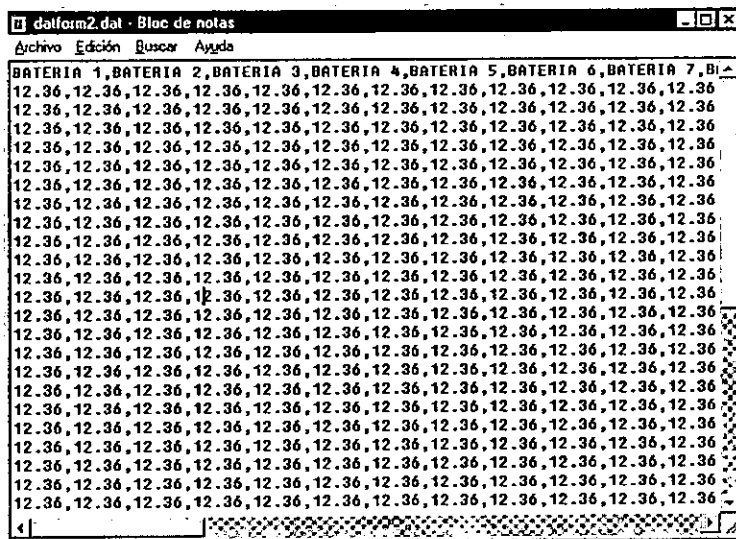


Figura IV.25 Archivo de datos después de la aplicación de la utilería PROCAR.

IV.9 Instalación del SADME

Para concluir presentamos el diagrama de instalación del sistema, (figura IV.26) para realizar la instalación, donde se indican las conexiones de las variables al SADME. Cabe aclarar que sólo se alambrió la conexión de corriente que consume el motor, por que la de regeneración aún no está disponible, sin embargo se mantiene habilitada para su uso.

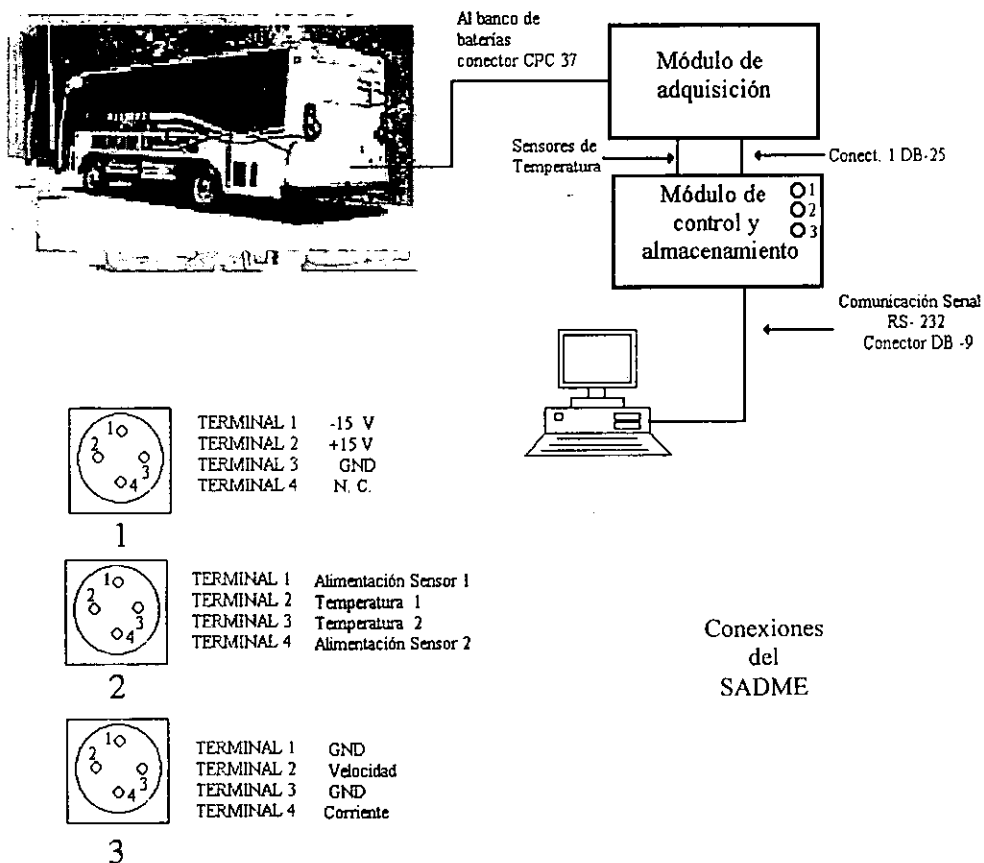


Figura IV.26 Diagrama de instalación del SADME.

Finalmente en la figura IV.27 observamos en la parte posterior del Minibús el SADME ya instalado.

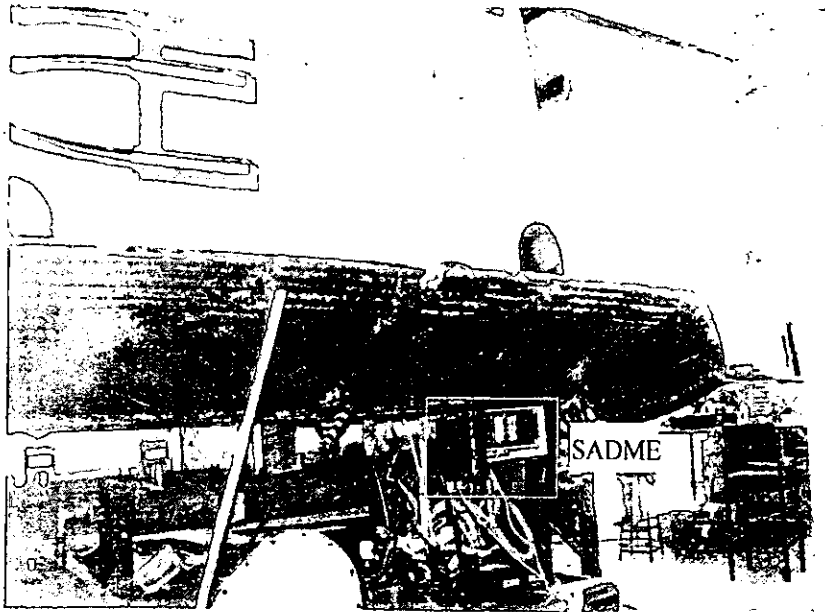


Figura IV.27 El SADME en el minibús eléctrico.

Como vimos en el capítulo, la integración de los módulos, consistió de dos partes fundamentales, lo referente al *hardware* y sus conexiones y al *software* utilizado, ya sea como parte de alguna de las tarjetas de los módulos o *interface* con el usuario.

En el siguiente capítulo veremos las pruebas, tanto de laboratorio como en el minibús, que se realizaron.

====CAPITULO VI====

Resultados y conclusiones

En esta parte final del trabajo presentamos los resultados obtenidos del desarrollo del proyecto, así mismo presentamos sus perspectivas y posibles mejoras

VI.1 Resultados

Desde el inicio del planteamiento del sistema se presentó como objetivo el desarrollar un sistema que cumpliera con las características recomendadas o pedidas para la supervisión de las variables del minibús eléctrico.

Como parte inicial se pretendió un monitoreo de 52 baterías del banco de alimentación del minibús eléctrico, pero esto tuvo que ser modificado debido a que se realizaron diferentes cambios en el mismo banco. En realidad el banco de alimentación continúa teniendo 52 baterías, pero como se mencionó en el capítulo IV, este banco se conecta en forma paralelo-serie. Sin embargo, los primeros programas se realizaron pensando en medir 52 baterías, un voltaje total, una velocidad, dos corrientes y posteriormente se pidió él adicionarle 2 sensores de temperatura, uno para cada subdivisión del banco.

Para poder desarrollar el proyecto fue necesario aprender el lenguaje de la tarjeta del módulo de control (*Little-G*), posteriormente lenguaje ensamblador propio del *COP8SAC44* y *Visual Basic*.

También la tarjeta del módulo de acondicionamiento (TMA) fue objeto de estudio, pues aunque fue desarrollado en el mismo Instituto de Ingeniería, no se tenía toda la documentación que se requería para saber de primera instancia su funcionamiento, por lo que se tuvieron que hacer cuidadosas pruebas para determinar la filosofía de operación y con que parámetros se contaba para su control.

Una de las partes que requirió mayor dedicación fue el desarrollo de la tarjeta de almacenamiento externo pues se trataban de conjuntar dos filosofías diferentes de operación, a saber, la forma en que funcionaba el puerto *PLCBus* de la tarjeta *Little-G*, y por lo mismo como se comunicaba con las tarjetas *interface* como la tarjeta del módulo de conversión analógico digital (TMCAD). Y por la parte del *COP8SAC44* el realizar la *interface* que

permitiera que este microcontrolador se comunicara con la *Little-G* y realizar la escritura en la *Thin Card RAM* (TCRAM).

Otra parte fundamental fue el cableado del banco de baterías, se contaba solamente con una sección de cableado. Se tuvo que conjuntar todos los cables en varios conectores, por lo que se tuvieron que armar, se consiguió el cable y se desmontó el banco del minibús, no sin antes determinar en dónde se realizarían las conexiones de los terminales correspondientes en las baterías.

El determinar la capacidad disponible de memoria fue también un hecho que influye en las expectativas finales del sistema, pues está estrechamente relacionada con el número de variables de las cuales se quiera hacer un muestreo. Y de ahí se partió, para darle aún más autonomía al sistema, en pensar en el desarrollo de un módulo externo de almacenamiento.

Como sistema, el Sistema de Adquisición de Datos del Minibús Eléctrico (SADME), presenta la flexibilidad de que ya sea en su conjunto, o en algunos de sus módulos, éstos pueden ser adecuados para tener la capacidad de realizar un muestreo de un mayor número de variables.

En sí, el TMA puede monitorear un banco de hasta 64 baterías, claro que habría que hacer adecuaciones en cuanto al tiempo en que se realizan los muestreos, tal vez introducirse un poco más en el software de desarrollo de la tarjeta *Little-G, Dynamic C*, con el propósito de aprovechar aun más la opción de programar partes de las rutinas a nivel ensamblador y verificar si se puede sacar una mayor ventaja en, por ejemplo, realizar los muestreos con intervalos de tiempo más reducidos, lo que permitiría un aumento de número de variables que en un cierto momento se pudieran necesitar.

En el caso de las pruebas de transmisión de datos se utilizó, inicialmente, el programa hiperterminal de las utilerías de *Windows*, y después se pensó en realizar un software más amigable y de fácil uso, éste fue desarrollado utilizando el lenguaje *Visual Basic 4*, con la ventaja de que trabaja bajo ambiente gráfico.

Cabe señalar que se trató de explotar al máximo las características de cada uno de los módulos, lo cual creemos que se logró en la integración final. También se trató reducir hasta un cierto punto el número de componentes externos, como serían los conectores entre los módulos y los conectores que van a las baterías.

Al suministrarle energía al SADME, éste automáticamente realiza el muestreo de las variables, lo que representa una ventaja; no obstante lo anterior, en el caso de finalizar el muestreo se necesita tener conectada la computadora personal al SADME, y esto tal vez represente una desventaja. Sin embargo, una vez que se ha terminado el muestreo, los datos permanecen almacenados en el SADME, aun después de haber retirado la alimentación, esto gracias a las baterías de respaldo con que se cuenta en el módulo de control (*Little-G*) y en caso de utilizarse la tarjeta TCRAM.

Finalmente, el trabajo desarrollado cumple con los objetivos inicialmente planeados; las pruebas realizadas y el análisis de los resultados obtenidos nos dejan ver que el sistema cumple cabalmente con las especificaciones del proyecto.

VI.2 Conclusiones

En este apartado se presentan brevemente las ventajas, recomendaciones y perspectivas del sistema desarrollado con la finalidad de presentar un aspecto general del mismo.

Bondades del sistema

- El Sistema de Adquisición de Datos del Minibús Eléctrico (SADME), es un sistema automático y autónomo de adquisición de datos, con una capacidad de manejo de 32 variables, y una capacidad de almacenamiento desde 4 a 12 horas.
- La adquisición total de las variables se realiza en un segundo.
- El sistema está provisto de un aislamiento eléctrico del banco de baterías, el cual le proporciona seguridad en caso de existir algún corto circuito en el banco, este aislamiento lo proporcionan los relevadores a los que están conectados las baterías.

- El sistema cuenta con una batería de respaldo que nos garantiza la integridad de los datos en caso de una falla en la alimentación.
- Existen dos formas de transferir los datos para procesarlos en una PC.
 - Por medio de un puerto serial con protocolo de comunicación RS-232, con conector DB-9 a 19200 bps.
 - Por medio de la tarjeta de almacenamiento externo TCRAM con capacidad de 1Mbyte.
- Cuenta con un software de *interface* con el operador del sistema, el cual le permite controlar la operación del mismo.
- Los datos adquiridos son almacenados en un archivo cuyo formato nos permite su proceso en una hoja de cálculo para el análisis pertinente.

Recomendaciones

- El sistema empieza operar con sólo alimentarlo, con la configuración predeterminada, pero para finalizar el muestreo es necesario establecer comunicación con la PC.
 - El sistema se alimenta con ± 15 V de corriente directa con una capacidad de mínima de 1 A.
 - El sistema requiere de un aislamiento eléctrico del *chasis*, para evitar interferencia o ruido que afecte la correcta adquisición de los datos.

Perspectivas del sistema

- Su composición modular nos permite detectar y corregir fallas de una forma más ágil, así como observar su comportamiento por etapas.
- El sistema se podría instalar en vehículos similares, claro, tomando en cuenta de cuantas baterías se compondrían los bancos y el tipo de conexiones que se realizan en el mismo.
- Se puede realizar el monitoreo de hasta 64 baterías, existen además 9 canales adicionales en la tarjeta del módulo de acondicionamiento, los cuales pueden servir para realizar el monitoreo de diferentes variables.

- Tal vez el TMA se podría rediseñar utilizando componentes de montaje superficial con lo que se reduciría el tamaño total del módulo y diseñándolo nuevamente de tal forma que se podría omitir el conector DB25, cuando se interconecta con el módulo de control.
- También se podría realizar un solo gabinete para todo el sistema, con la ventaja de que instalaría en lugares aún más reducidos.

Para finalizar, con el presente trabajo esperamos contribuir de alguna forma en el estudio del comportamiento del minibús eléctrico del Instituto de Ingeniería, lo permitirá que la gente encargada realice las modificaciones que crea convenientes para lograr un mejor desempeño del mismo.

BIBLIOGRAFIA

Bibliografía

Bibliografía

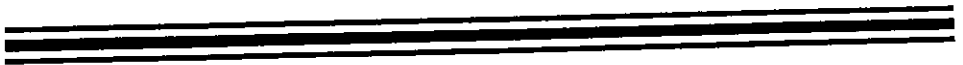
- SAE. *Electric Vehicle Design and Development*, SP-862, USA, 1991
- Wilson, Verne H. *8-bit Microcontroller Instruction Set Performance Evaluation. National Semiconductor*. June 10, 1994. USA
- Motorola. *M68HC05 Applications guide*. USA. 1996.
- Couch II, Leon. *Digital and analog Communication System*, Fourth Edition. Ed. Mcmillan. New York.
- Ashelsky, Louis. *Fundamentos de Tecnología Digital*. Ed. Limusa. México.
- Dempsy, John A. *Electrónica Digital básica con aplicaciones MSI*. Ed. Alfa y Omega, México.
- Kennedy. *Electronic Communication System*. 3ª. Edición, Mc Graw-Hill. México. 1984.
- *PLC Bus Expansion, Boards. Technical Reference Manual*. ZWorld engineering, November 1, 1994. USA.
- *Little- G. C-Programmable Miniature Controller*. Technical Reference Manual. ZWorld engineering. Revision 1.3. Noviembre 23, 1995. USA.
- Kelley/Phol. *Lenguaje C, introducción a la programación*. Addison-Wesley Iberoamericana. 1989. México.
- Ceballos, Fco. Javier. *Enciclopedia de Microsoft Visual Basic 4*. Alfa y Omega. 1997. México.
- Cornell, Gary. *Manual de Visual Basic 4 para Windows 95*. Osborne. Mc Graw-Hill. México. 1997
- *COP feature family User's manual. National Semiconductor*, September 1996. USA.
- Halvorson, Michael. *Aprenda Visual Basic Ya*. Mc Graw Hill. España. 1996.
- Fullea García, José atl. *El vehículo eléctrico, tecnologías y desarrollo y perspectivas del futuro*, EVE. Mc Graw Hill. España. 1997.
- W. Kernighan, Brian atl. *The C programming language*. Prentice-Hall software series. EEUU. 1978.

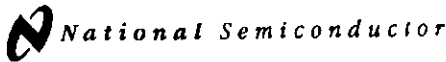
BIBLIOGRAFIA

- Información sobre Vehículos. <http://moon.inf.uji.es/aluweb/tarazona/es/>.
- *Electric Vehicle Data* base CD-ROM. <http://www.ddgn/carman/eucd2.htm>.

===== APENDICE A =====

Hojas
de
especificación





PRELIMINARY
August 1997

COP8SAA7/COP8SAB7/COP8SAC7

8-Bit One-Time Programmable (OTP) Microcontroller

General Description

The COP8Ax7 OTP microcontrollers are members of the COP8™ feature family using an 8-bit single chip core architecture. These devices are fabricated in National Semiconductor's high-density EPROM process, and offered on a variety of packages, temperature ranges and voltage ranges to satisfy a wide variety of applications.

Key Features

- Low cost 8-bit OTP microcontroller
- OTP program space with read/write protection (fully secured)
- Quiet Design (low radiated emissions)
- Multi-Input Wakeup pins with optional interrupts (4 to 8 pins)
- 8 bytes of user storage space in EPROM
- User selectable clock options
 - Crystal/Resonator options
 - Crystal/Resonator option with on-chip bias resistor
 - External oscillator
 - Internal R/C oscillator
- Internal Power-On Reset — user selectable
- WATCHDOG and Clock Monitor Logic — user selectable
- Up to 12 high current outputs

- One Timer (with 2 interrupts)
- MICROWIRE/PLUS™ Serial Interface
- Multi-Input Wake Up
- Software Trap
- Default VIS (default interrupt)
- 8-bit Stack Pointer SP (stack in RAM)
- Two 8-bit Register Indirect Data Memory Pointers
- True bit manipulation
- Memory mapped I/O
- BCD arithmetic instructions

Peripheral Features

- Multi-Input Wakeup Logic
- One 16-bit timer with two 16-bit registers supporting:
 - Processor Independent PWM mode
 - External Event counter mode
 - Input Capture mode
- Idle Timer
- MICROWIRE/PLUS Serial Interface (SPI Compatible)

I/O Features

- Software selectable I/O options
 - TRI-STATE® Output
 - Push-Pull Output
 - Weak Pull Up Input
 - High Impedance Input
- Schmitt trigger inputs on ports G and L
- Up to 12 high current outputs
- Pin efficient (i.e., 40 pins in 44-pin package are devoted to useful I/O)

Fully Static CMOS Design

- Low current drain (typically < 4 µA)
- Single supply operation: 2.7V to 5.5V
- Two power saving modes: HALT and IDLE

Temperature Ranges

0°C to +70°C, -40°C to +85°C, and -40°C to +125°C

Development Support

- Windowed packages for DIP and PLCC
- Real time emulation and full program debug offered by MetaLink Development System

Device	EPROM	RAM	Package and I/O	
			Package Types	Number of I/O
COP8SAC7	4k	128	20 DIP/SO	16
			28 DIP/SO	24
			40 DIP	36
			44 PLCC	40
COP8SAB7	2k	128	20 DIP/SO	16
			28 DIP/SO	24
COP8SAA7	1k	64	16 DIP/SO	12
			20 DIP/SO	16
			28 DIP/SO	24

CPU Features

- Versatile easy to use instruction set
- 1 µs instruction cycle time
- Eight multi-source vectored interrupts servicing
 - External interrupt
 - Idle Timer TO

TRI-STATE® is a registered trademark of National Semiconductor Corporation.
MICROWIRE/PLUS™, COP8™, MICROWIRE™ and WATCHDOG™ are trademarks of National Semiconductor Corporation.
COP8SAA7™ is a trademark of MetaLink Corporation.
PC® is a registered trademark of International Business Machines Corporation.

COP8SAA7/COP8SAB7/COP8SAC7 8-Bit One-Time Programmable (OTP) Microcontroller

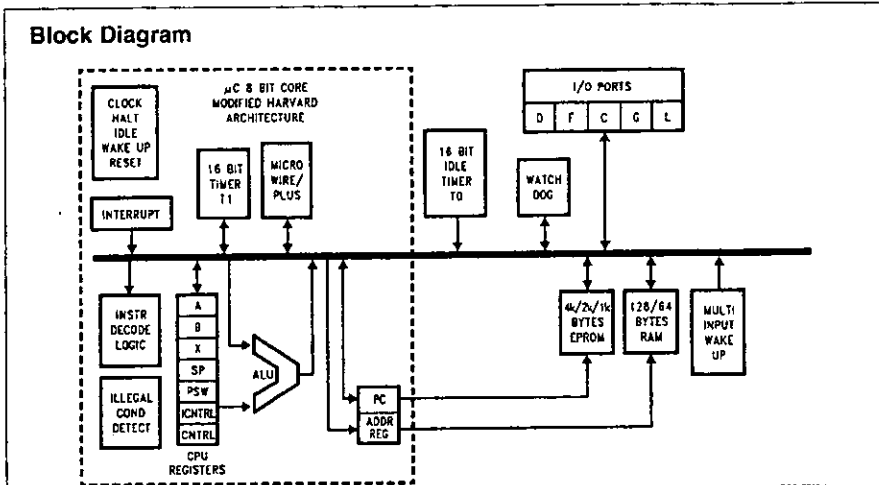


FIGURE 1. COP8Sx7 Block Diagram

General Description (Continued)

Key features include an 8-bit memory mapped architecture, a 16-bit timer/counter with two associated 16-bit registers supporting three modes (Processor Independent PWM generation, External Event counter, and Input Capture capabilities), two power saving HALT/IDLE modes with a multi-sourced wakeup/interrupt capability, on-chip R/C oscillator, high current outputs, user selectable options such as WATCHDOG™, Oscillator configuration, and power-on-reset.

1.1 EMI REDUCTION

The COP8Sx7 family of devices incorporates circuitry that guards against electromagnetic interference—an increasing problem in today’s microcontroller board designs. National’s patented EMI reduction technology offers low EMI clock circuitry, gradual turn-on output drivers (GTOs) and internal I_{CC} smoothing filters, to help circumvent many of the EMI issues influencing embedded control designs. National has achieved 15 dB–20 dB reduction in EMI transmissions when designs have incorporated its patented EMI reducing circuitry.

1.2 ARCHITECTURE

The COP8Sx7 family is based on a modified Harvard architecture, which allows data tables to be accessed directly from program memory. This is very important with modern microcontroller-based applications, since program memory is usually ROM or EPROM, while data memory is usually RAM. Consequently data tables usually need to be contained in ROM or EPROM, so they are not lost when the microcontroller is powered down. In a modified Harvard architecture, instruction fetch and memory data transfers can be overlapped with a two stage pipeline, which allows the next instruction to be fetched from program memory while the current instruction is being executed using data memory. This is not possible with a Von Neumann single-address bus architecture.

The COP8Sx7 family supports a software stack scheme that allows the user to incorporate many subroutine calls. This capability is important when using High Level Languages. With a hardware stack, the user is limited to a small fixed number of stack levels.

1.3 INSTRUCTION SET

In today’s 8-bit microcontroller application arena cost/performance, flexibility and time to market are several of the key issues that system designers face in attempting to build well-engineered products that compete in the marketplace. Many of these issues can be addressed through the manner in which a microcontroller’s instruction set handles processing tasks. And that’s why COP8 family offers a unique and code-efficient instruction set—one that provides the flexibility, functionality, reduced costs and faster time to market that today’s microcontroller based products require.

Code efficiency is important because it enables designers to pack more on-chip functionality into less program memory space (ROM/OTP). Selecting a microcontroller with less program memory size translates into lower system costs, and the added security of knowing that more code can be packed into the available program memory space.

1.3.1 Key Instruction Set Features

The COP8Sx7 family incorporates a unique combination of instruction set features, which provide designers with optimum code efficiency and program memory utilization.

Single Byte/Single Cycle Code Execution

The efficiency is due to the fact that the majority of instructions are of the single byte variety, resulting in minimum program space. Because compact code does not occupy a substantial amount of program memory space, designers can integrate additional features and functionality into the microcontroller program memory space. Also, the majority instructions executed by the device are single cycle, resulting in

General Description (Continued) (Continued)

minimum program execution time. In fact, 77% of the instructions are single byte single cycle, providing greater code and I/O efficiency, and faster code execution.

1.3.2 Many Single-Byte, Multifunction Instructions

The COPSAX7 instruction set utilizes many single-byte, multifunction instructions. This enables a single instruction to accomplish multiple functions, such as DRSZ, DCOR, JID, and LOAD/EXCHANGE instructions with post-incrementing and post-decrementing, to name just a few examples. In many cases, the instruction set can simultaneously execute as many as three functions with the same single-byte instruction.

JID: (Jump Indirect); Single byte instruction; decodes external events and jumps to corresponding service routines (analogous to "DO CASE" statements in higher level languages).

LAI: (Load Accumulator-Indirect); Single byte look up table instruction provides efficient data path from the program memory to the CPU. This instruction can be used for table lookup and to read the entire program memory for checksum calculations.

RETSK: (Return Skip); Single byte instruction allows return from subroutine and skips next instruction. Decision to branch can be made in the subroutine itself, saving code.

AUTOINC/DEC: (Auto-Increment/Auto-Decrement); These instructions use the two memory pointers B and X to efficiently process a block of data (analogous to "FOR NEXT" in higher level languages).

1.3.3 Bit-Level Control

Bit-level control over many of the microcontroller's I/O ports provides a flexible means to ease layout concerns and save board space. All members of the COP8 family provide the ability to set, reset and test any individual bit in the data memory address space, including memory-mapped I/O ports and associated registers. Three memory-mapped pointers handle register indirect addressing and software stack pointer functions. The memory data pointers allow the option of post-incrementing or post-decrementing with the data movement instructions (LOAD/EXCHANGE). And 15 memory-mapped registers allow designers to optimize the precise implementation of certain specific instructions.

1.4 PACKAGING/PIN EFFICIENCY

Real estate and board configuration considerations demand maximum space and pin efficiency, particularly given today's high integration and small product form factors. Microcontroller users try to avoid using large packages to get the I/O needed. Large packages take valuable board space and increase device cost, two trade-offs that microcontroller designs can ill afford.

The COP8 family offers a wide range of packages and do not waste pins: up to 90.9% (or 40 pins in the 44-pin package) are devoted to useful I/O.

Connection Diagrams

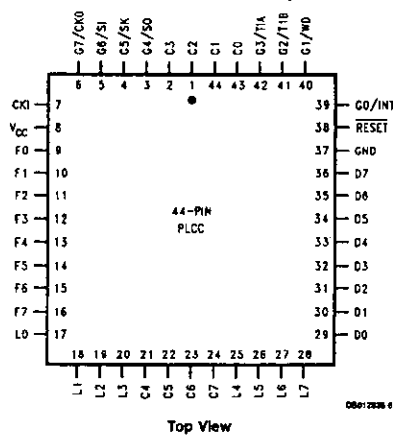
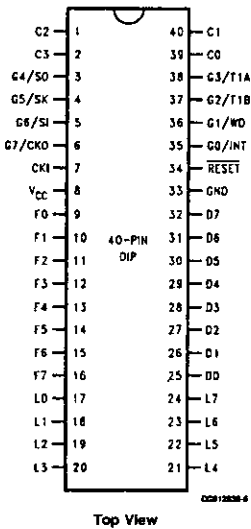
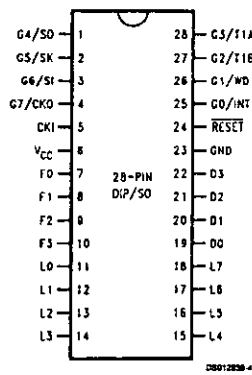
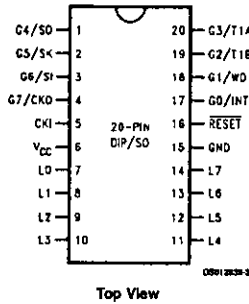
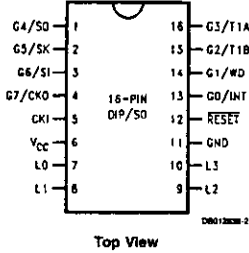
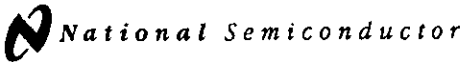


FIGURE 2. Connection Diagrams



June 1989

5404/DM5404/DM7404 Hex Inverting Gates

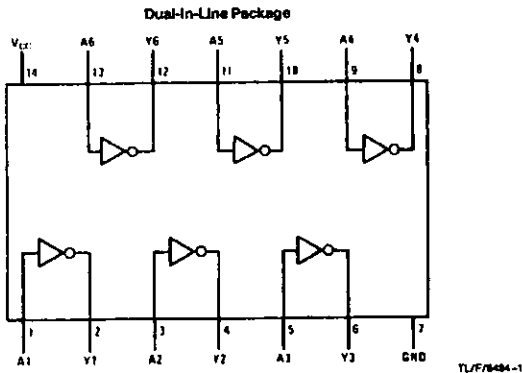
General Description

This device contains six independent gates each of which performs the logic INVERT function.

Features

- Alternate Military/Aerospace device (5404) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



Order Number 5404DMQB, 5404FMQB, DM5404J, DM5404W, DM7404M or DM7404AM
See NS Package Number J14A, M14A, N14A or W14B

Function Table

$$Y = \bar{A}$$

Inputs	Output
A	Y
L	H
H	L

H = High Logic Level
L = Low Logic Level

5404/DM5404/DM7404 Hex Inverting Gates

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	5.5V
Operating Free Air Temperature Range	
DM54 and 54	-55°C to +125°C
DM74	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Note: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	DM5404			DM7404			Units
		Min	Nom	Max	Min	Nom	Max	
V _{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH}	High Level Input Voltage	2			2			V
V _{IL}	Low Level Input Voltage			0.8			0.8	V
I _{OH}	High Level Output Current			-0.4			-0.4	mA
I _{OL}	Low Level Output Current			16			16	mA
T _A	Free Air Operating Temperature	-55		125	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

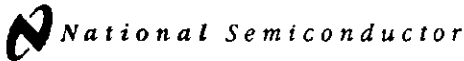
Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = -12 mA			-1.5	V
V _{OH}	High Level Output Voltage	V _{CC} = Min, I _{OH} = Max V _{IL} = Max	2.4	3.4		V
V _{OL}	Low Level Output Voltage	V _{CC} = Min, I _{OL} = Max V _{IH} = Min		0.2	0.4	V
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 5.5V			1	mA
I _{IH}	High Level Input Current	V _{CC} = Max, V _I = 2.4V			40	µA
I _{IL}	Low Level Input Current	V _{CC} = Max, V _I = 0.4V			-1.6	mA
I _{OS}	Short Circuit Output Current	V _{CC} = Max (Note 2)	DM54 -20 DM74 -18		-55	mA
I _{COH}	Supply Current with Outputs High	V _{CC} = Max		6	12	mA
I _{COL}	Supply Current with Outputs Low	V _{CC} = Max		18	33	mA

Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	Conditions	Min	Max	Units
t _{PLH}	Propagation Delay Time Low to High Level Output	C _L = 15 pF R _L = 400Ω		22	ns
t _{PHL}	Propagation Delay Time High to Low Level Output			15	ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time.



June 1989

5408/DM5408/DM7408 Quad 2-Input AND Gates

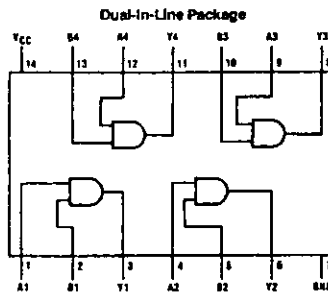
General Description

This device contains four independent gates each of which performs the logic AND function.

Features

- Alternate Military/Aerospace device (5408) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



Order Number 5408DMQB, 5408FMQB, DM5408J, DM5408W or DM7408N
See NS Package Number J14A, N14A or W14B

TL/F/6498-1

Function Table

$$Y = AB$$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = High Logic Level
L = Low Logic Level

5408/DM5408/DM7408 Quad 2-Input AND Gates

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	5.5V
Operating Free Air Temperature Range	
DM54 and 54	-55°C to +125°C
DM74	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Note: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	DM5408			DM7408			Units
		Min	Nom	Max	Min	Nom	Max	
V _{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH}	High Level Input Voltage	2			2			V
V _{IL}	Low Level Input Voltage			0.8			0.8	V
I _{OH}	High Level Output Current			-0.8			-0.8	mA
I _{OL}	Low Level Output Current			16			16	mA
T _A	Free Air Operating Temperature	-55		125	0		70	°C

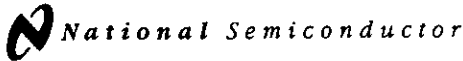
Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = -12 mA			-1.5	V
V _{OH}	High Level Output Voltage	V _{CC} = Min, I _{OH} = Max V _{IL} = Max	2.4	3.4		V
V _{OL}	Low Level Output Voltage	V _{CC} = Min, I _{OL} = Max V _{IH} = Min		0.2	0.4	V
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 5.5V			1	mA
I _{IH}	High Level Input Current	V _{CC} = Max, V _I = 2.4V			40	μA
I _{IL}	Low Level Input Current	V _{CC} = Max, V _I = 0.4V			-1.6	mA
I _{OS}	Short Circuit Output Current	V _{CC} = Max (Note 2)	DM54 -20		-55	mA
			DM74 -18		-55	
I _{OOH}	Supply Current with Outputs High	V _{CC} = Max		11	21	mA
I _{OOL}	Supply Current with Outputs Low	V _{CC} = Max		20	33	mA

Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	Conditions	Min	Max	Units
t _{PLH}	Propagation Delay Time Low to High Level Output	C _L = 15 pF R _L = 400Ω		27	ns
t _{PHL}	Propagation Delay Time High to Low Level Output			19	ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.
 Note 2: Not more than one output should be shorted at a time.



May 1989

54LS32/DM54LS32/DM74LS32 Quad 2-Input OR Gates

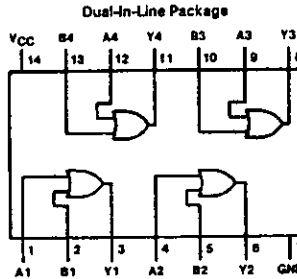
General Description

This device contains four independent gates each of which performs the logic OR function.

Features

- Alternate Military/Aerospace device (54LS32) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TU/F/6361-1

Order Number 54LS32DNQB, 54LS32FMQB, 54LS32LMQB,
DM54LS32J, DM54LS32W, DM74LS32M or DM74LS32N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$$Y = A + B$$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

H = High Logic Level
L = Low Logic Level

54LS32/DM54LS32/DM74LS32 Quad 2-Input OR Gates

Absolute Maximum Ratings (Nota)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage 7V
 Input Voltage 7V

Operating Free Air Temperature Range
 DM54LS and 54LS -55°C to +125°C
 DM74LS 0°C to +70°C
 Storage Temperature Range -65°C to +150°C

Note: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	DM54LS32			DM74LS32			Units
		Min	Nom	Max	Min	Nom	Max	
V _{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH}	High Level Input Voltage	2			2			V
V _{IL}	Low Level Input Voltage			0.7			0.8	V
I _{OH}	High Level Output Current			-0.4			-0.4	mA
I _{OL}	Low Level Output Current			4			8	mA
T _A	Free Air Operating Temperature	-55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

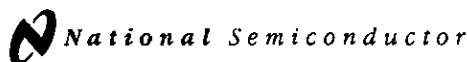
Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units	
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = -18 mA			-1.5	V	
V _{OH}	High Level Output Voltage	V _{CC} = Min, I _{OH} = Max	DM54	2.5	3.4	V	
		V _{IH} = Min	DM74	2.7	3.4		
V _{OL}	Low Level Output Voltage	V _{CC} = Min, I _{OL} = Max	DM54		0.25	0.4	V
		V _{IL} = Max	DM74		0.35	0.5	
		I _{OL} = 4 mA, V _{CC} = Min	DM74		0.25	0.4	
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 7V			0.1	mA	
I _{IH}	High Level Input Current	V _{CC} = Max, V _I = 2.7V			20	µA	
I _{IL}	Low Level Input Current	V _{CC} = Max, V _I = 0.4V			-0.36	mA	
I _{OS}	Short Circuit Output Current	V _{CC} = Max (Note 2)	DM54	-20	-100	mA	
			DM74	-20	-100		
I _{OCH}	Supply Current with Outputs High	V _{CC} = Max		3.1	6.2	mA	
I _{OCL}	Supply Current with Outputs Low	V _{CC} = Max		4.9	9.8	mA	

Switching Characteristics at V_{CC} = 5V and T_A = 25°C (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	R _L = 2 kΩ				Units
		C _L = 15 pF		C _L = 50 pF		
		Min	Max	Min	Max	
t _{PLH}	Propagation Delay Time Low to High Level Output	3	11	4	15	ns
t _{PHL}	Propagation Delay Time High to Low Level Output	3	11	4	15	ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.



May 1995

54F/74F240•54F/74F241•54F/74F244
Octal Buffers/Line Drivers with TRI-STATE Outputs

54F/74F240•54F/74F241•54F/74F244 Octal Buffers/Line Drivers with TRI-STATE® Outputs

General Description

The 'F240, 'F241 and 'F244 are octal buffers and line drivers designed to be employed as memory and address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC and board density.

Features

- TRI-STATE outputs drive bus lines or buffer memory address registers
- Outputs sink 64 mA (48 mA mil)
- 12 mA source current
- Input clamp diodes limit high-speed termination effects
- Guaranteed 4000V minimum ESD protection

Commercial	Military	Package Number	Package Description
74F240PC		N20A	20-Lead (0.300" Wide) Molded Dual-In-Line
	54F240DM (Note 2)	J20A	20-Lead Ceramic Dual-In-Line
74F240SC (Note 1)		M20B	20-Lead (0.300" Wide) Molded Small Outline, JEDEC
74F240SJ (Note 1)		M20D	20-Lead (0.300" Wide) Molded Small Outline, EIAJ
	54F240FM (Note 2)	W20A	20-Lead Cerpack
	54F240LM (Note 2)	E20A	20-Lead Ceramic Leadless Chip Carrier, Type C
74F241PC		N20A	20-Lead (0.300" Wide) Molded Dual-In-Line
	54F241DM (Note 2)	J20A	20-Lead Ceramic Dual-In-Line
74F241SC (Note 1)		M20B	20-Lead (0.300" Wide) Molded Small Outline, JEDEC
74F241SJ (Note 1)		M20D	20-Lead (0.300" Wide) Molded Small Outline, EIAJ
	54F241FM (Note 2)	W20A	20-Lead Cerpack
	54F241LM (Note 2)	E20A	20-Lead Ceramic Leadless Chip Carrier, Type C
74F244PC		N20A	20-Lead (0.300" Wide) Molded Dual-In-Line
	54F244DM (Note 2)	J20A	20-Lead Ceramic Dual-In-Line
74F244SC (Note 1)		M20B	20-Lead (0.300" Wide) Molded Small Outline, JEDEC
74F244SJ (Note 1)		M20D	20-Lead (0.300" Wide) Molded Small Outline, EIAJ
74F244MSA (Note 1)		MSA20	20-Lead Molded Shrink Small Outline, EIAJ Type II
	54F244FM (Note 2)	W20A	20-Lead Cerpack
	54F244LM (Note 2)	E20A	20-Lead Ceramic Leadless Chip Carrier, Type C

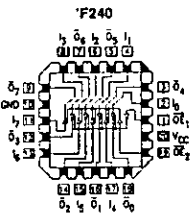
Note 1: Devices also available in 13" reel. Use Suffix = SCX, SJX and MSAX.

Note 2: Military grade device with environmental and burn-in processing. Use suffix = DMQB, FMQB and LMQB.

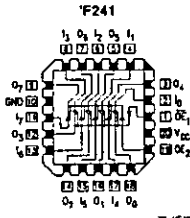
TRI-STATE® is a registered trademark of National Semiconductor Corporation.

Connection Diagrams

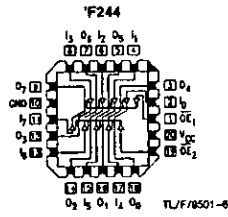
Pin Assignment for LCC



TL/F/9501-2

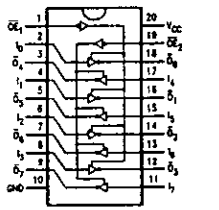


TL/F/9501-4

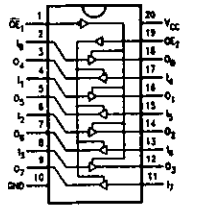


TL/F/9501-6

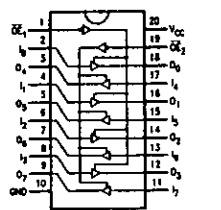
Pin Assignment for DIP, SOIC, SSOP and Flatpak



TL/F/9501-1

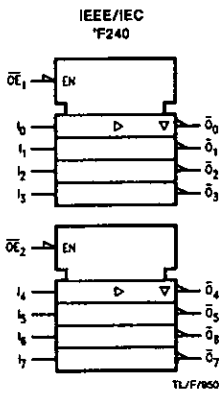


TL/F/9501-3

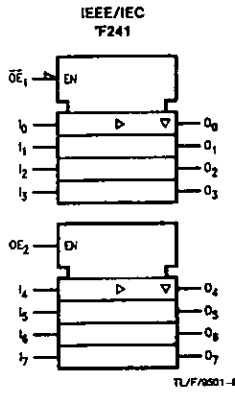


TL/F/9501-5

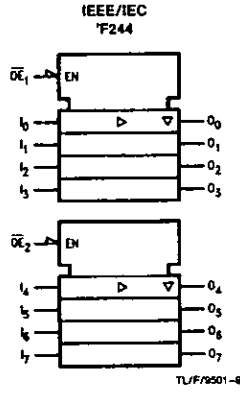
Logic Symbols



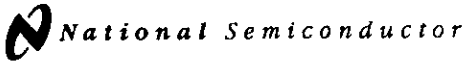
TL/F/9501-7



TL/F/9501-8



TL/F/9501-9



May 1995

54F/74F273 Octal D Flip-Flop

General Description

The 'F273 has eight edge-triggered D-type flip-flops with individual D inputs and Q outputs. The common buffered Clock (CP) and Master Reset (MR) inputs load and reset (clear) all flip-flops simultaneously.

The register is fully edge-triggered. The state of each D input, one setup time before the LOW-to-HIGH clock transition, is transferred to the corresponding flip-flop's Q output.

All outputs will be forced LOW independently of Clock or Data inputs by a LOW voltage level on the MR input. The device is useful for applications where the true output only is required and the Clock and Master Reset are common to all storage elements.

Features

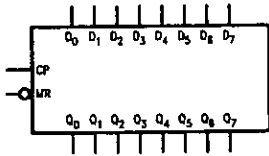
- Ideal buffer for MOS microprocessor or memory
- Eight edge-triggered D flip-flops
- Buffered common clock
- Buffered, asynchronous Master Reset
- See 'F377 for clock enable version
- See 'F373 for transparent latch version
- See 'F374 for TRI-STATE® version
- Guaranteed 4000V minimum ESD protection

Commercial	Military	Package Number	Package Description
74F273PC		N20A	20-Lead (0.300" Wide) Molded Dual-In-Line
	54F273DM (Note 2)	J20A	20-Lead Ceramic Dual-In-Line
74F273SC (Note 1)		M20B	20-Lead (0.300" Wide) Molded Small Outline, JEDEC
74F273SJ (Note 1)		M20D	20-Lead (0.300" Wide) Molded Small Outline, EIAJ
	54F273FM (Note 2)	W20A	20-Lead Cerpack
	54F273LM (Note 2)	E20A	20-Lead Ceramic Leadless Chip Carrier, Type C

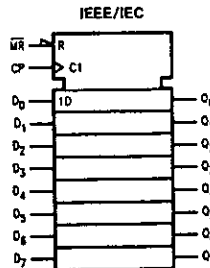
Note 1: Devices also available in 13" reel. Use suffix = SCX and SJX.

Note 2: Military grade device with environmental and burn-in processing. Use suffix = DMOB, FMOB and LMOB.

Logic Symbols



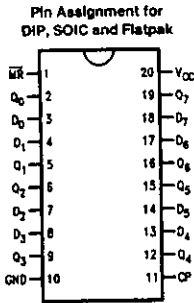
TLF/9511-3



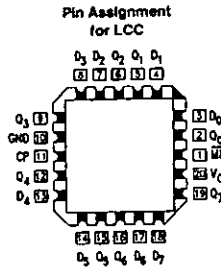
TLF/9511-5

TRI-STATE® is a registered trademark of National Semiconductor Corporation

Connection Diagrams



TL/F/8511-1



TL/F/8511-2

Unit Loading/Fan Out

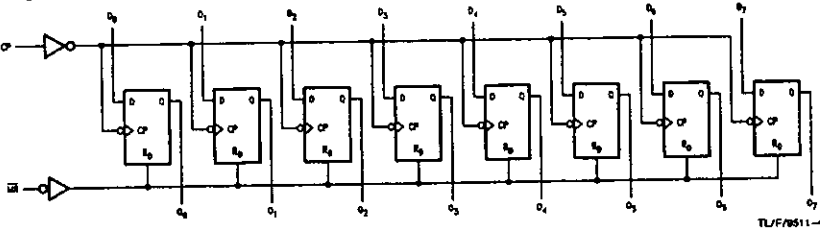
Pin Names	Description	64F/74F	
		U.L. HIGH/LOW	Input I_{IH}/I_{IL} Output I_{OH}/I_{OL}
D ₀ -D ₇	Data Inputs	1.0/1.0	20 μ A/ -0.6 mA
MR	Master Reset (Active LOW)	1.0/1.0	20 μ A/ -0.6 mA
CP	Clock Pulse Input (Active Rising Edge)	1.0/1.0	20 μ A/ -0.6 mA
Q ₀ -Q ₇	Data Outputs	50/33.3	-1 mA/20 mA

Mode Select-Function Table

Operating Mode	Inputs			Output
	MR	CP	D _n	Q _n
Reset (Clear)	L	X	X	L
Load '1'	H	↗	h	H
Load '0'	H	↘	l	L

H = HIGH Voltage Level steady state
 h = HIGH Voltage Level one setup time prior to the LOW-to-HIGH clock transition
 L = LOW Voltage Level steady state
 l = LOW Voltage Level one setup time prior to the LOW-to-HIGH clock transition
 X = Immaterial
 ↗ = LOW-to-HIGH clock transition
 ↘ = LOW-to-HIGH clock transition

Logic Diagram



TL/F/8511-4

Please note that this diagram is provided only for the understanding of logic operations and should not be used to estimate propagation delays.

==== APENDICE B =====

Listado de los programas desarrollados



PROGRAMA DE LA TARJETA DE CONTROL

```

.....
.*
.* Programa principal de la tarjeta Little-G
.*
.*
.*
.*
.....

```

```

/* Constantes globales *****/
typedef char BYTE; //Redefinición de variables
#define on 1
#define off 0
#define OR ||
#define AND &&
#define ch1 1 //8
#define ch2 2 //9
#define ch3 3 //10
#define ch4 4 //11 enable del ADG527
#define ch5 5 //13
#define ch6 6 //14
#define ch7 7 //15
#define ch8 8 //16
#define ch9 9 //17
#define ch10 10 //18
#define ch11 11 //19
#define ch12 12 //20 enable del 74138
#define ch13 13 //read activa/desactiva el latch
#define ch14 14 //write
#define TRUE 1
#define FALSE 0
#define QUEUE_SIZE 460800// 450 k Tamaño de la base de datos
#define TDE1 1 // cuenta del timer descendente
#define TIE1 5 //Interrupcion del timre 1

xdata DBASE[QUEUE_SIZE]; //tabla de la base de datos
xmem char InitDBase(void); //función que inicializa la base de datos
xmem void menu();
char DataGetchar(char *vDato);
char DataPushchar(char vDato);
int CHSUM(); //Realiza la suma de los datos a transmitir
xmem void Inicializa_ADC(void); //Inicializa el convertidor analógico digital
xmem void Inicializa_Tx(void); //Inicializa el puerto serial
xmem void act_muestreo();
xmem int TxDBase(void); //Envia encabezados de la transmisión
xmem void enviar(); //Transmite los datos almacenados en memoria
xmem int convert(); //Habilita un canal del convertidor solo para las baterias
xmem int convert1(); //Muestra temperatura
xmem int convert2(); //MUESTREO DE VOLTAJE TOTAL
xmem void conversion(); //Habilita convertidor para los demas canales
xmem void actual(); //Modifica la fecha
xmem void apagado(); //Apaga las salidas digitales
xmem void fecha_time(); //Lee el reloj
xmem int baterias (); //Se multiplexan las baterias habilitando las salidas digitales
xmem int checkbat(); //Checa el nivel de las baterias
xmem checmem(); //Checa capacidad de memoria
xmem void adc();
int prt1_init(int tc); // Funcion de habilitacion del timer 1

struct signal_rec adtab[12]; //Tabla para la conversion analogico digital
char DBase_inicializado; //Inicializacion de la base de datos
long DataIn, DataOut, DataEnd,k1;
char trss,trs;
int ii,a[60],*pa; //SE aumento en dos localidades el arreglo a
int bad; //indica si alguna bateria esta por debajo de algun nivel
int tiem_fec;
int tarjeta_rdi; /* Variable para definir dirección del ADC*/
float volc,batc; /* Variable para conversion*/
float temp;
int bandera,flag,flag1,flag3;
int opc;
int dia,mes,año,hora,min,seg;
char tp,ty;
shared long veces;

```

shared long cuenta;
 char cad_trans[6];
 struct tm tiempo;

```

.....
/* Programa principal */
.....

main()
{
    char SegsString[25], DataEndStr[25];
    char RxBuffer[25]; //caracter transmision
    int k,flag2;

    char opc;

    k1=0;
    cuenta = 0L;
    veces = 100L;
    tp=48;

    bad=0;
    flag=0;
    flag1=1;
    flag2=0;

    tiem_fec=TRUE;
    up_setout(ch14,off);
    InitDBase(); //inicializa la tabla de datos
    Inicializa_ADC();//inicializa el coovertidor
    Inicializa_Tx();//Define los parametros de transmision
    Dreset_z1rbuf();//Resetea el buffer de transmision

    while(1)
    {
        if(Dread_z1rch(RxBuffer)!=FALSE)
        {
            switch(*RxBuffer)
            {
                case '1': if(flag==0)
                {
                    act_muestreo();

                    *RxBuffer='';
                    Dreset_z1rbuf();
                    bandera=1;

                    while(bandera==TRUE)
                    {
                        if(Dread_z1rch(RxBuffer)!=FALSE)
                        {
                            switch(*RxBuffer)
                            {
                                case '1': veces=500L;//MUESTREANDO CADA 10 SEGUNDOS
                                    tp=48;

                                    TxLong(veces);
                                    TxChar(',');
                                    bandera=FALSE;
                                    break;

                                case '2': veces=2500L;//MUESTREANDO CADA 30 SEGUNDOS
                                    tp=49;

                                    TxLong(veces);
                                    TxChar(',');
                                    bandera=FALSE;
                                    break;

                                case '3': veces=5500L;//MUESTREANDO

```

CADA MINUTO

case '3': veces=5500L;//MUESTREANDO


```

        DataPushchar(tp);
        break;
    case 53: ty=253;
        DataPushchar(ty);
        DataPushchar(tp);

        break;
    case 54: ty=253;
        DataPushchar(ty);
        DataPushchar(tp);

        break;
    default :break;
}

bad=0;
up_setout(ch14,off);
TxString("Iniciado");
TxChar('.');
    checmem();

conversion();
baterias();
checkbat();
flag2=1;
cuenta=0L;
    prt1_init((9216000/20)/100);

flag=1;
flag3=1;
                                                                    break;
}
                                                                    case '3':    i{!(flag==1)AND (*RxBuffer=='3')}
{
TxString("Fin");
TxChar('.');
    IRES(TCR,TDE1);
                                                                    IRES(TCR,TIE1);
    tiem_fec=TRUE;
                                                                    cuenta=0L;

flag=0;
flag1=0;
                                                                    break;
}
                                                                    case '4':
i{!(flag==0) AND (flag1==0) AND (*RxBuffer =='4')}
{
    enviar();
    IniDBase();
    flag2=0;
    k1=0;
                                                                    break;
}
                                                                    case 'M':    i{!(flag==0)}
{
    menu();
                                                                    break;
}
case '5':i{!(flag==1) AND (*RxBuffer=='5') AND (tiem_fec==FALSE)}
{
    TxChar(0x41); //Transmite una A
    TxChar('.');
    TxLong(veces);
        TxChar('.');
        TxByte(dia);
    TxChar('/');
                                                                    TxByte(mes);
    TxChar('/');
                                                                    TxByte(año);
    TxChar('.');
                                                                    TxByte(hora);
    TxChar('.');
                                                                    TxByte(min);
    TxChar('.');
                                                                    TxByte(seg);
}

```



```

TxChar(',');
TxLong(k1);
TxChar(',');
break;
}
else if((flag==0) AND (*RxBuffer=='5') AND (flag2==1))
{
    TxChar(0x42);/* B manda cartacter de estado no se encuentra muestreando */
    TxChar(',');
    TxChar(0x43); // C No se han bajado datos de memoria
    TxChar(',');
    TxLong(veces);
    TxChar(',');
    TxLong(k1);
    TxChar(',');
    break;
}
else if((flag==0) AND (*RxBuffer=='5') AND (flag2==0))
{
    TxChar(0x42);/* B manda cartacter de estado no se encuentra muestreando */
    TxChar(',');
    TxChar(0x44); // D sin datos en memoria
    TxChar(',');
    TxLong(veces);
    TxChar(',');
    TxLong(k1);
    TxChar(',');
    break;
}
}
case T:    if((flag==0) AND (*RxBuffer=='T'))
{
    DIO;
    actual(); // Modifica la fecha
    break;
}

default: break;
} //end del switch*/
} //end if transmission*/
hitwd();
runwatch();
if (cuenta>=veces)
{
    checmern();
    conversion();
    baterias();
    checkbat();
} //end if de cuentas*/
} //end while*/
/*end main principal*/

// *****
// *** FUNCIONES DE LA BASE DE DATOS ***
// *****
xmem char InitDBase(void){
BYTE First_time;
    First_time=DBase_inicializado;
    DataIn=DataOut=0;
    DataEnd= QUEUE_SIZE;
    return (First_time);
}
//*****
xmem void conversion()
{
    int ij;

```

```

    DI();
    if(tiem_fec==TRUE)
    {
        fecha_time();
        dia=tiempo.tm_mday;
        mes=tiempo.tm_mon;
        año=tiempo.tm_year;
        hora=tiempo.tm_hour;
        min=tiempo.tm_min;
        seg=tiempo.tm_sec;
        DataPushchar(seg);
        DataPushchar(min);
        DataPushchar(hora);
        DataPushchar(dia);
        DataPushchar(mes);
        DataPushchar(año);

        tiem_fec=FALSE;
    }
    cuenta=0L;
k1++;
    jj=1;
    for (ii=1; ii<4;ii++)
    {
        /* Cerca las señales de entradas en una tabla analogica */
        add_sig_table(0,tarjeta,ii,8,0); /* Create Signal Table w/calibration */
        set_mux(0);
        mxdel18(0); /*retraso*/
        rd = ptad_rd12(adch(0));
        volc = ad_conv(0,rd);
        if(jj==1)
            temp=volc;
        else if(jj==2)
            temp=volc;
        else if(jj==3)
            temp=volc;
        else if(jj==4)
            temp=volc;

        if(((temp>0) OR (temp==0))){
            volc =volc*100;
            trs =volc/1;
            DataPushchar(trs);
        }
        else if(temp< 0)
        {
            volc=volc*(-100);
            trs= volc/1;
            DataPushchar(trs);
        }

        jj++;
        cuenta=0L;
    }
}
.....*/

char DataPushchar(char VData){
    char Aux;
    Aux=VData;

    if( DataIn < DataEnd){
        root2xmem(&Aux, DBASE+DataIn*1, 1 );

        DataIn++;
    }

    else
    {
        DataIn= 0;
    }
}
.....*/

```

```

char DataGetchar(char *vDato){
    int byResult;

    if(DataOut<DataIn ){
        xmem2root(DBASE+DataOut*1, vDato, 1 );

        if(DataOut<DataEnd)
            DataOut++;
        else
            DataOut=0;

        byResult=TRUE;
    }else{
        *vDato=0xFF;
        byResult=FALSE;
    }
    return byResult;
}
/*.....*/
xmem checkmem()
{ /*prueba valor original de k1 7940*/
    if(k1==7940)
    {
        InitDBase();
        tiem_fec=TRUE;
        k1=0;
        ty=253;
        DataPushchar(y);
        DataPushchar(p);
    }
}
/*.....*/
/******MENU PRINCIPAL DEL SISTEMA******/
/*.....*/
void menu()
{
    TxString("Menu");
    TxChar(',');
}
/*.....*/
/******SUBMENU DE ACTUALIZAR MUESTREO******/
/*.....*/
void act_muestreo()
{
    TxString("Actualizar");
    TxChar(',');
}

/*.....*/
/**** Inicializacion del convertidor analogico digital
/*.....*/
xmem void Inicializa_ADC(void)
{
    reset_pbuss(); /* Reset a todas las tarjetas conectadas al PLCBUS*/
    reset_pbuss_wait();
    tarjeta = plcadd_addr(3); /* J1 1-2 n.c., 3-4 n.c. = tarjeta #3 */
}

/*.....*/
/**** FUNCIONES DE TX DE DATOS ***
/*.....*/
xmem void Inicializa_Tx(void)
{
    char rbuf[512];
    char tbuf[512];
    int rsize=512; /*Tamaño del buffer de recepcion
    int tsize=512; /*Tamaño del buffer de transmision
    char mode=4; /*Modo de transmision 8 bits, sin paridad
    char baud=16; /*Velocidad de transmision en multiplos de 1200
    char ismodem=0; /*No se tiene conexion de modem
    char isecho=0; /*No se realiza eco despues de transmitir

    Dinit_z1(rbuf,tbuf,rsize,tsize,mode,baud,ismodem,isecho);
}

/*.....*/

```

```

xmem int TxString(char *cString)
{
    while(Dwrite_z1(cString,strlen(cString))!=1)
        NULL;

    return ;
}

// *****
xmem int TxByte(BYTE byData)
{
    char cString[15]; //Variable para transmitir.
    itoa(byData, cString); //Convierte de int a char para transmision
    while(Dwrite_z1(cString,strlen(cString))!=1)
        NULL;

    return ;
} // *****
xmem int TxLong(long loData)
{
    char cString[15]; //Variable para transmitir.

    ltoa(loData, cString); //Convierte de int a char para transmision
    while(Dwrite_z1(cString,strlen(cString))!=1)
        NULL;

    return ;
}

// *****
xmem int TxChar(char cData)
{
    while(Dwrite_z11ch(cData)!=1)
        NULL;

    return ;
}
//*****
//*****transmite los datos almacenados en memoria*****/
//*****
xmem void enviar()
{
    int i,ii,check_rep,muestras,flag4,ch;
    char e_dato,datof;
    rep=0;
    check=0;
    flag3=0;
    flag4=1;
    while((DataGetchar(&e_dato)!=FALSE) AND(flag4==1))
    {
        switch(e_dato)
        {
            case 253: if(e_dato==253)
            {
                TxChar(e_dato);//Envia caracter de inicio
                TxChar(',');
                DataGetchar(&e_dato);
                TxChar(e_dato);//Envia caracter de el numero de veces
                TxChar(',');
                i=1;
                ii=0;//Saca la fecha de la memoria
                while(ii=FALSE)
                {
                    DataGetchar(&e_dato);
                    TxByte(e_dato);
                    TxChar(',');
                    if(ii==5)
                    {
                        i=0;
                    }
                }
                ii++;
            }
        }
    }
}

```

```

    }
    rep=0;
    chek=0;
    muestras=0;

    break;
}

default:

    pa=&a[rep];
    *pa=e_dato;
    chek=chek+e_dato;
    TxByte(e_dato);
    TxChar('.');

    rep++;
    if(rep==58)
    {

        rep=0;
        muestras++;

        TxLong(chek);
        bandera=1;
        ch = CHSUM();
        chek=0;
        if(ch==1)
        {
            flag4=0;
        }
        break;
    }
}

    TxString("FinDatos ");
    TxChar(0x0D); //coloca un CR
    TxChar(0x0A); //coloca un LF

flag1=1;
flag4=0;
}
/*****
/*muestreo de baterias*/
xmem int baterias()
{
int i;
ii=0;

DI();
/*manda acero el amplificador de instrumentacion*/
up_setout(ch1,off); //Bit menos significativo
up_setout(ch2,off);
up_setout(ch3,off); //Bit más significativo
up_setout(ch4,on); //Enable

/*Habilita canal 2 del ADG 527A*/
up_setout(ch1,off); //Bit menos significativo
up_setout(ch2,on);
up_setout(ch3,off); //Bit más significativo
up_setout(ch4,on); //Enable

/*Habilita el muestreo del voltaje total de las baterias decodificador 74LS138 */
/*Habilita DRR3 y DDR4*/
up_setout(ch11,on);
up_setout(ch10,off);
up_setout(ch9,off);
up_setout(ch12,on);
convert2();

up_setout(ch11,off);
/*
    contador */
/* Multiplexaje de la tarjeta 1 de baterias*/
/*De la bateria 1 a la 16*/
i=0;
while(i!=4){
    if(i==0)
    {
        up_setout(ch5,off); //0000 1
        up_setout(ch6,off);
        up_setout(ch7,off);
        up_setout(ch8,off);
    }
}

```

```

convert();

up_setout(ch5,on); //0001 2
convert();

up_setout(ch5,off);//0010 3
up_setout(ch6,on);
convert();

up_setout(ch5,on); //0011 4
convert();

up_setout(ch5,off);//0100 5
up_setout(ch6,off);
up_setout(ch7,on);
convert();

up_setout(ch5,on);//0101 6
convert();

up_setout(ch5,off);//0110 7
up_setout(ch6,on);
convert();

up_setout(ch5,on);//0111 8
convert();

up_setout(ch5,off);//1000 9
up_setout(ch6,off);
up_setout(ch7,off);
up_setout(ch8,on);
convert();

up_setout(ch5,on); //1001 10
convert();

up_setout(ch5,off);//1010 11
up_setout(ch6,on);
convert();

up_setout(ch5,on); //1011 12
convert();

up_setout(ch5,off);//1100 13
up_setout(ch6,off);
up_setout(ch7,on);
convert();

up_setout(ch5,on);//1101 14
convert();

up_setout(ch5,off);//1110 15
up_setout(ch6,on);
convert();

up_setout(ch5,on);//1111 16
convert();
}
else if(i==i)
{
up_setout(ch9,on);//Habilita la segunda tarjeta

/*muestreo de la bateria 17 a la 26*/
/*muestreo de la bateria 1 al 26 del segundo banco*/

up_setout(ch5,off); //0000 17
up_setout(ch6,off);
up_setout(ch7,off);
up_setout(ch8,off);
convert();

up_setout(ch5,on); //0001 18
convert();

up_setout(ch5,off);//0010 19
up_setout(ch6,on);
convert();

up_setout(ch5,on); //0011 20
convert();

```

```

up_setout(ch5,off);//0100 21
up_setout(ch6,off);
up_setout(ch7,on);
convert();

up_setout(ch5,on);//0101 22
convert();

up_setout(ch5,off)//0110 23
up_setout(ch6,on);
convert();
up_setout(ch5,on)//0111 24
convert();

up_setout(ch5,off)//1000 25
up_setout(ch6,off);
up_setout(ch7,off);
up_setout(ch8,on);
convert();

up_setout(ch5,on); //1001 26
convert();

up_setout(ch5,off)//1010 1 27
up_setout(ch6,on);
convert();

up_setout(ch5,on); //1011 2 28
convert();

up_setout(ch5,off)//1100 3 29
up_setout(ch6,off);
up_setout(ch7,on);
convert();

up_setout(ch5,on)//1101 4 30
convert();

up_setout(ch5,off)//1110 5 31
up_setout(ch6,on);
convert();

up_setout(ch5,on)//1111 6 32
convert();
}

else if(i==2)
{
up_setout(ch9,off);//Habilita la tercera tarjeta
up_setout(ch10,on);
/* muestreo de la bateria 7 a la 22 del segundo banco*/

up_setout(ch5,off); //0000 7 33
up_setout(ch6,off);
up_setout(ch7,off);
up_setout(ch8,off);
convert();

up_setout(ch5,on); //0001 8 34
convert();

up_setout(ch5,off)//0010 9 35
up_setout(ch6,on);
convert();

up_setout(ch5,on); //0011 10 36
convert();

up_setout(ch5,off)//0100 11 37
up_setout(ch6,off);
up_setout(ch7,on);
convert();

up_setout(ch5,on)//0101 12 38
convert();

up_setout(ch5,off)//0110 13 39
up_setout(ch6,on);

```

```

convert();

up_setout(ch5,on);//0111 14 40
convert();

up_setout(ch5,off)//1000 15 41
up_setout(ch6,off);
up_setout(ch7,off);
up_setout(ch8,on);
convert();

up_setout(ch5,on); //1001 16 42
convert();

up_setout(ch5,off)//1010 17 43
up_setout(ch6,on);
convert();

up_setout(ch5,on); //1011 18 44
convert();

up_setout(ch5,off)//1100 19 45
up_setout(ch6,off);
up_setout(ch7,on);
convert();

up_setout(ch5,on)//1101 20 46
convert();

up_setout(ch5,off)//1110 21 47
up_setout(ch6,on);
convert();

up_setout(ch5,on)//1111 22 48
convert();
}
else if(i==3)
{
up_setout(ch9,on);//Habilita la segunda cuarta

/*muestreo de la bateria 23 a la 26*/
/*Del segundo banco*/
up_setout(ch5,off);//0000 23 49
up_setout(ch6,off);
up_setout(ch7,off);
up_setout(ch8,off);
convert();

up_setout(ch5,on); //0001 24 50
convert();

up_setout(ch5,off)//0010 25 51
up_setout(ch6,on);
convert();

up_setout(ch5,on); //0011 26 52
convert();

up_setout(ch5,off)//0100
up_setout(ch6,off);
up_setout(ch7,on);

}
i++;
}
/*Deshabilita decodificador 74LS138*/

up_setout(ch12,off);

/*Habilita canal 3 del ADG527*/
up_setout(ch1,on);
up_setout(ch2,on);
up_setout(ch3,off);
up_setout(ch4,on);
convert();

/*Habilita canal 4 del ADG527*/

```



```

up_setout(ch1,off);
up_setout(ch2,off);
up_setout(ch3,on);
convert1();

cuenta=0L;
up_setout(ch4,off);
apagado();
cuenta =0L;
EI();

}
/*****
xmem void apagado ()

up_setout(ch1,off);
up_setout(ch2,off);
up_setout(ch3,off);
up_setout(ch4,off); /*enable del ADG527*/
up_setout(ch5,off);
up_setout(ch6,off);
up_setout(ch7,off);
up_setout(ch8,off);
up_setout(ch9,off);
up_setout(ch10,off);
up_setout(ch11,off);
up_setout(ch12,off)/*enable del 74ls138*/
// up_setout(ch13,off);
}
/*****
xmem int convert()
{
int in_ch;
in_ch=5;
add_sig_table(0,tarjeta,in_ch,0,0); /* Create Signal Table w/calibration */
set_mux(0);
mxdel18(); /*retraso*/
rd = plad_rd12(adch(0));
batac = ad_conv(0,rd);

if ((batac>0) OR (batac==0)){
batac =batac*100;
trs =batac/1;
/*Compara el nivel de carga de la bateria*/
if((trs<100) OR (trs==100))
{
bad=1;
}
DataPushchar(trs);
}
else if(batac< 0)
{
batac=batac*(-100);
trs = batac/1;
/*Compara el nivel de carga de la bateria*/
if((trs<100) OR (trs==100))
{
bad=1;
}
DataPushchar(trs);
}

ii++;
}
/*****
xmem int convert1()
{
int in_ch;
in_ch=5;
add_sig_table(0,tarjeta,in_ch,0,0); /* Create Signal Table w/calibration */
set_mux(0);
mxdel18(); /*retraso*/
rd = plad_rd12(adch(0));
batac = ad_conv(0,rd);
if ((batac>0) OR (batac==0)){
batac =batac*100;
trs =batac/1;
trs=trs-10;
}
}

```

```

        DataPushchar(trs);
        }
        else if(bate< 0)
        {
            bate=bate*(-100);
            trs= bate/1;
            trs=trs-10;
            DataPushchar(trs);
        }
        ii++;
    }
    /*.....*/
    xmem int convert2()
    {
        int in_ch;
        in_ch=5;
        add_sig_table(0,tarjeta_in_ch,0,0); /* Create Signal Table w/calibration */
        set_max(0);
        mxdel18(); /*retraso*/
        rd = plad_rd12(adch(0));
        bate = ad_conv(0,rd);

        if ((bate>0) OR (bate==0)){
            bate =bate*100;
            trs =bate/1;
            /*Compara el nivel de carga de la bateria*/
            if((trs<215) OR (trs==215))
            {
                bad=1;
            }
            DataPushchar(trs);

            ii++;
        }
    }
    /*.....*/
    /*Realiza el checksum*/
    int CHSUM()
    {
        int con, chec, k, result;
        char v;
        char RxBuffer[30];
        result=0;
        k=0;
        con=1;
        while(con==1)
        {
            if(!Dread_zt1ch(RxBuffer)!=FALSE)
            {
                switch(*RxBuffer)
                {
                    case'A': bandera=0;
                            con=0;
                            break;

                    case'B':chec=0;
                            for(k=0;k<58;k++)
                            {
                                pa=&a[k];
                                v=*pa;

                                chec=chec+v;
                                TxByte(v);
                                TxChar(',');
                            }

                                TxLong(chec);

                                break;
                    case'C': bandera=0;//termina totalmente el bajar los datos
                            con=0;
                            result=1;
                            break;

                    case'D': TxChar(0x45);//transmite una E de espera
                            TxChar(',');
                }
            }
        }
    }

```

```

        *RxBuffer=""; //se queda en espera
        Dreset_z1rbuf();
        while(Dread_z1rch(RxBuffer)==FALSE)
        {
            NULL;
        }

        bandera=0;
        con=0;
        break;

    case 'S': TxChar(0x52);/* R manda cartacter de estado bajando datos */
             TxChar(',');

        default:break;
    }
}

return(result);
}

/*.....*/
/*.....*/
xmem int checkbat()
{
    if (bad==1)
    {
        up_setout(ch14,on);//indica el voltaje total o voltaje en alguna bateria
    }
}
/*.....*/
/* Al generar una interrupcion aumenta la variable cuenta */
/*.....*/

#INT_VEC PRT1_VEC contar
interrupt reti int contar()
{
    inport(TCR);
    inport(TMDR1L);
    EI();
    cuenta=cuenta+1;
}
/*.....*/
/*.....*/
/*.....*/
/****** Lee datos de fecha y hora de inicio de muestreo *****/
/*.....*/
xmem void fecha_time()
{
    tm_rd(&tiempo);
}

/*.....*/
/*habilita interrupciones y carga el valor del contador en timer! */
/*.....*/
int prt1_init(int tc)
{
    IRES(TCR,TDE1);
    IRES(TCR,TIE1);
    outport(TMDR1L, tc);
    outport(TMDR1H, tc >> 8);
    outport(RLDR1H, tc >> 8);
    outport(RLDR1L, tc);
    ISET(TCR, TDE1);
    ISET(TCR, TIE1);
    EI();
}

/*.....*/
/*Modifica la fecha*/
xmem void actual()
{
    char RxBuffer[25];
    int con,VAL,bandera;
}

```



```

                                Dreset_z1rbuf();
while(Dread_z1lch(RxBuffer)==FALSE)
{
    NULL;
}
    tiempo.tm_mday=*RxBuffer;

TxString("mes");
TxChar(',');
*RxBUFFER=' ';                                Dreset_z1rbuf();

while(Dread_z1lch(RxBuffer)==FALSE)
{
    NULL;
}
    tiempo.tm_mon=*RxBuffer;

TxString("año");
TxChar(',');
*RxBUFFER=' ';                                Dreset_z1rbuf();

while(Dread_z1lch(RxBuffer)==FALSE)
{
    NULL;
}
    tiempo.tm_year=*RxBuffer;

TxString("día");
TxChar(',');
*RxBUFFER=' ';                                Dreset_z1rbuf();

while(Dread_z1lch(RxBuffer)==FALSE)
{
    NULL;
}
    tiempo.tm_wday=*RxBuffer;
TxChar(0x4D);
TxChar(',');
bandera=FALSE;
}
tm_wt(&tiempo);

VAL=0;
con=0;
}
}
break;
}
}

}

.....
/*
/* Módulo que establece la comunicación entre la tarjeta de la TCAM y la Little-G
/*
/*
.....

void escribir();
void adc();

void TCAM()
{
up_setout(ch13,on); //latch activado
up_setout(ch14,on); //escritura desactivada

#asm
eti: ld b,0x22
        out0(BUSRESET),b
        in0 a,(BUSRD0)
        cp 0x01
        jp nz,eti
et1: ld b,0x77

```

```

        out0(BUSADR0),b
in0 a,(BUSRD0)
    cp 0x02
    jp nz,et1
et2: ld b,0x44
        out0(BUSADR1),b
    in0 a,(BUSRD0)
        cp 0x03
    jp nz,et2
et3: ld b,0x88
        out0(BUSADR2),b
    in0 a,(BUSRD0)
        cp 0x04
    jp nz,et3
#endasm
escribir();
}

void escribir()
{
up_setout(ch13,on);
up_setout(ch14,on); //desactivada la escritura

#asm
etg: ld b,0x00
        out0(BUSWR),b
    in0 a,(BUSRD0)
        cp 0x05
    jp nz,etg
gvv: ld b,0x44
        out0(BUSWR),b
    in0 a,(BUSRD0)
        cp 0x0E
    jp nz,gvv
#endasm
adc();
}

void adc()
{
    char a[30000],*pa;
    int cont,conver;
    float rd;
    int j,i,conta, bd_addr, in_ch=0, an_ch=0;
    float vol,tempc,tvol;
    char temp;
    char pxc,pyc;
    int pxi,pyi;

    reset_pbus();
    bd_addr = plead_addr(3);
    add_sig_table(0,bd_addr,in_ch,an_ch,1);
    set_mux(0);
    mxdel18();

    up_setout(ch13,off); //desactivado el latch para permitir la adquisicion
    up_setout(ch14,on);

    for(i=0; i<=3000;i++)
    {
        NULL;
    }

    for (i=0;i<30000;i++)
    {
        pa=&a[i];
        rd=ad_conv(0,plad_rd12(0));
        rd=100*rd;
        *pa=rd;
        // printf("n%d valor ",*pa);
        // printf("n%f valor ",rd);
    }
}

```

```

up_setout(ch13,on); //activando el latch
up_setout(ch14,on);

#asm
ettl: ld b,0x33
      out0(BUSWR),b
      in0 a,(BUSRD0)
      cp 0x0B
      jp nz,ettl
#endasm

up_setout(ch14,off); //activando la escritura
up_setout(ch13,on); //activando el latch para permitir la adquisicion

//for (j=0;j<35;j++)
//{
for (i=0;i<30000;i++)
|
    pa=&a[i];
/* TRANSFIERE EL DATO */

#asm
etd1: ld a,(hl)
      out0(BUSWR),a
      in0 a,(BUSRD0)
      cp 0xDC
      jp nz,etd1
#endasm

//}
}

up_setout(ch14,on); //desactivando la escritura
up_setout(ch13,on); //activando el latch

|

```

PROGRAMA EN VISUAL BASIC

```

.....
/* Programa Sistema 26.
/* Este programa es una aplicacion en Visual Basic y proporciona una interfaz
/* para que el usuario opere el sistema
/*
.....
/*BAJAR*/
Private Sub Command1_Click()

Dim Tx As String
Dim informacion As String
Dim auxiliar As Integer
Dim TIEMPO As Variant
Dim Segundos As Integer
Dim Minutos As Integer
Dim HORAS As Integer
Dim Dia As Integer
Dim Mes As Integer
Dim YEARD As Integer
Dim Velocidad As Integer
Dim variable As String
Dim I1 As Integer
Dim I2 As Integer
Dim V1 As Integer
Dim Dato As Variant
Dim T1 As Integer
Dim T2 As Integer
Dim ssuma As String
Dim tem As Variant
Dim Bajar_Auxiliar As String

Tx = "4"
espera = 500
informacion = EnviarInformacion(Tx)
Bajar_Auxiliar = informacion
ssuma = ""

If "y" = Mid(informacion, 1, 1) Then 'entonces Primer Dato primer if
MousePointer = 11
J = 0
auxiliar = InStr(informacion, ",")
titulo = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
TIEMPO = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
Segundos = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
Minutos = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
HORAS = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
Dia = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
Mes = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
YEARD = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
temp = 0
For i% = 0 To 57

auxiliar = InStr(informacion, ",")
variable = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
tem = Val(variable)
temp = temp + tem

Next i%
ssuma = Val(informacion)
Do
If ssuma = temp Then 'segundo if

```



```

TMP = TABLADATOS(Bajar_Auxiliar)
informacion = EnviarInformacion("A")
Bajar_Auxiliar = informacion

If "y" = Mid(informacion, 1, 1) Then 'entonces Primer Dato tercer if
    J = J + 1
    auxiliar = InStr(informacion, ",")
    titulo = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    TIEMPO = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    Segundos = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    Minutos = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    HORAS = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    Dia = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    Mes = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    YEARD = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    temp = 0

End If 'fin tercer if

temp = 0
ssuma = 0

If InStr(1, informacion, "FinDatos") Then 'cuarto if
    MsgBox "tabla completa"
    Exit Do
End If 'fin cuarto if

For ii% = 0 To 57

    auxiliar = InStr(informacion, ",")
    variable = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    tem = Val(variable)
    temp = temp + tem

Next ii%

ssuma = Val(informacion)

Else 'else del segundo if

Do
    informacion = EnviarInformacion("B")
    Bajar_Auxiliar = informacion

    For ii% = 0 To 57

        auxiliar = InStr(informacion, ",")
        variable = Mid(informacion, 1, auxiliar - 1)
        informacion = Mid(informacion, auxiliar + 1, Len(informacion))
        tem = Val(variable)
        temp = temp + tem

    Next ii%
    ssuma = Val(informacion)
    If ssuma = temp Then 'quinto if
        TMP = TABLADATOS(Bajar_Auxiliar)
        Exit Do

    End If 'fin quinto if

Loop

End If 'fin segundo if

```

```

Loop
End If 'fin primer if
  MousePointer = 0
End Sub

/*ESTADO DEL SISTEMA*/

Private Sub CommandE_Click()
  F_Menu.Enabled = True
  F_ESTADO.Visible = False
End Sub

/*FECHA*/

Private Sub Command1_Click()

  Dim informacion As String
  Dim auxiliar As Integer
  MousePointer = 11
  informacion = EnviarInformacion("M")
  informacion = EnviarInformacion("M")
  informacion = EnviarInformacion("T")
  auxiliar = InStr(informacion, ",")
  Fecha.FechaActual.Text = Mid(informacion, 1, auxiliar - 1)
  informacion = Mid(informacion, auxiliar + 1, Len(informacion))
  Fecha.HoraActual.Text = informacion
  Fecha.Modificar.Enabled = True
  MsgBox temp, 0, "FECHA Y HORA"
  MousePointer = 0
End Sub

/*MENU*/

Private Sub ESTADO_SIST_Click()
'On Error GoTo Error_ConPto_Conexcion

  Dim informacion As String
  Dim titulo As String      ' Menu
  Dim estado As String      ' Muestrando
  Dim modo As String        ' BD
  Dim hora As String        ' Hora de inicio
  Dim fecha As String       ' Fecha de inicio de muestreo
  Dim tiempo As Integer     ' Tiempo de muestreo
  Dim vuelta As Integer     ' Numero de muestras
  Dim auxiliar As Integer
  Dim infor As String

  'informacion = EnviarInformacion("5")
  F_ESTADO.Show
  'espera = 6000
  If "" = informacion Then
    'espera = 6000
    'informacion = ""
    informacion = EnviarInformacion("5")
  'Exit Sub
  End If
  auxiliar = InStr(informacion, ",")
  estado = Mid(informacion, 1, auxiliar - 1)
  informacion = Mid(informacion, auxiliar + 1, Len(informacion))

  If "A" = estado Then
    auxiliar = InStr(informacion, ",")
    tiempo = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    fecha = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    hora = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    vuelta = informacion
    F_ESTADO.ESTADO1.Text = fecha
    F_ESTADO.ESTADO2.Text = hora
    vuelta = ((7940 - vuelta) / 7940) * 100
  
```

APENDICE B

```

MsgBox "FECHA Y HORA DE INICIO DEL MUESTREO"
F_ESTADO.ESTADO1.Text = vuelta
F_ESTADO.ESTADO2.Text = "PORCIENTO"
MsgBox "PORCENTAJE DE MEMORIA DISPONIBLE "

```

```
End If
```

```
If "B" = estado Then
```

```

auxiliar = InStr(informacion, ",")
modo = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
auxiliar = InStr(informacion, ",")
tempo = Mid(informacion, 1, auxiliar - 1)
auxiliar = InStr(informacion, ",")
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
vuelta = Mid(informacion, 1, auxiliar - 1)
vuelta = ((7940 - vuelta) / 7940) * 100
F_ESTADO.ESTADO1.Text = vuelta
F_ESTADO.ESTADO2.Text = "PORCIENTO"
MsgBox "PORCENTAJE DE MEMORIA DISPONIBLE "

```

```
If "C" = modo Then
```

```

F_ESTADO.ESTADO2.Text = "%"
MsgBox "Con Datos en Memoria"

```

```
End If
```

```
MsgBox "NO MUESTREANDO"
```

```
End If
```

```
End Sub
```

```
/*PRESENTACION*/
```

```
Private Sub Command1_Click()
```

```

F_Menu.Show
F_Presentacion.Visible = False
espera = 100
End Sub

```

```
/*PUERTO*/
```

```
Private Sub ConPto_Conceccion_Click()
```

```
On Error GoTo Error_ConPto_Conceccion
```

```

Dim informacion As String
Dim titulo As String      ' Menu
Dim estado As String      ' Muestreando
Dim modo As String        ' BD
Dim hora As String        ' Hora de inicio
Dim fecha As String       ' Fecha de inicio de muestreo
Dim tempo As Integer      ' Tiempo de muestreo
Dim vuelta As Integer      ' Numero de muestras
Dim vuelta1 As Double     ' Numero total muestras
Dim auxiliar As Integer

```

```

ConPto_Puerto.DTREnable = True      ' Deshabilita la linea DTR (Lectura)
ConPto_Puerto.RTSEnable = True      ' Deshabilita la linea RTS (Peticon)
ConPto_Puerto.RThreshold = 1        ' # de chars recibidos antes de la comunicacion
ConPto_Puerto.SThreshold = 1        ' # chars minimo en la transmision antes de la comunicacion
ConPto_Puerto.Settings = "19200,n,8,1" ' Baud-Rate=9600 No-parid 8 bits de datos 1 bits de stop
ConPto_Puerto.InputLen = 1
ConPto_Puerto.CommPort = NumeroPuerto
ConPto_Puerto.PortOpen = True

```

```

If True = ConPto_Puerto.PortOpen Then
informacion = EnviarInformacion("M")
informacion = EnviarInformacion("S")
espera = 200
If "" = informacion Then

```

```
MsgBox "EL MODULO NO SE ENCUENTRA" + Chr(13) + Chr(10) + "CONECTADO POR EL PUERTO # " + Str(NumeroPuerto), 0, "MENSAJE DE PUERTO"
```

```
ConPto_Puerto.PorOpen = False
ConPto_Conexion.Enabled = True
ConPto_Desconexion.Enabled = False
F_Menu.iniciar_muestreo.Enabled = False
F_Menu.terminar_muestreo.Enabled = False
F_Menu.bajar_informacion.Enabled = False
F_Menu.datos_muestreo.Enabled = False
F_Menu.M.Enabled = False
ConPto_Comm1.Enabled = True
ConPto_Comm2.Enabled = True
ConPto_Comm3.Enabled = True
ConPto_Comm4.Enabled = True
Exit Sub
End If
```

```
MsgBox "COMUNICACION ESTABLECIDA POR PUERTO " + Str(NumeroPuerto), 0, "MENSAJE DE PUERTO"
```

```
ConPto_Desconexion.Enabled = True
ConPto_Conexion.Enabled = False
auxiliar = InStr(informacion, ",")
estado = Mid(informacion, 1, auxiliar - 1)
informacion = Mid(informacion, auxiliar + 1, Len(informacion))
```

```
If "A" = estado Then
    auxiliar = InStr(informacion, ",")
    tempo = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    fecha = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    hora = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    vuelta = informacion
    fecha_texto.Text = fecha
    hora_texto.Text = hora
    vuelta = ((7940 - vuelta) / 7940) * 100
    MsgBox "FECHA Y HORA DE INICIO DEL MUESTREO"
    fecha_texto.Text = vuelta
    hora_texto.Text = "PORCIENTO"
    MsgBox "PORCENTAJE DE MEMORIA DISPONIBLE "
    F_Menu.TiempoDeMuestreo.Enabled = False
    F_Menu.iniciar_muestreo.Enabled = False
    F_Menu.terminar_muestreo.Enabled = True
    F_Menu.bajar_informacion.Enabled = False
    F_Menu.datos_muestreo.Enabled = False
End If
```

```
If "B" = estado Then
    auxiliar = InStr(informacion, ",")
    modo = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    tempo = Mid(informacion, 1, auxiliar - 1)
    informacion = Mid(informacion, auxiliar + 1, Len(informacion))
    auxiliar = InStr(informacion, ",")
    vuelta = Mid(informacion, 1, auxiliar - 1)
    vuelta1 = Val(vuelta)
    vuelta1 = ((7940 - vuelta1) / 7940) * 100
    fecha_texto.Text = Str(vuelta1)
    hora_texto.Text = "PORCIENTO"
    MsgBox "PORCENTAJE DE MEMORIA DISPONIBLE "
```

```
If "C" = modo Then
    hora_texto.Text = "% "
    MsgBox "Con Datos en Memoria"
```

```
F_Menu.terminar_muestreo.Enabled = False
F_Menu.bajar_informacion.Enabled = True
F_Menu.datos_muestreo.Enabled = False
```

```
End If
```

```
If "D" = modo Then
    MsgBox "No Hay Datos en Memoria"
    F_Menu.terminar_muestreo.Enabled = False
    F_Menu.bajar_informacion.Enabled = False
```

```

F_Menu.datosdemuestreo.Enabled = True
End If
MsgBox "NO MUESTREANDO"
F_Menu.TiempoDeMuestreo.Enabled = True
F_Menu.iniciar_muestreo.Enabled = True
F_Menu.terminar_muestreo.Enabled = False
F_Menu.bajarinformacion.Enabled = False
End If

```

```

F_Menu.M.Enabled = True
Else
MsgBox "COMUNICACION NO ESTABLECIDA POR PUERTO " + Str(NumeroPuerto), 0, "MENSAJE DE PUERTO"
ConPto_Desconecion.Enabled = False
ConPto_Conecion.Enabled = True
F_Menu.iniciar_muestreo.Enabled = False
F_Menu.terminar_muestreo.Enabled = False
F_Menu.bajarinformacion.Enabled = False
F_Menu.datosdemuestreo.Enabled = False
F_Menu.M.Enabled = False
Exit Sub
End If

```

```

Select Case NumeroPuerto
Case 1
ConPto_Comm2.Enabled = False
ConPto_Comm3.Enabled = False
ConPto_Comm4.Enabled = False
Case 2
ConPto_Comm1.Enabled = False
ConPto_Comm3.Enabled = False
ConPto_Comm4.Enabled = False
Case 3
ConPto_Comm1.Enabled = False
ConPto_Comm2.Enabled = False
ConPto_Comm4.Enabled = False
Case 4
ConPto_Comm1.Enabled = False
ConPto_Comm2.Enabled = False
ConPto_Comm3.Enabled = False
End Select

```

Exit Sub

Error_ConPto_Conecion:

```

Select Case (Err)
Case 8000 To 8020
MsgBox "COMUNICACION NO ESTABLECIDA POR PUERTO " + Str(NumeroPuerto), 0, "MENSAJE DE PUERTO"
Exit Sub
Case Else
MsgBox "ERROR " + Str$(Err), 0, "ERROR"
End
End Select

```

End Sub

PROGRAMA DE LA TARJETA DE ALMACENAMIENTO EXTERNO

```

.....
.* Programa de transferencia de datos del Zworld a la tarjeta      *
.* TCRAM                                                         *
.* Ultima modificación 11 de mayo de 1998                       *
.* Este programa libera dos líneas del Cop (W y R) y usa los 4 pines *
.* que se adicionan al usar el cop8sac44, los cuales se usaran para la *
.* comunicación serial y para direccionar hasta 4Mbytes de memoria. *
.....

```

```

.TITLE INTERF51
.INCLD COP8SAC.INC
.SECT MAIN,ROM

.SET COUNT, 0X0F0
.SET DLYCNT, 0X0F1
.SET CONT, 0X0F2
.SET BAUD_Y, 0X0F3
.SET BAUD_X, 0X0F4
.SET BAUD_1, 0X0F5
.SET BAUD_2, 0X0F6
.SET BAUD_3, 0X0F7
.SET BAUD_4, 0X0F8
.SET DIR1,0XF9
.SET DIR2,0XFA
.SET DIR3,0XFB

```

```

RCVREG=0X00
XMTREG=0X01

```

```

.....
.* Esta parte del programa permite la inicialización de la comunicación *
.* con la tarjeta del COP                                         *
.....

```

```

INIC:
.ORG 0X00
LD PORTFC,#0XFF ;COMO SALIDAS
LD PORTGC,#0X07 ;COMO ENTRADAS G7-G3 / COMO SALIDAS G2-G0
LD PORTFD,#0X00 ;INICIO
LD PORTGD,#0X07 ;HABILITANDO LATCH Y BUFFER
LD DIR1,#0X00 ;INICIALIZANDO LOS
LD DIR2,#0X00 ;PUERTOS A LA DIRECCION
LD DIR3,#0X00 ;000000

```

```

H: IFBIT 7,PORTGP
JMP H
LD A,PORTGP
AND A,#0X70
IFNE A,#0X60 ;60h E0
JMP H
LD PORTFC,#0X00 ;COMO ENTRADAS PORTF
LD A,PORTFP
XOR A,#0X22 ;EL RESET QUE ENVIA ES PARA ESTA TARJETA
IFNE A,#0X00
JMP H
LD PORTFC,#0XFF ;COMO SALIDA PORTF
LD PORTFD,#0X01 ;CONTINUA RESET
LD PORTD,#0X04

```

```

ESPERA: IFBIT 7,PORTGP
JMP ESPERA
LD A,PORTGP
AND A,#0X70
IFNE A,#0X10 ;10h 90
JMP ESPERA
LD PORTFC,#0X00 ;COMO ENTRADAS PORTF

```

```

LD A,PORTFP
XOR A,#0X77
IFNE A,#0X00
JMP ESPERA
LD PORTFC,#0XFF ;COMO SALIDA PORTF
LD PORTFD,#0X02 ;CONTINUA 1ER NIBBLE
LD PORTD,#0X0A
ESPERA1: IFBIT 7,PORTGP
JMP ESPERA1
LD A,PORTGP
AND A,#0X70
IFNE A,#0X50 ;50h D0
JMP ESPERA1
LD PORTFC,#0X00 ;COMO ENTRADAS PORTF
LD A,PORTFP
XOR A,#0X44
IFNE A,#0X00
JMP ESPERA1
LD PORTFC,#0XFF ;COMO SALIDA PORTF
LD PORTFD,#0X03 ;CONTINUA SEGUNDO NIBBLE
LD PORTD,#0X0F
ESPERA2: IFBIT 7,PORTGP
JMP ESPERA2
LD A,PORTGP
AND A,#0X70
IFNE A,#0X30 ;30h B0
JMP ESPERA2
LD PORTFC,#0X00 ;COMO ENTRADA EL PORTF
LD A,PORTFP
XOR A,#0X88
IFNE A,#0X00
JMP ESPERA2
LD PORTFC,#0XFF ;COMO SALIDA EL PORTF
LD PORTFD,#0X04 ;CONTINUA 3ER NIBBLE
LD B,#0X30
    
```

```

.....
:* En esta parte del programa se toman los datos que corresponden *
:* a la direcci3n de inicio y nLmero de datos *
.....
    
```

```

ESPERA3: LD PORTFC,#0X00
IFBIT 7,PORTGP
JMP ESPERA3
LD A,PORTGP
AND A,#0X70
IFNE A,#0X70
JMP ESPERA3
LD A,PORTFP
X A,[B+]
LD PORTFC,#0XFF ;COMO SALIDA EL PORTF
LD PORTFD,#0X05 ;CONTINUA 1ER DATO
NOP
NOP
ESP: DEC A
IFNE A,#0X00
JMP ESP
    
```

```

.....
:* PROCESO DE LECTURA DE LA TCRAM *
.....
    
```

```

LD PORTGD,#0X00 ;DESHABILITAR LOS ENABLES
LD PORTLC,#0XFF ;COMO SALIDAS
LD PORTCC,#0XFF ;COMO SALIDAS
TERMINO: LD A,DIR1 ;CARGA EN LOS PUERTOS EL VALOR DE
    
```

```

X A,PORTD ;LA ULTIMA DIRECCION ACCESADA
LD A,DIR2 ;DE LA TARJETA
X A,PORTLD ;PARA QUE EN EL
LD A,DIR3 ;PROXIMO ACCESO
X A,PORTLC ;CONTINUE A PARTIR DE AHI
LD A,0x30 ;DETERMINA LA FUNCION A REALIZAR
IFNE A,#0X00 ;SEA LECTURA O ESCRITURA 0=ESCRIBIR 1=LEER
JMP CLEE
JMP WAIT
CLEE: IFBNE #0X01
      JMP 00
      JMP LEE
      NOP
LEE: JSR SERIAL
      LD B,#0X05
      LD PORTFC,#0X00
SIG: LD PORTGD,#0X01 ;LEER PARTE BAJA
      LD A,PORTFP
      X A,[B+]
      LD A,B
      IFNE A,04
      JMP PUERTO
      LD PORTFC,#0XFF ;COMO SALIDAS
      LD PORTFD,#0X09
      JMP 00
PUERTO: LD A,PORTD
        IFNE A,#0XFF
        JMP INCD
        LD PORTD,#0X00
        LD A,PORTLD
        IFNE A,#0XFF
        JMP INCL
        LD PORTLD,#0X00
        LD A,PORTCD
        IFNE A,#0X7F ;HASTA 4Mb DE DIRECCIONAMIENTO
        JMP INCC
        JMP ESPERA3 ;REGRESA A LA SELECCION DE FUNCIONES
    
```

```

.....
* PROCESO DE ESCRITURA DE LA TCRAM *
.....
    
```

```

WAIT: LD PORTFC,#0X00 ; COMO ENTRADAS EL PUERTO F
      IFBIT 7,PORTGP
      JMP WAIT
      LD A,PORTGP
      AND A,#0X70
      IFNE A,#0X70
      JMP WAIT
      LD A,PORTFP
      XOR A,#0X44
      IFNE A,#0X00
      JMP WAIT
      LD PORTFC,#0XFF ;COMO SALIDA PORTF
      LD PORTFD,#0X0E ;CONTINUA EL ALMACENAMIENTO DE DATOS
      NOP
      NOP
      NOP
ESCRIBE: LD B,#0X00
PRLT: LD PORTFC,#0X00 ;COMO ENTRADAS EL PUERTO F
      IFBIT 7,PORTGP
      JMP PRLT
      LD A,PORTGP
      AND A,#0X70
      IFNE A,#0X70
      JMP PRLT
      LD A,PORTFP
      XOR A,#0X33
      IFNE A,#0X00
    
```



```

JMP PRLT
LD PORTFC,#0XFF ;COMO SALIDA PORTF
LD PORTFD,#0X0B ;CONTINUA EL ALMACENAMIENTO DE DATOS
NOP
NOP
LD CONT,#0X00
SBIT 5,PORTCD
CUENTA: LD A,CONT
        IFNE A,#0X64
        JMP PRE1
        JMP ACTUAL
        NOP
        NOP
ACTUAL: LD A,PORTD
        X A,DIR1 ;REGISTRO DONDE GUARDA LA ULTIMA DIRECCION
        LD A,PORTLD
        X A,DIR2 ;REGISTRO DONDE GUARDA LA ULTIMA DIRECCION
        LD A,PORTCD
        X A,DIR3 ;REGISTRO DONDE GUARDA LA ULTIMA DIRECCION
        JMP ESPERA3
PRE1: SBIT 5,PORTCD
      LD PORTFC,#0X00 ;COMO ENTRADAS EL PUERTO F
      LD PORTGD,#0X01 ;HABILITA EL ENABLE 1
      RBIT 5,PORTCD
      LD A,PORTFP ;CARGA EL DATO EN A
; SBIT 5,PORTCD
      X A,[B+]
      LD PORTGD,#0X03 ;DESHABILITADA LOS DATOS
      LD PORTFC,#0X00 ;DESHABILITA LOS DATOS
      JSR PUERTO1
      LD A,B
      IFNE A,#0XFC ;CICLO DE 2n DATOS NO LLEGA 0xFF
      JMP PRE1
      LD A,CONT
      INC A
      X A,CONT
      JMP CUENTA
INCD: INC A
      X A,PORTD
      JMP SIG
INCL: INC A
      X A,PORTLD
      JMP SIG
INCC: INC A
      X A,PORTCD
      JMP SIG
PUERTO1: LD A,PORTD
          IFNE A,#0XFF
          JMP INCD1
          LD PORTD,#0X00
          LD A,PORTLD
          IFNE A,#0XFF
          JMP INCL1
          LD PORTLD,#0X00
          LD A,PORTCD
          IFNE A,#0X7F ;HASTA 4Mb DE DIRECCIONAMIENTO
          JMP INCC1
          LD PORTCD,#0X00
          RET
          JMP ESPERA3 ;REGRESA A LA SELECCION DE FUNCIONES
INCD1: INC A
      X A,PORTD
      RET
INCL1: INC A
      X A,PORTLD
      RET
INCC1: INC A
      X A,PORTCD
      RET
    
```

```

.....
* COMUNICACION SERIAL CON UN BAUD RATE DE 1200 BPS
*
.....

```

```

SERIAL: LD BAUD_Y,#0X83 ;34+6X=820 uS
          LD BAUD_X,#0X83 ;32+6X=820 uS
          LD BAUD_1,#0X84 ;26+6X=820
          LD BAUD_2,#0XFF ;retraso largo
          LD BAUD_3,#0X40 ;17+6X=410 uS, ANTERIOR 40h
          LD BAUD_4,#0XA3 ;45+6X=1025 uS, ANTERIOR A3
          NOP

```

```

MAIN: LD PORTGC,#0X04 ;G2 COMO SALIDA, G3 COMO ENTRADA 0100
; SBIT 1,PORTFD ;Apaga los display
; LD PORTD,#0X00 ; " " "
; RBIT 1,PORTFD ; " " "
; SBIT 0,PORTFD ; " " "
          SBIT 3,PORTGD; SBIT 1,PORTCP Coloca un 1 en Rx
          SBIT 2,PORTGD
          JMP TALK ; Brinca a un ciclo de espera de start bit

```

```

USER: LD A,BAUD_3 ; 104 uS para 9600
        X A,DLYCNT

```

```

REDO_2: DRSZ DLYCNT
        JP REDO_2
        JP TALK
        NOP

```

```

TALK: CLR A ;Limpia el acumulador
        X A,RCVREG ;Limpio el registro de recepcion.
        IFBIT 3,PORTGP; IFBIT 1,PORTCP Checo bit de inicio
        JP USER ;Retardo 820/2 uS
        JSR DELAY4 ;retardo de 820+820/4 uS

```

```

RCVR: LD A,#0X08 ;Cargo datos de 8 bits
        X A,COUNT ;Se han cargado 8 bits

```

```

R_NEXT: RC ;Limpio el carry
          LD A,RCVREG ;cargo dato a recibir
          RRC A ;lo roto a la derecha
          X A,RCVREG ;salvo el bit recibido
          IFBIT 3,PORTGP; IFBIT 1,PORTCP checo si se mando un 1
          SBIT 7,RCVREG ;prendo bit menos significativo
          JSR DELAYY ;retardo de 820 uS
          DRSZ COUNT ;decremento numero de bits
          JP R_NEXT ;voy a recibir siguiente bit

```

```

R_OVER: LD A,RCVREG ;se termino recepcion
          X A,XMTREG ;envio el dato recibido

```

```

XMTR: LD A,#0X08 ;cargo numero de bits a transmitir
        X A,COUNT
        LD XMTREG,#0X50 ;Cargo una P para transmitir
        RBIT 2,PORTGD ;Envio bit de inicio
        JSR DELAY1 ;Retraso de 820 uS

```

```

X_NEXT: RC ;limpio el carry
          LD A,XMTREG ;cargo dato a transmitir
          RRC A ;envio bit menos significativo
          X A,XMTREG ;salvo bits restantes
          IFC ;checo si el carry es 1
          SBIT 2,PORTGD ;si es 1, prendo bit 2 del puerto C
          IFNC. ;checo si el carry es cero
          RBIT 2,PORTGD ;mando un cero por el puerto C
          JSR DELAYX ;retardo
          DRSZ COUNT ;decremento el numero de bits
          JP X_NEXT ;voy a transmitir siguiente bit
          SBIT 2,PORTGD ;mando bit de paro
          JSR DELAY1 ;retardo de xxx uS
          JMP USER ; voy a talk a esperar otro dato

```

```

DELAYY: LD A,BAUD_Y ;

```

```

                JP SAVE
DELAYX: LD     A, BAUD_X ;
                JP SAVE
DELAY4: LD     A, BAUD_4 ;
                JP SAVE
DELAY1: LD     A, BAUD_1 ;
                JP SAVE
DELAY2: LD     A, BAUD_2 ;
SAVE: X       A, DLYCNT ;
REDO_1: DRSZ  DLYCNT ;
                JP REDO_1
                RET
.END INIC
    
```

**PROGRAMA DE CONVERSION DE FORMATO PARA ARCHIVOS
ALMACENADOS EN LA TARJETA DE ALMACENAMIENTO EXTERNO**

```

.....
:* Este programa permite la conversi3n de un archivo almacenado en la TCRAM al formato de un archivo transmitido
:* serialmente por el sistema para poder procesarlo en una hoja de c3lculo
.....

```

```

Private Sub FicheroAbrir_Click()
    Dim Filtrors As String

```

```

'si ocurre un error ejecutar ManipularErrorAbrir
On Error GoTo ManipularErrorAbrir
'Generar un error cuando se pulse cancelar
CommonDialog1.CancelError = 1
'Visualizar la caja de dialogo
CommonDialog1.ShowOpen
'CommonDialog1.filetitle contiene el nombre del fichero elegido
Text1.Text = CommonDialog1.FileName
'variable a usar

```

```

Dim dato As String
Dim i As Long
Dim j As Integer
Dim k As Integer
Dim num As Double
Dim cadena As String
Dim Temp As String

```

```

Temp = ""
'obtener un numero libre de fichero
NumFichero = FreeFile
'abrir el fichero para leer
Open CommonDialog1.FileName For Binary As NumFichero

```

```

'si ocurre un error ejecutar ManipularErrorAbrir
On Error GoTo ManipularErrorAbrir
'Generar un error cuando se pulse cancelar
CommonDialog2.CancelError = 1
'Visualizar la caja de dialogo
CommonDialog2.ShowSave
'CommonDialog1.filetitle contiene el nombre del fichero elegido
Text2.Text = CommonDialog2.FileName

```

```

'obtiene el siguiente numero libre
Fichnum = FreeFile
'abrir el fichero para guardar
Open CommonDialog2.FileName For Binary As Fichnum

```

```

MousePointer = 11

```

```

For k = 1 To 26
    Temp = "BATERIA" + Str(k) + ","
    Put #Fichnum, , Temp
Next k
Temp = vbCrLf
Put #Fichnum, , Temp
j = 1
For i = 1 To LOF(NumFichero)

```

```

'se almacena en la variable dato
dato = Input(1, #NumFichero)
'Colocar el titulo
num = Asc(dato)

```

```

num = num / 100
'convierte el dato en una cadena
cadena = Str(num)
cadena = cadena + ","

If j = 26 Then
    cadena = cadena + vbCrLf
    Put #Fichnum, , cadena
    j = 0
Else
    Put #Fichnum, , cadena

End If
'guardar el dato en el archivo nuevo
'escribe en el fichero
j = j + 1
Next i
' For j = 0 To 7940
' For H = 1 To 58
' TEMP = Str(26) + ","
' Put #NumFichero, , TEMP
' Next H
TEMP = vbCrLf
'Put #NumFichero, , TEMP
Next j
'cierra el fichero

Close NumFichero
Close Fichnum
MousePointer = 0

SalirAbrir:
Exit Sub

ManipularErrorAbrir:
'Manipula el error
If Err.Number = cdICancel Then Exit Sub
MsgBox Err.Description
Resume SalirAbrir

End Sub

Private Sub FicheroGuardar_Click()
Dim Filters As String
'si ocurre un error ejecutar ManipularErrorAbrir
On Error GoTo ManipularErrorAbrir
'Generar un error cuando se pulse cancelar
CommonDialog2.CancelError = 1
'Visualizar la caja de dialogo
CommonDialog2.ShowSave
'CommonDialog1.filetitle contiene el nombre del fichero elegido
Text2.Text = CommonDialog2.FileTitle
SalirGuardar:
Exit Sub

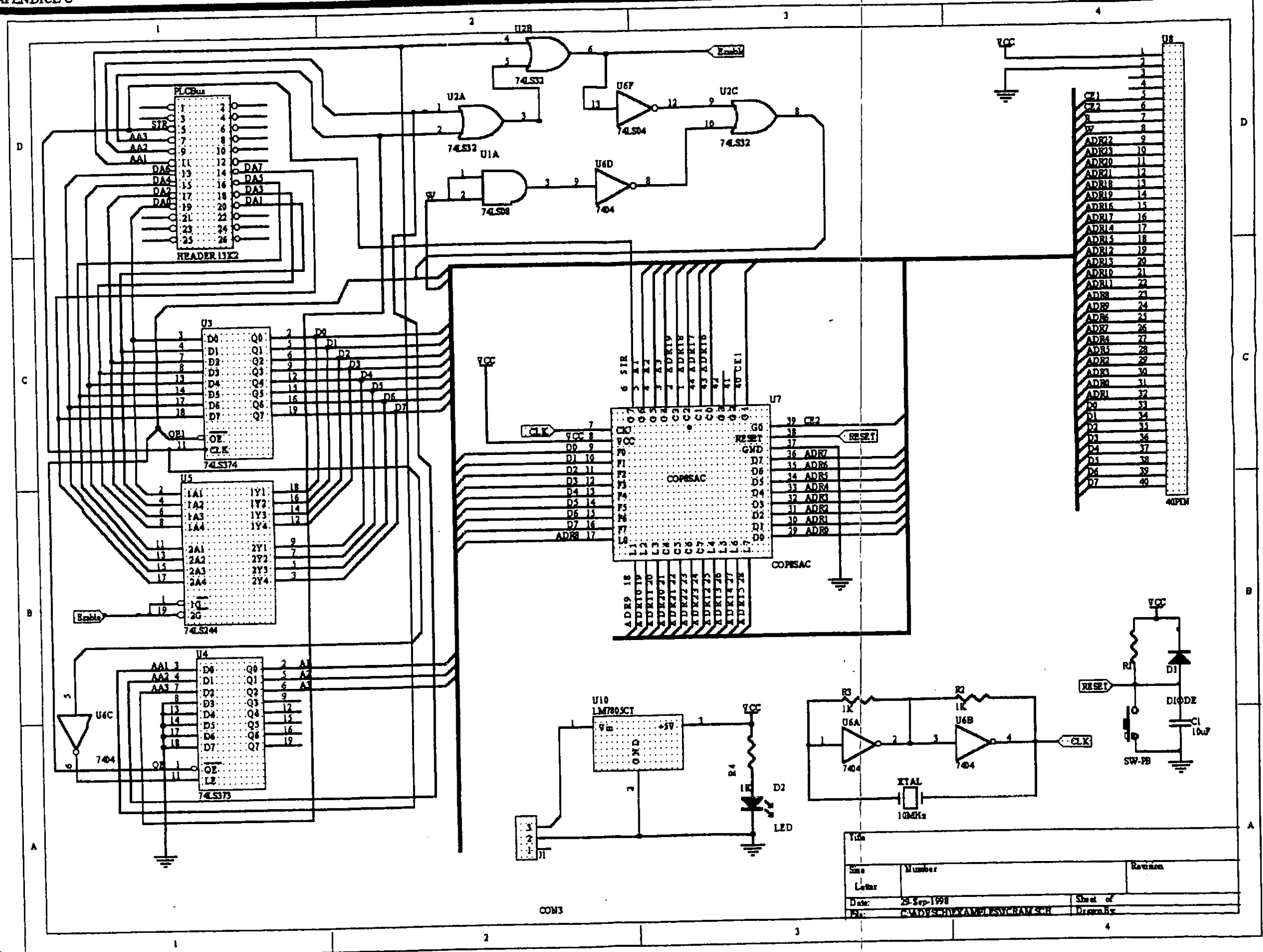
ManipularErrorAbrir:
'Manipula el error
If Err.Number = cdICancel Then Exit Sub
MsgBox Err.description
Resume SalirGuardar
End Sub

```

==== APENDICE C =====

Diagramas
electrónicos





T10m		
Size	Number	Revision
Letter		
Date:	25-Sep-1998	Sheet of
By:	CADSYS-DIVISION/ESV/RAV/SCH	Drawn By

==== APENDICE D =====

Manual
de
operación



Manual de operación

El sistema cuenta con un programa de *interface* con el usuario (Sistema 26), este programa está diseñado para programar la operación y tener acceso a la información del SADME, es un desarrollo en el lenguaje *Visual Basic*.

El programa nos permite las siguientes opciones:

- Configurar el puerto.
- Inicializar el muestreo.
- Terminar el muestreo
- Modificar fecha.
- Bajar los datos.
- Salvar los datos a un archivo.
- Reinicializar el sistema.
- Visualizar el estado del sistema.
- Salir.

El programa inicia con una ventana de presentación la cual se muestra en la figura D.1.



Figura D.1 Presentación del programa de interfaz con el usuario.

El menú del programa se presenta de una manera visual y amigable, esta ventana contiene las funciones que se pueden desarrollar, como se muestra en la figura D.2.

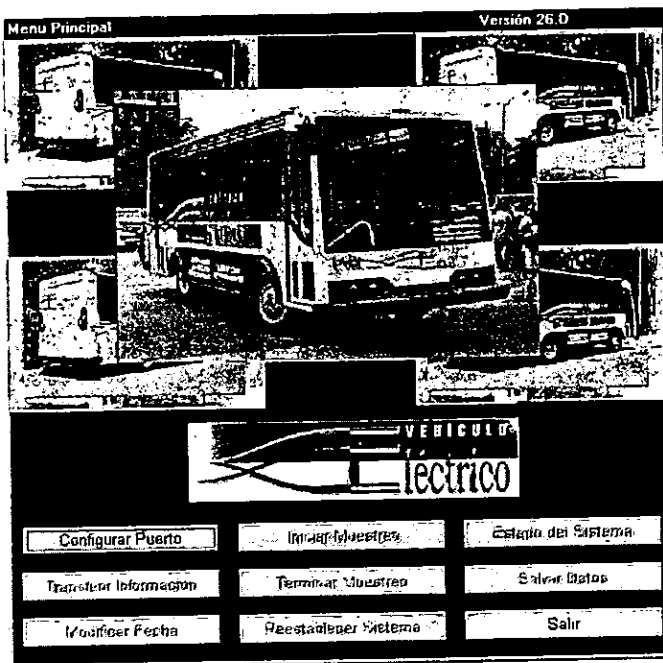


Figura D.2 Menú principal.

El primer paso a seguir para poner en operación el sistema de monitoreo es establecer la comunicación serial. Primero hay que asegurarse que exista la conexión física, la cual consiste en conectar un cable serial entre el puerto de la PC y el puerto del sistema. Ya habiendo considerado esto, se selecciona la opción «CONFIGURAR PUERTO», el cual desplegará un menú en el que podemos escoger el puerto por el cual se hará la comunicación.

Ya elegido éste se selecciona la opción «CONEXIÓN». La figura D.3 muestra la ventana de conexión al puerto.

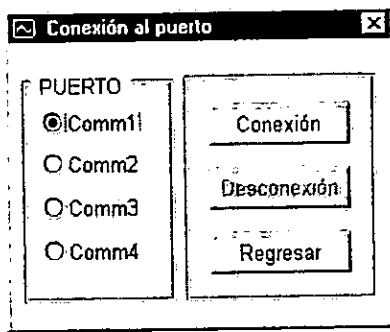


Figura D.3 Conexión al puerto.

Si la conexión se establece con éxito el programa despliega una ventana de diálogo en la cual informa de ello, figura D.4.

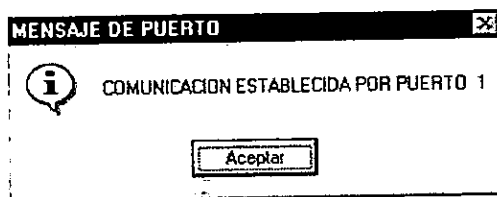


Figura D.4 Mensaje de comunicación establecida.

Si por el contrario la conexión no se pudo realizar, la ventana de diálogo que aparece notificará el error, y su posible causa. Ver figura D.5

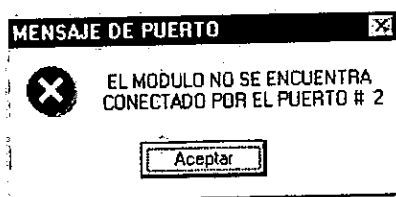


Figura D.5 Mensaje de error en la comunicación.

Una vez establecida la conexión se deberá seleccionar la opción «REGRESAR». En este momento el usuario podrá acceder a las demás opciones del programa y realizar las acciones necesarias. La figura D.6 muestra la ventana de menú principal después de hecha la conexión.

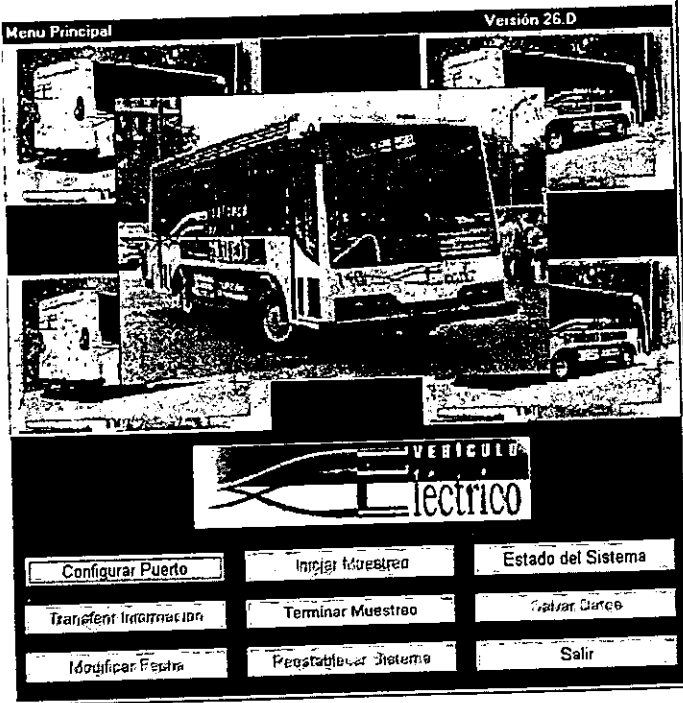


Figura D.6 Menú principal con las opciones activadas.

Las opciones que el usuario podrá realizar dependen del estado del sistema, el cual puede que esté o no operando. Si el caso es que ya está operando, puede terminar el muestreo, verificar el estado general del sistema o salir del programa. Si no está operando también puede verificar su estado, pero además puede también iniciar un muestreo, o reiniciar el sistema.

Cuando el sistema se encuentra haciendo un muestreo el usuario puede consultar su estado, esto se logra seleccionando la opción «ESTADO DEL SISTEMA», el cual nos

permitirá visualizar información como: porcentaje de memoria disponible, si el sistema está o no operando, la fecha y hora de inicio del muestreo.

La figura D.7 nos muestra la ventana que se despliega cuando el sistema está operando, y la figura D.8 la ventana que despliega cuando el sistema no está operando.

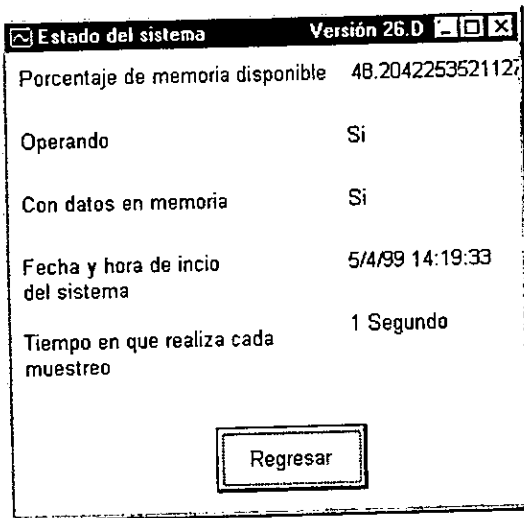


Figura D.7 Estado del sistema cuando está operando.

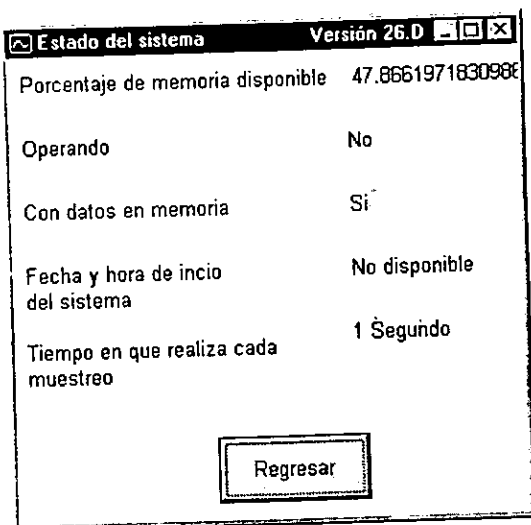


Figura D.8 Estado del sistema cuando no está operando.

Una vez terminado un muestreo, lo cual se puede lograr seleccionando la opción «TERMINAR MUESTREO» del menú principal, el sistema está en disposición de realizar la transferencia de los mismos desde la memoria del sistema hacia la memoria temporal de la PC.

Para realizar este proceso se deberá pulsar el botón «TRANSFERIR INFORMACIÓN» del menú, el cual desplegará una ventana en la cual se desplegará el avance del proceso en forma gráfica, y un botón para cancelar la operación, figura D.9.

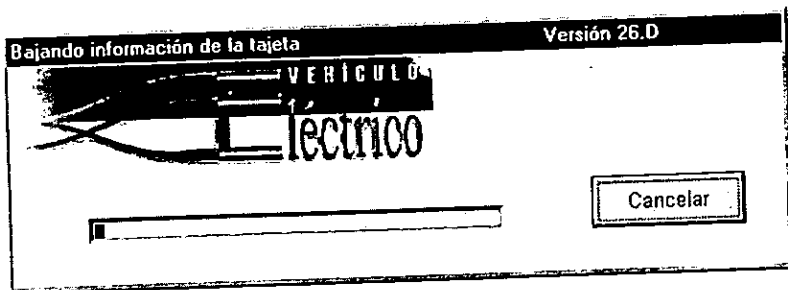


Figura D.9 Bajando información de la tarjeta.

Si se selecciona la opción «CANCELAR» el programa abortará el proceso y mandará un aviso, en el cual se despliega que se deberán bajar nuevamente los datos. Este proceso se ilustra en la figura D.10.

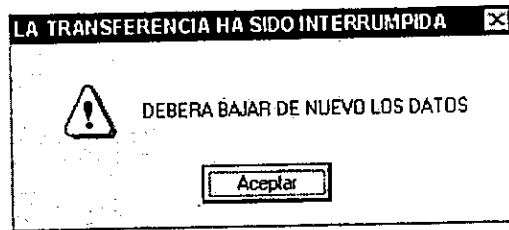


Figura D.10 La transferencia ha sido interrumpida.

En caso que la operación no se interrumpa, al termino del proceso el programa despliega una ventana informando que los datos han sido transferidos, figura D.11.

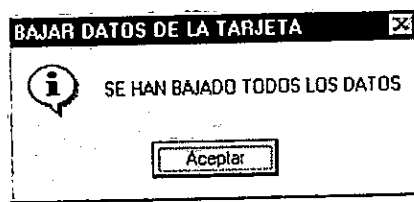


Figura D.11 Los datos han sido transferidos.

Estos datos podrán ser guardados en un archivo con formato, ya sea en el disco duro de la PC o en cualquier unidad de disco flexible con que cuente la misma. El proceso para lograr esto es el siguiente: se deberá seleccionar la opción «SALVAR DATOS», el cual mostrará una caja de diálogo en la cual se nos permitirá nombrar el archivo en el que se desee salvar la información, así como la unidad y el subdirectorio que el usuario elija, en la figura D.12 se puede ver la ventana correspondiente a este proceso.

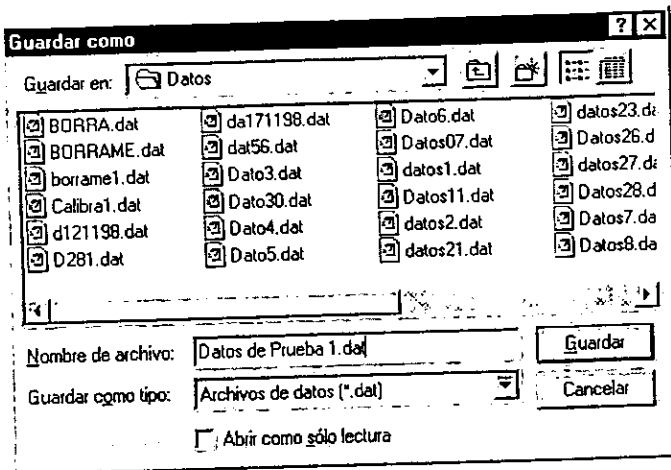


Figura D.12 Caja de diálogo *Guardar como*.

En caso que ya exista un archivo con el mismo nombre y ubicación, el programa desplegará una caja de diálogo en la que se podrá elegir entre dar otro nombre o sobre escribir el archivo especificado, figura D.13.

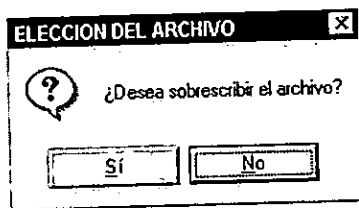


Figura D.13 Sobrescribir archivo.

En el caso que no se deseen bajar los datos en ese instante, sino realizar un nuevo muestreo, el sistema sabrá reconocer los distintos muestreos, ya que al iniciarlos envía información acerca de la hora de inicio. Al momento de ser bajados los datos y almacenados en un archivo, el programa los diferenciará por medio de una cabecera.

El programa también tiene la opción de verificar la fecha y hora del sistema, y en caso de así requerirse, se podrá modificar. Esta opción está disponible seleccionando la opción «MODIFICAR FECHA». Las figuras D.14 y D.15 nos muestra las ventanas que se despliegan

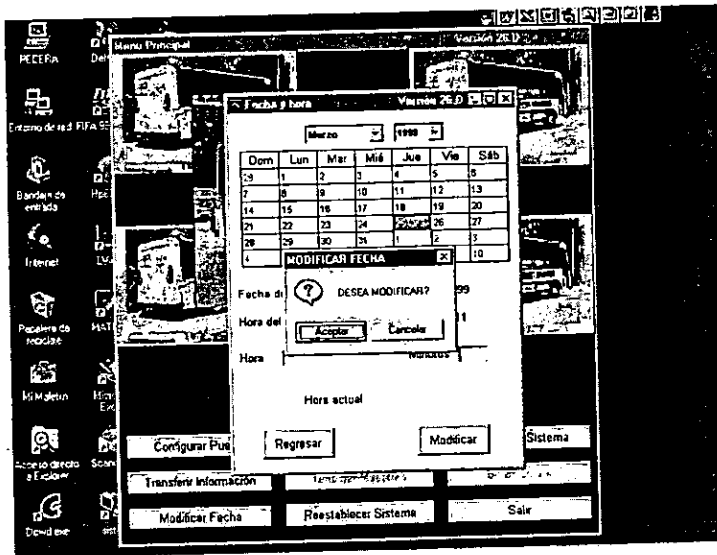


Figura D.14 Modificar fecha.

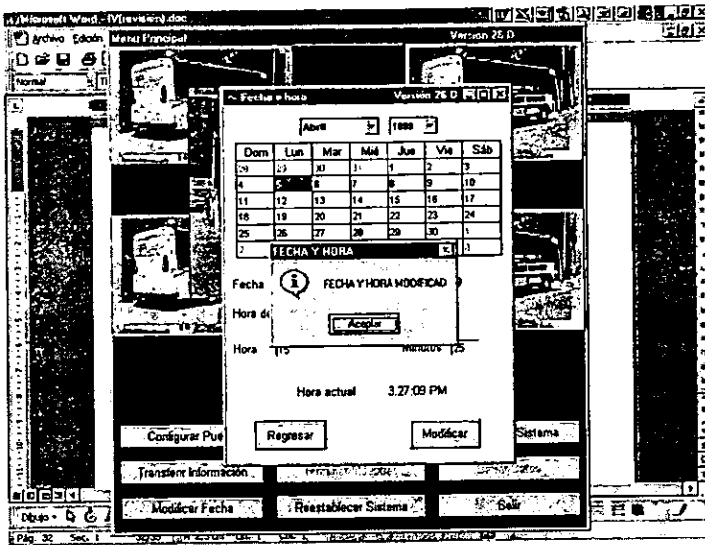


Figura D.15 Mensaje de fecha modificada.

Una vez ya realizadas las funciones necesarias para el respaldo de los datos y la puesta a punto del sistema, podemos salir del programa, para esto existe la función «SALIR», la cual al presionar el botón, finalizará la comunicación por el puerto serial, figura D.16.

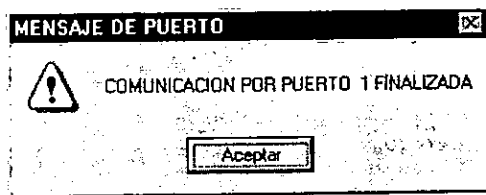


Figura D.16 Mensaje de puerto, comunicación finalizada.

Después de esto se avisa que la sesión ya terminó, figura D.17.



Figura D.17 Mensaje de puerto, fin de sesión.

Programa de conversión al formato del Sistema 26

Al utilizar la opción del almacenamiento externo, la información permanece en el formato en el cual es almacenada originalmente, esto significa que no cuenta con un formato apropiado para poder procesarse en una hoja de cálculo. Por esto se desarrolló una aplicación en *Visual Basic* la cual procesa el archivo obtenido de la *TCRAM* y devuelve un archivo que contiene el formato requerido para procesarse en dicha hoja de cálculo.

El *software* consiste en un programa que necesita el nombre del archivo de entrada y el archivo de salida. Al ejecutar la aplicación se despliega la ventana de la figura D.18, la cual nos muestra dos ventanas, una para el *archivo de origen* y otra para el *archivo destino*.

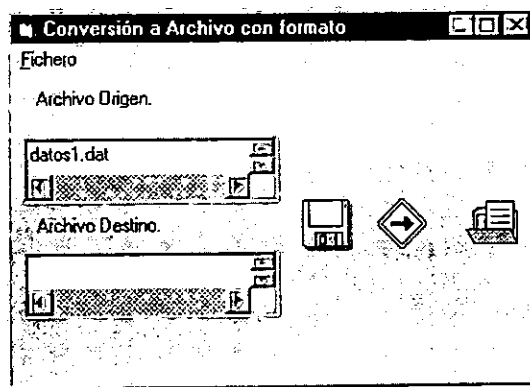


Figura D.18 Programa de aplicación para dar *formato a archivos*.

Haciendo uso del menú *fichero*, podemos elegir la opción abrir con la cual accedamos a la caja de diálogo mostrada en la figura D.19, la cual nos permite elegir el subdirectorio donde se encuentra el archivo origen.

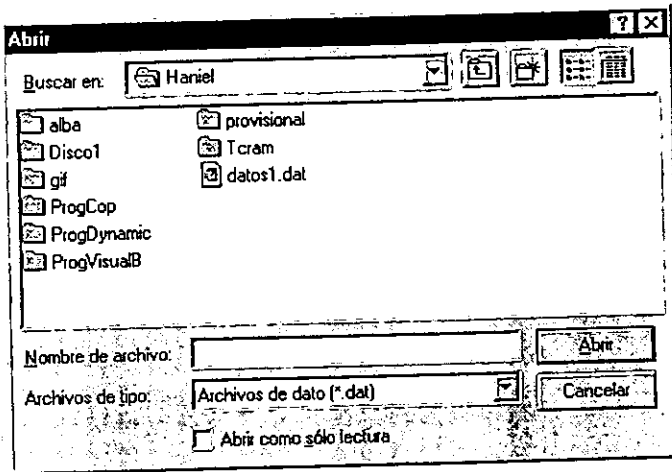


Figura D.19 Caja de diálogo Abrir.

Después de elegir el archivo origen automáticamente se abre la caja de diálogo *Guardar como*, donde escogeremos el subdirectorio y el nombre del *archivo destino*, esta ventana es mostrada en la figura D.20.

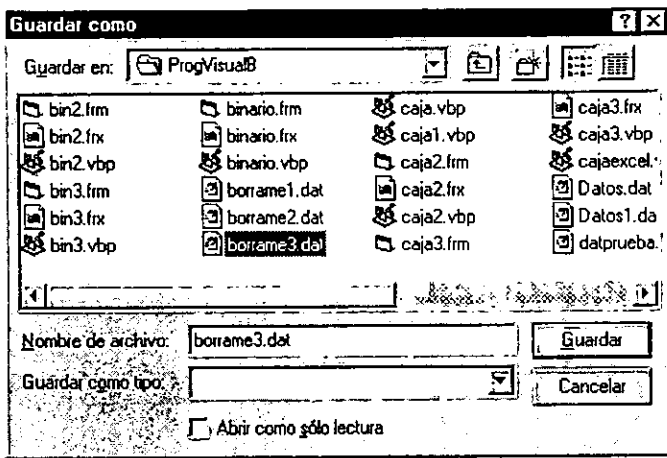


Figura D.20 Caja de diálogo *Guardar como*.

Una vez terminado el proceso aparece en las cajas de texto los nombres del archivo origen y del archivo destino (figura D.21).

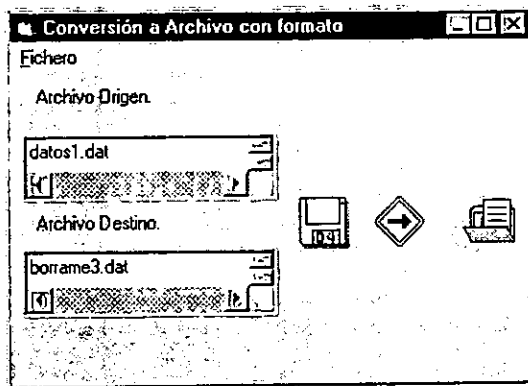
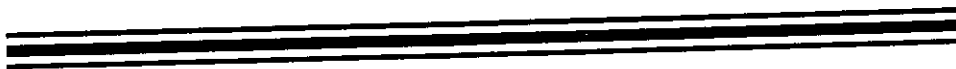


Figura D.21 Aspecto final de la ventana de la aplicación.

Ya con esto el archivo destino puede ser procesado de manera normal a como se procesa un archivo adquirido directamente por el sistema.

===== APENDICE E =====

Tablas
de
calibración



Número de Betaría

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
219	12.715	12.471	12.767	12.438	12.76	12.408	12.733	12.41	12.761	12.438	12.768	12.438	12.712	12.408	12.765	12.412	12.767	12.423	12.756	12.423	12.75	12.414	12.743	12.413	12.758	12.4
220	12.773	12.528	12.825	12.495	12.818	12.465	12.791	12.467	12.819	12.495	12.826	12.495	12.77	12.464	12.824	12.469	12.825	12.46	12.814	12.479	12.808	12.471	12.801	12.469	12.816	12.457
221	12.831	12.585	12.883	12.552	12.876	12.522	12.849	12.523	12.877	12.552	12.885	12.552	12.828	12.521	12.882	12.526	12.883	12.536	12.872	12.536	12.867	12.528	12.859	12.526	12.875	12.513
222	12.889	12.642	12.942	12.608	12.935	12.578	12.907	12.58	12.935	12.608	12.943	12.609	12.886	12.578	12.94	12.582	12.941	12.593	12.93	12.593	12.925	12.584	12.917	12.583	12.933	12.57
223	12.947	12.699	13	12.665	12.993	12.635	12.965	12.637	12.994	12.665	13.001	12.666	12.944	12.634	12.999	12.639	13	12.65	12.989	12.65	12.983	12.641	12.975	12.639	12.991	12.627
224	13.005	12.756	13.058	12.722	13.051	12.682	13.023	12.693	13.052	12.722	13.06	12.722	13.002	12.681	13.057	12.686	13.058	12.707	13.047	12.706	13.041	12.698	13.034	12.696	13.049	12.683
225	13.063	12.813	13.117	12.779	13.109	12.748	13.081	12.75	13.11	12.779	13.118	12.779	13.06	12.748	13.115	12.752	13.116	12.763	13.105	12.763	13.1	12.754	13.062	12.753	13.108	12.74
226	13.121	12.87	13.175	12.836	13.168	12.805	13.14	12.807	13.168	12.836	13.176	12.836	13.118	12.804	13.174	12.809	13.175	12.82	13.163	12.82	13.158	12.811	13.15	12.809	13.166	12.797
227	13.179	12.927	13.233	12.892	13.226	12.862	13.198	12.863	13.227	12.892	13.235	12.893	13.176	12.861	13.232	12.866	13.233	12.877	13.222	12.877	13.216	12.868	13.208	12.866	13.224	12.853
228	13.237	12.984	13.291	12.949	13.284	12.918	13.256	12.92	13.285	12.949	13.293	12.95	13.234	12.917	13.29	12.922	13.291	12.934	13.28	12.933	13.274	12.925	13.266	12.923	13.283	12.91
229	13.295	13.041	13.35	13.006	13.342	12.975	13.314	12.977	13.343	13.006	13.351	13.006	13.282	12.974	13.348	12.979	13.35	12.99	13.338	12.99	13.332	12.981	13.325	12.98	13.341	12.966
230	13.353	13.098	13.408	13.063	13.401	13.032	13.372	13.033	13.402	13.063	13.409	13.063	13.35	13.031	13.407	13.036	13.408	13.047	13.386	13.047	13.391	13.038	13.383	13.036	13.399	13.023
231	13.411	13.155	13.466	13.12	13.459	13.088	13.43	13.09	13.46	13.12	13.468	13.12	13.408	13.087	13.465	13.093	13.466	13.104	13.454	13.103	13.449	13.095	13.441	13.093	13.457	13.08
232	13.469	13.212	13.525	13.176	13.517	13.145	13.488	13.147	13.518	13.176	13.526	13.177	13.466	13.144	13.523	13.149	13.524	13.16	13.513	13.16	13.507	13.151	13.499	13.15	13.516	13.136
233	13.528	13.268	13.583	13.233	13.576	13.202	13.547	13.203	13.576	13.233	13.584	13.234	13.525	13.201	13.582	13.206	13.583	13.217	13.571	13.217	13.565	13.208	13.557	13.206	13.574	13.193
234	13.586	13.325	13.641	13.29	13.634	13.258	13.605	13.26	13.635	13.29	13.643	13.29	13.583	13.257	13.64	13.263	13.641	13.274	13.629	13.274	13.624	13.265	13.615	13.263	13.632	13.25
235	13.644	13.382	13.699	13.347	13.692	13.315	13.663	13.317	13.693	13.347	13.701	13.347	13.641	13.314	13.698	13.319	13.699	13.331	13.687	13.33	13.682	13.321	13.674	13.32	13.69	13.306
236	13.702	13.439	13.758	13.403	13.75	13.372	13.721	13.373	13.751	13.404	13.759	13.404	13.699	13.371	13.756	13.376	13.758	13.387	13.746	13.387	13.74	13.378	13.732	13.376	13.749	13.363
237	13.76	13.496	13.816	13.46	13.809	13.428	13.779	13.43	13.809	13.46	13.818	13.461	13.757	13.427	13.815	13.433	13.816	13.444	13.804	13.444	13.798	13.435	13.79	13.433	13.807	13.419
238	13.818	13.553	13.874	13.517	13.867	13.485	13.837	13.487	13.868	13.517	13.876	13.518	13.815	13.484	13.873	13.489	13.874	13.501	13.862	13.501	13.856	13.491	13.848	13.49	13.865	13.476
239	13.876	13.61	13.933	13.574	13.925	13.542	13.895	13.543	13.926	13.574	13.934	13.574	13.873	13.541	13.931	13.546	13.932	13.558	13.92	13.557	13.915	13.548	13.906	13.546	13.923	13.533
240	13.934	13.667	13.991	13.631	13.983	13.598	13.953	13.6	13.984	13.631	13.992	13.631	13.931	13.597	13.99	13.603	13.991	13.614	13.979	13.614	13.973	13.605	13.965	13.603	13.982	13.589
241	13.992	13.724	14.049	13.687	14.042	13.655	14.012	13.657	14.043	13.688	14.051	13.688	13.989	13.654	14.048	13.659	14.049	13.671	14.037	13.671	14.031	13.661	14.023	13.66	14.04	13.646
242	14.05	13.781	14.108	13.744	14.1	13.711	14.07	13.713	14.101	13.744	14.109	13.745	14.047	13.711	14.106	13.716	14.107	13.728	14.095	13.727	14.089	13.718	14.081	13.716	14.098	13.702
243	14.108	13.838	14.166	13.801	14.158	13.768	14.128	13.77	14.159	13.801	14.167	13.802	14.105	13.767	14.164	13.773	14.166	13.784	14.153	13.784	14.148	13.775	14.139	13.773	14.156	13.759
244	14.166	13.895	14.224	13.858	14.216	13.825	14.186	13.827	14.217	13.858	14.226	13.858	14.163	13.824	14.223	13.829	14.224	13.841	14.212	13.841	14.206	13.831	14.197	13.83	14.215	13.816
245	14.224	13.952	14.282	13.915	14.275	13.881	14.244	13.883	14.276	13.915	14.284	13.915	14.221	13.881	14.281	13.886	14.282	13.898	14.27	13.898	14.264	13.888	14.255	13.886	14.273	13.872
246	14.282	14.009	14.341	13.971	14.333	13.938	14.302	13.94	14.334	13.972	14.342	13.972	14.279	13.937	14.339	13.943	14.341	13.955	14.328	13.954	14.322	13.945	14.314	13.943	14.331	13.929
247	14.34	14.066	14.399	14.028	14.391	13.995	14.36	13.997	14.392	14.028	14.401	14.029	14.337	13.994	14.398	13.999	14.399	14.011	14.386	14.011	14.38	14.002	14.372	14	14.389	13.986
248	14.398	14.123	14.457	14.085	14.449	14.051	14.419	14.053	14.45	14.085	14.459	14.086	14.395	14.051	14.456	14.056	14.457	14.068	14.445	14.068	14.439	14.058	14.43	14.056	14.448	14.042
249	14.456	14.18	14.516	14.142	14.508	14.108	14.477	14.11	14.509	14.142	14.517	14.142	14.453	14.107	14.514	14.113	14.515	14.125	14.503	14.124	14.497	14.115	14.488	14.113	14.506	14.089
250	14.515	14.237	14.574	14.199	14.566	14.165	14.535	14.167	14.567	14.199	14.575	14.199	14.511	14.164	14.572	14.169	14.574	14.182	14.561	14.181	14.555	14.172	14.546	14.17	14.564	14.155

Tabla E.1 Tabla de valores esperados para los factores obtenidos en la calibración por canal (continuación).

APENDICE E

Número de Bateria

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
165	9.5796	9.3962	9.6188	9.3711	9.6136	9.3487	9.593	9.3499	9.6142	9.3712	9.6198	9.3714	9.5774	9.3482	9.6178	9.3518	9.6187	9.3598	9.6103	9.3596	9.6064	9.3533	9.6006	9.3521	9.6124	9.3426
166	9.6376	9.4531	9.6771	9.4279	9.6718	9.4054	9.6512	9.4066	9.6724	9.428	9.6781	9.4282	9.6355	9.4048	9.6761	9.4065	9.6769	9.4165	9.6686	9.4163	9.6646	9.41	9.6588	9.4087	9.6706	9.3992
167	9.6957	9.5101	9.7354	9.4947	9.7301	9.4621	9.7093	9.4633	9.7307	9.4948	9.7364	9.485	9.6935	9.4615	9.7344	9.4652	9.7352	9.4733	9.7268	9.4731	9.7228	9.4666	9.717	9.4654	9.7289	9.4559
168	9.7538	9.567	9.7937	9.5415	9.7884	9.5187	9.7674	9.5199	9.789	9.5415	9.7947	9.5418	9.7516	9.5182	9.7927	9.5218	9.7935	9.53	9.7851	9.5298	9.781	9.5233	9.7752	9.5221	9.7871	9.5125
169	9.8118	9.624	9.852	9.5983	9.8466	9.5754	9.8256	9.5766	9.8472	9.5983	9.853	9.5986	9.8096	9.5748	9.851	9.5785	9.8518	9.5867	9.8433	9.5865	9.8392	9.58	9.8334	9.5788	9.8454	9.5691
170	9.8699	9.6809	9.9103	9.6551	9.9049	9.632	9.8837	9.6333	9.9055	9.6551	9.9113	9.6554	9.8677	9.6315	9.9083	9.6352	9.9101	9.6434	9.9016	9.6432	9.8975	9.6367	9.8916	9.6365	9.9036	9.6257
171	9.9279	9.7379	9.9685	9.7118	9.9632	9.6887	9.9419	9.6899	9.9638	9.7119	9.9696	9.7122	9.9257	9.6881	9.9676	9.6919	9.9684	9.7002	9.9598	9.7	9.9557	9.6934	9.9497	9.6921	9.9619	9.6823
172	9.986	9.7948	10.027	9.7686	10.021	9.7453	10	9.7466	10.022	9.7687	10.028	9.769	9.9838	9.7448	10.026	9.7485	10.027	9.7569	10.018	9.7567	10.014	9.7501	10.008	9.7488	10.02	9.739
173	10.044	9.8517	10.085	9.8254	10.08	9.802	10.058	9.8033	10.08	9.8255	10.086	9.8258	10.042	9.8014	10.084	9.8052	10.085	9.8136	10.076	9.8134	10.072	9.8068	10.066	9.8055	10.078	9.7956
174	10.102	9.9087	10.143	9.8822	10.136	9.8587	10.116	9.8599	10.139	9.8823	10.145	9.8826	10.1	9.8581	10.142	9.8619	10.143	9.8703	10.135	9.8701	10.13	9.8634	10.124	9.8622	10.137	9.8522
175	10.16	9.9656	10.202	9.939	10.196	9.9153	10.174	9.9166	10.197	9.9381	10.203	9.9394	10.158	9.9147	10.201	9.9186	10.202	9.9271	10.193	9.9269	10.189	9.9201	10.182	9.9189	10.195	9.9088
176	10.218	10.023	10.26	9.9958	10.254	9.972	10.233	9.9733	10.255	9.9959	10.261	9.9962	10.216	9.9714	10.259	9.9753	10.26	9.9838	10.251	9.9836	10.247	9.9768	10.241	9.9755	10.253	9.9654
177	10.276	10.08	10.318	10.053	10.313	10.029	10.291	10.03	10.313	10.053	10.319	10.053	10.274	10.028	10.317	10.032	10.318	10.041	10.309	10.04	10.305	10.034	10.299	10.032	10.311	10.022
178	10.334	10.136	10.377	10.109	10.371	10.085	10.349	10.087	10.372	10.109	10.378	10.11	10.332	10.085	10.376	10.089	10.376	10.097	10.368	10.097	10.363	10.09	10.357	10.089	10.37	10.079
179	10.392	10.193	10.435	10.166	10.429	10.142	10.407	10.143	10.43	10.166	10.436	10.167	10.39	10.141	10.434	10.145	10.435	10.154	10.426	10.154	10.421	10.147	10.415	10.146	10.428	10.135
180	10.45	10.25	10.493	10.223	10.488	10.199	10.465	10.2	10.488	10.223	10.494	10.223	10.448	10.198	10.492	10.202	10.493	10.211	10.484	10.21	10.48	10.204	10.473	10.202	10.486	10.182
181	10.509	10.307	10.552	10.28	10.546	10.255	10.523	10.257	10.546	10.28	10.553	10.28	10.506	10.255	10.55	10.259	10.551	10.267	10.542	10.267	10.538	10.26	10.532	10.259	10.544	10.249
182	10.567	10.364	10.61	10.337	10.604	10.312	10.581	10.313	10.605	10.337	10.611	10.337	10.564	10.311	10.609	10.315	10.61	10.324	10.601	10.324	10.596	10.317	10.59	10.316	10.603	10.305
183	10.625	10.421	10.668	10.393	10.662	10.369	10.64	10.37	10.663	10.393	10.669	10.394	10.622	10.368	10.667	10.372	10.668	10.381	10.659	10.381	10.654	10.374	10.648	10.372	10.661	10.362
184	10.683	10.478	10.726	10.45	10.721	10.425	10.698	10.427	10.721	10.45	10.728	10.451	10.68	10.425	10.725	10.429	10.726	10.438	10.717	10.437	10.713	10.43	10.706	10.429	10.719	10.418
185	10.741	10.536	10.785	10.507	10.779	10.482	10.756	10.483	10.78	10.507	10.786	10.507	10.738	10.481	10.784	10.485	10.785	10.494	10.775	10.494	10.771	10.487	10.764	10.486	10.777	10.475
186	10.799	10.592	10.843	10.564	10.837	10.539	10.814	10.54	10.838	10.564	10.844	10.564	10.796	10.538	10.842	10.542	10.843	10.551	10.833	10.551	10.829	10.544	10.823	10.542	10.836	10.532
187	10.857	10.649	10.901	10.621	10.895	10.595	10.872	10.597	10.896	10.621	10.902	10.621	10.854	10.595	10.9	10.599	10.901	10.608	10.892	10.608	10.887	10.6	10.881	10.599	10.894	10.588
188	10.915	10.706	10.96	10.677	10.964	10.652	10.93	10.653	10.964	10.677	10.961	10.678	10.912	10.651	10.958	10.655	10.959	10.665	10.95	10.664	10.945	10.657	10.939	10.656	10.952	10.645
189	10.973	10.763	11.018	10.734	11.012	10.709	10.988	10.71	11.013	10.734	11.019	10.735	10.971	10.708	11.017	10.712	11.018	10.721	11.008	10.721	11.004	10.714	10.997	10.712	11.011	10.702
190	11.031	10.82	11.076	10.791	11.07	10.765	11.047	10.767	11.071	10.791	11.077	10.791	11.029	10.765	11.075	10.769	11.076	10.778	11.066	10.778	11.062	10.77	11.055	10.769	11.069	10.758
191	11.089	10.877	11.134	10.848	11.128	10.822	11.105	10.823	11.129	10.848	11.136	10.848	11.087	10.821	11.133	10.825	11.134	10.835	11.125	10.834	11.12	10.827	11.113	10.826	11.127	10.815
192	11.147	10.934	11.193	10.905	11.187	10.879	11.163	10.88	11.187	10.905	11.194	10.905	11.145	10.878	11.192	10.882	11.193	10.891	11.183	10.891	11.178	10.884	11.172	10.882	11.185	10.871
193	11.205	10.991	11.251	10.961	11.245	10.935	11.221	10.937	11.246	10.961	11.252	10.962	11.203	10.935	11.25	10.939	11.251	10.948	11.241	10.948	11.237	10.94	11.23	10.939	11.244	10.928
194	11.263	11.048	11.309	11.018	11.303	10.992	11.279	10.993	11.304	11.018	11.311	11.019	11.261	10.991	11.308	10.995	11.309	11.005	11.299	11.005	11.295	10.997	11.288	10.996	11.302	10.985
195	11.321	11.105	11.368	11.075	11.361	11.049	11.337	11.05	11.362	11.075	11.369	11.075	11.319	11.048	11.367	11.052	11.367	11.062	11.358	11.061	11.353	11.054	11.346	11.052	11.36	11.041
196	11.379	11.162	11.426	11.132	11.42	11.103	11.385	11.107	11.42	11.132	11.427	11.132	11.377	11.105	11.425	11.109	11.426	11.118	11.416	11.118	11.411	11.111	11.404	11.109	11.418	11.098
197	11.437	11.218	11.484	11.189	11.478	11.162	11.453	11.163	11.479	11.189	11.485	11.189	11.435	11.161	11.483	11.165	11.484	11.175	11.474	11.175	11.469	11.167	11.463	11.166	11.477	11.155
198	11.496	11.275	11.543	11.245	11.536	11.218	11.512	11.22	11.537	11.245	11.544	11.245	11.493	11.218	11.541	11.222	11.542	11.232	11.532	11.232	11.528	11.224	11.521	11.222	11.535	11.211
199	11.554	11.332	11.601	11.302	11.595	11.275	11.57	11.277	11.595	11.302	11.602	11.303	11.551	11.274	11.6	11.279	11.601	11.289	11.591	11.289	11.586	11.281	11.579	11.279	11.593	11.268
200	11.612	11.389	11.659	11.359	11.653	11.332	11.628	11.333	11.654	11.359	11.66	11.359	11.609	11.331	11.658	11.336	11.659	11.345	11.649	11.345	11.644	11.337	11.637	11.336	11.651	11.324
201	11.67	11.446	11.717	11.416	11.711	11.388	11.686	11.39	11.712	11.416	11.719	11.416	11.667	11.388	11.716	11.392	11.717	11.402	11.707	11.402	11.702	11.394	11.695	11.393	11.71	11.381
202	11.728	11.503	11.776	11.472	11.769	11.445	11.744	11.447	11.77	11.473	11.777	11.473	11.725	11.444	11.775	11.449	11.776	11.459	11.765	11.458	11.761	11.451	11.753	11.449	11.768	11.438
203	11.786	11.56	11.834	11.529	11.828	11.502	11.802	11.503	11.828	11.529	11.835	11.53	11.783	11.501	11.833	11.506	11.834	11.515	11.824	11.515	11.819	11.507	11.812	11.506	11.826	11.494
204	11.844	11.617	11.892	11.586	11.886	11.558	11.86	11.56	11.887	11.586	11.894	11.586	11.841	11.558	11.891	11.562	11.892	11.572	11.882	11.572	11.877	11.564	11.87	11.563	11.884	11.551
205	11.902	11.674	11.951	11.643	11.944	11.615	11.919	11.617	11.945	11.643	11.952	11.643	11.899													

Número de Bateria

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
111	6.4445	6.3211	6.4708	6.3042	6.4673	6.2882	6.4535	6.29	6.4677	6.3042	6.4715	6.3044	6.443	6.2888	6.4702	6.2912	6.4707	6.2966	6.4651	6.2965	6.4625	6.2922	6.4586	6.2914	6.4665	6.285
112	6.5025	6.378	6.5291	6.361	6.5256	6.3458	6.5116	6.3466	6.526	6.361	6.5298	6.3612	6.501	6.3454	6.5285	6.3479	6.529	6.3533	6.5234	6.3532	6.5207	6.3489	6.5168	6.3481	6.5247	6.3416
113	6.5606	6.435	6.5874	6.4178	6.5838	6.4025	6.5698	6.4033	6.5942	6.4178	6.5881	6.418	6.5591	6.4021	6.5868	6.4046	6.5873	6.4101	6.5816	6.4099	6.5789	6.4056	6.575	6.4047	6.583	6.3983
114	6.6186	6.4919	6.6457	6.4746	6.6421	6.4591	6.6279	6.46	6.6425	6.4746	6.6464	6.4748	6.6171	6.4587	6.645	6.4612	6.6456	6.4668	6.6399	6.4666	6.6371	6.4623	6.6332	6.4614	6.6413	6.4549
115	6.6767	6.5488	6.704	6.5314	6.7004	6.5158	6.686	6.5166	6.7008	6.5314	6.7047	6.5316	6.6752	6.5154	6.7039	6.5179	6.7039	6.5235	6.6981	6.5234	6.6953	6.5189	6.6913	6.5181	6.6995	6.5115
116	6.7347	6.6058	6.7623	6.5882	6.7586	6.5724	6.7442	6.5733	6.7591	6.5882	6.763	6.5884	6.7332	6.5721	6.7616	6.5746	6.7622	6.5802	6.7564	6.5801	6.7536	6.5756	6.7495	6.5748	6.7578	6.5681
117	6.7928	6.6627	6.8206	6.6449	6.8169	6.6291	6.8023	6.63	6.8173	6.645	6.8213	6.6452	6.7913	6.6287	6.8199	6.6313	6.8205	6.637	6.8146	6.6368	6.8118	6.6323	6.8077	6.6315	6.816	6.6248
118	6.8509	6.7197	6.8789	6.7017	6.8752	6.6858	6.8605	6.6866	6.8756	6.7018	6.8796	6.702	6.8493	6.6854	6.8782	6.688	6.8788	6.6937	6.8729	6.6935	6.87	6.689	6.8659	6.6881	6.8743	6.6814
119	6.9089	6.7766	6.9372	6.7585	6.9334	6.7424	6.9186	6.7433	6.9339	6.7586	6.9379	6.7588	6.9074	6.742	6.9365	6.7446	6.9371	6.7504	6.9311	6.7503	6.9282	6.7457	6.9241	6.7448	6.9325	6.738
120	6.967	6.8336	6.9955	6.8153	6.9917	6.7991	6.9767	6.8	6.9921	6.8154	6.9962	6.8156	6.9654	6.7987	6.9948	6.8013	6.9954	6.8071	6.9893	6.807	6.9864	6.8024	6.9823	6.8015	6.9908	6.7946
121	7.025	6.8905	7.0538	6.8721	7.05	6.8557	7.0349	6.8566	7.0504	6.8722	7.0545	6.8724	7.0235	6.8553	7.0531	6.858	7.0537	6.8639	7.0476	6.8637	7.0447	6.8591	7.0405	6.8582	7.0491	6.8512
122	7.0831	6.9475	7.1121	6.9289	7.1082	6.9124	7.093	6.9133	7.1087	6.929	7.1128	6.9292	7.0815	6.912	7.1114	6.9147	7.112	6.9206	7.1068	6.9204	7.1029	6.9157	7.0986	6.9149	7.1073	6.9079
122	7.0831	6.9475	7.1121	6.9289	7.1082	6.9124	7.093	6.9133	7.1087	6.929	7.1128	6.9292	7.0815	6.912	7.1114	6.9147	7.112	6.9206	7.1068	6.9204	7.1029	6.9157	7.0986	6.9149	7.1073	6.9079
124	7.1992	7.0614	7.2287	7.0425	7.2247	7.0257	7.2093	7.0266	7.2252	7.0426	7.2294	7.0428	7.1976	7.0253	7.2279	7.028	7.2286	7.034	7.2223	7.0339	7.2193	7.0291	7.215	7.0282	7.2238	7.0211
125	7.2573	7.1183	7.2869	7.0993	7.283	7.0824	7.2674	7.0833	7.2835	7.0994	7.2877	7.0996	7.2556	7.082	7.2862	7.0847	7.2869	7.0908	7.2806	7.0906	7.2775	7.0858	7.2732	7.0849	7.2821	7.0777
126	7.3153	7.1753	7.3452	7.1561	7.3413	7.139	7.3256	7.14	7.3417	7.1562	7.3461	7.1564	7.3137	7.1386	7.3445	7.1414	7.3452	7.1475	7.3388	7.1473	7.3358	7.1425	7.3314	7.1416	7.3403	7.1344
127	7.3734	7.2322	7.4035	7.2129	7.3995	7.1957	7.3837	7.1966	7.4	7.213	7.4044	7.2132	7.3717	7.1953	7.4028	7.1981	7.4034	7.2042	7.3971	7.2041	7.394	7.1932	7.3886	7.1983	7.3986	7.191
128	7.4314	7.2892	7.4618	7.2687	7.4678	7.2524	7.4419	7.2533	7.4583	7.2698	7.4627	7.27	7.4298	7.2519	7.4611	7.2547	7.4617	7.2609	7.4553	7.2608	7.4522	7.2559	7.4478	7.2549	7.4569	7.2476
129	7.4895	7.3461	7.5201	7.3265	7.5161	7.309	7.5	7.31	7.5165	7.3265	7.521	7.3268	7.4878	7.3086	7.5194	7.3114	7.52	7.3177	7.5135	7.3175	7.5104	7.3126	7.5059	7.3116	7.5151	7.3042
130	7.5476	7.403	7.5784	7.3833	7.5743	7.3657	7.5581	7.3666	7.5748	7.3833	7.5793	7.3835	7.5459	7.3652	7.5777	7.3681	7.5783	7.3744	7.5718	7.3742	7.5686	7.3682	7.5641	7.3683	7.5734	7.3608
131	7.6056	7.46	7.6367	7.4401	7.6326	7.4223	7.6163	7.4233	7.6331	7.4401	7.6376	7.4403	7.6039	7.4219	7.636	7.4248	7.6366	7.4311	7.63	7.431	7.6269	7.4259	7.6223	7.425	7.6316	7.4175
132	7.6637	7.5168	7.695	7.4969	7.6909	7.479	7.6744	7.4799	7.6913	7.4969	7.6959	7.4971	7.662	7.4786	7.6943	7.4814	7.6949	7.4878	7.6883	7.4877	7.6851	7.4826	7.6805	7.4817	7.6899	7.4741
133	7.7217	7.5738	7.7533	7.5537	7.7491	7.5356	7.7326	7.5366	7.7496	7.5537	7.7542	7.5539	7.72	7.5352	7.7526	7.5381	7.7532	7.5446	7.7465	7.5444	7.7433	7.5393	7.7387	7.5383	7.7481	7.5307
134	7.7798	7.6308	7.8116	7.6105	7.8074	7.5923	7.7907	7.5933	7.8079	7.6105	7.8125	7.6107	7.778	7.5919	7.8108	7.5948	7.8115	7.6013	7.8048	7.6011	7.8015	7.596	7.7969	7.595	7.8064	7.5873
135	7.8378	7.6878	7.8699	7.6672	7.8657	7.649	7.8488	7.6499	7.8661	7.6673	7.8708	7.6675	7.8361	7.6485	7.8691	7.6515	7.8698	7.668	7.863	7.6679	7.8598	7.6627	7.8551	7.6617	7.8699	7.4741
136	7.8959	7.7447	7.9282	7.724	7.9239	7.7056	7.907	7.7066	7.9244	7.7241	7.9291	7.7243	7.8941	7.7052	7.9274	7.7081	7.9281	7.7148	7.9213	7.7146	7.918	7.7094	7.9132	7.7084	7.9229	7.7006
137	7.954	7.8017	7.9865	7.7808	7.9822	7.7623	7.9651	7.7633	7.9827	7.7809	7.9874	7.7811	7.9522	7.7618	7.9857	7.7648	7.9864	7.7715	7.9795	7.7713	7.9762	7.766	7.9714	7.7651	7.9812	7.7572
138	8.012	7.8586	8.0448	7.8376	8.0404	7.8189	8.0233	7.8199	8.0409	7.8377	8.0457	7.8379	8.0102	7.8185	8.044	7.8215	8.0447	7.8282	8.0377	7.828	8.0344	7.8227	8.0296	7.8217	8.0394	7.8138
139	8.0701	7.9156	8.1031	7.8944	8.0987	7.8756	8.0814	7.8756	8.0992	7.8945	8.104	7.8947	8.0683	7.8751	8.1023	7.8782	8.103	7.8849	8.086	7.8848	8.0826	7.8794	8.0878	7.8784	8.0977	7.8704
140	8.1281	7.9725	8.1614	7.9512	8.157	7.9323	8.1395	7.9333	8.1575	7.9513	8.1623	7.9515	8.1263	7.9318	8.1606	7.9349	8.1613	7.9417	8.1542	7.9415	8.1509	7.9361	8.146	7.9351	8.1559	7.9271
141	8.1862	8.0295	8.2197	8.008	8.2152	7.9889	8.1977	7.9879	8.2157	8.0081	8.2206	8.0083	8.1844	7.9885	8.2189	7.9915	8.2196	7.9984	8.2125	7.9982	8.2091	7.9928	8.2042	7.9918	8.2142	7.9837
142	8.2443	8.0864	8.278	8.0648	8.2735	8.0456	8.2588	8.0466	8.274	8.0649	8.2789	8.0651	8.2424	8.0451	8.2772	8.0482	8.2779	8.0551	8.2707	8.0549	8.2673	8.0495	8.2624	8.0484	8.2724	8.0403
143	8.3023	8.1433	8.3363	8.1216	8.3318	8.1022	8.314	8.1033	8.3323	8.1217	8.3372	8.1219	8.3004	8.1018	8.3355	8.1049	8.3362	8.1118	8.329	8.1117	8.3255	8.1062	8.3205	8.1051	8.3307	8.0969
144	8.3604	8.2009	8.3946	8.1784	8.39	8.1589	8.3721	8.1599	8.3905	8.1785	8.3955	8.1787	8.3585	8.1584	8.3937	8.1616	8.3945	8.1686	8.3872	8.1684	8.3837	8.1629	8.3787	8.1618	8.389	8.1535
145	8.4184	8.2572	8.4529	8.2352	8.4483	8.2156	8.4302	8.2166	8.4488	8.2353	8.4538	8.2355	8.4165	8.2151	8.452	8.2182	8.4528	8.2253	8.4455	8.2251	8.442	8.2195	8.4369	8.2185	8.4472	8.2102
146	8.4765	8.3142	8.5112	8.292	8.5066	8.2722	8.4884	8.2733	8.5071	8.2921	8.5121	8.2923	8.4746	8.2717	8.5103	8.2749	8.5111	8.282	8.5037	8.2818	8.5002	8.2762	8.4951	8.2752	8.5055	8.2668
147	8.5345	8.3711	8.5695	8.3488	8.5648	8.3289	8.5465	8.3299	8.5654	8.3489	8.5704	8.3491	8.5326	8.3284	8.5686	8.3316	8.5693	8.3367	8.5619	8.3386	8.5584	8.3329	8.5533	8.3318	8.5637	8.3234
148	8.5926	8.4281	8.6277	8.4056	8.6231	8.3855	8.6047	8.3866	8.6236	8.4056	8.6287	8.4059	8.5907	8.385	8.6269	8.3883	8.6276	8.3955	8.6202	8.3953	8.6166	8.3896	8.6115	8.3885	8.622	8.38
149	8.6507	8.486	8.686	8.4624	8.6813	8.4422	8.6628	8.4433	8.6819	8.4624	8.687	8.4627	8.6487	8.4417	8.6852	8.445	8.6859	8.4522	8.6784	8.452	8.6748	8.4463	8.6697	8.4452	8.6802	8.4367
150	8.7087	8.542	8.7443	8.5192	8.7396	8.4989	8.7209	8.4999	8.7402	8.5192	8.7453	8.5195	8.7088	8.4984	8.7435	8.5016	8.7442	8.5089	8.7367	8.5087	8.7331	8.503	8.7278	8.5019	8.7385	8.4933
151	8.7668	8.5989	8.8026	8.576	8.7979	8.5555	8.7791	8.5566	8.7984	8.576	8.8036	8.5763	8.7648	8.555	8.8018											

APENDICE E

Número de Bateria

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
56	3.2513	3.189	3.2646	3.1805	3.2628	3.1729	3.2558	3.1733	3.263	3.1805	3.2649	3.1806	3.2505	3.1727	3.2642	3.1739	3.2645	3.1767	3.2617	3.1766	3.2603	3.1744	3.2584	3.174	3.2624	3.1708
57	3.3093	3.246	3.3228	3.2373	3.3211	3.2296	3.314	3.23	3.3213	3.2373	3.3232	3.2374	3.3086	3.2294	3.3225	3.2306	3.3228	3.2304	3.3199	3.2333	3.3186	3.2311	3.3166	3.2307	3.3206	3.2274
58	3.3674	3.3029	3.3811	3.2941	3.3793	3.2862	3.3721	3.2866	3.3795	3.2941	3.3815	3.2942	3.3666	3.286	3.3808	3.2873	3.3811	3.2901	3.3782	3.29	3.3768	3.2876	3.3748	3.2874	3.3789	3.2841
59	3.4254	3.3598	3.4394	3.3509	3.4376	3.3429	3.4302	3.3433	3.4378	3.3509	3.4398	3.351	3.4247	3.3427	3.4391	3.344	3.4394	3.3468	3.4364	3.3468	3.435	3.3445	3.433	3.3441	3.4371	3.3407
60	3.4836	3.4168	3.4977	3.4077	3.4958	3.3995	3.4884	3.4	3.4961	3.4077	3.4981	3.4078	3.4827	3.3993	3.4974	3.4007	3.4977	3.4036	3.4947	3.4035	3.4932	3.4012	3.4911	3.4008	3.4954	3.3973
61	3.5415	3.4737	3.556	3.4645	3.5541	3.4562	3.5465	3.4566	3.5543	3.4645	3.5564	3.4646	3.5408	3.456	3.5557	3.4573	3.556	3.4603	3.5529	3.4602	3.5514	3.4579	3.5493	3.4574	3.5537	3.4539
62	3.5996	3.5307	3.6143	3.5213	3.6124	3.5129	3.6047	3.5133	3.6126	3.5213	3.6147	3.5214	3.5988	3.5127	3.614	3.514	3.6143	3.517	3.6112	3.5169	3.6097	3.5146	3.6075	3.5141	3.6119	3.5106
63	3.6577	3.5876	3.6726	3.578	3.6706	3.5695	3.6628	3.57	3.6709	3.5781	3.673	3.5782	3.6568	3.5693	3.6723	3.5707	3.6726	3.5737	3.6694	3.5737	3.6679	3.5712	3.6657	3.5708	3.6702	3.5672
64	3.7157	3.6446	3.7309	3.6348	3.7289	3.6262	3.7209	3.6266	3.7291	3.6349	3.7313	3.635	3.7149	3.626	3.7306	3.6274	3.7309	3.6305	3.7277	3.6304	3.7261	3.6279	3.7239	3.6275	3.7284	3.6238
66	3.8318	3.7585	3.8475	3.7484	3.8454	3.7395	3.8372	3.74	3.8457	3.7485	3.8479	3.7486	3.831	3.7393	3.8471	3.7407	3.8475	3.7439	3.8441	3.7438	3.8425	3.7413	3.8403	3.7408	3.8449	3.737
66	3.8318	3.7585	3.8475	3.7484	3.8454	3.7395	3.8372	3.74	3.8457	3.7485	3.8479	3.7486	3.831	3.7393	3.8471	3.7407	3.8475	3.7439	3.8441	3.7438	3.8425	3.7413	3.8403	3.7408	3.8449	3.737
66	3.8318	3.7585	3.8475	3.7484	3.8454	3.7395	3.8372	3.74	3.8457	3.7485	3.8479	3.7486	3.831	3.7393	3.8471	3.7407	3.8475	3.7439	3.8441	3.7438	3.8425	3.7413	3.8403	3.7408	3.8449	3.737
67	3.8899	3.8154	3.9058	3.8052	3.9037	3.7962	3.8953	3.7966	3.9039	3.8053	3.9062	3.8054	3.889	3.7969	3.9054	3.7974	3.9058	3.8007	3.9024	3.8006	3.9008	3.798	3.8984	3.7975	3.9032	3.7937
68	3.948	3.8724	3.9641	3.862	3.962	3.8528	3.9535	3.8533	3.9622	3.8621	3.9645	3.8622	3.9471	3.8526	3.9637	3.8541	3.9641	3.8574	3.9606	3.8573	3.959	3.8547	3.9566	3.8542	3.9615	3.8533
69	4.006	3.9293	4.0224	3.9188	4.0202	3.9095	4.0116	3.91	4.0205	3.9188	4.0228	3.919	4.0051	3.9082	4.022	3.9108	4.0223	3.9141	4.0189	3.914	4.0172	3.9114	4.0148	3.9109	4.0197	3.9069
71	4.1221	4.0432	4.139	4.0324	4.1368	4.0228	4.1279	4.0233	4.137	4.0324	4.1394	4.0326	4.1212	4.0226	4.1386	4.0241	4.1389	4.0276	4.1364	4.0275	4.1336	4.0247	4.1312	4.0242	4.1362	4.0202
71	4.1221	4.0432	4.139	4.0324	4.1368	4.0228	4.1279	4.0233	4.137	4.0324	4.1394	4.0326	4.1212	4.0226	4.1386	4.0241	4.1389	4.0276	4.1364	4.0275	4.1336	4.0247	4.1312	4.0242	4.1362	4.0202
72	4.1802	4.1001	4.1973	4.0892	4.195	4.0794	4.186	4.08	4.1953	4.0892	4.1977	4.0893	4.1792	4.0792	4.1969	4.0808	4.1972	4.0843	4.1936	4.0842	4.1919	4.0814	4.1894	4.0809	4.1945	4.0768
73	4.2382	4.1571	4.2556	4.146	4.2533	4.1361	4.2442	4.1366	4.2535	4.146	4.256	4.1461	4.2373	4.1369	4.2552	4.1375	4.2555	4.141	4.2519	4.1409	4.2501	4.1381	4.2475	4.1376	4.2527	4.1334
75	4.3544	4.271	4.3722	4.2596	4.3698	4.2494	4.3605	4.25	4.3701	4.2596	4.3726	4.2597	4.3534	4.2492	4.3717	4.2508	4.3721	4.2545	4.3683	4.2544	4.3665	4.2515	4.3639	4.2509	4.3683	4.2466
76	4.4124	4.3279	4.4305	4.3164	4.4281	4.3061	4.4186	4.3066	4.4283	4.3164	4.431	4.3165	4.4114	4.3068	4.43	4.3075	4.4304	4.3112	4.4266	4.3111	4.4247	4.3082	4.4221	4.3076	4.4275	4.3033
76	4.4124	4.3279	4.4305	4.3164	4.4281	4.3061	4.4186	4.3066	4.4283	4.3164	4.431	4.3165	4.4114	4.3068	4.43	4.3075	4.4304	4.3112	4.4266	4.3111	4.4247	4.3082	4.4221	4.3076	4.4275	4.3033
76	4.4124	4.3279	4.4305	4.3164	4.4281	4.3061	4.4186	4.3066	4.4283	4.3164	4.431	4.3165	4.4114	4.3068	4.43	4.3075	4.4304	4.3112	4.4266	4.3111	4.4247	4.3082	4.4221	4.3076	4.4275	4.3033
78	4.5285	4.4418	4.5471	4.43	4.5446	4.4194	4.5349	4.42	4.5449	4.43	4.5476	4.4301	4.5275	4.4191	4.5466	4.4209	4.547	4.4246	4.5431	4.4245	4.5412	4.4215	4.5385	4.421	4.544	4.4165
79	4.5866	4.4988	4.6054	4.4868	4.6029	4.4761	4.593	4.4766	4.6031	4.4868	4.6069	4.4869	4.5656	4.4758	4.6049	4.4775	4.6053	4.4814	4.6013	4.4813	4.5994	4.4782	4.5967	4.4777	4.6023	4.4731
79	4.5866	4.4988	4.6054	4.4868	4.6029	4.4761	4.593	4.4766	4.6031	4.4868	4.6069	4.4869	4.5656	4.4758	4.6049	4.4775	4.6053	4.4814	4.6013	4.4813	4.5994	4.4782	4.5967	4.4777	4.6023	4.4731
80	4.6447	4.5557	4.6636	4.5436	4.6611	4.5327	4.6512	4.5332	4.6614	4.5436	4.6642	4.5437	4.6436	4.5325	4.6632	4.5342	4.6636	4.5381	4.6596	4.538	4.6576	4.5349	4.6548	4.5343	4.6605	4.5297
81	4.7027	4.6127	4.7219	4.6003	4.7194	4.5994	4.7083	4.59	4.7197	4.6004	4.7225	4.6005	4.7017	4.5891	4.7215	4.5909	4.7219	4.5948	4.7178	4.5947	4.7159	4.5916	4.713	4.591	4.7188	4.584
82	4.7608	4.6696	4.7802	4.6571	4.7777	4.646	4.7674	4.6466	4.778	4.6572	4.7808	4.6573	4.7597	4.6458	4.7798	4.6476	4.7802	4.6515	4.7761	4.6514	4.7741	4.6483	4.7712	4.6477	4.777	4.643
83	4.8188	4.7266	4.8385	4.7139	4.8359	4.7027	4.8256	4.7033	4.8362	4.714	4.8391	4.7141	4.8177	4.7024	4.8381	4.7042	4.8385	4.7083	4.8343	4.7082	4.8323	4.705	4.8294	4.7044	4.8353	4.6996
85	4.9349	4.8405	4.9551	4.8275	4.9524	4.816	4.9419	4.8166	4.9528	4.8276	4.9557	4.8277	4.9338	4.8157	4.9546	4.8176	4.9551	4.8217	4.9508	4.8216	4.9487	4.8183	4.9458	4.8177	4.9518	4.8129
86	4.993	4.8974	5.0134	4.8843	5.0107	4.8727	5	4.8733	5.011	4.8844	5.014	4.8845	4.9919	4.8724	5.0129	4.8743	5.0134	4.8784	5.009	4.8783	5.007	4.875	5.004	4.8744	5.0101	4.8695
87	5.0511	4.9543	5.0717	4.9411	5.069	4.9293	5.0581	4.93	5.0693	4.9412	5.0723	4.9413	5.0499	4.929	5.0712	4.9309	5.0717	4.9352	5.0673	4.9351	5.0652	4.9317	5.0621	4.9311	5.0683	4.9261
88	5.1091	5.0113	5.13	4.9979	5.1272	4.986	5.1163	4.9866	5.1276	4.998	5.1306	4.9981	5.108	4.9867	5.1295	4.9876	5.1299	4.9919	5.1255	4.9918	5.1234	4.9884	5.1203	4.9878	5.1266	4.9827
89	5.1672	5.0682	5.1883	5.0547	5.1855	5.0427	5.1744	5.0433	5.1858	5.0547	5.1889	5.0549	5.166	5.0424	5.1878	5.0443	5.1882	5.0486	5.1838	5.0485	5.1816	5.0451	5.1785	5.0444	5.1848	5.0393
90	5.2252	5.1252	5.2466	5.1115	5.2438	5.0993	5.2326	5.1	5.2441	5.1115	5.2472	5.1117	5.2241	5.099	5.2461	5.101	5.2465	5.1054	5.242	5.1052	5.2398	5.1018	5.2367	5.1011	5.2431	5.096
91	5.2833	5.1821	5.3049	5.1683	5.302	5.156	5.2907	5.1563	5.3024	5.1683	5.3055	5.1685	5.2821	5.1557	5.3044	5.1577	5.3048	5.1621	5.3003	5.162	5.2981	5.1585	5.2949	5.1578	5.3014	5.1526
92	5.3413	5.2391	5.3632	5.2251	5.3603	5.2126	5.3488	5.2133	5.3606	5.2251	5.3638	5.2253	5.3401	5.2123	5.3627	5.2143	5.3631	5.2188	5.3585	5.2187	5.3563	5.2152	5.3531	5.2145	5.3596	5.2092
93	5.3994	5.296	5.4215	5.2819	5.4186	5.2683	5.407	5.27	5.4189	5.2819	5.4221	5.2821	5.3982	5.269	5.421	5.271	5.4214	5.2755	5.4167	5.2754	5.4145	5.2718	5.4113	5.2712	5.4179	5.2658
94	5.4575	5.353	5.4798	5.3387	5.4768	5.3259	5.4651	5.3266	5.4772	5.3387	5.4804	5.3389	5.4562	5.3256	5.4792	5.3277	5.4797	5.3323	5.475	5.3321	5.4727	5.3285	5.4694	5.3278	5.4761	5.3225
95	5.5155	5.4099	5.5381	5.3955	5.5351	5.3826	5.5233	5.3833	5.5354	5.3955	5.5387	5.3957	5.5143	5.3823	5.5375	5.3844	5.538	5.389	5.5332	5.388						

Número de Batería	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
3	0.1742	0.1708	0.1749	0.1704	0.1748	0.17	0.1744	0.17	0.1748	0.1704	0.1749	0.1704	0.1741	0.17	0.1749	0.17	0.1749	0.1702	0.1747	0.1702	0.1747	0.1701	0.1746	0.17	0.1748	0.1699
4	0.2322	0.2278	0.2332	0.2272	0.2331	0.2266	0.2306	0.2267	0.2331	0.2272	0.2332	0.2272	0.2322	0.2266	0.2332	0.2267	0.2332	0.2269	0.233	0.2269	0.2329	0.2267	0.2327	0.2267	0.233	0.2265
5	0.2903	0.2847	0.2915	0.284	0.2913	0.2833	0.2907	0.2833	0.2913	0.284	0.2915	0.284	0.2902	0.2833	0.2914	0.2834	0.2915	0.2836	0.2912	0.2836	0.2911	0.2834	0.2909	0.2834	0.2913	0.2831
6	0.3483	0.3417	0.3498	0.3408	0.3496	0.34	0.3488	0.34	0.3496	0.3408	0.3498	0.3408	0.3483	0.3399	0.3497	0.3401	0.3498	0.3404	0.3495	0.3403	0.3493	0.3401	0.3491	0.3401	0.3495	0.3397
7	0.4064	0.3986	0.4081	0.3976	0.4078	0.3966	0.407	0.3967	0.4079	0.3976	0.4081	0.3976	0.4063	0.3966	0.408	0.3967	0.4081	0.3971	0.4077	0.3971	0.4075	0.3969	0.4073	0.3968	0.4078	0.3964
8	0.4645	0.4556	0.4664	0.4544	0.4661	0.4533	0.4651	0.4533	0.4661	0.4544	0.4664	0.4544	0.4644	0.4532	0.4663	0.4534	0.4664	0.4538	0.466	0.4538	0.4658	0.4535	0.4655	0.4534	0.4661	0.453
9	0.5225	0.5125	0.5247	0.5111	0.5244	0.5099	0.5233	0.51	0.5244	0.5112	0.5247	0.5112	0.5224	0.5099	0.5246	0.5101	0.5247	0.5105	0.5242	0.5105	0.524	0.5102	0.5237	0.5101	0.5243	0.5096
10	0.5806	0.5695	0.583	0.5679	0.5826	0.5666	0.5814	0.5667	0.5827	0.5679	0.583	0.568	0.5805	0.5666	0.5829	0.5668	0.5829	0.5673	0.5824	0.5672	0.5822	0.5669	0.5819	0.5668	0.5826	0.5662
11	0.6386	0.6264	0.6413	0.6247	0.6409	0.6232	0.6385	0.6233	0.6409	0.6247	0.6413	0.6248	0.6385	0.6232	0.6412	0.6235	0.6412	0.624	0.6407	0.624	0.6404	0.6236	0.64	0.6235	0.6408	0.6228
12	0.6967	0.6834	0.6985	0.6815	0.6982	0.6799	0.6977	0.68	0.6982	0.6815	0.6996	0.6816	0.6965	0.6799	0.6995	0.6801	0.6995	0.6807	0.6989	0.6807	0.6986	0.6802	0.6982	0.6802	0.6991	0.6795
13	0.7548	0.7403	0.7578	0.7383	0.7574	0.7366	0.7558	0.7367	0.7575	0.7383	0.7579	0.7384	0.7546	0.7365	0.7578	0.7368	0.7578	0.7374	0.7572	0.7374	0.7569	0.7369	0.7564	0.7368	0.7573	0.7361
14	0.8128	0.7973	0.8161	0.7961	0.8157	0.7932	0.814	0.7933	0.8157	0.7961	0.8162	0.7962	0.8126	0.7932	0.8161	0.7935	0.8161	0.7942	0.8154	0.7941	0.8151	0.7936	0.8146	0.7935	0.8156	0.7927
15	0.8709	0.8542	0.8744	0.8519	0.874	0.8499	0.8721	0.85	0.874	0.8519	0.8745	0.8519	0.8707	0.8498	0.8743	0.8502	0.8744	0.8509	0.8737	0.8509	0.8733	0.8503	0.8728	0.8502	0.8739	0.8493
16	0.9289	0.9111	0.9327	0.9087	0.9322	0.9065	0.9302	0.9067	0.9323	0.9087	0.9328	0.9087	0.9287	0.9065	0.9326	0.9068	0.9327	0.9076	0.9319	0.9076	0.9315	0.907	0.931	0.9069	0.9321	0.9059
17	0.987	0.9681	0.991	0.9665	0.9905	0.9632	0.9884	0.9633	0.9906	0.9655	0.9911	0.9655	0.9868	0.9631	0.9909	0.9635	0.991	0.9643	0.9902	0.9643	0.9897	0.9637	0.9892	0.9635	0.9904	0.9626
18	1.045	1.025	1.0493	1.0223	1.0488	1.0199	1.0465	1.02	1.0488	1.0223	1.0494	1.0223	1.0448	1.0198	1.0492	1.0202	1.0493	1.0211	1.0484	1.021	1.048	1.0204	1.0473	1.0202	1.0486	1.0192
19	1.1031	1.082	1.1076	1.0791	1.107	1.0765	1.1047	1.0767	1.1071	1.0791	1.1077	1.0791	1.1029	1.0765	1.1075	1.0769	1.1076	1.0778	1.1066	1.0778	1.1062	1.077	1.1065	1.0769	1.1069	1.0758
20	1.1612	1.1389	1.1659	1.1359	1.1653	1.1332	1.1628	1.1333	1.1654	1.1359	1.166	1.1359	1.1609	1.1331	1.1658	1.1336	1.1659	1.1345	1.1649	1.1345	1.1644	1.1337	1.1637	1.1336	1.1651	1.1324
21	1.2192	1.1959	1.2242	1.1927	1.2235	1.1898	1.2209	1.19	1.2236	1.1927	1.2243	1.1927	1.2189	1.1898	1.2241	1.1902	1.2242	1.1912	1.2231	1.1912	1.2226	1.1904	1.2219	1.1903	1.2234	1.1891
22	1.2773	1.2528	1.2825	1.2495	1.2818	1.2465	1.2791	1.2467	1.2819	1.2495	1.2826	1.2495	1.277	1.2464	1.2824	1.2469	1.2825	1.248	1.2814	1.2479	1.2808	1.2471	1.2801	1.2469	1.2816	1.2457
23	1.3353	1.3098	1.3408	1.3063	1.3401	1.3032	1.3372	1.3033	1.3402	1.3063	1.3409	1.3063	1.335	1.3031	1.3407	1.3036	1.3408	1.3047	1.3386	1.3047	1.3391	1.3038	1.3383	1.3036	1.3399	1.3023
24	1.3934	1.3667	1.3991	1.3631	1.3983	1.3598	1.3953	1.36	1.3984	1.3631	1.3992	1.3631	1.3931	1.3597	1.399	1.3603	1.3991	1.3614	1.3979	1.3614	1.3973	1.3605	1.3965	1.3603	1.3982	1.3589
25	1.4515	1.4237	1.4574	1.4199	1.4566	1.4165	1.4535	1.4167	1.4567	1.4199	1.4575	1.4199	1.4511	1.4164	1.4572	1.4169	1.4574	1.4182	1.4561	1.4181	1.4555	1.4172	1.4546	1.417	1.4564	1.4155
26	1.5095	1.4806	1.5157	1.4767	1.5149	1.4731	1.5116	1.4733	1.515	1.4767	1.5159	1.4767	1.5092	1.473	1.5155	1.4736	1.5157	1.4749	1.5144	1.4748	1.5137	1.4738	1.5128	1.4737	1.5147	1.4722
27	1.5676	1.5376	1.574	1.5334	1.5731	1.5298	1.5688	1.53	1.5732	1.5335	1.5742	1.5335	1.5672	1.5297	1.5738	1.5303	1.574	1.5316	1.5726	1.5316	1.572	1.5305	1.571	1.5303	1.5729	1.5288
28	1.6256	1.5945	1.6323	1.5902	1.6314	1.5865	1.6279	1.5867	1.6315	1.5903	1.6325	1.5903	1.6253	1.5864	1.6321	1.587	1.6323	1.5883	1.6308	1.5883	1.6302	1.5872	1.6292	1.587	1.6312	1.5864
29	1.6837	1.6514	1.6906	1.647	1.6897	1.6431	1.686	1.6433	1.6908	1.6471	1.6918	1.6471	1.6833	1.643	1.6904	1.6436	1.6906	1.6451	1.6891	1.645	1.6884	1.6439	1.6874	1.6437	1.6884	1.642
30	1.7417	1.7084	1.7489	1.7038	1.7479	1.6998	1.7442	1.7	1.748	1.7038	1.7491	1.7039	1.7414	1.6997	1.7487	1.7003	1.7488	1.7018	1.7473	1.7017	1.7466	1.7006	1.7456	1.7004	1.7477	1.6987
31	1.7998	1.7653	1.8072	1.7606	1.8062	1.7564	1.8023	1.7567	1.8063	1.7606	1.8074	1.7607	1.7994	1.7563	1.807	1.757	1.8071	1.7585	1.8066	1.7585	1.8048	1.7573	1.8038	1.7571	1.806	1.7553
32	1.8579	1.8223	1.8655	1.8174	1.8645	1.8131	1.8605	1.8133	1.8646	1.8174	1.8657	1.8175	1.8574	1.813	1.8653	1.8137	1.8654	1.8152	1.8638	1.8152	1.8631	1.814	1.8619	1.8137	1.8642	1.8119
33	1.9159	1.8792	1.9238	1.8742	1.9227	1.8697	1.9186	1.87	1.9228	1.8742	1.924	1.8743	1.9155	1.8696	1.9236	1.8704	1.9237	1.872	1.9221	1.8719	1.9213	1.8707	1.9201	1.8704	1.9225	1.8685
34	1.974	1.9362	1.9821	1.931	1.981	1.9264	1.9767	1.9267	1.9811	1.931	1.9823	1.9311	1.9735	1.9263	1.9819	1.927	1.982	1.9287	1.9803	1.9286	1.9795	1.9273	1.9783	1.9271	1.9807	1.9251
35	2.032	1.9931	2.0403	1.9878	2.0392	1.9831	2.0349	1.9833	2.0394	1.9878	2.0406	1.9879	2.0316	1.9829	2.0401	1.9837	2.0403	1.9854	2.0386	1.9854	2.0377	1.984	2.0365	1.9838	2.039	1.9818
36	2.0901	2.0501	2.0986	2.0446	2.0975	2.0397	2.093	2.04	2.0976	2.0446	2.0989	2.0447	2.0896	2.0396	2.0984	2.0404	2.0986	2.0421	2.0968	2.0421	2.0959	2.0407	2.0947	2.0405	2.0972	2.0384
37	2.1482	2.107	2.1569	2.1014	2.1558	2.0964	2.1512	2.0967	2.1559	2.1014	2.1572	2.1015	2.1477	2.0963	2.1567	2.0971	2.1569	2.0989	2.155	2.0988	2.1542	2.0974	2.1529	2.0971	2.1555	2.095
38	2.2062	2.164	2.2152	2.1582	2.214	2.153	2.2093	2.1533	2.2142	2.1582	2.2155	2.1583	2.2057	2.1529	2.215	2.1537	2.2152	2.1556	2.2133	2.1555	2.2124	2.1541	2.2111	2.1538	2.2138	2.1516
39	2.2643	2.2209	2.2735	2.215	2.2723	2.2097	2.2674	2.21	2.2724	2.215	2.2738	2.2151	2.2638	2.2096	2.2733	2.2104	2.2735	2.2123	2.2715	2.2123	2.2706	2.2108	2.2682	2.2105	2.272	2.2083
40	2.3223	2.2779	2.3318	2.2718	2.3306	2.2664	2.3256	2.2667	2.3307	2.2718	2.3321	2.2719	2.3218	2.2662	2.3316	2.2671	2.3318	2.269	2.3298	2.269	2.3288	2.2675	2.3274	2.2672	2.3303	2.2649
41	2.3804	2.3348	2.3901	2.3286	2.3888	2.323	2.3837	2.3233	2.389	2.3286	2.3904	2.3287	2.3798	2.3229	2.3899	2.3238	2.3901	2.3258	2.388	2.3257	2.387	2.3241	2.3856	2.3238	2.3885	2.3215
42	2.4384	2.3918	2.4484	2.3854	2.4471	2.3797	2.4419	2.38	2.4472	2.3854	2.4487	2.3855	2.4379	2.3795	2.4482	2.3805	2.4484	2.3825	2.4463	2.3824	2.4453	2.3808	2.4438	2.3805	2.4468	2.3781
43	2.4965	2.4487	2.5067	2.4422	2.5054	2.4363	2.5	2.4367	2.5055	2.4422	2.507	2.4423	2.4959	2.4362	2.5065	2.4371	2.5067	2.4392	2.5045	2.4392	2.5035	2.4375	2.502	2.4372	2.	