

9
21.

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO DE UNA COMPUTADORA DE
VIAJE PARA CAMIONES, AUTOBUSES Y
VEHICULOS DE MENSAJERIA

T E S I S

T E S I S
QUE PARA OBTENER EL TITULO DE :
INGENIERO EN COMPUTACION
P R E S E N T A N :
ARCINIEGA ESPINOSA ELIZABETH
FONSECA CELIS FRANCISCO EDGARDO

DIR. DE TESIS : M. I. JUAN CARLOS ROA BEIZA

CIUDAD UNIVERSITARIA

1997



TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecemos a la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por la formación adquirida, a nuestros profesores por su enseñanza y dedicación, en especial a nuestro director de tesis Juan Carlos Roa Beiza por su dirección, apoyo y amistad.

A Dios, por todas las bendiciones dadas y por permitirme cumplir esta meta.

A mi Madre, por sus consejos y acertados regaños, que me daban la pauta cuando dudaba, y volvía a retomar la meta trazada. Por ese carácter de superación tesonera que me ha demostrado por siempre y ha sido un ejemplo en todo momento para no claudicar.

A mi Padre, por su gran paciencia y apoyo, por su buen humor ante la vida. Por esos consejos tan pequeños en palabras pero tan ricos en contenido, y por enseñarme a través de la vida con hechos y no sólo con frases. Además por todas las desveladas que le hice y hago pasar.

A mi Tía y Madrina Hermelinda, por ser una mujer ejemplar y con unas agallas que admiro por todos los retos que le pone la vida. Por ser un espíritu que no admite derrotas y porque tiene un corazón de oro y alma de roble.

A mi hermano por soportarme y a mis amigas por sus palabras de aliento.

A mi Doctor consentido Sergio Saavedra Amaro al que considero mi amigo. Le doy las gracias por todos sus regaños y por mostrarme otra visión de la vida, pero sobre todo a su sencillez y gran corazón que sabe esconder muy bien.

A Francisco, por su gran entrega en todos los proyectos que hemos hecho juntos y a su paciencia, pero sobre todo a su apoyo incondicional en los buenos y malos momentos de mi vida.

Elizabeth

A mi familia por su amor y sacrificio.

A la Dirección General de Incorporación y Revalidación de Estudios, por su apoyo, especialmente para el Dr. Luis I. Solórzano Flores y para la Mat. Olivia Suárez.

Al Ing. Miguel Castillo por su apoyo.

A mis compañeros de Tesis : Juan, Ricardo, Elizabeth y Francisco.

A la Escuela Superior de Sistemas, su personal y sus alumnos

Al Instituto Mundo, su personal y sus alumnos

A mis compañeros de trabajo

A los amigos

María Elena

A mi Madre, por sus sacrificios al apoyarme con mi carrera, por su paciencia, y por todos aquellos buenos y malos momentos que pasamos.

A mi Tía, por escucharme y comprender mis miedos y anhelos.

A mis Hermanos, por aguantar mis arranques.

A mis amigos y amigas por sus consejos y deseos para que concluyera este proyecto.

A Eli a quien gracias a su contagiosa constancia y ganas de llevar a cabo los proyectos, convierten los sueños en realidad.

Francisco E.

A mis Padres quienes me han apoyado en todo momento para verme formado en un hombre de bien.

A mis Hermanos a quienes soporto con gusto y alegría con quienes he pasado muy buenos momentos momentos.

A mi Abuela Lupe quien nos consintió cuando éramos chicos pero justo ahora cuando más lo necesita debemos de apoyarla con todo nuestro cariño.

A mi Abuela Gela (+) quien desgraciadamente no pudo ver terminada mi carrera, pero se que le hubiera dado mucha alegría.

A todos mis Tíos y Primos a quienes estimo con agrado. por la amistad que tenemos.

A todos mis Amigos que a lo largo de mi vida he tenido la suerte de conocerlos y día con día nuestra amistad se ha ido fortaleciendo.

"gracias"

Ricardo

Deseo mostrar mi agradecimiento a la Mat. Olivia Suárez y al Ing. Miguel Castillo por todo el apoyo que me han brindado y por contar con su amistad.

Gracias a mi familia, mis Abuelas, mis Tias, mis Hermanos y mis Primos, por creer en mí lo suficiente para ayudarme a crecer durante todos estos años.

Agradezco también a mis amigos por sus palabras de aliento y motivación.

Un agradecimiento especial a Mercedes, por su paciencia y comprensión. Soy muy afortunado por tenerla en mi vida.

J. Manuel

Índice

Introducción

CAPITULO I

ANTECEDENTES Y CONCEPTOS BASICOS

1.1 Perfil de las vías terrestres en México.....	5
1.2. Administración y Operación de Vehiculos en Empresas Públicas y Privadas.....	17
1.3 Normas y Reglamentos del Transporte en México.....	21
1.4 Empleo de los Microprocesadores en la Industria Automotriz.....	33
1.5 Desarrollo e Implementación de las Computadoras de Viaje y las Cajas Negras.....	43
1.5.1 Desarrollo de las Computadoras de Viaje.....	43
1.5.2. Implementación de la Computadora de Viaje.....	45
1.5.3 Cajas Negras (Registro de Datos de Vuelo).....	49
1.6 Teoría de las bases de datos relacionales.....	52
1.6.1 Tipos de Enfoques de las Bases de Datos.....	52
1.6.2 Operaciones de Almacenamiento y Recuperación de Información.....	52
1.6.3 Características de los Sistemas Manejadores de Bases de Datos Relacionales.....	65
1.6.4 Ventajas de los Sistemas Manejadores de Bases de Datos Relacionales.....	65
1.6.5 Poder de los Sistemas Manejadores de Bases de Datos Relacionales.....	65
1.6.6 Diccionario de Datos Dinámico basado en el Modelo Relacional.....	65
1.6.7 Modelado Entidad-Relación.....	66
1.6.8 Escenario de diseño de un diagrama Entidad-Relación.....	74
1.6.9 Normalización.....	74
1.6.10 Uso Múltiple del Mismo Atributo.....	78
1.6.11. Múltiples Ocurrencias de un mismo Hecho.....	81

CAPITULO II

SENSORES

2.1 Conceptos Básicos.....	84
2.1.1 Variables Físicas.....	84
2.1.2 Entradas y Salidas Analógicas.....	85
2.1.3 Transductores.....	89
2.1.4 Actuadores.....	91
2.1.5 Obtención y Proceso de Datos.....	93
2.2 Areas de Aplicación.....	97
2.3 Tipos de sensores a emplear en el sistema.....	100
2.3.1 Nivel de combustible.....	100
2.3.2 Temperatura del motor.....	101
2.3.3 Velocidad del vehículo.....	104
2.4 Elección de los sensores y lugares óptimos de trabajo.....	106

CAPITULO III

EL MICROCONTROLADOR MC68HC11

3.1. Características del Microcontrolador MC68HC11.....	112
3.1.1 La Unidad Central de Procesamiento (CPU) y Registros.....	114
3.1.2 Las Memorias Internas.....	118
3.1.3 Puertos de Entradas/Salidas en Paralelo.....	122
3.1.4 Interface Periférica Serial Sincrona (SPI).....	125
3.1.5 Interface Periférica Serial Asíncrona (SCI).....	128
3.1.6 Convertidor Analógico Digital.....	132
3.1.7 Reloj Principal e Interrupción de Tiempo Real.....	147
3.1.8 Función de captura de entrada.....	162
3.1.9 Funciones de Comparación de Salida.....	165
3.1.10 El Acumulador de Pulsos.....	168
3.2 Configuración y Modos de Operación.....	176

3.2.1 Configuración del Sistema.....	176
3.2.2 Los Modos de Operación.....	177
3.3 Modelo de Programación e Instrucciones.....	181
3.3.1 Modos de Direccionamiento.....	182
3.3.2 El Conjunto de Instrucciones.....	186

CAPITULO IV

SISTEMA DE COMUNICACIONES, REGISTRO Y MANEJO DE DATOS

4.1 Canales de comunicación para microcomputadoras.....	197
4.1.1 Conceptos Básicos.....	197
4.1.2 Tipos de Transmisión.....	200
4.1.3 Estándares de Comunicación.....	205
4.2 Comunicación entre el Microcontrolador y la PC.....	220
4.2.1 Selección del canal de comunicación entre la PC y el MCU.....	220
4.3 Características y Selección de la Base de Datos Relacional.....	222
4.3.1 Características.....	222
4.3.2 Selección de la base de datos.....	236
4.4 Características y Selección de la Herramienta Visual.....	257
4.4.1 Características.....	257
4.4.2 Selección del lenguaje.....	263

CAPITULO V

DISEÑO DEL SISTEMA

5.1 Desarrollo e implementación a nivel hardware.....	268
5.1.1 Diseño del sistema mínimo del MC68HC11.....	269
5.1.2 Módulo de Recolección de Datos.....	285
5.1.3 Registro de las variables físicas.....	287
5.1.4 Transmisión y recepción de datos.....	292
5.1.5 Acoplamiento entre módulos.....	297
5.2 Diseño del Software.....	298

5.2.1 Programación del Microcontrolador.....	298
5.2.2 Descripción del control de comunicaciones de Visual Basic 4.0.....	299
5.2.3 Descripción del front-end (Interface gráfica con el usuario).....	305
5.2.4 Diseño y Construcción de la Base de Datos.....	321
5.2.5 Diseño y construcción del Front-End.....	332
5.3 Areas de aplicación.....	351
Conclusiones.....	352

Bibliografía

Apéndice A. Códigos de programación.

Apéndice B. Hojas de especificación de componentes electrónicos.

Apéndice C. Manual del usuario.

INTRODUCCION

Objetivo General

Desde la aparición de los circuitos integrados en los primeros años de la década de los sesenta, han tenido lugar grandes cambios en la forma de diseñar y de fabricar sistemas digitales. Las continuas mejoras en la tecnología, han incrementado enormemente el rango y la complejidad de las funciones que se pueden incorporar en éstas pequeñas pastillas de material semiconductor. La velocidad en el tratamiento de datos, la exactitud en su funcionamiento y la cada vez mayor miniaturización de los microprocesadores los hace adecuados para un gran número de aplicaciones.

La computadora de viaje es un sofisticado elemento electrónico que ha hecho recientemente su aparición en los automóviles técnicamente más desarrollados. En general, está constituida por un sistema de sensores y un microcontrolador electrónico, estos elementos permiten el monitoreo constante de las condiciones del vehículo, al mismo tiempo que influyen directamente en su desempeño al controlar algunos parámetros como es la mezcla de aire-combustible, la temperatura del motor, sistema de frenos, etc.

Por otro lado, dentro del área de cómputo se han desarrollado potentes herramientas, como son los manejadores de bases de datos y las interfaces gráficas de usuario, que permiten el diseño e implantación de sistemas robustos y veloces en muy poco tiempo.

Cada vez es más común el desarrollo de sistemas que involucren aspectos combinados de cómputo y electrónica para resolver los problemas más complejos que se presentan en la sociedad moderna.

En la actualidad, muchas empresas del transporte en México no cuentan con mecanismos eficientes para su administración y menos aún con elementos que les permitan tomar decisiones para incrementar la seguridad en sus operaciones cotidianas.

El objetivo del presente trabajo es desarrollar un sistema basado en un microcontrolador, a través del cual se registren todas las situaciones por la que atraviesa un vehículo en su recorrido.

A diferencia de las computadoras de viaje actuales que solo intervienen en el funcionamiento del vehículo, el dispositivo desarrollado registrará la información más representativa durante un recorrido (similar a una "caja negra" en los aviones) para posteriormente transferir esta información hacia una computadora personal, en la cual se realizará la interpretación y análisis de la información a través de un sistema que emplea bases de datos relacionales y una interfaz visual para el usuario.

El sistema propuesto, pretende conformarse como un mecanismo conveniente para dotar a las empresas del transporte en México de elementos que les permitan realizar una mejor administración de sus recursos al mismo tiempo de elevar la seguridad y eficiencia en sus operaciones.

CAPITULO I

ANTECEDENTES Y CONCEPTOS BASICOS

1.1 Perfil de las vías terrestres en México

Milenios antes de Cristo, hombres con fisonomía mongoloide y de costumbres nómadas, aprovecharon la formación de gruesas capas de hielo sobre el casquete boreal de nuestro mundo para atravesar del extremo oriental de la península de Chukosky, en Siberia, al extremo Noroccidental de América, en Alaska.

Se supone que estos grupos humanos cruzaron el helado estrecho de Bering persiguiendo a los animales para su alimento, o bien en busca de mejores climas; así, remontaron la corriente del Yukon y se internaron más al Sur, en un lento, pero constante incursionar hacia nuevas tierras.

Otra teoría sostiene que grupos polinesios llegaron a este continente por las costas del Océano Pacífico. Cualquiera que haya sido el grupo de pobladores, ignoraba encontrarse con un nuevo mundo de más de 42 millones de kilómetros cuadrados de extensión.

Por los lugares donde se han encontrado los utensilios que esos hombres utilizaron, ha sido posible establecer los rumbos de sus migraciones y el nivel de su desarrollo cultural.

Más tarde, y después de una milenaria evolución independiente, se destacan en este continente dos grandes zonas de desarrollo cultural, Mesoamérica y la región Andina.

Mesoamérica fue la zona en donde diversos pueblos compartieron aspectos como los religiosos, sistema de medición de tiempo-calendario-métodos agrícolas, organización politicosocial y su escritura que dieron como resultado las grandes culturas prehispánicas mexicanas, entre las que sobresalen: La Olmeca, La Maya, La Tolteca y La Mexica o Azteca.

Uno de los pueblos que forjaron una ciudad fueron los Aztecas, fundada en 1325 tras una larga y penosa peregrinación; para 1519 fecha en que la conocieron Hernán Cortés y su grupo de conquistadores, era grande, opulenta y organizada. La estratégica traza urbana de la lacustre

ciudad suscitó admiración entre los europeos, según conceptos tomados de las Cartas de Relación del propio Hernán Cortés.

En sus orígenes los aztecas se asentaron en un pequeño islote del gran lago, pero como la población iba en aumento, terminaron por ocupar y unir las islas del lago, relleno con tierra las partes intermedias hasta alcanzar la mancha urbana dimensiones considerables. Tenochtitlán tenía de 80 a 100 mil casas y un promedio de cinco habitantes por casa.

Considerando el poder de los Mexicas, los cuales entablaron batallas con los pueblos de los alrededores; estos proyectaron caminos por donde debían llegar con puntualidad los tributos impuestos de los pueblos sojuzgados; transitar sin obstáculos los correos que mantenían informado al Emperador.

En 1519 llegó la expedición de Hernán Cortés a lo que hoy es Veracruz, ahí confirmó las noticias del esplendor del Imperio Azteca y del enorme poder del rey Moctezuma II, quien había seguido sus pasos informando por los veloces correos Mexicas.

En la costa oriental, empezó a tener conocimiento de Moctezuma; recibió solicitudes de algunos caciques para que los liberara de la "tiranía azteca", circunstancia que sería su mejor aliada.

En agosto del mismo año Hernán Cortés partió hacia México, con un gran número de aliados cempoaltecas para entablar tratos con Moctezuma. En su camino hacia México Cortés preparó su ataque a Cholula, ciudad que sería dominada tras una larga y fiera batalla. A su llegada a Amecameca, al pie del Popocatepetl en donde siguieron uniéndose de mexicanos sojuzgados, arribaron a Ayotzingo, junto al lago de Chalco, y el 18 de noviembre de 1519 hicieron su entrada a México Tenochtitlán.

Consumada la caída de Tenochtitlán, se inició la expansión de la Conquista, y ya para 1522 Cortés ordenó la construcción del camino de Tenochtitlán a Veracruz que lo iba a comunicar con

Europa; este camino fue el primero en transformarse en carretera en 1531. Hacia 1540, cien recuas de mulas podían transitar a lo largo del camino, entre Veracruz y la capital.

Las rutas más importantes dentro del Valle de México fueron en su mayoría las mismas de los tiempos prehispánicos pero, con los intereses comerciales de los españoles, se multiplicaron los caminos vecinales, los tributarios y los de grandes peregrinaciones religiosas.

El Virrey Antonio de Mendoza ordenó la construcción de importantes caminos, entre ellos, el que va de México a Acapulco, ya que este puerto era el único del Pacífico autorizado a comerciar con las Filipinas. Por ese camino entraban productos como seda cruda, floja y torcida, tejidos de seda, telas finas de algodón, almizcle, perlas, muebles finos y especias, entre otros. Los caminos que se abrieron dentro del territorio nacional durante la Colonia, estaban auspiciados por el sistema de consulados, establecidos al concluir la conquista.

Para 1532 se descubrieron las primeras minas y se inició un auge que tuvo que integrar una red vial con el propósito de extraer y transportar la producción de las minas. Los difíciles obstáculos orográficos del país no lo fueron para la explotación minera, cuyos caminos, se estipulaba, debían ser conservados para el tránsito de recuas y carretas por los propios explotadores de las minas.

De 1726 a 1739 el incremento demográfico fue permanente en todo el siglo. La población calculada para 1793 era cuatro millones 800 mil habitantes; a fines del siglo aumento un millón, pues en 1803 había cinco millones 837 mil y en 1810 rebasaba los seis millones 200 mil.

En la época Independiente fue desde sus comienzos una gran guerra de clases de los trabajadores del campo y las minas, dirigida por cleros rurales, la mayoría criollos.

En esta época de constante lucha, los caminos y transportes heredados de la Colonia sirvieron para que los ejércitos de ambos bandos transportaran víveres y material de combate. Aquellas brechas fueron acondicionadas para que por ellas pudieran transitar las carretas y diligencias,

tanto de servicio público como particular, además de las famosas recuas que tanto se utilizaron en la época anterior.

A consecuencia de la guerra, durante este período no fue posible ocuparse de atender tales vías de tránsito, cuya condición fue empeorando conforme paso el tiempo.

La última fase de la campaña de Independencia termina con la entrada a la ciudad de México del Ejército Trigarante con Agustín de Iturbide al frente, el 27 de septiembre de 1821.

Iturbide tomó a su cargo la dirección de los asuntos públicos, nombrado por una Junta Gubernamental, la cual lo designó como su presidente y posteriormente coronado Emperador en 1822. Abdicó en 1823, se ausentó del país y a su regreso fue preso y fusilado en 1824. Después de una lucha de 11 años quedando el país en un estado socioeconómicamente en ruinas, por los desajustes políticos, disminución demográfica y abandono de las tierras.

El 4 de octubre de 1824 se promulga la Constitución Federal de los Estados Unidos Mexicanos y el día 1º del mismo mes, Guadalupe Victoria fue electo presidente de la nueva República.

A pesar de encontrarse el país en medio de un desastre político y por ende económico y social, es en estos tiempos difíciles cuando empieza a surgir la preocupación de mejorar las dañadas vías terrestres e inclusive, de proporcionar al país los caminos de fierro, cuyo uso empezaba a intensificarse en las naciones europeas y americanas que habían abierto sus puertas a la revolución industrial.

El primer antecedente relativo a la construcción de un ferrocarril mexicano tuvo lugar en 1837, fecha en que el gobierno del general Anastasio Bustamante, concede la autorización a don Francisco Arrillaga para construir un ferrocarril y enlazar a la ciudad de México con Veracruz, el puerto marítimo más importante del país.

De este proyecto no se tendió ni un kilómetro de vía, pero quedo como constancia del primer intento para establecer una línea férrea en el país.

El interés de unir México con Veracruz se mantuvo latente y en 1842, el presidente Antonio López de Santa Anna, impuso a los acreedores del viejo camino carretero de Perote a Veracruz, la obligación de construir un ferrocarril de dicho puerto al río de San Juan, Veracruz, a cambio de algunos privilegios para el transporte de carga y pasajeros. El 16 de septiembre de 1850 es fecha histórica para los anales ferroviarios, pues se inauguro el primer tramo de la vía del primer ferrocarril en México, con una extensión de 13 kilómetros, entre Veracruz y el Molino.

Fueron varias las concesiones otorgadas por diferentes gobiernos para la construcción de esta vía. Por decreto de octubre de 1853, se firmaron contratos con don Juan Laurie Rickards, para la construcción de la ruta México a Veracruz con la ramal a Puebla, el cual caducó casi tres años después.

En 1824 fue suprimido el derecho de avería que, junto con los peajes, financiaba la reparación de los caminos; hacia la misma época.

El gobierno del presidente Juárez se empeño en remediar el enorme problema que representaba la comunicación terrestre. Así el 19 de diciembre de 1867 destinó una partida de un millón 200 mil pesos del presupuesto nacional de egresos para la construcción y conservación de caminos del país a cargo de la Secretaría de Fomento, medida que constituyó la primera en su género tomada por un Mandatario (figura 1.1.a). A ésta surgieron otras decisiones, las cuales se acompañaron por leyes regulatorias que se referían al uso del suelo, creación de dependencias encargadas de vigilancia y conservación de los caminos, reglamentos para la administración general, otorgamiento de concesiones para la construcción de nuevas vías de comunicación, elaboración de cartas de la República Mexicana y creación de institutos en este ramo.

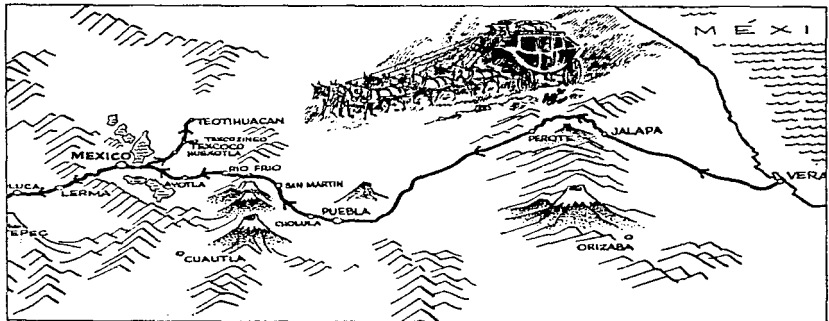


Figura 1.1.a. Ruta de Veracruz a la Cd. de México

Después del fallecimiento del presidente Benito Juárez, en la red carretera destacaban los siguientes caminos (figura 1.1.b.) :

- De México a Puebla, Jalapa y Veracruz
- De Orizaba a Córdoba
- De Huejutla a Tampico
- De México a Querétaro y San Luis Potosí
- De Victoria a Matamoros
- De Monterrey a Piedras Negras
- De Aguascalientes a Zacatecas, Durango y Mazatlán
- De Ciudad de Maíz a Tampico
- De Guanajuato a Lagos y Aguascalientes
- De México a Cuernavaca, Chilpancingo y Acapulco
- De México a Toluca, Morelia y Guadalajara

- De Zapotlan a Colima y Manzanillo
- De Guadalajara a San Blas
- De Tonalá a San Cristóbal y Frontera
- De Campeche a Calkini, Mérida y Progreso.

El fallecimiento de Juárez significó la pérdida del único líder civil capaz de dominar al militarismo renaciente, que trabajaba furtivamente dominado por la figura de Porfirio Díaz, quien se venía haciendo notorio desde diez años atrás. Presintiendo que el presidente Lerdo de Tejada intentaría reelegirse, Díaz volvió a levantarse en armas. Formado en la lucha por la Reforma y contra la intervención extranjera. Desde la toma de la presidencia en 1877, Díaz impulsó constantemente a los ferrocarriles; se destacaba entonces que México había pasado de los caminos de herradura a los de hierro.

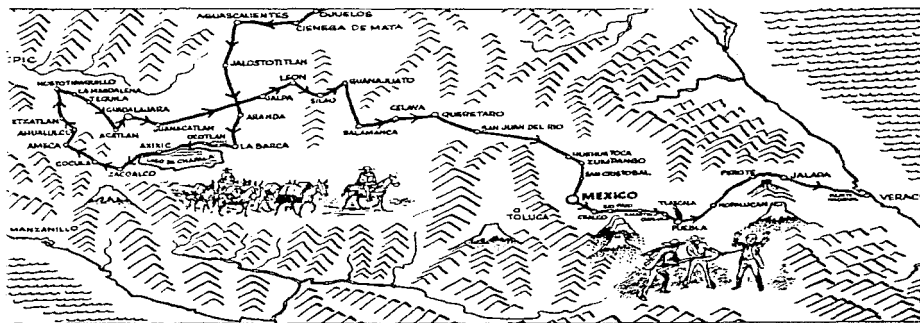


Figura 1.1.b. Principales caminos de México

La creación de Ferrocarriles Nacionales de México, tuvo lugar en 1908, al fusionarse en una sola compañía; por iniciativa y control del gobierno.

Al término del Porfiriato existían en el país, ocho líneas de vía ancha en construcción, 49 de vía angosta y otras 13 sin terminar, además de líneas estables y pequeñas líneas particulares.

Durante el período del Porfiriato, el esfuerzo en materia de comunicación estuvo volcado sobre los ferrocarriles, poco se realizó en materia de caminos; la construcción de estos no sobrepasó los mil kilómetros y el objetivo principal era alimentar las estaciones de ferrocarriles y, en cuantía comunicar zonas que carecían de medios de transporte. El descuido era tal que carreteras que unían poblados eran intransitables.

El gobierno consideraba la construcción de nuevos caminos comunicadores de regiones importantes y conservación de los ya existentes.

El transporte de carga por algunos caminos seguía realizándose con mulas, carros y carretas de poco volumen, lo que hacía muy lento y costoso el traslado de mercancías; la transportación de pasajeros quedaba a cargo de la diligencias, la litera, el guayín y el caballo. La clasificación del camino dependía de la cantidad del ganado que transitara; un buen camino era aquel que soportaba una recua de 100 mulas.

En 1910 se inició la Revolución Mexicana. Las batallas se libraban principalmente a los centros ferroviarios de mayor importancia, ya que el dominio del único sistema de comunicación decidía la suerte de las contiendas, provocando una gran tensión en el sistema ferroviario nacional.

El saldo de la lucha armada fue desastroso para los ferrocarriles, ya que todo el sistema sufrió las consecuencias de los combates y fue intensamente usado por los bandos contendientes, dado que facilitaba el desplazamiento masivo de tropas de un lugar a otro y a largas distancias.

El automóvil había hecho su aparición en México en 1906, trayendo consigo la revolución de los viejos conceptos del transporte; sin embargo, en nuestro país no significó ninguna mejora para los caminos existentes; como estaban continuaron prestando servicio a los vehículos de motor y a los de tracción animal.

En el México de 1925, los automovilistas se limitaban a transitar por las calles y calzadas urbanas. El transporte de personas y mercancías de una ciudad a otra, tenía que hacerse utilizando el ferrocarril, muy deteriorado en aquellos días.

Consolidada la Revolución a partir del primer gobierno preconstitucional de don Venustiano Carranza, en 1914, la conciencia camionera se hizo presente, pero las difíciles condiciones que atravesaba el país no permitían la puesta en marcha de una política encauzada a mejorar y extender los caminos.

Durante los siguientes gobiernos revolucionarios, hasta el de Alvaro Obregón, se palpa un impulso real a la labor caminera, ya que el gobierno adquirió maquinaria especial para la construcción de carreteras. Se celebró en México el Primer Congreso de Caminos y surgió el proyecto de legislación para promover la construcción y conservación de los caminos de México.

En el momento que el gobierno del general Plutarco Elías Calles creó la Comisión Nacional de Caminos, en 1925, significó el punto de partida de la obra caminera con que contamos hoy en día. Pocos meses después de iniciar su gestión, el Presidente obtuvo recursos para comenzar la tarea, implantando un impuesto de la venta de gasolina y encargó a la nueva institución, la administración y aplicación de estos fondos.

Cuando se iniciaba la rehabilitación de los primeros caminos para automóviles, el país tenía 15 millones de habitantes. La producción era incipiente en general y los transportes se fueron desarrollando paralelamente a las vías de comunicación.

Por lo que respecta a la evolución de los transportes carreteros, gracias a la construcción de los caminos que se consideraron como los más importantes, se comunicaron tres áreas: La ciudad de México con las de Pachuca, Puebla, Toluca y Acapulco; la de Mérida con el Puerto de Progreso y Valladolid, y la de Monterrey con Nuevo Laredo.

Los primeros vehículos que transitaron por nuestro territorio tenían poca potencia y capacidad para la carga y pasajeros pues la velocidad máxima era de 40 km./hr.

Es entre 1925 y 1930 cuando se realizaron los primeros mil 420 kilómetros de carretera que unían los puntos arriba mencionados; en este último año se había incrementado el tráfico automovilístico el uno por ciento del territorio nacional. En la siguiente década se agregaron a la red ocho mil 500 kilómetros, con lo cual quedaba comunicado el nueve por ciento del área de la República por el automóvil y el camión. En esos años se utilizaron los primeros autobuses de pasajeros para 20 personas (figura 1.1.c).



Figura 1.1.c. Principales carreteras en 1930

La línea Estrella Roja se fundó en 1925 para proporcionar servicio de México a Cuernavaca con 10 unidades. En 1929 la Alianza Camionera Veracruzana Flecha de Oro, estableció la ruta de Perote a Veracruz.

Para el 19 de febrero de 1940 fue publicado en el Diario Oficial de la Federación en la que se estipulaba cuales son estas vías generales de comunicación, caminos, ferrocarriles, puentes, las corrientes fluviales navegables, mares territoriales, lagunas, lagos, rutas del servicio postal, así como el espacio aéreo en el que transiten las aeronaves (Figura 1.1.d.).



Figura 1.1.d. Principales carreteras en 1940

Dicha ley vino a cubrir una necesidad imprescindible en el ámbito de las comunicaciones, especialmente en los aspectos de planeación, dando preferencia a las zonas de mayor potencialidad económica, así como a las vías de enlace alimentadoras de troncales.

Quando se terminó de unir por medio de la red de carreteras la capital de la República con las de los estados, ciudades fronterizas y puertos principales, ante el empuje de las fuerzas económicas y sociales, el crecimiento demográfico y la expansión de la industria automotriz, las carreteras se saturaron, motivando que el Gobierno Federal construyera en 1950 un nuevo camino entre Cuernavaca y Amacuzac.

Las carreteras de altas especificaciones que operan en la República Mexicana fueron proyectadas para mejorar los índices de seguridad, de velocidad y economía. Cuentan con mayores especificaciones geométricas de curvatura, mayor distancia de visibilidad, pendientes moderadas, mas señalamientos, mayores dimensiones en sus carriles y acotamientos, características que las convierten en mejores y más seguras.

Tienen el control de acceso a lo largo de su recorrido, ya que el derecho de vía se encuentra protegida por alambradas y cuenta con tramos de desnivel para cruce de peatones, de ferrocarriles y de otros caminos y su trazo permite a los usuarios el ahorro en tiempo y dinero.

El tramo de Amacuzac a Iguala fue el primero que se construyó de este tipo en 1950, a continuación se mencionan las primeras carreteras que se construyeron a partir de esta según su cronología.

- De México a Cuernavaca en 1952
- De Cuernavaca a Amuzac en 1954
- De México a Palmillas en 1958
- De México a Puebla en 1962
- De la Pera a Cuautla en 1965
- De México a Tecamac en 1967
- De Entronque Morelos a Pirámides en 1967
- De Querétaro a Irapuato en 1968
- De Zapotlanejo a Guadalajara en 1969
- De Chipalilla a Compostela en 1973

1.2. Administración y Operación de Vehículos en Empresas Públicas y Privadas

Dentro de las empresas públicas y privadas que manejan automóviles y camiones para el transporte ya sea de artículo o personas, se debe de llevar un control vehicular tanto en cuestión administrativa como de la unidad para poder saber el estado de la misma.

Existen diferentes métodos de llevar el control dentro de las empresas, este puede ser simplemente llevando una bitácora donde se van anotando los gastos que van haciendo durante el recorrido del trayecto o también existen controles muy sofisticados como computadores de viaje que van realizando un muestreo de una serie de variables (velocidad, consumo de combustible, tiempos de conducción, etc.) durante el transcurso del viaje y posteriormente poder ser analizados.

Hoy en día las empresas que transportan personas llevan un control muy riguroso de la operación y el control del vehículo, como por ejemplo el conductor no debe de rebasar una velocidad de 90 Kilómetros por hora, aparte la Policía Federal de Caminos realizan una serie de exámenes durante el recorrido del conductor para saber que este no va en estado de ebriedad entre otros factores que se revisan.

Dentro de las empresas públicas y privadas existen diferentes tipos de vehículos, cada uno de los cuales presentan ciertas características para su operación, como son automóviles, transporte comercial (camionetas Pick-up, de redilas o con cajas distintas acondicionadas al giro de la empresa) y camiones de carga y de pasajeros entre otros.

La administración y la operación de cada uno de estos vehículos va de acuerdo a su diseño y también dependiendo de la distancia que se tenga que hacer. En el caso de automóviles, el conductor debe tener una mínima experiencia en el manejo, no obstante, debe conocer que revisión mínima debe practicar a la unidad que operará, estas son :

- Revisión de niveles de aceite, líquidos refrigerante, líquido de frenos, nivel de combustible, fugas de líquidos o aceite y bandas.
- Revisión de presión de inflado de llantas, según lo recomendado por el fabricante.
- Revisión de funcionamiento de luces en general, limpiadores y bocina.
- Realizar un pequeño recorrido de prueba para checar el funcionamiento de embrague, frenos, suspensión y dirección.
- Llenar el tanque de combustible al finalizar el turno.

Habiendo verificado el conductor las recomendaciones anteriores, tendrá plena confianza de la unidad que opera y su conducción será mejor.

Para los vehículos tipo Pick-up, la revisión a practicar contempla los puntos anteriores además de las siguientes, ya que para este caso los vehículos casi siempre llevarán artículos para su transporte. Se menciona a continuación las revisiones adicionales:

- Revisar que la carga a transportar esté debidamente asegurada.
- Revisar que estén visibles los señalamientos de precaución en el vehículo.
- Si la carga sobrepasa el ancho o largo del vehículo, colocar señalamientos que indiquen el límite de ésta.

Para este tipo de vehículos el chofer debe de contar con una experiencia más amplia, ya que el transportar carga de diferente tipo, representa mayor cuidado en la conducción para evitar maltrato de la misma.

Para la operación de un camión, el conductor deberá contar con una experiencia a toda prueba, ya que el manejo de estas unidades requiere de toda la destreza posible de una persona para evitar percances de lamentables consecuencias.

La revisión para estos vehículos, serán las que se mencionaron anteriormente, no obstante la mayoría de estas unidades cuentan con máquinas diesel, lo que representa algunas otras operaciones de revisión, las cuales son:

- Purgar tanque y filtros de combustible.
- Purgar tanque de aire
- Revisar y retirar obstrucciones en el panal del radiador.

Con lo anterior se tendrá plena seguridad que la unidad no sufrirá avería en operación en los elementos revisados.

Aparte de la revisión que realiza a la unidad el conductor debe de tener un amplio conocimiento de mecánica en general para poder solucionar un problema en cualquier momento que se le presente.

La infraestructura para transitar en muchas de las ciudades del país, no se planeó debidamente, es decir acorde a las necesidades y asentamientos de la población, ya que se presenta una deficiencia de caminos, principalmente para los camiones, teniendo que utilizar en la mayoría de los casos vías diseñadas para automóviles y camionetas, como es el caso del Circuito Interior y Periférico en la Ciudad de México. Lo anterior es causa del excesivo tránsito que se presenta en horas "pico", en las vías donde podrían transitar camiones.

Se ha mencionado la experiencia requerida por los operadores en los diferentes vehículos, sin embargo, además de ello deben contar con ciertos requerimientos para poder conducir con seguridad y dentro de los reglamentos y normas dictadas por el gobierno. Estos requerimientos son los siguientes:

- Conocer el reglamento de tránsito vigente.
- Conocer los señalamientos de tránsito que regulan la operación de las vías de comunicación.

- Contar con la licencia de manejo respectiva de acuerdo al vehículo, que exige la Secretaría de Comunicaciones y Transporte.

Además de contar con los requerimientos, experiencia y conocimientos anteriores, es conveniente capacitarlos con algún curso en el que se indiquen ciertas recomendaciones a seguir para reducir el riesgo de accidentes.

Es por ello que se debe elegir al personal adecuado para la conducción de las unidades, y a la vez capacitarlo y supervisarlo continuamente, previniendo con ello desperfectos continuos en las unidades y accidentes de cualquier magnitud.

Las computadoras de viaje son el método más confiable dentro de las empresas; el objetivo principal es el de dar seguridad economía y confort. Estas buscan conocer en sus vehículos más a detalle el funcionamiento del motor y esto es posible obteniendo una serie de datos durante el trayecto como por ejemplo:

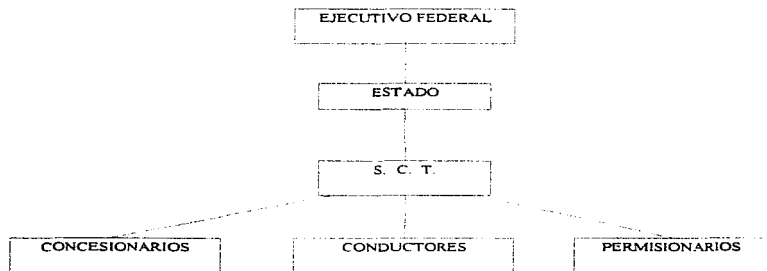
- Velocidad media del viaje
- Consumo instantáneo de combustible
- Distancia recorrida
- Fecha
- Cronómetro
- Distancia hasta el punto de destino
- Hora estimada de llegada a destino
- Consumo medio de combustible
- Hora
- Tiempo parcial
- Pausa del sistema

Unas computadoras de viaje tienen una pantalla gráfica fluorescente al vacío con un alto nivel de contraste que facilita la lectura de los datos vitales que se despliegan en la pantalla para poderse ver a simple vista.

1.3 Normas y Reglamentos del Transporte en México

Es de gran importancia dar a los transportes un sitio en el desarrollo político, social, económico, industrial, y de proyección al exterior, considerando que es vital para la infraestructura del país ya que las diversas modalidades del transporte, exigen un sustento jurídico dinámico de actualización a través del cual los ordenamientos vigentes así como la elaboración y proposición de nuevas leyes y disposiciones garanticen el desarrollo del país en este ámbito.

Corresponde al ejecutivo federal el control y vigilancia de las vías generales de comunicación y de los medios de transporte que operan en ellas, facultades que le confiere la fracción I del artículo 89 de la Constitución Política Federal y los Artículos 10o. (fracciones X, XI, XII y XIV) de la Ley Orgánica de la Administración Pública Federal, así como el 1o. (fracciones VI y VII), 2o. y 3o. de la Ley de Vías Generales de Comunicación y Capítulo II, título segundo del Libro Segundo y demás relativas al propio ordenamiento legal.



El avance tecnológico de la industria automotriz, ha permitido la fabricación de vehículos cada vez más potentes, más veloces y con mayor capacidad, integrados con dispositivos evolucionados

según el avance tecnológico en el campo mecánico, eléctrico, electrónico, de comunicaciones y computación.

Los caminos que integran la red nacional se han visto mejorados en sus especificaciones, y características de diseño y construcción, que permiten el tránsito de vehículos con las características anteriores, además, para gran parte de los desplazamientos de personas y mercancías en el país se utilizan carreteras federales y para salvaguardar vidas y bienes, es necesario contar con un ordenamiento que permita el adecuado tránsito de la circulación vial.

El Reglamento de Tránsito en Carreteras Federales cita que un vehículo puede transitar en carreteras federales equipado mínimamente con:

- Luces: faros principales, lámparas posteriores, reflectantes, indicadoras de frenaje, direccionales, gálibo, de identificación, demarcadoras, de retroceso y reflectantes, así como, lámparas de gálibo y lámparas demarcadoras laterales, luces de vehículos estacionados, equipo adicional de lámparas y reflectantes obligatorio para determinados vehículos, lámparas adicionales en grúas y vehículos de servicio mecánico, indicadores de peligro en carga sobresaliente posterior, lámparas opcionales.
- Faros: faros de niebla y luces auxiliares, dispositivos acústicos y ópticos en vehículos de servicio de emergencia
- Frenos, bocinas y dispositivos e instrumentos de advertencia en los sistemas de frenado.
- Silenciador de escape
- Espejos retrovisores, cristales y limpiadores
- Dispositivos para casos de emergencia
- Llantas de refacción
- Equipo de aire acondicionado
- Equipo adicional para vehículos de servicio público federal

El Reglamento de Tránsito en Carreteras Federales en su título primero especifica diversas definiciones de las que se ofrecen aquí las más relevantes.

- **ACOTAMIENTO.** Faja comprendida entre la orilla de la superficie de rodamiento y de la corona de un camino, que sirve para dar más seguridad al tránsito y para estacionamiento eventual de vehículos.
- **AUTOMOVIL, COCHE.** Vehículo de motor con cuatro ruedas con capacidad hasta de nueve personas, incluido el conductor.
- **CALLE, VIA URBANA.** Vía pública comprendida dentro de una zona urbana y que forme parte de una carretera federal.
- **CAMION.** Vehículo de cuatro ruedas o más, destinado al transporte de carga.
- **CARRETERA, CAMINO.** Vía pública de jurisdicción federal situada en zonas rurales y destinada principalmente al tránsito de vehículos.
- **CARRIL.** Una de las fajas de circulación en que puede estar dividida la superficie de rodamiento de una vía, marcada, con anchura suficiente para la circulación en fila de vehículos de motor de cuatro ruedas.
- **CEDER EL PASO.** Tomar todas las precauciones del caso, inclusive detener la marcha si es necesario, para que otros vehículos no se vean obligados a modificar bruscamente su dirección o su velocidad.
- **CONDUCTOR.** Persona que lleva el dominio del movimiento del vehículo.
- **CRUCE.** Intersección de un camino con vía férrea.
- **DISPOSITIVOS PARA EL CONTROL DEL TRANSITO.** Señales, marcas semáforos y otros medios que se utilizan para regular y guiar el tránsito.
- **GLORIETA.** Intersección de varias vías donde el movimiento vehicular es rotatorio al rededor de una isleta central.
- **LUCES ALTAS.** Las que emiten los faros principales de un vehículo para obtener largo alcance en la iluminación de la vía.
- **LUCES BAJAS.** Las que emiten los faros principales para iluminar a corta distancia.
- **LUCES DE ESTACIONAMIENTO.** Las de baja intensidad emitida por dos faros accesorios colocados en el frente y parte posterior del vehículo y que pueden ser de haz fijo o intermitente.

- **LUCES DE FRENO.** aquellas que emiten el haz por la parte posterior del vehículo, cuando se oprime el pedal de freno.
- **LUCES DE MARCHA ATRAS.** Las que iluminan el camino, por la parte posterior del vehículo, durante su movimiento hacia atrás.
- **LUCES DIRECCIONALES.** Las de haces intermitentes, emitidos simultáneamente por una lámpara delantera y otra trasera según la dirección que vaya a tomar el vehículo.
- **ÓMNIBUS, AUTOBUS.** Vehículo de motor destinado al transporte de más de nueve personas.
- **VIAS DE ACCESO CONTROLADO.** Aquellas en que la entrada o salida de vehículos se efectúa en lugares específicamente determinados.

Clasificación de los vehículos:

- automóviles
 - motocicletas
 - otros
 - ómnibuses
 - bicicletas
 - camiones
 - remolques
- Luces:

Los vehículos de motor de 4 o más ruedas deberán estar provistos de dos faros principales delanteros, que cuando estén encendidos emitan una luz blanca, colocada simétricamente y al mismo nivel, uno a cada lado del frente del vehículo y lo más alejado posible de la línea de centro y a una altura no mayor de 1.40 m, ni menor de 0.60 m., deben ser operados con facilidad y manipulados como luz baja deben permitir ver personas o vehículos a una distancia de 30 m al frente, el haz que proyecte no debe incidir en los ojos de algún conductor que se acerque en sentido opuesto, manipulados como luz alta deben permitir la visibilidad a 100 m de frente, el indicador de luces debe encender siempre que este en uso la luz alta y permanecer apagado bajo cualquier otra circunstancia.

Una de las lámparas posteriores o un dispositivo aparte deberá estar construido y colocado de manera que ilumine con luz blanca la placa posterior de identificación claramente a 15 m atrás. Las lámparas rojas posteriores y la luz blanca de placa deberán estar conectadas de manera que enciendan simultáneamente con los faros principales delanteros o las luces de estacionamiento.

Los vehículos automotores, los de semirremolque o los de remolque deben estar provistos en su parte posterior de dos o más reflectantes rojos a una altura no menor de 0.35 m ni mayor de 1.50 m, visibles en la noche a 100 m de distancia.

Las lámparas indicadoras de frenaje deben emitir luz roja al aplicar los frenos y ésta debe ser visible bajo la luz solar desde una distancia de 90 m atrás.

Los autobuses y camiones de 2 m de ancho o más deberán portar un equipo adicional de lámparas y reflectantes, al frente dos lámparas de galibo, una a cada lado y tres lámparas de identificación, a cada lado dos lámparas demarcadoras una anterior y otra posterior, a cada lado de estas dos reflectantes.

Las luces indicadoras de vehículos estacionados deben ser dos lámparas delanteras colocadas simétricamente y lo más alejado posible de la línea del centro del vehículo, altura entre 0.35 m y 1.60 m, visible a 300 m por delante, deben encender simultáneamente con las lámparas rojas posteriores que hacen las veces de luces de estacionamiento.

Cualquier vehículo de motor podrá estar provisto de uno o dos faros buscadores de niebla, una o dos lámparas auxiliares de conducción.

- Frenos y dispositivos de advertencia:

Los frenos deberán ser fácilmente accionados por el conductor del vehículo desde su asiento, deben conservarse en buen estado de funcionamiento, estar ajustados de manera que actúen uniformemente en todas las ruedas. Los frenos de servicio permiten reducir la velocidad del

vehículo e inmovilizarlo de modo seguro, rápido y eficaz, cualesquiera que sean las condiciones de carga y de la pendiente de la vía por la que circula. Los frenos de estacionamiento permiten mantener inmóvil al vehículo sin importar las condiciones en que este se encuentre, actuarán en las ruedas o cuando menos sobre una rueda de cada lado de su plano longitudinal medio, sin importar el agotamiento de la fuente de energía o fugas de cualquier especie.

Los vehículos deberán estar provistos de una bocina que emita un sonido audible desde una distancia de 60 m en circunstancias normales, quedando prohibido instalar bocinas u otros dispositivos de advertencia que emitan sonidos irrazonablemente fuertes o agudos, podrán estar provistos de un dispositivo de alarma contra robos. El vehículo debe estar provisto de un silenciador de escape, conectado permanentemente para evitar ruidos excesivos

- Espejos retrovisores, cristales y limpiadores:

Los espejos retrovisores deben permitir al conductor ver la circulación detrás de su vehículo. El parabrisas, ventana posterior, ventanillas y aletas laterales deben mantenerse libres de cualquier material opaco que obstruya la visibilidad del conductor de esta manera el parabrisas deberá estar provisto de un dispositivo que le libre de lluvia, nieve, humedad, etc., que limite la visibilidad.

- Llantas:

La llantas deben garantizar la seguridad del vehículo y proporcionar su adecuada adherencia sobre el pavimento aunque se encuentre mojado, además deberá llevar una llanta de refacción inflada a presión adecuada.

- Equipo de aire acondicionado:

El aire acondicionado deberá estar construido e instalado de manera que al accionarse no despidan sustancias tóxicas o inflamables.

- **Placas:**

Los vehículos de autotransporte dedicados al servicio público federal deberán ser matriculados en la Secretaría de Comunicaciones y Transportes, de acuerdo al domicilio del propietario, dichos propietarios para poder matricularlos deberán presentar la solicitud correspondiente que contenga: nombre, domicilio del propietario, marca, modelo, tipo, número de fábrica (serie), número del motor del vehículo y cualquier otro requisito informativo que señale la Secretaría y que permita determinar si el vehículo en cuestión puede o no ser matriculado legalmente. A dicha solicitud deberá acompañarse lo siguiente:

- a) Concesión o permiso correspondiente para la prestación del servicio público federal de autotransporte
- b) Certificado del Registro Federal de Automóviles
- c) Comprobante de pago de los derechos de matrícula correspondiente
- d) Baja de la matrícula anterior cuando se trate de vehículos usados

Las placas del servicio público federal, serán entregadas por la Secretaría de Comunicaciones y Transportes, previo el pago a la Federación de las cuotas correspondientes y tendrán la vigencia que señale dicha Secretaría, vencido el plazo correspondiente, deberá efectuarse el canje de las mismas. En caso de extravío, mutilación o ilegibilidad de las placas, tarjeta de circulación o calcomanía se deberá solicitar la reposición de los documentos.

- **Licencias para Conducir:**

Para conducir un vehículo destinado a la prestación de un servicio público federal de autotransporte será necesario obtener la licencia federal correspondiente de conductor, expedida por la Secretaría de Comunicaciones y Transportes en cuyo caso es necesario cumplir con los siguientes requisitos:

- a) Tener 21 años, acta de nacimiento, cartilla o pasaporte

- b) Demostrar a satisfacción de la Secretaría de Comunicaciones y Transportes, que posee la experiencia y capacidad suficientes para conducir un vehículo de servicio público federal.
 - c) Conocer y saber interpretar los preceptos de la Ley de Vías Generales de Comunicación y Disposiciones reglamentarias de la misma en materia de circulación de vehículos y seguridad en los caminos
 - d) Cubrir el derecho de la expedición de la licencia
 - e) La licencia tendrá una vigencia de diez años; pero deberá revalidarse cada dos años o reexpedirla al término de su vigencia
- Obligaciones de los Concesionarios y Permisarios de Servicio Público Federal

Ninguna persona o empresa deberá permitir la conducción de vehículos de servicio público federal a quien no tenga la licencia correspondiente, es decir, los concesionarios y Permisarios de servicio público federal de autotransporte están obligados a vigilar escrupulosamente que el manejo y control de sus vehículos quede encomendado sólo a conductores que cuenten con la experiencia, capacidad, pericia y condiciones físico-mentales adecuadas.

Por otra parte el marco jurídico del autotransporte federal esta contenido en el titulo segundo, capitulo II, de la Ley de Vías Generales de Comunicación y Reglamentos de la Secretaria de Comunicaciones y Transportes en lo relativo a la explotación de Caminos, es en ella donde se establecen sus modalidades y características.

Entre los múltiples asuntos que se norman, sobresalen los siguientes:

Para la prestación del servicio público en los caminos de jurisdicción federal, se requiere obtener una concesión por parte del gobierno federal a través de la Secretaría de Comunicaciones y Transportes, con las excepciones que la propia ley establece. Sólo pueden explotar los caminos de jurisdicción federal los mexicanos por nacimiento y las sociedades constituidas por éstos conforme a las leyes del país.

El límite en el número de vehículos explotados por persona no deberá ser mayor de cinco para evitar el acaparamiento por parte de éstos.

Además se regulan en esta ley los diversos tipos de servicio en materia de autotransporte y las modalidades especiales que cada uno tiene en su prestación, así como las formalidades que deben cumplirse para la obtención de concesiones y permisos que deben realizar los particulares para la explotación de un servicio determinado.

En dicha ley se establece que cualquier modificación al régimen jurídico, deberá procurar el aumento en la eficiencia y economía del servicio mediante su prestación más organizada, suficiente y rentable para garantizar un buen servicio.

Vivimos en un sistema de gobierno en el que el Estado tiene la obligación de proporcionar a la comunidad una eficiente y segura infraestructura en lo que se refiere a transporte ya que con ello se garantiza la comunicación en la totalidad del país.

En nuestra legislación el Derecho Administrativo separa por una parte los servicios públicos administrativos y por otra a las empresas del Estado.

Los servicios públicos admitidos dentro de un sistema liberal son los de Defensa Nacional, de Policía, de Comunicaciones, de Prestaciones Exigidas por la Vida Urbana, de Puertos Marítimos, Aéreos, de Irrigación, etc., estos servicios se realizan en algunos casos por el Estado y en otras mediante concesiones y un buen número por Organismos Descentralizados.

Existen servicios públicos manejados por particulares ya que el Estado no tiene capacidad económica suficiente para satisfacer las necesidades colectivas que demandan las diversas comunidades del país.

Un particular puede prestar un servicio público sólo mediante una concesión (privilegio o derecho que se obtiene del Estado para una explotación), ya que el simple aprovechamiento de la vía o la

realización, en ella, de una actividad que no satisfaga la definición de servicio público no constituye en sí misma un servicio público y, por lo tanto, no requiere de autorización en forma de concesión, sino en forma de permiso (figura jurídica con características distintas a las de la concesión).

Definiciones:

Entenderemos por transporte el conjunto de diversos medios de traslado de personas y mercancías.

Servicio público es un acto del Estado para satisfacer las necesidades de la colectividad.

Los servicios públicos son organizaciones que forman la estructura misma del Estado.

El acto jurídico que se denomina concesión puede ser clasificado en:

- a) Concesión de bienes del Estado
- b) Concesión de Servicio público
- c) Concesión de obra pública y/o servicio

Por medio de la concesión de servicio público el Estado lega al particular parte de sus facultades, ello no significa que renuncie a todas sus prerrogativas ya que sólo dará al concesionario las facultades necesarias para que pueda prestar el servicio reservándose el poder del control en todo momento. Las prerrogativas más comunes que un concesionario puede obtener son:

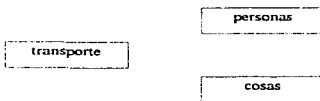
- 1) ocupación del dominio público
- 2) establecimiento de restricciones y servidumbres
- 3) expropiación por causa de utilidad pública
- 4) exclusividad del servicio
- 5) exención de impuestos

- 6) seguridad (policia)
- 7) subvenciones
- 8) garantia de intereses
- 9) cláusula de no mejores condiciones en el otorgamiento de nuevas concesiones
- 10) derecho de preferencia y ampliación de servicio

Los permisos no están sujetos a un plazo de vigencia como las concesiones, en general son indefinidos y en su otorgamiento no existe facultad discrecional por parte de las autoridades competentes.

En el caso de las terminales de autotransporte federal de pasajeros, éstas son lugares en las que los vehículos de cada ruta inician y terminan su recorrido, constituyéndose como parte integrante de la vía general de comunicación, en ellas se presenta una diversidad de servicios. Las instalaciones en sí, se enmarcan en la fracción I del artículo segundo de la Ley de Vías Generales de Comunicación y Reglamentos.

Una clasificación de transporte:



La responsabilidad en el contrato de transporte terrestre no recibirá el mismo tratamiento en el transporte de cosas que en el transporte de personas.

El artículo 590 fracción VIII del Código de Comercio, se relaciona con el artículo 123 del reglamento del capítulo de Explotación de Caminos de la Ley de Vías Generales de Comunicación, en ellos se menciona que la empresa es responsable de todos los daños, salvo en

caso de que la cosa transportada provoque algún daño por sí misma, por caso de fuerza mayor o por caso fortuito.

En cuanto a transporte de personas, el conductor tiene la obligación de trasladar sin algún daño al pasajero por lo que los concesionarios o permisionarios de servicios públicos de transporte deben contratar un seguro de viajero o crear un fondo de garantía que cubra cualquier responsabilidad objetiva del transportista, la aseguradora intermedia en el riesgo y el viajero recibe la protección debida. La empresa responderá también por los efectos que lleve consigo el viajero.

Los concesionarios están obligados a proporcionar a los usuarios, locales comerciales y/o especiales para ello, estos deberán contar con instalaciones adecuadas.

Es preciso mencionar que toda modificación a las leyes, normas, reglamentos, etc. previamente establecidos deben mejorar la calidad del servicio y la seguridad de cosas y personas.

En nuestro caso se espera mejorar la calidad del servicio en cuanto a la seguridad del transporte, por medio de la construcción de una computadora de viaje que permita el registro de los parámetros más importantes para su estudio y análisis.

Cabe mencionar que para la construcción de la misma es necesario conocer las limitantes jurídicas ya que se pone en juego el transporte de vidas humanas y aún en el caso de transporte de cosas la calidad en tiempo de entrega, economía y seguridad del objeto se hace necesaria.

1.4 Empleo de los Microprocesadores en la Industria Automotriz.

La electrónica está especialmente dotada para cambiar la faz de nuestra civilización. He aquí una cosa que algunas personas saben a ciencia cierta y otras muchas lo intuyen. La revolución que ha representado el descubrimiento y utilización de los semiconductores y, posteriormente los proyectos que han dado paso a la informática, la cibernética y la robótica tienen un porvenir inmediato que llegará a cambiar las costumbres, los hábitos y las ideas de los seres humanos y, con todo ello, la sociedad que hemos conocido hasta el momento. El cambio se está realizando ya en nuestros días.

De igual modo, también el automóvil, que es herramienta fundamental y que hace posible nuestra sociedad de hoy, con sus grandes aglomeraciones urbanas, está siendo invadida, para bien, por los productos que la electrónica elabora, los cuales vienen a mejorar sus funciones, ya sea en la parte de tracción ó mecánica, o bien desde el punto de vista de la seguridad, en éstas nuestras carreteras cada día más concurridas. A continuación presentamos algunos ejemplos.

El problema del encendido ha podido ser resuelto de una manera satisfactoria con la colaboración de la electrónica. La supresión del ruptor mecánico ha supuesto un paso de gigante para conseguir no solamente eliminar un constante trabajo de entretenimiento en los antiguos motores, sino la seguridad de una permanente y correcta puesta a punto durante miles y miles de kilómetros, lo que se ha traducido en una economía de consumo muy sustanciosa. La inyección de gasolina, otro importante factor de seguridad y alto rendimiento, puede servirnos como el ejemplo más sobresaliente en el que la aplicación de la electrónica adquiere una vital importancia. En comparación con el carburador, que se halla regido exclusivamente por el flujo de aire que circula por su conducto, la inyección tiene en cuenta la temperatura del motor, el número de revoluciones por minuto a que gira el cigüeñal, la composición química de los gases de escape, el estado de depresión en el conducto de admisión, etc., para determinar la cantidad exacta de gasolina que tiene que aportar al aire aspirado.

Tales cálculos que, como se advierte, pueden combinarse entre sí y dar por resultado una extraordinaria cantidad de variantes, deben ser procesadas por un minúsculo procesador que determine con suma precisión la cantidad de gasolina que en todo momento resulte más conveniente aportar para todo el conjunto de condiciones dadas. El resultado es que todos los motores dotados de inyección electrónica de combustible son sencillamente más potentes, pero mucho más económicos que los motores dotados con los procedimientos tradicionales de alimentación.

Pero la electrónica está presente además en casi todos los circuitos y partes que componen al automóvil. En la parte eléctrica no hay que decir, desde el generador que es posible gracias a su puente electrónico de diodos rectificadores y que mejora su funcionamiento gracias a la acción ejercida por un diminuto regulador electrónico, hasta los limpiaparabrisas, los indicadores de giro, los tableros de instrumentos actuales con chequeo automático de los principales circuitos, etc., son una muestra de ello. En lo que respecta a la parte mecánica, se estudia también la posibilidad de conseguir una distribución variable automática, provista de avance, por medio de la cuál se optimaría en alto grado el funcionamiento del motor a cualquier régimen de giro elegido. Podríamos poner todavía más ejemplos, pero estimamos que basta con lo dicho para damos cuenta de la importancia que ésta nueva pero ya madura tecnología representa en la industria del automóvil

Las nuevas técnicas y los nuevos procedimientos se van imponiendo poco a poco en el ya complejo mundo de la mecánica del automóvil. Con lentitud pero con seguridad, los automóviles modernos van incorporando todos aquellos adelantos que han sido conquistados por otros sectores industriales, sobre todo si estos adelantos tienen algo que ver con la economía de consumo y con la atenuación de la contaminación atmosférica, lo primero exigido cada vez más por el comprador del vehículo y lo segundo por los reglamentos de los ayuntamientos de las ciudades afectadas.

Desde los años setenta estamos viviendo un importante progreso en la incorporación al automóvil de todas estas técnicas y los microprocesadores forman parte fundamental en los sistemas de control electrónico de los vehículos.

Los microprocesadores, como en la realidad la gran mayoría de las máquinas realizadas por el hombre, está construida a imagen y semejanza humanas. La forma de actuar de un microprocesador tiene mucho que ver con la forma de actuar del hombre frente a un problema semejante. Veamos un ejemplo : En la figura 1.4.a, lo que podría ser un esquema muy simple de lo que parece y ocurre en la percepción y reacción del cerebro humano.

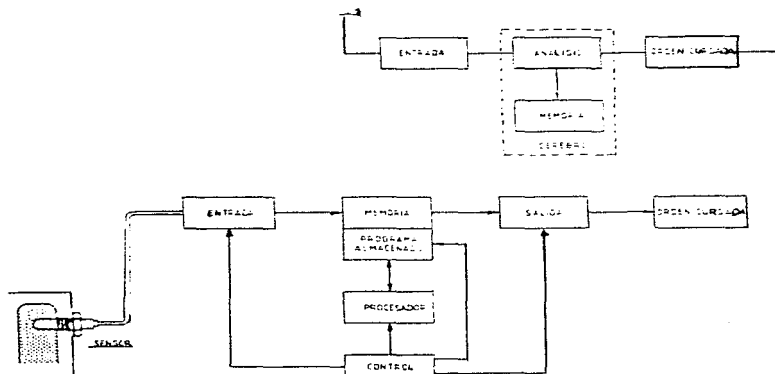


Figura 1.4.a Estructura básica de un microprocesador

Uno de nuestros sentidos aprecia una acción e inmediatamente le da entrada a lo que podríamos llamar un bloque de análisis, el cual consulta con la memoria para saber que se hace normalmente en estos casos o similares. Elabora una respuesta y cursa una orden de conducta frente a la acción.

Una unidad de control con microprocesador está diseñada para que pueda reaccionar de forma parecida. En el caso de las máquinas lógicas de éste tipo usadas en el automóvil, los sentidos equivalentes a nuestra vista, tacto, oído, etc., están sustituidos por los sensores o captadores que se encuentran en contacto con lo que se trata de controlar. Esta información controlada por los sensores, pasa a la memoria, la cual consulta con su programa almacenado. Según los datos obtenidos, el ordenador actúa de acuerdo con las ordenes de su programa, y toma decisiones de cálculo que una unidad de control vigila. Una vez tomada la decisión, esta pasa en forma de impulsos eléctricos por la unidad de salida para materializar una orden activa que puede ser, por ejemplo, la de excitar el devanado de un solenoide ó bobina y producir con ello una corriente magnética lo suficientemente intensa para abrir una válvula durante una fracción muy pequeña de segundo, y dejar salida libre a la gasolina a través del inyector. Los microprocesadores están capacitados para cumplir este proceso en unidades de tiempo extraordinariamente breves, de solamente algunos nanosegundos, es decir, fracciones de 0.00000001 segundos (1×10^{-9}), por lo que su capacidad de reacción es prácticamente instantánea.

Dado que en una unidad de control, podemos almacenar órdenes y acciones de control, a un dispositivo de este tipo se le pueden grabar en la memoria ordenes como la siguiente : "Cuando la temperatura de un sensor mande una señal (equivalente a un estado de temperatura muy determinado y exacto) poner en marcha un ventilador y esperar cinco segundos; comprobar si el sensor de aire anuncia que la corriente de aire se ha establecido. Si es así, abrir una válvula para entrada de agua fría en el circuito. Si no hay aire ó no hay agua encender una luz testigo y poner en funcionamiento una alarma cuando hayan transcurrido 10 segundos."

Un trabajo como éste resulta tremendamente sencillo para una máquina incluso tan simple como la que hemos descrito de una forma muy simplificada. Realmente a un microordenador se le puede

exigir una reacción diferente para cada estado de temperatura que un sensor puede convertir en diferentes estados eléctricos; puede combinar la necesidad de que intervengan un número elevado de ventiladores con los más diversos caudales y una aportación de diferentes conductos de agua a diferentes temperaturas para modular la temperatura final del líquido que el sensor controla. Todo ello no es más que la adición de instrucciones a una memoria dotada cada vez con mayor capacidad de almacenamiento.

De una forma muy elemental podemos pensar en el microprocesador como una unidad de control que posee un entrada de datos a la que llega constantemente la información producida por los sensores encargados de efectuar algún tipo de medición. En la figura 1.4.b tenemos un esquema funcional simplificado de lo que son estas unidades de control.

En la figura, por ejemplo, tenemos un captador en una zona del cigüeñal por medio del cuál se proporciona el dato numérico de rev/min. a que se encuentra en cada momento el cigüeñal.

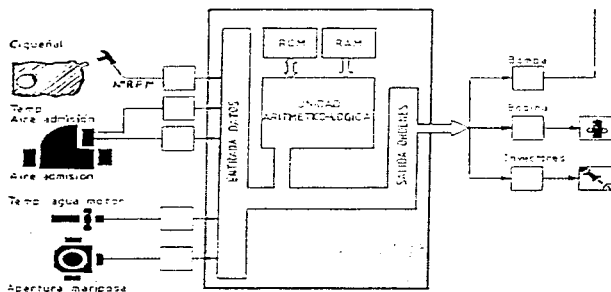


Figura 1.4.b. Esquema funcional simplificado de un microprocesador estudiado para el control de la inyección de gasolina.

También tenemos otro sensor que mide constantemente la temperatura del agua de refrigeración, y por último, tenemos otro que proporciona información permanente de determinadas posiciones angulares de la mariposa del acelerador. Todas éstas son las informaciones que se están ofreciendo de una manera constante a la unidad de control mientras está en funcionamiento. Por otro lado observamos que se posee una salida de órdenes, a través de las cuáles se indica al inyector la cantidad exacta de combustible que tiene que aportar; se dan órdenes a la bobina de encendido para que aplique la corriente de alta tensión en el momento más favorable; se mandan órdenes a una bomba para que éste obtenga las presiones más convenientes, etcétera.

Entre la entrada de datos y la salida de órdenes existe un conjunto de dispositivos "lógicos" capaces de combinar las diferentes posibilidades de las informaciones que se reciben: Son las memorias (ROM y RAM) y la unidad aritmética-lógica que analiza y elabora las órdenes resultantes de las informaciones que recibe en cada momento. Cuando el motor está a 70 °C, por ejemplo, es un dato que pasa a elaborarse y compararse con lo que el dispositivo tiene que responder cuando se trata de éste caso. Ello lo ha de combinar con la situación, por ejemplo, de la apertura de la mariposa. Si la mariposa está cerrada, el caudal del paso del aire es pequeño, y también las rev/min, de modo que se está procediendo al arranque del motor, la unidad aritmética-lógica, de acuerdo con las instrucciones de memoria en estos casos, elabora una orden de dosificación a los inyectores que es diferente de la que tendría que mandar en las mismas circunstancias si la temperatura del motor fuera inferior a los 20 °C.

La electrónica puede trabajar de dos maneras conceptualmente diferentes: una de ellas consiste en trabajar comparando siempre unos valores que recibe con otros valores fijos que se encuentran en sus circuitos establecidos de antemano. Por ejemplo un circuito admite un determinado voltaje muy preciso; en el momento en el que llega a un voltaje superior al esperado, la máquina produce una reacción que se transmite a un mando (o circuito de potencia). De igual modo podemos trabajar con valores de frecuencias, u otros, que la máquina siempre está comparando con valores establecidos para transmitir o no una señal que determine una reacción deseada. Esto es lo que da origen a los circuitos analógicos en los que, en general, los voltajes no están limitados

en amplitud (salvo en el caso de no sobrecargar el circuito, como es natural) y resulta un procedimiento muy adecuado para llevar a cabo los circuitos de radio, TV, etc., que son una importante rama de la electrónica.

Pero también existe otra forma conceptualmente diferente de actuar que está constituido por los circuitos digitales en los que las señales de operación están determinadas por el hecho de que el voltaje en un determinado elemento solamente puede hallarse en dos estados que consisten en el paso de corriente o en la interrupción del paso de corriente, es decir, en estado de conducción ó en estado de bloqueo. Desde un punto de vista eléctrico ello equivale a la presencia de unos relés que estuvieran activados o desactivados. A primera vista podría parecer que los circuitos digitales tienen muy pocas posibilidades frente a las ventajas que parecen presentar los circuitos analógicos. En el caso de los microprocesadores nada es más erróneo. Un aparato provisto de la técnica digital puede llegar a realizar trabajos de control y de cálculo que pueden ser llevados a un alto grado de sofisticación y muy por encima de las posibilidades de lo que en éste campo podrían realizar diseños con circuitos analógicos. En el terreno de los aparatos de control la aplicación de los diseños digitales nos sirven perfectamente para gobernar máquinas de una forma similar a lo que podría lograrse con la presencia de relés, pero con un precio de costo y un ahorro de volumen formidablemente menores.

Por supuesto, el microprocesador, o con más propiedad la unidad electrónica de control, representa la cúspide de la aplicación de la electrónica al automóvil hasta el momento. A pesar de los importantes trabajos de control que ya se le han encomendado, sobre todo en instalaciones de encendido y de inyección de gasolina, el futuro de estos conjuntos electrónicos para centralizar los más diversos controles del automóvil parece que acaba de empezar pues en un futuro se observará la gran variedad de funciones que pueden llegar a realizar para aumentar la seguridad, el correcto funcionamiento y control de otras partes importantes del automóvil como pueden ser los frenos, los cambios de luces, la suspensión, antirrobo, cambios automáticos, etcétera.

Los sistemas de inyección electrónica de combustible para los motores en los vehículos, son ahora capaces de incrementar la precisión en sus funciones de control gracias al progreso de los

microprocesadores en su rendimiento y su eficiencia. Es de esperarse que en un futuro, los sistemas de control de los vehículos sean dependientes del desarrollo de los microprocesadores, así también como de las mejoras en las técnicas y las teorías del control.

Existen algunos aspectos a considerar en las tendencias futuras de los controles en los automotores, a continuación mencionamos los más importantes.

a) **Control Multifuncional.** A diferencia de la función específica del control electrónico en la inyección de combustible que se produjo en los primeros desarrollos, el control del encendido, la velocidad, frenos, etc. han sido introducidos en los mercados recientemente. Estas tendencias de un control multifuncional seguramente seguirán expandiéndose en los próximos desarrollos.

b) **Control Sofisticado.** Es de esperarse que las funciones específicas del control se vuelvan más sofisticadas. En el área de la **fuel injection** (inyección electrónica de combustible), la inyección secuencial de cilindro por cilindro se vuelve más común y cada vez se perfeccionan los sistemas con la correcta mezcla de aire-combustible en un cilindro individual, gracias a los nuevos sistemas. Pero no solo se tiene que incrementar la precisión y exactitud en las funciones individuales de control, sino también se requieren altas velocidades en los controles para superar los comportamientos transitorios de los motores.

c) **Tamaño Compacto.** El tamaño y peso de los controladores se ha ido reduciendo, gracias a las nuevas técnicas de construcción. Esto permite aprovechar mucho mejor los espacios interiores en los vehículos.

d) **Bajos costos.** Gracias a los grandes avances en las técnicas de construcción de los microprocesadores, su alto rendimiento viene acompañado de un costo muy bajo, esto permite indudablemente su rápida incorporación a las áreas de control en los vehículos.

Con el incremento en la diversidad de variables a controlar y la alta sofisticación del control de los parámetros en funciones específicas, se requiere que los controladores sean capaces de operar una

gran cantidad de parámetros y a una elevada velocidad. Por ejemplo, en el control de la inyección electrónica de combustible (fuel injection) se requiere además de un alto grado de sofisticación, rapidez en las operaciones de tiempo real.

En el pasado, el gran número de parámetros de entrada y salida, que debía operar un microcontrolador requería de almacenar (a través de un programa) grandes cantidades de información, misma que debía recuperar en algún momento para realizar algunas funciones de control. Lo anterior tenía una gran limitante, pues en operaciones de tiempo real, la eficiencia del sistema bajaba considerablemente.

Por lo tanto se requiere que los próximos controles a desarrollar cuenten con unidades especialmente diseñadas para una función específica, tales como los coprocesadores matemáticos, los cuales se encargan de realizar todas las operaciones y funciones matemáticas necesarias, disminuyendo así la carga de trabajo del procesador principal y obteniendo una mejor respuesta en tiempo real.

La gran velocidad que se requiere, puede satisfacerse con los actuales microprocesadores de 16 bits de longitud de palabra, un tamaño superior no es necesario en un futuro próximo en vista de que la exactitud en el control es realizable con los actuadores y sensores con que se cuenta actualmente, por el contrario si se manejara una longitud de palabra mayor se estaría desperdiciando una cantidad considerable de información y el costo del controlador sería más elevado.

En resumen y a nivel de conclusión podemos decir que el desarrollo de los nuevos microcontroladores empleados para el control de los motores, deben de responder a las necesidades que se tengan en la industria automotriz.

El desarrollo actual en el área de la electrónica y más específicamente en el área de los microprocesadores, ha alcanzado los siguientes resultados :

- **Desarrollo de controles multifuncionales .** Los microcontroladores actuales, están dotados con unidades especialmente destinadas a una tarea específica, esto permite mejores respuestas en tiempo real de operación, reducción de el tamaño del **software** (programas de computadora) y un alto grado de sofisticación.
- En virtud de que el **CPU** (Unidad Central de Proceso de una computadora), se dedica más a tareas de operación y control, las capacidades de cómputo han podido ser extendidas, sin perder eficiencia en la operación.
- Se han llegado a alcanzar puntos óptimos de funcionalidad entre el **CPU** y las unidades adicionales como memorias y periféricos. En el caso de los automóviles es importante la relación entre el microprocesador, los sensores y los actuadores.
- Gracias a las grandes capacidades de procesamiento alcanzadas en los últimos años, se han podido incorporar las últimas teorías del control a aplicaciones reales, logrando así aumentar las capacidades en los vehículos.

1.5 Desarrollo e Implementación de las Computadoras de Viaje y las Cajas Negras.

1.5.1 Desarrollo de las Computadoras de Viaje.

Vamos a realizar el estudio de una serie de aparatos que tienen por misión coadyuvar a la mejora de la conducción del automóvil. El más importante de estos aparatos y al que se le da preferencia es el procesador de viaje, también conocido con el nombre de computadora de viaje.

La computadora de viaje es un aparato en el que interviene fundamentalmente la electrónica, ya que debe disponer de una serie de circuitos integrados que son básicos en cualquiera de los tipos de procesadores que se conocen hasta ahora. En su consecuencia se trata de una realización altamente sofisticada, aunque en el fondo se trata de un sencillo conjunto de circuitos si se le compara, por ejemplo, con una computadora personal y no digamos con el típico computador de oficina, casos éstos que corresponden a aparatos capaces de modificar la información proporcionada por medio de un cambio de programa, cosa que no se puede realizar, evidentemente, en una computadora de viaje ya que ésta trabaja siempre en virtud de un solo programa que su diseñador le introduce en su momento.

El procesador, en general, es un aparato de alta tecnología electrónica. La comprensión de cómo puede llegar a realizar su trabajo no puede llevarse a cabo sin un conocimiento relativamente profundo no sólo de lo que es la electrónica en general y la función que realizan sus principales y más conocidos componentes, tales como los diodos, los transistores, los tiristores, los transistores monounión, etc., sino también con un cierto conocimiento de la lógica que utilizan para combinar conjuntamente las características de los componentes con su posibilidad de tomar decisiones frente a la presencia de varios parámetros que se les ofrecen.

Tales cálculos que, como se advierte, pueden combinarse entre sí y dar por resultado una extraordinaria cantidad de variantes, deben ser procesadas por una minúscula computadora que

determine con suma precisión la cantidad de gasolina que en todo momento resulte más conveniente aportar para todo el conjunto de condiciones dadas.

Con la información proporcionada por el procesador el conductor podrá deducir si su velocidad resulta la más económica, es decir, si está gastando más gasolina de la que sería preciso, o si la misma velocidad la podríamos regular de tal modo que consumiendo lo mismo hiciéramos a su vez mayor cantidad de metros en el mismo periodo de tiempo.

A continuación, presentamos algunos de los procesadores de viaje de los que van provistos algunos importantes modelos de automóviles, y las funciones que con ellos se pueden realizar:

Alfa Romeo.

Modelos: Alfetta y Quadrifoglio.

Funciones: Reloj, duración del trayecto, consumo instantáneo, consumo medio, consumo total, velocidad promedio.

B.M.W.

Modelos: 745i, 635CSi, 725i, 528i, 728i, 735i, etc.

Funciones: Reloj, cronómetro, avisador, temperatura exterior, consumo instantáneo, consumo medio, aviso antes de superar el consumo programado, autonomía, distancia hasta el lugar de destino, tiempo para la llegada, velocidad de promedio, aviso antes de rebasar la velocidad programada, seguro antirrobo.

FORD.

Modelos: Granada GL, Granada Ghia, Sierra Ghia, Granada L, Granada N, Sierra GL.

Funciones: Reloj, avisador, indicación de la fecha, duración del trayecto, horas de servicio del motor, consumo instantáneo, consumo medio, consumo total, autonomía, trayecto recorrido, velocidad media, aviso antes de rebasar la velocidad programada.

LEYLAND.

Modelos: Jaguar XJ 5.3, Vanden Plas, Jaguar XJ 4.2, Sovereigh.

Funciones: Reloj, duración del trayecto, consumo instantáneo, consumo medio, autonomía, velocidad promedio.

RENAULT.

Modelos: R-11 Electronic.

Funciones: Sistema de aviso electrónico con voz sintetizada de la computadora.

Modelos: R-18, R-14.

Funciones: Reloj, temperatura exterior, carburante restante en el depósito, autonomía, consumo medio, consumo instantáneo, velocidad promedio, distancia recorrida.

Modelo: R-5.

Funciones: Reloj, consumo instantáneo, consumo medio, consumo total, velocidad promedio, tiempo transcurrido después de la salida.

VOLKSWAGEN.

Modelos: Golf GTI, Scirocco GTI.

Funciones: Reloj, duración del trayecto, temperatura exterior, temperatura del aceite, consumo medio, trayecto recorrido, velocidad promedio.

1.5.2. Implementación de la Computadora de Viaje.

Desde un punto de vista electrónico, las computadoras de viaje o procesadores de viaje son aparatos que trabajan combinando, de una manera que se ha establecido de antemano en su memoria permanente, los datos de:

- a) consumo de combustible,
- b) velocidad del vehículo,
- c) tiempo.

Estos datos son suficientes para llevar a cabo las funciones de información así como la exactitud de los datos proporcionados depende de la perfección de los sensores (o captadores) y de la propia computadora.

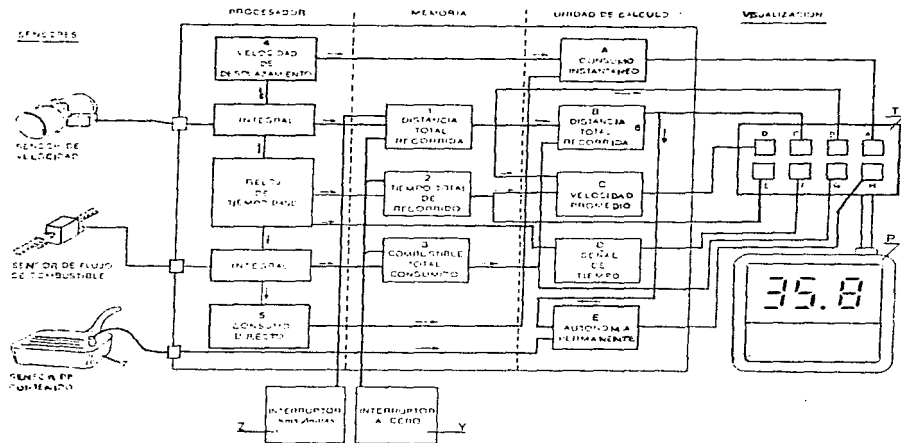


Figura 1.5.2.a. Esquema de bloques de la constitución interior de una computadora de viaje

En la figura 1.5.2.a, se tiene un esquema que reproduce en líneas generales todo el funcionamiento del conjunto de una de estas computadoras y de sus órganos anexos. Tenemos señalado en este dibujo, y en primer lugar, la zona de los sensores. El primero señalado es el sensor de velocidad, por medio del cual se toma el dato de los kilómetros recorridos y de la

velocidad a que se desplaza el vehículo. Se encuentra el sensor correspondiente al flujo de combustible que consiste en un medidor de flujo por medio del cual se establecen señales eléctricas proporcionales a la cantidad de combustible que pasa por el conducto de alimentación del carburador.

El tercer sensor, que no es general en todas las computadoras es el que controla el contenido total del depósito de combustible.

Con estos tres datos exteriores, en el caso presentado en la figura 1.5.2.a citada, tenemos la situación de los sensores principales que se encuentran colocados en el motor, la transmisión y el depósito de gasolina. Veremos enseguida la forma cómo actúan estos sensores en la unidad electrónica.

La unidad electrónica consta de las cuatro partes principales siguientes:

- a) Procesador.
- b) Memoria.
- c) Unidad de cálculo.
- d) Unidad de visualización.

El procesador dispone ante todo de un reloj muy exacto, del tipo de oscilador de cuarzo, el cual constituye el bloque de "tiempo base" que se aprecia en la figura. Cada uno de los impulsos eléctricos enviados por los sensores de velocidad y flujo de combustible deben ser procesados en esta unidad por los grupos integrales dando como resultado unos valores que pasan a la memoria en unos casos, y en otros directamente a la unidad de cálculo.

A este respecto es conveniente fijarse en la dirección de las flechas colocadas en los circuitos o "bus" en los que intervienen estos valores. Siendo cinco valores básicos, tres de los cuales quedan permanentemente disponibles en la memoria y cuyos valores son:

1. Distancia total recorrida.
2. Tiempo total de recorrido.
3. Combustible total consumido.

Los datos que se suministran a la unidad de cálculo de una manera directa desde la misma unidad de proceso son los siguientes:

4. La velocidad de desplazamiento.
5. El consumo directo.

La unidad del cálculo dispone de información permanente que procede de los puntos que se han dicho y que pueden verse en la figura 1.5.2.a, de arriba a abajo, de la siguiente forma:

- A. Consumo instantáneo. Es el resultado de dividir la velocidad instantánea de desplazamiento (4) por el flujo de combustible (5) que se está produciendo en el mismo momento.
- B. Consumo promedio. Es el resultado de dividir el combustible total consumido (3) por la distancia total recorrida (1) lo que da como resultado el consumo promedio que podremos considerar con el número 6 de las funciones (pues la computadora podrá elaborar este dato en la unidad de cálculo).
- C. Velocidad promedio. La unidad de cálculo se nutre aquí de los datos proporcionados por el bloque de memoria de la "distancia total recorrida" (1) dividida por el "tiempo total de recorrido" (2) con lo que en este bloque se tiene permanente información de la velocidad promedio a que se está realizando el viaje.
- D. Señal de tiempo. La unidad de tiempo base del procesador se transmite directamente a este bloque dando las señales correspondientes a tiempo real.
- E. Autonomía permanente. de acuerdo con el volumen de combustible o contenido total del depósito (7) y el consumo medio (6) que se está realizando hasta el momento, traducida a

kilómetros que pueden ser recorridos con el resto del combustible que va quedando en el depósito.

Todos estos datos se pueden visualizar en una pantalla (P) según la posición de un conductor (T) que pone en contacto cada uno de los terminales de cada circuito con la pantalla. En el caso concreto de una computadora de viaje como el esbozado en este esquema de bloques tendríamos que la información proporcionada por cada uno de los ocho contactos, sería el siguiente:

- A. consumo instantáneo,
- B. distancia total recorrida,
- C. consumo promedio,
- D. velocidad de promedio,
- E. tiempo total de recorrido,
- F. información del tiempo real,
- G. combustible total consumido hasta el momento,
- H. autonomía con el combustible de que se dispone en el depósito.

También cabe destacar la presencia de los mandos auxiliares Y y Z. El primero es el interruptor que pone a cero las memorias para poder iniciar el viaje con todos los parámetros limpios a cero. En cuanto al mando Z es un interruptor por medio del cual se pasa la información al circuito conveniente para que se haga la transformación de las unidades de medición en kilómetros o millas, para ser utilizado el aparato indistintamente en países anglosajones o del continente.

1.5.3 Cajas Negras (Registro de Datos de Vuelo).

En los procedimientos de vuelo se ha incluido siempre el registro de algunos de los parámetros medidos por los instrumentos y de cualquier incidente anormal que tenga relación con los motores, y las funciones del avión en general. El método más primitivo, y que aún sigue en vigor, consiste en efectuar las anotaciones correspondientes en el libro o cuaderno de registro correspondiente del avión. Sin embargo, según los sistemas se volvían más sofisticados en su

modo de funcionar, el número de parámetros que hay que vigilar aumentaba, y esto imponía limitaciones al método de registro mediante anotaciones en el libro del avión. Además, la cuestión de cómo se podían recuperar datos de importancia para la investigación de accidentes empezó a tomarse cada vez más en cuenta. Por este motivo, se declaró obligatorio (en 1958 en los EE.UU. y en 1965 en Gran Bretaña) equipar a los aviones de determinadas categorías con un sistema automático de registro de vuelo (caja negra), que comprendía un registro de datos de vuelo y un registro de conversaciones en cabina. El número de parámetros que había que registrar aumentó progresivamente, a partir del grupo básico común establecido, de acuerdo con las clases de aviones agrupados por pesos y, por tanto, de acuerdo con el grado de complejidad de aviones y sistemas. Como la mayoría de los parámetros se miden derivando una señal de los elementos sensores o de los transductores de los sistemas de instrumentos ya existentes en los aviones, con arreglo al número de instrumentos empleados en el avión, virtualmente no hay límite para el número de parámetros que se pueden registrar. El valor relativo de los registros de datos de vuelo (cajas negras), según la experiencia de los investigadores, se ha indicado con respecto a dos categorías generales de accidentes. La primera categoría es aquella en la cual algún fallo del avión o sus sistemas ha sido la causa principal del accidente. Al tratar de establecer cual fue el fallo primario, el análisis de los restos del accidente constituye la principal fuente de información, apoyada por el conocimiento de los datos registrados antes del accidente, como tiempo, altitud, velocidad y aceleración.

La segunda categoría de accidentes está constituida por aquellos en los cuales la causa es de tipo operativo, es decir, aquellos en los cuales no ha tenido lugar un fallo de ingeniería ni una degradación de las prestaciones del avión. Como ejemplo se pueden citar la pérdida de control debido al mal manejo del avión por la tripulación, las malas condiciones atmosféricas, o diversos factores de navegación. En estos casos, el papel del análisis de los restos del accidente se limitará a establecer los datos concernientes a la configuración del avión en el momento del impacto, y a obtener una idea general de la altitud y velocidad del avión en ese momento. En estos casos, el papel del registrador de datos de vuelo será más importante, porque proporcionará una historia cronológica de las principales maniobras de vuelo.

1.5.3.1. Funciones primarias y de interfaz de las computadoras en los aviones.

- Referencias y navegación inerciales
- Aviso de proximidad al suelo (GPWS)
- Control automático de vuelo (AFCS)
- Computar datos de aire o manométricos
- Control del motor y presentaciones de datos de vigilancia
- Presentación del sistema de instrumentos de vuelo
- Presentación de estado de sistemas (status)
- Registro y adquisición de datos de vuelo
- Alertar a la tripulación
- Control ambiental
- Navegación radioeléctrica y control del tráfico aéreo (ATC)
- Presentación de control de mantenimiento y datos asociados

Una computadora digital de datos de vuelo es, básicamente un ordenador que procesa las señales eléctricas, correspondientes a la altitud, velocidad y régimen de ascenso y descenso, computando las presiones de aire en los tubos de pitot y sondas estáticas montadas al exterior del avión; de ahí el término de datos de aire, o datos manométricos, que se emplean frecuentemente. Las señales eléctricas las producen unos transductores de presión electromecánica, situados dentro de la computadora, que están conectados adecuadamente a los tubos de pitot y sondas estáticas. Las señales de salida que se computan finalmente, se transmiten entonces a altímetros accionados mediante servos, anemómetros indicadores de Mach y variómetros del panel de instrumentos de la cabina de los pilotos, así como a todos los sistemas que dependen de los datos de aire para su funcionamiento.

1.6 Teoría de las bases de datos relacionales.

1.6.1 Tipos de Enfoques de las Bases de Datos

Existen enfoques alternativos para visualizar y manejar datos a un nivel lógico independientemente de la estructura física de soporte en la que se basen.

Los enfoques de base de datos que existen son los siguientes:

- **Modelo Jerárquico**
- **Modelo de Red**
- **Modelo Relacional**

El enfoque de base de datos de interés para nuestro estudio, es el **Modelo Relacional**, por lo cual es el que a continuación se desarrolla y explica en detalle:

1.6.1.1 Modelo Relacional

Este modelo está basado en el concepto matemático de la relación. El modelo fue propuesto por primera vez por E. F. Codd en 1970 y ha tenido una gran influencia y un gran valor. Se utiliza, exclusivamente, en el contexto de los sistemas de gestión de bases de datos. Con frecuencia, aunque no necesariamente, se expresa en función de formas normales.

1.6.2 Operaciones de Almacenamiento y Recuperación de Información

En la sociedad moderna la tarea del almacenamiento y recuperación de vastas cantidades de información ha sido absorbida casi por completo por sistemas de cómputo. No paso mucho tiempo, para que una herramienta utilizada exclusivamente en el área de las matemáticas, pasará en la actualidad, a ser utilizada para gestionar grandes colecciones de información llamadas bases de datos. Agencias de gobierno de los Estados Unidos como la Internal Revenue Service, la

Social Security Administration, y el National Crime Information Center, mantienen bases de datos, así como la industria privada (registros de personal) y otras organizaciones (registros médicos, registros de crédito). El acceso de las computadoras a esta información almacenada plantea varias preguntas de gran importancia para la sociedad moderna:

- ¿Qué tan fiel es esta información, y cómo pueden ser corregidos estos errores?
- ¿Quién tiene acceso a la información?
- ¿Cómo es prevenido el acceso inadecuado o ilegal?

Estas preguntas se encuentran actualmente bajo estudio por investigadores de la informática, legisladores, y por aquellos con algún interés en las bases de datos.

Las operaciones de almacenamiento y recuperación se dividen en dos clases:

1. Operaciones de Conjuntos
2. Operaciones Relacionales

1.6.2.1 Operaciones de Conjuntos

Las operaciones estándar de conjuntos unión, intersección, y diferencia permiten manipular información de la base de datos. Estas tres funciones permiten utilizar sentencias **SELECT** (SELECCIÓN) para verificar la integridad de una base de datos después de que se ha efectuado una actualización (**UPDATE**), una inserción (**INSERT**), o un borrado (**DELETE**). Pueden ser útiles cuando se están transfiriendo datos a una tabla histórica, por ejemplo, y se quiere verificar que los datos correctos estén en la tabla histórica antes de borrarlos de la tabla original.

Las Operaciones de Conjuntos son las siguientes:

- Unión
- Intersección
- Diferencia
- Producto

Cada una de ellas se describe y ejemplifica en los siguientes subíndices:

1.6.2.1.1. Unión

La operación UNION, a través del operador \cup (unión), es considerada como una operación de conjuntos. Esta operación es conmutativa y asociativa. Simbólicamente se representa de la siguiente manera:

$$S \cup T = \{x | x \in S \vee x \in T\}$$

El operador UNIÓN acepta como entrada dos tablas con las mismas columnas en el mismo orden, y produce como resultado todas las columnas y todos los renglones de ambas tablas. Si existe algún renglón con la misma información en ambas tablas, en la tabla que se genera como resultado de aplicar el operador \cup , ese renglón aparecerá una sola vez. A continuación se muestra un ejemplo de unión entre dos tablas, en las figuras 1.6.2.1.1.a, 1.6.2.1.1.b y 1.6.2.1.1.c:

EMPLEADOS_ANTIGUOS

Número_de_Empleado	Nombre_de_Empleado	Sueldo
1	Elizabeth Arciniega E.	12,000
2	Francisco E. Fonseca C.	9,500
3	Elena Patton Anton	10,000

Figura 1.6.2.1.1.a Tabla EMPLEADOS_ANTIGUOS

EMPLEADOS_NUEVOS

Número_de_Empleado	Nombre_del_Empleado	Sueldo
3	Elena Patton Anton	10,000
4	Luis Colorado Martínez	9,500
5	Patricia Sánchez Calderón.	10,000

Figura 1.6.2.1.1.b Tabla EMPLEADOS_NUEVOS.

La operación UNIÓN (\cup) entre las dos tablas anteriores da como resultado la siguiente tabla:

UNIÓN

Número de Empleado	Nombre del Empleado	Sueldo
1	Elizabeth Arciniega E.	12,000
2	Francisco E. Fonseca C.	9,500
3	Elena Patton Anton	10,000
4	Luis Colorado Martínez	9,500
5	Patricia Sánchez Calderón.	10,000

Figura 1.6.2.1.1 c Tabla UNIÓN

1.6.2.1.2 Intersección

La operación INTERSECCIÓN de Conjuntos da como resultado un conjunto que resulta de combinar elementos comunes a dos conjuntos S y T, por ejemplo, generalmente expresados como:

$$S \cap T$$

La intersección, a través del operador \cap (INTERSECCIÓN), es considerada como una operación de conjuntos, conmutativa y asociativa. Simbólicamente se representa de la siguiente manera:

$$S \cap T = \{x \mid x \in S \text{ y } x \in T\}$$

El operador de intersección selecciona de ambas tablas los renglones que posean exactamente la misma información en todas las columnas. A continuación se muestra un ejemplo de intersección entre dos tablas, en las figuras 1.6.2.1.2.a, 1.6.2.1.2.b y 1.6.2.1.2.c:

EMPLEADOS_ANTIGUOS

Número de Empleado	Nombre de Empleado	Sueldo
1	Elizabeth Arciniega E.	12,000
2	Francisco E. Fonseca C.	9,500
3	Elena Patton Anton	10,000

Figura 1.6.2.1.2.a: Tabla EMPLEADOS_ANTIGUOS

EMPLEADOS_NUEVOS

Número_de_Empleado	Nombre_del_Empleado	Sueldo
3	Elena Patton Anton	10,000
4	Luis Colorado Martinez	9,500
5	Patricia Sánchez Calderón.	10,000

Figura 1.6.2.1.2.b Tabla EMPLEADOS_NUEVOS.

La operación INTERSECCIÓN (\cap) entre las dos tablas anteriores da como resultado la siguiente tabla:

INTERSECCIÓN

Número_de_Empleado	Nombre_del_Empleado	Sueldo
3	Elena Patton Anton	10,000

Figura 1.6.2.1.2.c Tabla INTERSECCIÓN.

1.6.2.1.3 Diferencia

La DIFERENCIA de Conjuntos es una operación entre dos conjuntos S y T, por ejemplo, dando como resultado el conjunto S-T que consta de los elementos que están en S pero no en T. Formalmente se representa de la siguiente manera:

$$S - T = \{x | x \in S \text{ y } x \notin T\}$$

La diferencia de conjuntos es una generalización del concepto del complementario de un conjunto y, como tal, se le llama, a veces, complementario relativo de T con respecto a S.

DEPARTAMENTO

Clave_de_Depto.	Nombre_del_Depto.
RHM	Recursos Humanos
ADM	Administración
MER	Mercadotecnia
DIS	Diseño
SIS	Sistemas

Figura 1.6.2.1.3.a Tabla DEPARTAMENTO.

EMPLEADO

No_Empleado	Nombre_del_Empleado	Sueldo	Clave_de_Depto.
1	Elizabeth Arciniega E.	12,000	RHM
2	Francisco E. Fonseca C.		ADM
3	Elena Patton Anton.	10,000	
4	Luis Colorado Martínez	9,500	
5	Patricia Sánchez Calderón	10,000	ADM

Figura 1.6.2.1.3 b Tabla EMPLEADO.

El operador de diferencia (-) acepta como entrada dos tablas que tengan al menos una columna en común, en donde la tabla resultante tendrá todas las columnas de la primer tabla y los renglones que no aparezcan en la segunda tabla. A continuación se muestra un ejemplo de diferencia entre dos tablas, en las figuras 1.6.2.1.3.a, 1.6.2.1.3.b y 1.6.2.1.3.c:

La operación DIFERENCIA (-) entre las dos tablas anteriores da como resultado la siguiente tabla:

DIFERENCIA

Clave_de_Depto.	Nombre_del_Depto.
MER	Mercadotecnia
DIS	Diseño
SIS	Sistemas

Figura 1.6.2.1.3.c Tabla DIFERENCIA.

1.6.2.1.4 Producto

El PRODUCTO de conjuntos es mejor conocido como Producto Cartesiano. El Producto Cartesiano, de dos conjuntos S y T, es el conjunto de todos los pares ordenados de la forma (s, t), con la propiedad de que s es un miembro de S y t es un miembro de T. Esto se escribe generalmente, como S X T. Formalmente se expresa de la siguiente manera:

$$S \times T = \{(s, t) | (s \in S) \vee (t \in T)\}$$

Cuando se efectúa una consulta multi-tabla, que no establece explícitamente una condición que determina cuales renglones en las tablas serán recuperados, se crea un Producto Cartesiano. Un Producto Cartesiano consiste en cualquier posible combinación de renglones de las tablas. Este es usualmente un conjunto de resultados muy grande, difícil de manejar, y además los datos obtenidos, raramente serán útiles y confiables. A continuación se muestra un ejemplo de producto entre dos tablas, en las figuras 1.6.2.1.4.a, 1.6.2.1.4.b y 1.6.2.1.4.c:

EMPLEADO

No_Empleado	Nombre_del_Empleado	Sueldo	Clave_de_Depto.
1	Elizabeth Arciniega E.	12,000	RHM
2	Francisco Fonseca C.		ADM

Figura 1.6.2.1.4 a Tabla EMPLEADO.

DEPARTAMENTO

Clave_de_Depto.	Nombre_Depto.
RHM	Recursos Humanos
ADM	Administración
MÉR	Mercadotecnia
DIS	Diseño

Figura 1.6.2.1.4 b Tabla DEPARTAMENTO.

PRODUCTO_CARTESIANO

No_Emp.	Nombre_Empleado	Sueldo	Clave_Depto.	Nombre_Depto.
1	Elizabeth Arciniega E.	12,000	RHM	Recursos Humanos
1	Elizabeth Arciniega E.	12,000	RHM	Administración
1	Elizabeth Arciniega E.	12,000	RHM	Mercadotecnia
1	Elizabeth Arciniega E.	12,000	RHM	Diseño
2	Francisco Fonseca C.		ADM	Recursos Humanos
2	Francisco Fonseca C.		ADM	Administración
2	Francisco Fonseca C.		ADM	Mercadotecnia
2	Francisco Fonseca C.		ADM	Diseño

Figura 1.6.2.1.4 c Tabla PRODUCTO_CARTESIANO

Aunque existen sólo 2 renglones en la tabla EMPLEADO y 4 en la tabla DEPARTAMENTO, el efecto de esta operación es el multiplicar los renglones de una tabla por los de la otra, y recuperar así 8 renglones:

Nótese que algunos de los datos desplegados en los renglones concatenados son erróneos. Por ejemplo, las columnas No_Empleado y Clave_de_Empleado de la tabla EMPLEADO, indican el departamento al que pertenece un cierto empleado, en tanto, el valor de la columna Nombre_del_Depto. de la tabla DEPARTAMENTO podría ser para una clave de departamento diferente (sólo el primero y el sexto registro del producto Cartesiano son correctos).

1.6.2.2 Operaciones Relacionales

Una Operación Relacional involucra la manipulación de una o más tablas o relaciones para obtener como resultado otra tabla.

Las Operaciones de Relación son las siguientes:

1. Selección
2. Proyección
3. Join

Cada una de ellas se describe y ejemplifica en los siguientes subincisos:

1.6.2.2.1 Selección

En la terminología relacional, la selección está definida como la obtención de un subconjunto horizontal de renglones de una sola tabla que satisface una condición en particular.

El operador de selección acepta una sola tabla como entrada, y produce como resultado todas las columnas que contiene la tabla de entrada y los renglones que sean especificados por el usuario.

Las condiciones de selección pueden ser de varios grados de complejidad y pueden incluir a los operadores lógicos AND, OR y NOT (pueden ser utilizados paréntesis para indicar precedencia de operación).

Las comparaciones pueden realizarse con valores literales, valores contenidos en las columnas, o expresiones matemáticas que involucren valores literales de las columnas. A continuación se muestra un ejemplo de la selección, en las figuras 1.6.2.2.1.a y 1.6.2.2.1.b:

EMPLEADO

No_Empleado	Nombre_del_Empleado	Sueldo	Clave_de_Depto.
1	Elizabeth Arciniega E.	12,000	RHM
2	Francisco E. Fonseca C.		ADM
3	Elena Patton Anton.	10,000	
4	Luis Colorado Martínez	9,500	
5	Patricia Sánchez Calderón	10,000	ADM

Figura 1.6.2.2.1.a Tabla EMPLEADO

La SELECCIÓN de todos los empleados que laboran en el departamento de Administración produce el siguiente resultado:

SELECCIÓN

No_Empleado	Nombre_del_Empleado	Sueldo	Clave_de_Depto.
2	Francisco E. Fonseca C.		ADM
5	Patricia Sánchez Calderón	10,000	ADM

Figura 1.6.2.2.1.b Tabla SELECCIÓN.

1.6.2.2.2 Proyección

En la terminología relacional, la Proyección está definida como la obtención de un subconjunto vertical de las columnas de una sola tabla que retiene los renglones únicos.

EMPLEADO

No_Empleado	Nombre_del_Empleado	Sueldo	Clave_de_Depto.
1	Elizabeth Arciniega E.	12,000	RHM
2	Francisco E. Fonseca C.		ADM
3	Elena Patton Anton.	10,000	
4	Luis Colorado Martínez	9,500	
5	Patricia Sánchez Calderón	10,000	ADM

Figura 1.6.2.2.2.a. Tabla EMPLEADO.

La PROYECCIÓN de las columnas No. Empleado y Nombre del Empleado, produce el siguiente resultado:

PROYECCIÓN

No_Empleado	Nombre_del_Empleado
1	Elizabeth Arciniega E.
2	Francisco E. Fonseca C.
3	Elena Patton Anton.
4	Luis Colorado Martínez
5	Patricia Sánchez Calderón

Figura 1.6.2.2.2.b Tabla PROYECCIÓN.

El operador de proyección recibe como entrada una tabla, y produce como resultado sólo aquellas

columnas especificadas por el usuario y todos los renglones de la tabla. El orden en el cual aparecen las columnas, es el que se indica cuando se hace la proyección. El número de columnas que se pueden proyectar es como máximo el mismo número de columnas de la tabla y como mínimo una sola columna. A continuación se muestra un ejemplo de la proyección, en las figuras 1.6.2.2.2.a y 1.6.2.2.2.b:

1.6.2.2.3 Join

Un join ocurre cuando una o más tablas son conectadas por medio de una o más columnas en común, creando una nueva tabla de resultados.

Conceptualmente, la primera etapa de un join es la creación de un producto Cartesiano. Para refinar o restringir este producto Cartesiano y eliminar renglones de datos que carezcan de sentido, se incluye una cláusula WHERE con una condición válida en una sentencia SELECT de Structured Query Language (Lenguaje Estructurado de Consultas)

El operador JOIN acepta como entrada dos o más tablas, teniendo cada una al menos una columna en común con las otras tablas, y produce como resultado a todas las columnas de las tablas de entrada, y los renglones se concatenan con aquellos renglones cuyos valores en las tablas de entrada cumplen con la condición que indica el usuario para hacer el Join.

DEPARTAMENTO

Clave_de_Depto.	Nombre_Depto.
RHM	Recursos Humanos
ADM	Administración
MER	Mercadotecnia
DIS	Diseño

Figura 1.6.2.2.3.a Tabla DEPARTAMENTO.

EMPLEADO

No._Empleado	Nombre_del_Empleado	Sueldo	Clave_de_Depto.
1	Elizabeth Arciniega E.	12,000	RHM
2	Francisco E. Fonseca C.		ADM
3	Elena Patton Anton.	10,000	
4	Luis Colorado Martinez	9,500	
5	Patricia Sánchez Calderón	10,000	ADM

Figura 1.6.2.2.3 b Tabla EMPLEADO.

El JOIN efectuado con la columna en común (Código_de_Departamento), muestra el resultado siguiente:

JOIN

Clave_de_Depto.	Nombre_Depto.	No._Emp.	Nombre_del_Emp.	Sueldo
RHM	Recursos Humanos	1	Elizabeth	12,000
ADM	Administración	2	Francisco	
ADM	Administración	5	Patricia	10,000

Figura 1.6.2.2.3 c Tabla JOIN.

Los operadores de relación para indicar las condiciones del join pueden ser >, <, =, !=. Las columnas en común sólo se muestran una vez. En la figura 1.6., se muestra un ejemplo de la operación join. A continuación se muestra un ejemplo de la proyección, en las figuras 1.6.2.2.3.a, 1.6.2.2.3.b y 1.6.2.2.3.c:

1.6.3 Características de los Sistemas Manejadores de Bases de Datos Relacionales

- Representación de datos a través de tablas.
- Desarrollo de aplicaciones a través de herramientas de alta productividad.
- Flexibilidad en el mantenimiento de las estructuras y de los datos, así como en el tipo de consultas.
- Diccionario de Datos integrado.
- Soporte a todos los operadores Relacionales.

1.6.4 Ventajas de los Sistemas Manejadores de Bases de Datos Relacionales

- Simplicidad
 - ⇒ Son fáciles de usar.
 - ⇒ Es fácil realizar inserciones y actualizaciones de datos.
 - ⇒ Es sencillo cambiar la estructura de los datos.
 - ⇒ La navegación es responsabilidad del RDBMS, no del programador.

1.6.5 Poder de los Sistemas Manejadores de Bases de Datos Relacionales

- Prácticamente todas las consultas son posibles de realizar.

1.6.6 Diccionario de Datos Dinámico basado en el Modelo Relacional

La descripción de la base de datos es representada dinámicamente, al nivel lógico, como datos ordinarios, de tal forma que los usuarios autorizados puedan aplicar el mismo lenguaje relacional para consultarla.

1.6.6.1 Lenguaje de Datos Comprensible

No importa cuantos lenguajes y modos interactivos se soporten, por lo menos un lenguaje debe ser soportado, con una sintaxis bien definida, que soporte interactivamente y por programa lo siguiente:

- Definición de Datos
- Manipulación de Datos
- Control de Transacciones
- Reglas de Integridad
- Vistas
- Reglas de Autorización

1.6.7 Modelado Entidad-Relación

El modelado Entidad-Relación es una técnica usada desde hace un número de años para ilustrar gráficamente datos y las relaciones existentes entre estos. Esta técnica es un componente del modelado de datos y aunque no está restringido a ningún paradigma en especial de base de datos, este es muy efectivo para estructuras de datos que se apegan a el modelo relacional. Desafortunadamente, hasta muy recientemente, el modelado Entidad-Relación solo había sido utilizado casi por completo en el modelado de bases de datos en ambientes mainframe y de rango medio. Con la reciente disponibilidad de herramientas CASE de bajo costo para plataformas PC, esta muy efectiva técnica de análisis y diseño es ahora mucho más fácilmente disponible para los desarrolladores.

Existen varias versiones del modelado Entidad-Relación incorporando símbolos un tanto diferentes, pero la mayoría de ellas expresan los mismos principios generales. Muchas de las herramientas CASE basadas en PC soportan ahora diagramas de Entidad-Relación, y el ejemplo de diseño presentado aquí, puede ser fácilmente implantado utilizando muchas o la gran mayoría de estas herramientas.

Hay muchos usos comunes para los diagramas de Entidad-Relación (ERD), incluyendo la representación gráfica de los requerimientos de los datos y soporte en la construcción de

aplicaciones. Dependiendo de la herramienta CASE utilizada, los diagramas de Entidad-Relación pueden asistir también en el proceso de encapsulación de métodos con datos en objetos de datos. Adicionalmente, algunas herramientas CASE soportan la integración del modelo de datos Entidad-Relación a ambientes cliente servidor, y la generación de un esquema de base de datos para la creación física de la base de datos.

1.6.7.1 Entidad

En primer lugar, una entidad es un conjunto de cosas reales o abstractas (una persona, lugar, cosa, evento o transacción basada en el tiempo), acerca de la cual es importante almacenar información. Aunque este no es siempre el caso, una entidad puede a menudo ser pensada como una tabla de una base de datos o como un archivo. Las entidades son representadas generalmente dentro de un diagrama Entidad-Relación como rectángulos o como cajas con las esquinas redondeadas. De acuerdo a ciertas convenciones, el nombre de la entidad se escribe dentro del símbolo de entidad en singular, a un nivel de **Entity Display Level** (Despliegue a Nivel de Entidad), tal y como se muestra en la figura 1.6.7.1.a. Cada entidad debe tener un nombre único dentro del sistema, lo que indica que no pueden existir dos entidades dentro del mismo sistema con el mismo nombre.



EMPLEADO

Figura 1.6.7.1.a Despliegue de una Entidad a Nivel de Entidad.

La entidad dentro de un modelo relacional se representa por medio de una tabla, en donde el nombre de la tabla corresponde al nombre de la entidad. Además, cada tabla debe contener una columna que identifique de forma única a cada renglón de ésta. Esta columna recibe el nombre de llave primaria (PK), la cual no puede contener valores nulos, ni duplicados. Lo anterior se muestra en las figuras 1.6.7.1.b y 1.6.7.1.c:

PRODUCTO

Clave_de_Depto.
PK
ADM
MER
DIS

Figura 1.6.7.1 b Entidad como una Tabla (a).

EMPLEADO

No._Empleado
PK
1
2
3
4

Figura 1.6.7.1 c Entidad como una Tabla (b).

1.6.7.2 Atributo

Un atributo es información individual acerca de una entidad. Representa un tipo de característica, cualidad o propiedad asociada con un conjunto de cosas reales o abstractas (gente, lugares, eventos, etc.). Aunque los atributos normalmente no son representados en los diagramas Entidad-Relación, estos son definidos como información detallada dentro de las entidades. Los atributos en el modelo relacional se representan como columnas o campos en una tabla, tal y como se muestra en la figura 1.6.7.2.a:

PRODUCTO

Código_del_Producto	Descripción
PK	
100860	Diskettes
100861	Discos Duros
100870	Scanners
100899	Impresoras

Figura 1.6.7.2.a Atributos representados como columnas o campos en una tabla.

1.6.7.3 Relación

Una relación es una liga semántica entre entidades, o entre una entidad y sí misma. Las relaciones pueden ser puntos comunes de referencia, asociaciones, conexiones, indicaciones de propiedad, o semejanzas directas entre entidades. Las relaciones están representadas en el diagrama de Entidad-Relación como líneas conectando entidades, estas relaciones deben caer dentro del alcance del sistema, es decir, acerca de estas, el sistema deberá mantener, correlacionar y desplegar información. Una relación es un objeto "lógico" que es representado como uno o varios atributos llave foránea.

1.6.7.4 Nombres en Singular

Los nombres de entidades siempre son en singular. El hacerlo de esta manera facilita la interpretación del modelo con sentencias declarativas como "UN VUELO <Transporta> cero o más PASAJEROS" y "UN PASAJERO <es transportado por> un VUELO". Cuando nombramos el conjunto de instancias representadas por la entidad, estamos proporcionando un nombre a una instancia. Cada instancia de la entidad PASAJERO es un pasajero individual, no un conjunto de "pasajeros".

Los nombres de atributos son en singular también, p.ej., "Nombre_de_Persona",

RFC_del_Empleado”, “Aumento_en_Bono_del_Empleado”. La razón para nombrar atributos en singular es en parte cuestión de una política para nombrar de manera consistente, y en parte para evitar ciertas clases de errores de normalización. Se podría tener la sensación intuitiva de que algo está incorrecto si se ve un nombre de atributo como “Hobbies_de_la_Persona”. ¿Cuántos? ¿Como distinguimos uno de otro? ¿Cómo los representaríamos en una tabla de instancias de ejemplo? Y así sucesivamente. La sensación de que algo está errado es correcta.

1.6.7.5 Ocurrencia

Una ocurrencia es una instancia específica de una entidad. Una ocurrencia de una entidad se refiere a menudo a un registro o renglón de una tabla.

1.6.7.6 Cardinalidad

La cardinalidad es un aspecto de una relación que indica si una ocurrencia en una entidad se relaciona con sólo una ocurrencia en otra entidad, o con más de una ocurrencia en esa entidad. Los tipos más comunes de cardinalidad utilizados en un diagrama de Entidad-Relación son los siguientes:

- Relaciones Uno a Uno (1:1)
- Relaciones Uno a Muchos o Muchos a Uno (1:M) o (M:1)
- Relaciones Muchos a Muchos (M:M)

Ciertos estándares indican que una cardinalidad definida en el lado “uno” de una relación sea representada como una sola línea sólida o una línea sólida con una línea vertical unida al símbolo de entidad. Una cardinalidad definida en el lado “muchos” de una relación es representada como una línea múltiple, ramificación, punta de flecha o círculo unida al símbolo de entidad.

1.6.7.6.1 Relaciones Uno a Uno

Las relaciones uno a uno dentro del modelo relacional se representan dibujando la llave primaria (PK) de la tabla con más renglones como llave foránea (FK) en la otra tabla, tal y como se muestra en la figuras 1.6.7.6.1.a y 1.6.7.6.1.b:

PRODUCTO

Código_del_Producto
PK
100860
100861
100870
100899

Figura 1.6.7.6.1.a Tabla PRODUCTO.

PRECIO

Clave_de_Precio	Número_de_Producto
PK	FK, ND
1	100870
2	100876
3	100101
4	100860

Figura 1.6.7.6.1.b Tabla PRECIO

1.6.7.6.2 Relaciones Uno a Muchos o Muchos a Uno

Las relaciones uno a muchos o muchos a uno entre dos entidades se modelan dibujando la llave primaria (PK) de la tabla que tiene la correspondencia de uno como llave foránea (FK) en la

otra tabla, tal y como se muestra en las figuras 1.6.7.6.2.a y 1.6.7.6.2.b:

CLIENTE

Número_de_Cliente
PK
100
101
102
103

Figura 1.6.7.6.2.a Tabla CLIENTE.

ORDEN

Número_de_Orden	Número_de_Cliente
PK	FK
600	101
601	101
602	102
603	
604	103

Figura 1.6.7.6.2.a Tabla ORDEN.

1.6.7.6.3 Relaciones Muchos a Muchos

Las relaciones muchos a muchos entre dos entidades se modelan dibujando una tercer tabla, la cual se compone de una llave primaria (PK) compuesta de dos columnas, las cuales son llaves primarias y foráneas a la vez, tal y como se muestra en las figuras 1.6.7.6.3.a, 1.6.7.6.3.b y 1.6.7.6.3.c:

ORDEN

Número_de_Orden
PK
600
602
603
604

Figura 1.6.7.6.3 a Tabla ORDEN.

PRODUCTO

Código_del_Producto
PK
100860
100861
100870
100899

Figura 1.6.7.6.3 b Tabla PRODUCTO.

ORDEN/PRODUCTO

Número_de_Orden	Código_del_Producto
PK	
FK	FK
600	100860
601	100861
603	100899
604	101860

Figura 1.6.7.6.3 c Tabla ORDEN/PRODUCTO.

1.6.8 Escenario de diseño de un diagrama Entidad-Relación

Hay un número de tareas que deben ser llevadas a cabo durante el desarrollo de un diagrama Entidad-Relación y un modelo de datos completamente normalizado.

Estas tareas incluyen las siguientes:

- Identificación y definición de entidades
- Identificación de llaves primarias (PK)
- Identificación de llaves foráneas (FK)
- Normalización
- Definición de atributos
- Definición de relaciones
- Validación de datos a procesar
-

1.6.9 Normalización

La meta de la normalización es asegurar que exista una sola manera de conocer un hecho, de acuerdo a la teoría matemática, para asegurar que cualquier porción de información del mundo real esté representada sólo una vez en un modelo Entidad-Relación.

El proceso de normalización de un modelo es aquel que implica el remover todas las estructuras de datos que proporcionan múltiples caminos para llegar al mismo hecho concreto. Otra forma de ver a la normalización es como un método para controlar y eliminar la redundancia en el almacenamiento de datos. La normalización es una parte muy importante del buen diseño de base de datos.

1.6.9.1 Normalización y Reglas del Negocio

Desde el punto de vista de la perspectiva del negocio, la meta de la normalización de un modelo de información es el asegurar que las reglas correctas del negocio sean registradas en el modelo y que las reglas que no son correctas sean removidas o revisadas. Este es un trabajo para la gente y no para una teoría matemática.

1.6.9.2 Problemas comunes en el Diseño

Los problemas comunes en el diseño son aquellos a los que nos tenemos que enfrentar cuando nos encontramos en la etapa de normalización de un modelo de datos, y que de alguna manera ya han sido identificados y analizados, por lo cual existen una serie de reglas a seguir para llegar a la obtención de un modelo de datos bien normalizado. Los problemas comunes en el diseño más importantes son los siguientes:

1. Grupos de Repetición
2. Uso Múltiple del mismo Atributo
3. Múltiples Ocurrencias del Mismo Hecho

1.6.9.2.1 Grupos de Repetición

Hay un problema con el modelo mostrado en la figura 1.6.9.2.1.a ¿Puede verse cuál es este?

EMPLEADO
Empleado_Id
Empleado_Nombre
Empleado_Dirección
Nombres_de_Hijos

Figura 1.6.9.2.1.a Entidad EMPLEADO

Una instancia de la tabla podría ayudar para ejemplificar:

EMPLEADO

Emp_Id	Emp_Nombre	Emp_Dirección	Nombres_de_Hijos
E1	Elizabeth	Polanco	Lisa
E2	Francisco	Insurgentes	Hugo, Paco, Luis
E3	Elena	Reforma	
E4	Luis	Insurgentes	Bar
E5	Patricia	Toluca	

Figura 1.6.9.2.1.b Ejemplo de Instancia de la tabla EMPLEADO para la Figura 1.6.

Es probable que el problema ya se haya visualizado sin la ayuda de la tabla. Existe una discordancia con el atributo llamado "Nombres_de_Hijos". Cuando se discutió acerca de entidades y atributos en el punto 1.6. (Nombres en Singular), se dijo que todos los nombres deben estar en singular. Aquí vemos uno que no lo está. La pregunta obvia podría ser, "¿Cuántos nombres necesitamos registrar?" "¿Cuánto espacio debemos reservar para nombres en cada renglón en la base de datos?" y "¿Qué es lo vamos a hacer si tenemos más nombres que espacio libre?"

Este diseño es una violación a lo que se conoce como *First Normal Form* (1NF) (Primera Forma Normal). La primera forma normal es una definición básica de la "estructura" de un diseño. Esta dice que los renglones y columnas de datos deben formar una tabla bidimensional sin una estructura anidada dentro de cada celda. Cada valor de datos en nuestra base de datos debe ser atómico, sin listas, ocurrencias repetidas o estructura interna. (Piénsese en cada valor en una columna como un valor en una "celda" en una hoja de cálculo como Microsoft Excel o Lotus 1-2-3.)

Con el fin de arreglar el diseño anterior, debemos remover de algún modo la lista de nombres de niños. Una manera de hacerlo es convertir la entidad que no está en primera forma normal, en la estructura mostrada en la figura 1.6.9.2.1.c.



Figura 1.6.9.2.1.c Modelo alterado, construido a partir de la figura 1.6

Ahora podemos representar los nombres de los hijos como datos únicos. (En términos de la estructura física de registro por empleado, esto puede resolver varias cuestiones de la asignación de espacio p.ej., el desperdicio de espacio en la estructura de registros para empleados que no tienen hijos o a la inversa, el decidir cuanto espacio asignar para empleados con familias).

EMPLEADO

Emp_Id	Emp_Nombre	Emp_Dirección
E1	Elizabeth	Polanco
E2	Francisco	Insurgentes
E3	Elena	Reforma
E4	Luis	Insurgentes
E5	Patricia	Toluca

Figura 1.6.9.2.1.d Instancia de la tabla EMPLEADO

HUJO

Emp_id	Hijo_id	Nombre_de_Hijo
PK		
E1	H1	Lisa
E2	H1	Hugo
E2	H2	Paco
E2	H3	Luis
E4	H1	Bart

Figure 1.6.9.2.1.e Instancia de la tabla HUJO para la Figura 1.6.

Se ha descubierto un atributo que tiene múltiples valores (lista o grupo de repetición), por lo que se crea una nueva entidad y se relaciona con la entidad original mediante una relación M:1.

1.6.10 Uso Múltiple del Mismo Atributo

Aquí está otro ejemplo de un problema con un diseño. ¿Puede verse cuál es este?

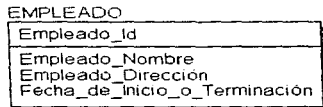


Figura 1.6.10 a Ejemplo de error en el diseño.

EMPLEADO

Emp_Id	Emp_Nombre	Emp_Dirección	Fecha_de_Inicio_o_Terminación
E1	Elizabeth	Polanco	Enero 10 de 1988
E2	Francisco	Insurgentes	Mayo 22 de 1986
E3	Elena	Reforma	Marzo 15 de 1985
E4	Luis	Insurgentes	Septiembre 30 de 1988
E5	Patricia	Toluca	Abril 22 de 1987
E6	Miriam	Palo Alto	Octubre 15 de 1988

Figura 1.6.10 b Ejemplo de Instancia de la tabla EMPLEADO para la Figura 1.6.

El problema aquí, por supuesto, es que tenemos un solo atributo, "Fecha_de_Inicio_o_Terminación" el cual puede representar uno de dos hechos, y no hay manera de saber a que hecho representa. Asimismo, no hay forma de registrar ambas fechas: la fecha de inicio, "la fecha en la cual el EMPLEADO comenzó a trabajar", y la fecha de terminación, "la fecha en la cual un EMPLEADO dejó la compañía", en situaciones en las cuales ambas fechas son conocidas.

EMPLEADO
Empleado_Id
Empleado_Nombre
Empleado_Dirección
Fecha_de_Inicio_o_Terminación
Tipo_de_Fecha

Figura 1.6.10.c Modelo mejorado construido a partir del mostrado en la Figura 1.6.:

La solución es no requerir de un solo atributo para que juegue una doble tarea, sino por el contrario, permitir que atributos por separado, soporten hechos separados. Abajo está nuestro primer intento por corregir el problema. Este no es totalmente correcto. Aquí, lo que es necesario es determinar, de que clase de fecha se trata, a partir del atributo "Tipo_de_Fecha". Mientras esto

puede ser eficiente en términos de la conservación del espacio físico de la base de datos, bien puede desencadenar confusión en la lógica para efectuar consultas.

En realidad lo que se ha creado es otro tipo de error de normalización (el atributo Tipo_de_Fecha no depende del atributo Empleado_Id para su existencia). Estos errores son en ocasiones fáciles de ver cuando encontramos un atributo soportando más de un hecho.

En la figura a continuación, se muestra una modificación al diagrama anterior. Este es un mejor camino que el anterior. Aquí, todos los atributos soportan hechos separados.

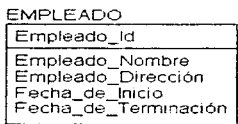


Figura 1.6.10.d Modelo "arreglado" construido a partir del mostrado en la Figura 1.6.:

EMPLEADO

Emp_Id	Emp_Nombre	Emp_Dir	Fecha_Inicio	Fecha_Terminación
E1	Elizabeth	Polanco	Ene 10 de 1988	
E2	Francisco	Insurgentes	May 22 de 1986	
E3	Elena	Reforma	Mar 15 de 1985	
E4	Luis	Insurgentes	Sep 30 de 1988	
E5	Patricia	Toluca	Abr 22 de 1987	
E6	Miriam	Palo Alto	Oct 15 de 1988	Nov 30 de 1988

Figura 1.6.10.e Ejemplo de Instancia de la tabla EMPLEADO para la Figura 1.6

Cada una de las dos situaciones que hemos visto fue lo que se conoce como error de primera forma normal. Al cambiar las estructuras nos hemos asegurado de que un atributo aparezca solo una vez en la entidad, y de que soporte un solo hecho. Aquí tenemos una definición para la primera forma normal:

Definición Informal: Una entidad está en primera forma normal si cada uno de sus atributos posee un significado único y no más de un valor por cada instancia.

Si estamos seguros de que todos los nombres de entidades y atributos están en singular, y que ningún atributo puede soportar múltiples hechos, entonces se habrá dado un gran paso para asegurar que un modelo de datos está en primera forma normal.

1.6.11. Múltiples Ocurrencias de un mismo Hecho

Aquí está una estructura con otro problema:

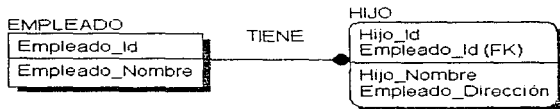


Figura 1.6.11.a Ejemplo de error de normalización

Si colocáramos el atributo "Empleado_Dirección" en la entidad HIJO como se muestra en la figura 1.6.11.a, tendríamos la sensación intuitiva de que algo es incorrecto. "Empleado_Dirección" es información acerca del EMPLEADO, no información acerca del HIJO. Otra manera de decir esto es que el valor de "Empleado_Dirección" no depende totalmente del valor de la llave primaria de la entidad HIJO, sino solo en parte de esta (la parte "Empleado_Id").

Colocando "Empleado_Dirección" de nuevo en la entidad EMPLEADO, como se muestra en la

figura 1.6.11.b, aseguramos que el modelo está al menos en segunda forma normal.



Figura 1.6.11.b Ejemplo de error de normalización.

Una entidad viola la segunda forma normal, si un hecho puede ser determinado conociendo solo parte de la llave primaria de la entidad. Si un hecho puede ser determinado conociendo el valor de algún atributo no llave de la entidad, entonces la tercera forma normal es violada. Por ejemplo, habría un error de tercera forma normal si la entidad HIJO en la figura 1.6.11.b contuviera los atributos "Fecha_de_Nacimiento" y "Edad" como atributos no llave. "Edad" es funcionalmente dependiente de "Fecha_de_Nacimiento", conociendo el valor del atributo "Fecha_de_Nacimiento" y la fecha actual, podemos derivar el valor "Edad" del HIJO.

Para resumir las tres formas normales se tiene la siguiente regla. Esta ha sido atribuida al Dr. E. F. Codd, el padre de la Teoría Relacional. "Una entidad está en tercera forma normal si cada atributo no llave depende de la llave primaria de la entidad; dicha de otro modo, cada atributo debe no depender de nada excepto del identificador único de su entidad (PK) para un valor."

CAPITULO II

SENSORES

2.1 Conceptos Básicos

2.1.1. Variables Físicas

La industria moderna exige el control en la fabricación de los diversos productos obtenidos. Los procesos son muy variados y abarcan muchos tipos de productos : la industria automotriz, la fabricación de los productos derivados del petróleo, los productos alimenticios, la industria cerámica, las centrales generadoras de energía, la siderurgia, los tratamientos térmicos, la industria papelera, la industria textil, etc.

En todos estos procesos es absolutamente necesario controlar y mantener constantes algunas magnitudes, tales como la presión, el caudal, el nivel, la temperatura, el pH, la conductividad, la velocidad, la humedad, el punto de rocío, etc. Los instrumentos de medición y control permiten el mantenimiento y regulación de éstas constantes en condiciones más idóneas que las que el propio operador podría realizar.

En los inicios de la era industrial, el operario llevaba a cabo un control manual de éstas variables utilizando sólo instrumentos simples como manómetros, termómetros, válvulas manuales, etc., este tipo de control era suficiente por la relativa simplicidad de los procesos. Sin embargo, la gradual complejidad con que éstos se han ido desarrollando ha exigido su automatización progresiva por medio de los instrumentos de medición y control. Estos instrumentos han ido liberando al operario de su función de actuación física directa en la planta y al mismo tiempo, le han permitido una labor única de supervisión y vigilancia del proceso desde centros de control situados en el propio proceso o bien en las salas aisladas separadas; así mismo, gracias a los instrumentos ha sido posible fabricar productos complejos en condiciones estables de calidad y de características, condiciones que al operario le serían difíciles o imposibles de conseguir, realizando exclusivamente un control manual.

Los procesos industriales a controlar pueden dividirse ampliamente en dos categorías : procesos continuos y procesos discontinuos. En ambos tipos, deben mantenerse en general las variables (presión, caudal, nivel, temperatura, etc.) bien en un valor deseado fijo, bien en un valor variable con el tiempo de acuerdo con una relación predeterminada, o bien guardando una relación determinada con otra variable.

El sistema de control que permite este mantenimiento de las variables puede definirse como aquel que compara el valor de la variable o condición a controlar con un valor deseado y toma una acción de corrección de acuerdo con la desviación existente sin que el operario intervenga en absoluto.

El sistema de control exige que para que ésta comparación y la subsiguiente corrección sean posibles, se incluya una unidad de medida, una unidad de control, un elemento final de control y el propio proceso.

Además de la presión, el caudal, el nivel y la temperatura, existen otras muchas variables que también son de interés industrial y que pueden clasificarse como físicas y químicas. Las variables físicas son aquellas relacionadas con las causas físicas que actúan sobre un cuerpo, con su movimiento o bien con las propiedades físicas de las sustancias. Entre éstas podemos mencionar el peso, la velocidad, la densidad, el peso específico, la humedad y el punto de rocío, la viscosidad y la consistencia, la llama, el oxígeno disuelto, la turbidez y la radiación solar. Las variables químicas están relacionadas con las propiedades químicas de los cuerpos o con su composición. Entre ellas se encuentran la conductividad, el pH, redox, y la composición de los gases en una mezcla.

2.1.2 Entradas y Salidas Analógicas

Frecuentemente una microcomputadora (que es un sistema digital) debe interactuar con variables externas de Entrada/Salida de naturaleza analógica en lugar de naturaleza digital.

Una función básica de los dispositivos de E/S (Entrada/Salida) es transformar la información de varias formas no eléctricas a una forma eléctrica digital, tal como se requiere para ser procesada por un microprocesador. Así, por ejemplo, un LED (diodo emisor de luz) transforma los valores lógicos 0 y 1 en dos niveles de intensidad luminosa (oscuridad e iluminación). Cuando las señales eléctricas como las no eléctricas que intervienen en la interfaz de E/S son primariamente digitales, como en conmutadores, LEDs y dispositivos similares, se dice que el dispositivo es digital de E/S. Hay otra clase importante y numerosa de dispositivos que tratan con señales externas analógicas, es decir, que varían continuamente; éstos se denominan dispositivos de E/S analógicos.

Generalmente se requieren dispositivos analógicos de entrada cuando una cantidad física continuamente variable debe medirse con precisión, mientras que los dispositivos analógicos de salida son necesarios para controlar magnitudes físicas continuamente variables. Muchas aplicaciones de los microprocesadores utilizan medidas y control analógicos.

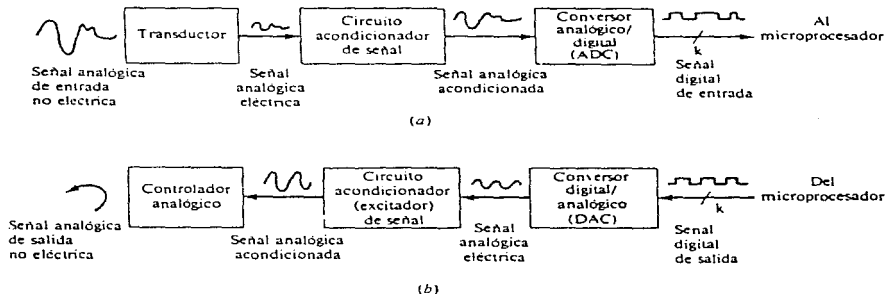


Figura 2.1.2.a Interfases analógicas de una microcomputadora:
(a) circuito de entrada; (b) circuito de salida

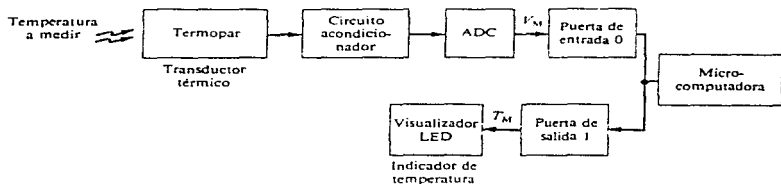
En la figura 2.1.2.a se ilustran los requisitos globales para realizar una interfaz entre un microprocesador y variables analógicas. En primer lugar la señal analógica no eléctrica de entrada se convierte en una señal eléctrica proporcional (analógica) por medio de un transductor o sensor. La salida del transductor usualmente debe ser modificada por circuitos electrónicos que adaptan las características eléctricas del transductor a las del dispositivos que alimentan. Las funciones habituales efectuadas por los circuitos que acondicionan las señales son filtrado de ruido y amplificación de voltaje. A continuación la señal eléctrica acondicionada se convierte en una señal digital proporcional por medio de un **convertor analógico/digital (ADC)** controlado por hardware o por software. La señal digital resultante puede ser transferida a la computadora central para su almacenamiento o procesamiento.

Un conjunto de operaciones similares se necesitan para realizar la interfaz de un microprocesador con un dispositivo analógico de salida como lo indica el inciso "b" de la figura 2.1.2.a. Las señales digitales de salida del microprocesador primero se convierten en señales eléctricas analógicas proporcionales por medio de un **convertor digital/analógico (DAC)**. La salida del DAC es acondicionada por circuitos apropiados (circuitos excitadores de salida, por ejemplo) y transferida al dispositivo analógico de salida.

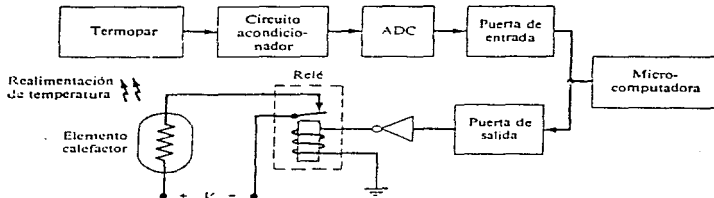
Debería observarse que en muchos casos de interés, las magnitudes analógicas de salida pueden controlarse directamente por señales digitales, con lo que en estos casos puede hacerse la interfaz de la misma forma que en los dispositivos de salida digitales.

Los instrumentos digitales de medida, por ejemplo, emplean unidades visualizadoras digitales de salida, tales como LEDs o CRTs, para mostrar cantidades analógicas correspondientes a magnitudes tales como voltaje, corriente, temperatura y presión; en lugar de unidades analógicas de salida tales como las producidas por indicadores electromecánicos con el movimiento de agujas y diales (analógicos). Con los visualizadores digitales se obtiene mayor precisión, y éstos son inherentemente más fáciles de conectar a un microprocesador. Es decir, hay menos necesidad de dispositivos analógicos de salida.

En el inciso "a" de la figura 2.1.2.b se muestra un termómetro basado en un microprocesador, que está diseñado para procesar la variable analógica temperatura. Utiliza un dispositivo analógico de entrada, un sensor de temperatura denominado termopar y un dispositivo de salida digital, como es un visualizador LED de siete segmentos, que proporciona los valores de interés de temperatura en forma digital (decimal). En ésta figura, se ilustra el papel de un microprocesador en la medida de una magnitud analógica. El control de una señal analógica a través de un microprocesador se ilustra con el sistema de control de temperatura o termostato en el inciso "b" de la figura 2.1.2.b. Aquí el objetivo del sistema es regular la temperatura de un horno eléctrico.



(a)



(b)

Figura 2.1.2.b Interfaces de un microprocesador para temperatura: (a) medida de temperatura; (b) control de temperatura.

La temperatura del horno se sube o baja conmutando (conectando o desconectando) un elemento eléctrico (resistencia) de caldeo. Como una resistencia de caldeo requiere corrientes eléctricas bastante elevadas (Amperes), se utiliza un réle electromagnético para conmutar la corriente de caldeo; el réle se conecta directamente a la puerta de salida del microprocesador. La temperatura del horno producida por el elemento de caldeo es detectada por el termopar y transmitida al microprocesador. Si la temperatura es demasiado alta, el programa del microprocesador hace que el réle se abra cortando la corriente al elemento calefactor, de éste modo la temperatura del horno bajará. Por el contrario, si se detecta una temperatura baja, el microprocesador conecta al elemento de caldeo. Este es un ejemplo de un sistema de control en lazo cerrado, en los que las desviaciones en el parámetro controlado, en este caso la temperatura del horno, son constantemente monitoreadas y corregidas.

2.1.3 Transductores

Los transductores se definen como dispositivos que convierten energía o información de una forma a otra. Se emplean extensamente en el trabajo de medición porque no todas las cantidades que se necesitan medir se pueden mostrar con tanta facilidad como otras.

En términos generales, los tres elementos principales de un transductor son :

1. El elemento de detección o sensor, que responde a la magnitud (o cambios en la magnitud) de la variable que se está midiendo.
2. El modificador de la señal, que recibe la señal de salida del elemento de detección y la modifica mediante amplificación o conformación adecuada de su onda. Cuando la señal sale del modificador, debe estar en forma adecuada para registrarse ó mostrarse.
3. El dispositivo de salida, que pueden ser medidores, pantallas, monitores, grabadoras de cinta, registradores y microcontroladores.

La figura 2.1.3.a muestra el diagrama de bloques de un transductor.



Figura 2.1.3.a Diagrama de bloques de un transductor.

En las áreas de aplicación de los microprocesadores, los transductores tienen la función de convertir variables de entrada no eléctricas (usualmente analógicas) a eléctricas. Se conocen un gran número de métodos de transducción, muchos de los cuales de gran ingenio. La mayoría de los transductores son sencillos, muchos utilizan resistencias cuyo valor es alterado por la variación del parámetro de interés.

Muy diversos transductores reciben información del mundo exterior en forma de señales luminosas variables en el tiempo, siendo convertidas por ellos en señales eléctricas. Los dispositivos de este tipo son denominados transductores ópticos.

Los transductores ópticos son adecuados especialmente para interfaces de microprocesadores con dispositivos mecánicos que involucran movimientos que deben ser medidos, ya que este tipo de transductores elimina la necesidad de contactos mecánicos entre las partes móviles y el sensor. Muchos transductores ópticos son construidos a partir de materiales semiconductores cuya resistencia es función de la intensidad de la luz a la que están expuestos.

La transformación en señal eléctrica (o transducción) de una magnitud física particular usualmente puede realizarse de muchas formas, que difieren grandemente en sus requisitos de interfaz.

Los transductores son muy diversos, y están basados en una gran variedad de fenómenos eléctricos. Una clase importante de transductores depende de la posibilidad de que la variable analógica de interés altere la resistencia del transductor. La relación entre la señal de salida del

transductor y la señal de entrada analógica original puede no ser lineal y compleja. Sin embargo, la potencia de los microprocesadores puede usarse fácilmente para sobreponerse a éste problema; por ejemplo, almacenando en forma tabular las características del transductor en la memoria del sistema, a partir de la cual puede recuperarse el valor deseado con las instrucciones adecuadas de búsqueda. Los actuadores electromecánicos, tales como solenoides y motores de pasos, permiten a los microprocesadores controlar variables de salida tales como la posición y la velocidad de un objeto. Estos dispositivos son componentes clave en un amplio rango de dispositivos de E/S, incluyendo impresoras, unidades de memoria de disco y equipos de control de procesos industriales.

La conversión analógico/digital y digital/analógico implica una serie de compromisos entre los circuitos y los programas de control, y también entre la velocidad de operación y precisión digital. Generalmente los DACs son circuitos construidos alrededor de resistencias ponderadas. Se usan muchas técnicas para realizar conversores ADCs, incluyendo el método de conversión directa totalmente realizada con circuitos, el método rampa y el método de aproximaciones sucesivas. Los dos últimos pueden realizarse con rutinas ADC usando circuitos de interfaz relativamente sencillos.

2.1.4 Actuadores

En muchas aplicaciones un microprocesador controla variables mecánicas tales como la posición u orientación de un objeto. Esto requiere un dispositivo de salida, generalmente llamado actuador éste puede transformar señales de control electrónicas en movimientos mecánicos. Un réle puede considerarse como un tipo sencillo de actuador de control de la posición de un conmutador mecánico. Movimientos más complicados se controlan con dispositivos electromecánicos más potentes, de los que los solenoides y motores son los más importantes.

Un solenoide es un tipo de actuador bastante más sencillo, que, similarmente a un réle, es un tipo de electroimán. Contiene una bobina cilíndrica de hilo esmaltado que se activa cuando pasa una corriente a través de él. Esto crea un fuerte campo magnético en la cavidad interna del solenoide,

que ejerce una fuerza de atracción en un émbolo deslizante de hierro dulce. De esta forma, el solenoide, cuando se activa, absorbe hacia su interior el émbolo; cuando se desactiva, por conmutarse a corte la corriente que fluye a través de la bobina, un muelle de recuperación hace que el émbolo salga del interior de la bobina.

La figura 2.1.4.a muestra un solenoide utilizado para actuar como cerradura de una puerta. El movimiento del émbolo del solenoide se transmite al pestillo deslizante por medio de una palanca. También puede utilizarse un solenoide para producir un movimiento de rotación con ayuda de un trinquete. Generalmente los movimientos de rotación, así como movimientos lineales grandes, se efectúan con ayuda de motores eléctricos. Los motores eléctricos se utilizan para controlar formas de movimiento más complicados.

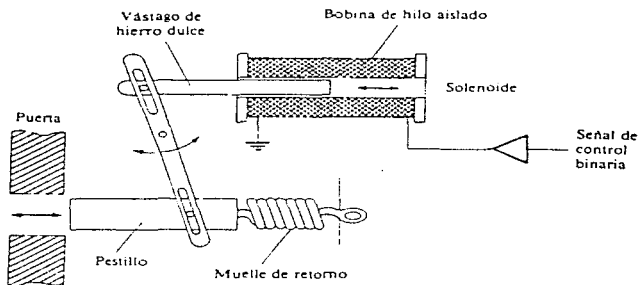


Figura 2.1.4.a. Actuador solenoide utilizado en una cerradura de una puerta.

La figura 2.1.4.b muestra un motor que controla el movimiento hacia adelante y hacia atrás de una herramienta a lo largo de un camino lineal.

El movimiento de rotación del eje del motor se convierte en movimiento lineal por medio de un eje roscado (una guía con rosca), en el que se acopla (como una tuerca) un orificio de la herramienta. La dirección de movimiento de la herramienta puede cambiarse invirtiendo la dirección del movimiento del motor eléctrico; esto último usualmente se efectúa invirtiendo la dirección del flujo de corriente eléctrica que atraviesa el motor.

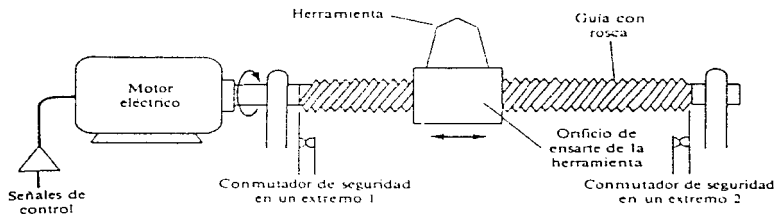


Figura 2.1.4.b Utilización de un motor eléctrico para controlar el movimiento de una herramienta a lo largo de un camino lineal.

Se utilizan conmutadores de seguridad en los extremos de la barra con rosca para desconectar el motor si el cabezal de la herramienta llega a ellos. Pueden utilizarse otros conmutadores para detener la herramienta en posiciones intermedias; en esta posición son particularmente útiles los transductores ópticos o magnéticos.

2.1.5 Obtención y Proceso de Datos

El elemento esencial de un sistema moderno de obtención de datos es el elemento sensor, el cuál proporciona una señal eléctrica que indica la variable física que se está midiendo. La señal puede ser un voltaje, resistencia o frecuencia analógicos, o una representación digital de cualesquiera de estas cantidades en forma de serie de pulsos eléctricos.

El objeto de cualquier sistema de obtención y proceso de datos es coleccionar los datos, procesarlos en la forma deseada y registrar los resultados de manera adecuada para almacenamiento, presentación o proceso adicional subsecuente; por lo tanto un potenciómetro registrador es un sistema simple de obtención de datos que puede usarse a fin de coleccionar datos de temperatura de termopares. En este caso, los puntos de datos deben de leerse en la carta registradora. Un sistema más complejo puede convertir la señal analógica de voltaje del termopar en una señal digital equivalente, útil para operar un registrador impresor, de modo que el valor numérico de la temperatura se imprime en una hoja de papel. Dicho sistema es mucho más complicado que el registrador simple, debido al proceso de conversión de analógico a digital. Sin embargo es fácil ver que la salida digital tiene muchas ventajas. Los elementos principales de cualquier sistema de proceso de adquisición de datos se muestran en el diagrama de bloques de la figura 2.1.5.a. Esta figura presenta un sistema dividido en tres partes fundamentales. La primera es la etapa de entrada, la cual consta de sensores apropiados y un circuito de entrada, además de circuitos adicionales y el acondicionamiento de la señal como sea necesario. (amplificadores, filtros, etc.).

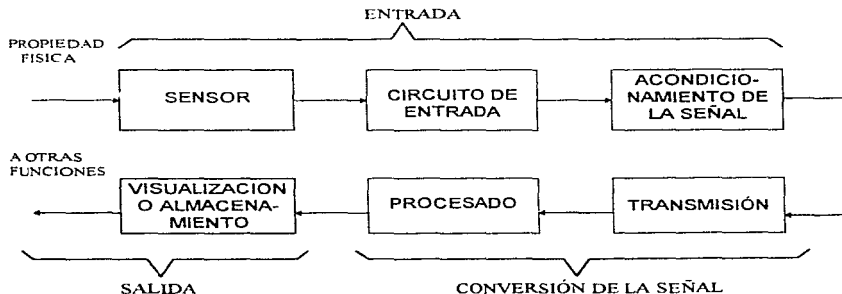


Figura 2.1.5.a Sistema general de adquisición de conjuntos de datos

La segunda es la etapa de conversión de la señal, en la cuál la información se prepara para transmisión, lo mismo que el tipo de transmisión-recepción y cualesquiera procesadores necesarios de datos; un ejemplo de ésta última es la conversión de una señal de la forma analógica a digital. La parte final es una etapa de salida con dos funciones primarias: mostrar y almacenar datos.

Los ejemplos incluyen el despliegue de la información y el almacenamiento en forma impresa en una hoja de papel, en forma gráfica o en un papel adecuado y almacenamiento en cinta magnética u otro medio. La etapa de salida debe incluir circuitos apropiados de acoplamiento para convertir los datos en una forma que puedan usarse a fin de emplear una impresora, cinta magnética, etc.

Es muy raro que los datos que se van a coleccionar contengan solo una variable experimental, generalmente un sistema de procesamiento y obtención de datos debe estar equipado para captar y analizar canales múltiples de entrada de datos. Este proceso de obtención debe realizarse por medio de un canal parecido al que se muestra en la figura 2.1.5.b para cada variable que se va a estudiar. El costo de dicho sistema puede ser bastante alto, debido a la duplicación de equipo, de modo que normalmente se emplea un barredor (o explorador) /programador para trabajo con canales múltiples.

El barredor es un dispositivo que muestra los canales de datos, de modo que solo se necesita una etapa de conversión y salida. El equipo disponible actualmente, hace posible cualquier secuencia particular de un gran número de canales de datos puestos a la discreción del personal de laboratorio; por lo tanto, el sistema puede programarse con objeto de coleccionar cualquier gama de variables en cualquier orden, y el término barredor/programador es bastante apropiado.

Muchas situaciones experimentales requieren la obtención de los datos a intervalos regulares o con alguna secuencia particular de tiempo. El sistema de obtención puede realizar ésta función de sincronización en forma automática por la incorporación de un reloj digital y un estándar de tiempo en el barredor o en las etapas de conversión.

Por último puede ser ventajoso aplicar el acondicionamiento de la señal a la salida del dispositivo que está funcionando como el barredor/programador. Este acondicionamiento puede ser: amplificación, conversión de analógica en digital para algunos canales, filtrado, análisis de distorsión o armónico de la forma de onda de algunas señales, etc. Cuando todos estos elementos se combinan, dan como resultado un sistema de procesamiento y obtención de datos muy flexible. Como se muestra en la figura. Debe de observarse que la característica programable del barredor también se usa normalmente en la etapa convertidora. Esto es esencial, ya que algunos canales pueden requerir acondicionar la señal mientras que otros no.

El uso de un sistema flexible como el descrito anteriormente, depende de muchos factores, de los cuales el costo no es el menor. El progreso de la industria de los semiconductores, en particular el desarrollo de los microprocesadores, ha reducido de forma considerable el costo de dichos sistemas. Los sistemas de procesamiento y de datos que usan microprocesadores para realizar la función de barredor/programador se han difundido ampliamente.

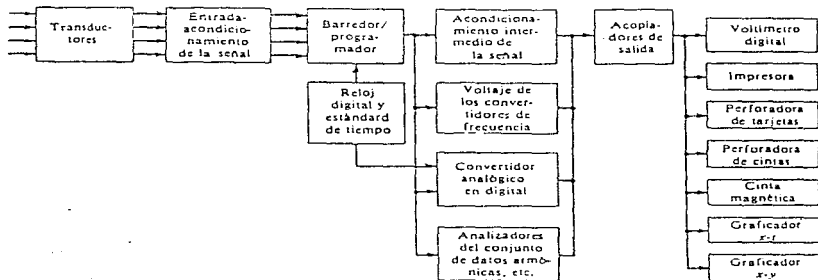


Figura 2.1.5.b Esquema de un sistema multicanal para obtención de datos.

2.2. Areas de Aplicación

En la vida cotidiana encontramos que nos rodean innumerables tipos de sensores, desde el reloj electrónico que nos despierta, el horno de microondas, la máquina rasuradora eléctrica, la plancha, la licuadora, el metro, los autobuses, etc.

En base a los tipos de sensores existentes podemos mencionar algunas de las áreas importantes donde intervienen los sensores :

- Sensores en la medición de parámetros físicos.- Consideremos primero los de fuerza que tienen una gran cobertura en el diseño y análisis de vehículos veloces, máquinas sometidas a altos regímenes de trabajo, estructuras gigantescas, y grúas entre otras muchas aplicaciones. Los transductores de desplazamiento lineal como el radar, el cual se emplea en distintos campos como la navegación aérea y marítima, topografía, meteorología y control de tráfico cuyas frecuencias oscilan entre los 30MHz y los 30GHz. En distancias inferiores a 100m se utilizan los ultrasonidos (30 a 40 KHz) cuya velocidad de propagación es mucho menor: 340 m/s en el aire, 1450 m/s en el agua, y del orden de los 4000 m/s en los metales. Si la distancia a medir es de unos pocos metros puede ser útil un potenciómetro como sensor y las pequeñas medidas podrán realizarse con bastante precisión empleando sensores resistivos, inductivos y capacitivos. Dos sistemas de medida de desplazamiento angular de mayor uso son los sensores del tipo inductivo: el Syncro y el Resolver que son empleados con frecuencia en radar para la posición de una antena giratoria o para obtener una indicación de la posición de los alerones de un avión.
- Sensores de proximidad.- Los sensores de proximidad sensibles a materiales ferromagnéticos con bobina pueden sustituir a los contactos de tungsteno en los automóviles que incorporan sistemas complementarios estáticos de encendido como sistema generador de impulsos para la medición y regulación de las rpm de un motor. Los sensores de proximidad por efecto Hall utilizan potencia para su funcionamiento es por ello que son de interés en sistemas lógicos complejos, en general son compatibles con circuitería TTL, este modelo es diez veces más

rápido que el anterior, esta característica lo hace explotable en circuitos interconectados a sistemas lógicos, triács, SCR, etc. Asimismo es capaz de realizar detecciones estáticas que permiten su uso como detector de posición o fin de carrera. Su empleo es tan extenso que se presentan modelos comerciales en formato dual-in-line. Este modelo de detectores de proximidad, utilizados con actuadores de magnetismo permanente, permite la fabricación de pulsadores, botoneras, teclados, exentos de contactos mecánicos y por lo tanto de rebotes. Una aplicación elemental de los detectores de proximidad por radiación es la detección de focos de calor por lo que es usado en el foco de un incendio con ambiente rodeado de humo, en motores o rodamientos de vehículos ferroviarios. En cuanto a los de radiación visible tienen mayor aplicación en la detección de intrusos con luz ambiente, en control de puertas y ascensores, control de tráfico, estacionamientos tanto de vehículos como de peaje, etc.

- Sensores en medida de espesores.- Los sensores son empleados ampliamente en la industria donde se requiere realizar un control de calidad de tipo mecánico sobre el producto acabado. Un caso típico de necesidad de utilización de un control de espesores de materia es el de un proceso de fabricación continuo de un material que se lamina o se produzca en bandas como la laminación de metales en frío o en caliente, producción de papel, fabricación de papel de estaño, obtención de diversos plásticos para recubrimientos de superficies, etc. en todos estos casos precisa controlar continuamente el espesor del material acabado de forma que si se produce algún fallo en la cadena de producción se pueda parar ésta automáticamente, a fin de evitar la pérdida inútil de material o bien se pueda realizar una regulación continua de parámetros del proceso tales como separación de la banda, separación de rodillos, etc. cuyos efectos se puedan medir en función del espesor del material resultante. En los últimos años su uso se ha incrementado en la fabricación de semiconductores para depositar las capas sobre obleas semiconductoras de diversos tipos de materiales, tanto metálicos (oro, aluminio), como dieléctricos.
- Sensores para el caudal de fluidos.- En la mayor parte de las operaciones realizadas en los procesos industriales y en las efectuadas en laboratorio es muy importante la medición de los caudales de líquidos o gases, incluyendo el caudal de fluidos con partículas suspendidas.

Existen varios métodos para medir el caudal según sea el tipo de fluido, la presión deseada, el control requerido y tipo de caudal volumétrico o másico deseado, algunos elementos a mencionar son la placa-orificio o diafragma, la tobera, el tubo Venturi, tubo de Pitot, tubo Annuar, fuelle, turbinas, incluso sondas ultrasonicas, entre otros.

- **Sensores en medidas de vibraciones.**- Actualmente las medidas de vibraciones son de mucha utilidad; tal es el caso del estudio dinámico de estructuras estáticas (obras públicas, edificios, etc), estudio de fatiga y estabilidad en materiales sometidos a este tipo condiciones (aeronaves, automóviles etc.), los movimientos vibratorios de un sismo o erupción volcánica, así como la manufactura de los aparatos que los sensan o la influencia en el hombre de este tipo de agente (higiene mental).
- **Sensores de iluminación y calorimetría de sólidos.**- En las lámparas incandescentes la emisión de la luz depende de la tensión RMS aplicada. El fototransistor es sensible a cambios de luz y se utiliza como monitor de esta lámpara, estos sistemas se usan en lugares donde se debe mantener constante un nivel de iluminación a pesar de las variaciones de tensión de red. Los fotoacopladores cubren campos como detección de objetos con luz, transmisión de información analógica y digital, circuitos de telecomunicaciones, control completo de ondas de CA, disparo de series de SCR en altas tensiones, aislamiento de circuitos trifásicos, diseño de fuentes de alimentación aisladas, así como operar señales lógicas TTL.
- **Sensores en análisis de gases.**- En los últimos años una de las áreas de aplicación más importantes tanto para la industria como para los seres vivos es la contaminación. Los detectores de conductividad térmica son prácticos en la detección de gases puros.

Como es posible observar las áreas de aplicación de los sensores no tiene límites y puede ser aplicada en todas las ramas de la ciencias según la evolución de la misma y la imaginación de aquellos que los manipulan.

2.3 Tipos de sensores a emplear en el sistema

Las principales variables físicas involucradas en el sistema desarrollado en el presente trabajo son nivel de combustible, temperatura del motor y velocidad del vehículo. A partir de éstas, es posible determinar directa o indirectamente muchas otras mediciones durante el recorrido.

Para nuestro propósito, es importante considerar las características principales y el funcionamiento de algunos sensores comerciales que se emplean actualmente en la industria automotriz y que tratan con las variables antes mencionadas.

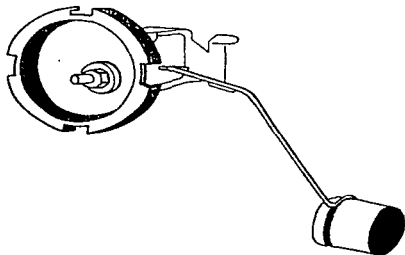
La mayoría de los dispositivos estudiados a continuación, se encuentran incorporados en los vehículos actuales, principalmente en aquellos provistos con sistemas de inyección electrónica de combustible (Fuel Injection) controlada por microprocesador.

2.3.1 Nivel de combustible.

Dentro del área de Ingeniería, la medida de nivel de líquidos en un tanque puede realizarse por varios métodos según sea el material almacenado, el tipo de tanque y la precisión deseada, así por ejemplo tenemos los siguientes tipos : Flotador, Desplazamiento, Presión diferencial, Conductivo, Capacitivo, Ultrasónico y Radiación.

Algunos de estos sensores, son sumamente precisos y sofisticados, sin embargo, en la industria automotriz, generalmente se emplean sensores del tipo flotador, ya que son de bajo costo, su funcionamiento es muy sencillo y satisfacen las necesidades actuales.

El sensor de nivel de combustible (flotador) tiene la función básica de medir la cantidad de combustible en el tanque correspondiente, en un instante determinado. Este sensor funciona bajo el principio de continuidad y opera a través de un potenciómetro (resistencia variable). En la siguiente figura, se muestra el diagrama general de un sensor de nivel y sus partes fundamentales.



2.3.1 a Medidor de nivel de combustible

Podemos observar, en la figura anterior, que el elemento flotador variará su posición de acuerdo al nivel de combustible que tenga el tanque en ese momento. Esta variación de posición se traducirá en un movimiento de la varilla o brazo que hace el papel del cursor del potenciómetro. De ésta forma, se obtendrá una variación en el valor de la resistencia. Finalmente a través de un divisor de voltaje será posible obtener en las terminales, una señal de voltaje cuya magnitud será directa ó inversamente proporcional (dependiendo de la posición de las terminales) al combustible en el tanque.

2.3.2 Temperatura del motor

Existen varios sensores para medir la temperatura del motor en los vehículos, muchos de estos emplean mediciones indirectas, es decir, miden la temperatura de las diferentes sustancias que se encuentran en contacto directo con el motor, por ejemplo, la temperatura del líquido refrigerante.

de los gases de escape o del aceite lubricante; por otro lado también se han diseñado algunos sensores que van montados sobre la superficie del motor y registran directamente el valor de la temperatura.

En la mayoría de los vehículos, la temperatura del motor se obtiene con el sensor de temperatura del refrigerante, también conocido como **CTS (coolant temperature sensor)**. Este sensor mide la temperatura del motor en forma indirecta, a través de la temperatura del líquido refrigerante que se emplea para enfriar el motor. El sensor CTS consiste en una resistencia eléctrica (termistor) que baja su valor conforme se incrementa la temperatura del refrigerante. Este tipo de sensor es alimentado con 5 volts de referencia por un cable, ese voltaje regresa a tierra a través de la computadora al cerrarse el circuito. Como la resistencia del termistor se altera con la temperatura, el voltaje retornado se modificará, en consecuencia, el cambio en el voltaje consumido es advertido por el microprocesador, el cual lo transforma en datos de temperatura.

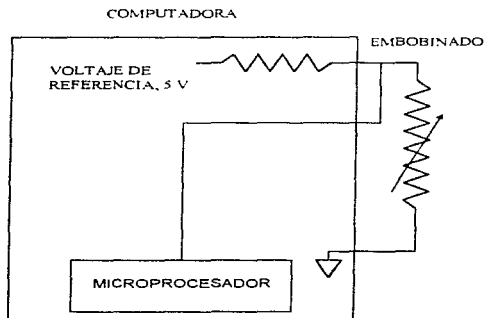


Figura 2.3.2 a Sensor de temperatura (CTS)

Cuando el dispositivo, por medio del cual ha de medirse la temperatura, es un interruptor, éste sólo tiene dos posiciones: abierto o cerrado en este caso la computadora envía un voltaje de referencia de 5 volts. Al estar el motor frío el interruptor esta cerrado y la computadora sensa 0 volts ya que el voltaje de referencia es aterrizado a través del motor. Al alcanzar el motor una determinada temperatura, el interruptor, que es de estado sólido, abre para que el voltaje de retorno sea de 5 volts. El interruptor permanece abierto hasta que la temperatura cae por debajo de un valor predeterminado entonces el voltaje es nuevamente aterrizado. La gráfica 2.3.2.b muestra un interruptor de temperatura del refrigerante.

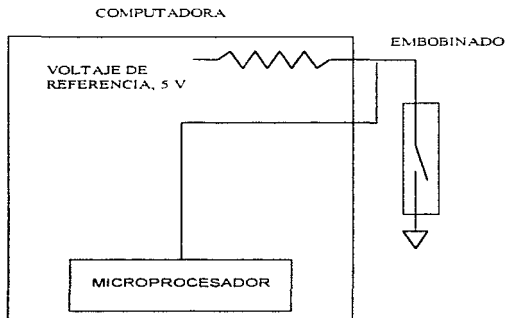


Figura 2.3.2.b Circuito sensor de temperatura del tipo interruptor

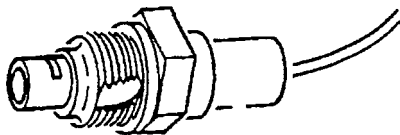


Figura 2.3.2.c. Aspecto típico del sensor de temperatura

2.3.3 Velocidad del vehículo

El sensor de velocidad o VSS (Vehicle Speed Sensor), proporciona al microprocesador, el dato de la velocidad a la que se desplaza el vehículo, esta información es muy importante para el correcto funcionamiento de muchos subsistemas: velocímetro electrónico, control de gasolina en desaceleración, referencia de velocidad en aceleración, registro de código de falla diversos, odómetro electrónico, control de acelerador automático, frenos antibloqueo, enganche mecánico del convertidor de torsión, control de marcha mínima, etc. En la siguiente tabla se listan algunos tipos de sensores de velocidad comunes.

SENSORES DE VELOCIDAD

TIPO	VOLTAJE	RESPUESTA	LOCALIZACIÓN
Mecánico	Analógico	Pulsos	En el chirrón del velocímetro o en la flecha del cardán.
Mecánico	Digital	Pulsos	En el chirrón del velocímetro o en la flecha del cardán.
Mecánico	Digital	Pulsos	Entre el extremo del chicote en el tablero de instrumentos y su montura.

Tabla 2.3.3 a Tipos de sensores de velocidad comunes

Como se aprecia en la tabla anterior, existen varios tipos de sensores de velocidad, el más común utiliza un punto botón metálico giratorio montado en el chirrión o eje, esta pieza incide sobre una sonda magnética con cada giro para producir en un momento un voltaje o impulso (con corriente alterna), dicho voltaje es analógico y es convertido a digital por medio de un módulo externo de encendido antes de llegar a la Microcomputadora.

Otro tipo de sensor mecánico utiliza un micro-interruptor normalmente abierto formado de dos platinos y fijado cerca del eje rotatorio. Entre los platinos existe una diferencia de potencial de 5 volts (referencia recibida desde la computadora, uno de los platinos funciona como tierra). Montado en el eje rotatorio esta un imán que al pasar cerca del micro-interruptor empuja a los platinos a que contacten. Cuando los platinos están separados el micro-interruptor esta abierto, la computadora siente que no hay consumo de voltaje de referencia, por lo tanto detecta los 5 volts. Cuando el imán hace que contacten los platinos el micro-interruptor se cierra, la Unidad Electrónica de Control sensa 0 volts, esto significa que el voltaje de referencia esta siendo aterrizado. Aquí se genera un pulso digital y a medida que el eje rotatorio pasa una y otra vez sobre el micro-interruptor, se genera una frecuencia digital que es interpretada como la velocidad del vehículo. Este tipo de sensor no requiere de convertidor analógico digital ya que su señal es digital.

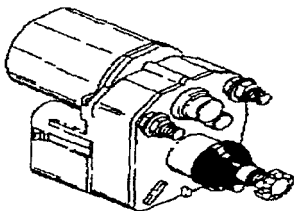


Figura 2.3.3.a Sensor tipo de velocidad

2.4 Elección de los sensores y lugares óptimos de trabajo

Cuando se pretende realizar la medición de una cantidad no eléctrica convirtiéndola a una forma eléctrica, se debe seleccionar un sensor adecuado (o combinación de ellos) para llevar a cabo esta conversión. El primer paso en el procedimiento de selección es definir con claridad la naturaleza de la cantidad que se va a medir. Esto debe incluir el conocimiento del rango de magnitudes y frecuencias que se espera exhiba la variable.

Cuando se ha establecido el problema, se deben examinar los principios del sensor disponible para medir la cantidad deseada. Si uno o más principios de sentido son capaces de producir una señal satisfactoria, se debe decidir si se usa un dispositivo comercial o si se intenta construir uno. Si hay sensores comerciales disponibles, a un precio razonable, la elección probablemente será comprar uno de ellos. Por otro lado, si no se fabrican sensores como los requeridos, se tendrán que diseñar, construir y calibrar. Cuando se examinan las especificaciones de un sensor en especial, se deben considerar los siguientes puntos:

1. **Rango.** Debe ser lo suficientemente grande de tal manera que abarque las magnitudes esperadas en la cantidad a ser medida.

Sensibilidad. El dispositivo sensor debe de producir una señal de salida lo suficientemente alta por cada unidad de entrada medida, con el fin de obtener datos significativos.
2. **Efecto de carga.** Dado que los sensores siempre consumirán algo de energía del efecto físico que se está probando, se debe determinar si la absorción es despreciable o si se deben aplicar factores de corrección para compensar las lecturas.
3. **Respuesta a la frecuencia.** El sensor debe ser capaz de responder a la velocidad máxima de cambio del efecto observado.

4. Formato de salida eléctrica. La forma eléctrica de la salida del sensor debe ser compatible con el resto del sistema de medición.
5. Impedancia de salida. Debe de tener un valor que lo haga compatible con las siguientes etapas eléctricas del sistema.
7. Potencia requerida. Dado que los sensores pasivos requieren de alimentación externa, se debe asegurar que existen fuentes adecuadas para su operación.

Medio Físico. El sensor debe poder resistir las condiciones ambientales a las que estará sujeto durante su operación. La temperatura, la humedad y las substancias químicas corrosivas pueden dañar algunos dispositivos.

8. Errores. Los errores inherentes a la operación del sensor o aquellos originados por las condiciones de operación deben ser lo suficientemente pequeños o controlables para permitir obtener datos significativos.
9. Aspectos eléctricos. Se deben considerar la longitud y el tipo de cable a utilizar, cuál es la relación señal-ruido cuando se combinan con amplificadores y las limitaciones de la respuesta a la frecuencia.

Como se mencionó anteriormente, para el registro de las variables que se pretenden, es necesario contar con tres tipos de sensores que son : Nivel de Combustible, Temperatura y Velocidad. Dentro de la diferente gama de vehículos que se pueden encontrar en el mercado automotriz, la mayor parte de los sensores son generalmente los mismos y son colocados en los mismos lugares respectivamente; es por ello que la selección de los sensores así como los lugares de trabajo, se basaron en la tecnología automotriz actual, obteniendo de esta forma las siguientes ventajas:

- Simplificación en el desarrollo del sistema, puesto que no es necesario diseñar sensores nuevos o especiales para cada aplicación, además de que ya existe en el mercado información técnica para el empleo de cada dispositivo.
- Precisión y seguridad en la obtención de resultados, ya que el funcionamiento de los sensores está probado en la práctica
- Reducción en el costo del sistema.

A continuación se mencionan los sensores y los lugares de trabajo seleccionados, para cada una de las variables a medir :

- Nivel de combustible- Después del análisis de los diferentes tipos de sensores existentes y considerando los puntos anteriores, la medición de esta variable, se realizará a través de un sensor del tipo Flotador, este tipo de sensor, se encuentra en la mayoría de los automóviles y son colocados siempre dentro del depósito de combustible.

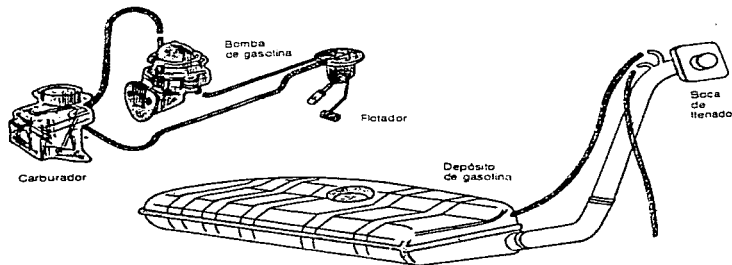


Figura. 2.4.a Lugar del montaje del sensor de nivel de combustible (flotador)

- **Sensor de Temperatura.** - Para el desarrollo del sistema, se ha elegido el sensor de temperatura del tipo CTS (coolant temperature sensor), es decir, mide la temperatura del motor a través del líquido refrigerante. Este sensor va montado a uno de los costados del motor donde se toma la temperatura del sistema de refrigeración, como se puede ver en la figura 2.4.b. En algunos motores, éste sensor también se encuentra localizado uno de los costados del radiador. Cualquiera de los dos casos anteriores es recomendable, pues dependiendo del modelo del vehículo vienen diseñados para ser montados en un lugar o en el otro.

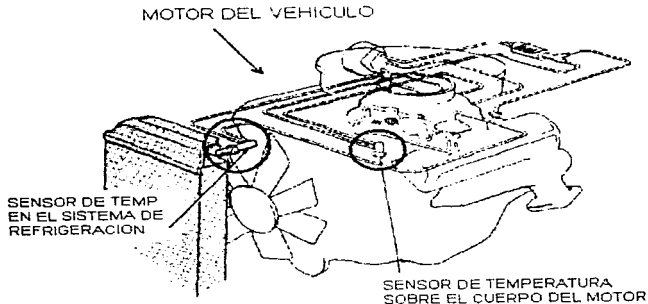


Figura. 2 4 b Montaje del sensor de temperatura en el vehículo.

- **Sensor de velocidad.** - El sensor de velocidad es uno de los más importantes dentro del sistema, ya que en base a éste se obtendrán directa o indirectamente la mayoría de las variables que requerimos como son: la velocidad, la distancia y el tiempo de recorrido entre otras. Para nuestro caso, se ha optado por emplear un sensor del tipo mecánico, que se alimenta con 5 volts de D.C. y proporciona una respuesta digital en forma de pulsos. Este sensor va acoplado

a través de una serie de engranes a una de las ruedas delanteras (ver figura 2.4.c); cada vez que la rueda gira, el sensor manda una serie de pulsos por cada vuelta, mismos que serán registrados en la computadora de viaje.

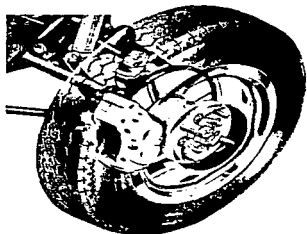


Figura 2.4.c Montaje del sensor de velocidad

Existen otras formas de instalar el sensor de velocidad, esta técnica consiste en cortar el cable del velocímetro y montar un acoplamiento para el sensor. Este caso no es muy recomendable pues puede ocasionar problemas después de un determinado tiempo de uso, además su montaje requiere de cierto cuidado para no dañar el cable del velocímetro.

En el presente trabajo, se manejan las variables físicas más representativas, que nos permiten definir las condiciones más importantes durante el trayecto de un vehículo. Todos los dispositivos elegidos, cumplen satisfactoriamente con los requerimientos que se tienen, sin embargo, el diseño del sistema permite ser adaptado para manejar más variables y emplear sensores más sofisticados, dependiendo de los niveles de exactitud y seguridad que se deseen alcanzar.

CAPITULO III

EL MICROCONTROLADOR MC68HC11

3.1. Características del Microcontrolador MC68HC11

En la presente sección se dará una descripción breve de la arquitectura y operación del microcontrolador MC68HC11 de motorola, el cual se destaca por ser un circuito avanzado de alta escala de integración (LSI), que combina en una misma pastilla un microprocesador, memorias, puertos, un convertidor analógico-digital (A/D) y otros circuitos electrónicos sofisticados. Su manejo de información es de 8 bits y puede alcanzar velocidades de hasta 2 MHz (con un cristal de 8 MHz). El microcontrolador (MCU) está fabricado con tecnología **IICMOS** (de las siglas en inglés de Semiconductor de Oxido de metal Complementario de alta densidad), lo que le permite brindar en un tamaño muy pequeño circuitos de alta velocidad, bajo consumo de potencia y alta inmunidad al ruido externo. Los dispositivos contenidos en el MCU permiten elaborar diseños de sistemas electrónicos específicos de manera más rápida que tomando elementos por separado, ofreciendo además, beneficios en costos, tamaño funcionalidad y facilidad de manejo. La información presentada en este capítulo fue obtenida de los manuales del fabricante, por lo que si se desea información más profunda, podrá ser consultada en las referencias bibliográficas.

En el chip se incluyen circuitos de auto-monitoreo para la protección contra errores del sistema los cuales son: Un sistema de vigilancia para la adecuada operación del computador (COP) trabaja para evitar las posibles fallas de software en que se incurra, un sistema de monitoreo de reloj que genera un restablecimiento del sistema en caso de que la señal del reloj se pierda o sea muy lenta y un circuito de detección de códigos de operación ilegales que proporcionará una interrupción no mascarable si es detectado un código de operación ilegal.

Otra de las virtudes de este microcontrolador es que cuenta con dos controladores por software WAIT y STOP, para el ahorro de energía y los cuales están disponibles para un control adicional de la potencia. Estos tipos de operación hacen que la familia de microcontroladores MC68HC11 sean especialmente atractivos para la industria automotriz y para aplicaciones de sistemas portátiles.

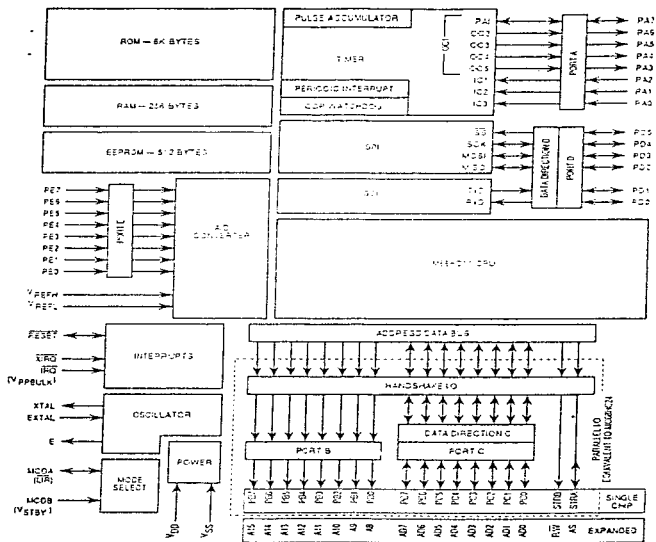


Figura 3.1 Diagrama de bloques del microcontrolador MC68HC11

En la figura 3.1 se muestra el diagrama de bloques del microcontrolador MC68HC11, en él, se presentan los subsistemas de mayor importancia. La mayor diferencia que se encuentra entre los distintos chips de esta familia de microcontroladores radica en sus memorias y el tamaño de estas. Más adelante se presentaran estos distintos tipos.

Las características que presenta el MCU pueden ser básicamente de dos tipos; las de composición electrónica (hardware) y las de su capacidad de instrucciones a ejecutar para programación (software) las características correspondientes a la programación se describirán en el subcapítulo referente al conjunto de instrucciones.

3.3.1 La Unidad Central de Procesamiento (CPU) y Registros

La Unidad Central de Procesamiento es responsable de ejecutar todas las instrucciones de su secuencia programada. El MC68HC11 puede ejecutar todas las instrucciones del MCU M6800 y M6801 (compatibilidad de código fuente) y más de 90 nuevas instrucciones. Contiene más de 256 instrucciones, usa un mapa de código de operación en algunas de las nuevas instrucciones y, que están especificadas con un byte de selección de página antes del byte del código a ejecutar.

Su arquitectura permite especificar todos los periféricos y dispositivos de entrada/salida como localidades de memoria en un mapa de 64 kbytes. No hay instrucciones especiales para diferenciar la entrada/salida de aquellas que son usadas para la memoria. Esta arquitectura es llamada "mapa de memoria", lo cual no incrementa el tiempo de ejecución al acceder un operando desde una localidad de memoria externa.

La CPU ofrece nuevas capacidades comparándola con el primer M6800 y M6801. El cambio más grande es la adición de un segundo registro indexado de 16 bits (Y). Se incluyen nuevas instrucciones en la manipulación de los bits, lo que permite que un bit o un conjunto de bits sea más fácil de manipular en cualquier localidad del espacio de memoria. Además se incluyen dos nuevas instrucciones para dividir 16 bits entre 16 bits. También se permite el cambio del

contenido de cualquiera de los registros índice con el contenido del acumulador doble. Se han reasignado varias instrucciones para realizar una aritmética más completa de 16 bits.

Acumuladores (A, B y D)

Los acumuladores A y B son acumuladores de propósito general de 8 bits que contienen operandos y resultados de cálculos aritméticos o manipulación de datos. Algunas instrucciones utilizan estos dos acumuladores de 8 bits como un acumulador doble de 16 bits (llamado acumulador D). La mayoría de las operaciones utiliza al acumulador A o B de manera intercambiable, sin embargo hay notables excepciones. Las instrucciones ABX y ABY adicionan el contenido de B a los registros X o Y, esto no es válido para A. Las instrucciones TAP y TPA transfieren datos entre A y CCR, pero no hay una instrucción equivalente para B. El ajuste decimal después de haber realizado operaciones aritméticas es diferente entre A y B, mientras que las instrucciones de suma, resta y comparación involucran a ambos operandos (A y B) únicamente en una dirección.

Registros Índice (X y Y)

Los registros X y Y son registros de 16 bits usados para el modo de dirección indexada a los que se les agrega un offset de 8 bits que se incluye en una parte de la instrucción. En otros casos, las instrucciones involucran el registro Y tomando un byte extra del código objeto y un ciclo extra de ejecución comparada a la instrucción usada en el registro X. Las instrucciones de intercambio XGDX y XGDY, ofrecen un camino simple para cargar un valor índice en un acumulador doble, esto permite una capacidad aritmética más poderosa que la de los mismos registros índice. El direccionamiento requiere de menos bytes de código objeto que la correspondiente instrucción usando direccionamiento extendido.

Apuntador de pila (Stack Pointer SP)

La CPU automáticamente soporta a la pila que puede ser localizada en cualquiera de los 64 KBytes del espacio de direcciones. Normalmente el Apuntador de Pila es inicializado por una de las primeras instrucciones en un programa de aplicación. Cada vez que un byte es colocado sobre la pila, el SP automáticamente se decrementa, y cada vez que un byte es extraído de la pila, el SP automáticamente se incrementa. El SP es usado para llamadas de subrutinas, interrupciones y para almacenamiento temporal de datos.

Cuando una subrutina es llamada y se realiza el salto, la dirección de la siguiente instrucción después del salto automáticamente es colocada en la pila. Cuando la subrutina es finalizada, el Apuntador de la Pila proporciona la dirección donde continúa la ejecución del programa.

Siempre que una interrupción ocurre, la instrucción termina y la dirección próxima es almacenada en la pila, todos los registros de la CPU son almacenados en la pila y la ejecución continúa en la dirección especificada por el vector para la interrupción solicitada y que posea la más alta prioridad. Después de completar una rutina de interrupción, una instrucción de retorno es ejecutada y los registros salvados en orden inverso para continuar.

Otro uso común de la pila es almacenamiento temporal de datos. Un ejemplo podría ser que una subrutina utilizara el acumulador A. El usuario podría colocar el acumulador A sobre la pila cuando se ejecute la subrutina y sacarlo justamente antes de abandonar esta subrutina. Es un método simple para asegurar los registros que se utilizarán con el mismo valor al retornar de una subrutina. Lo más importante del apuntador de pila es que es completamente automático.

Contador de Programa (Program Counter PC)

El PC es un registro de 16 bits que contiene la dirección de la siguiente instrucción a ser ejecutada.

Registro de Condición (CCR)

Este registro es de 8 bits, contiene cinco indicadores de estado, dos bits de interrupción mascarable y un bit de paro. Su nombre se debe a los cinco bits de estado, ya que reflejan el resultado de la última instrucción ejecutada en la CPU, por lo que es el registro de mayor uso.

En los primeros M6800 y M6801 no hubo indicadores de interrupción de paro. Los bits de las banderas son los siguientes:

- Acarreo/temporal (C).- Es 1 si existió acarreo durante la última operación.
- Sobreflujo (V).- Es 1 si existió sobreflujo aritmético en la última operación.
- Cero (Z).- Es 1 si la última op. aritmética, lógica o de manipulación de datos es cero.
- Negativo (N).- Es 1 si la última operación aritmética de manipulación de datos o de lógica es negativa.
- Bandera de medio acarreo (M).- Es 1 si existe acarreo entre los bits 3 y 4 de la Unidad Lógica Aritmética para las instrucciones ADD, ABA y ADC.
- Máscara de interrupción (I).- El bit I es fijado por hardware o por instrucción del programa deshabilitando todas las fuentes de interrupción mascarable.
- Máscara de interrupción (X).- El bit X se fija por hardware (reset o XIRQ) y se limpia por programa (TAP o RTI).
- Deshabilitación de paro (S).- El bit S se fija cuando la instrucción de paro fue deshabilitada.

Las banderas de medio acarreo son usadas en operaciones BCD. Los bits de estado N, Z, V y C permiten realizar saltos en función a las operaciones previas.

3.1.2 Las Memorias Internas

Número de parte	EPROM	ROM	EEPROM	RAM	CONFIG 2	Comentarios
MC68HC11A8	-	-	512	256	\$0F	Familia en base a este dispositivo
MC68HC11A1	-	-	512	256	\$0D	A8 con ROM deshabilitada
MC68HC11A0	-	-	-	256	\$0C	A8 con ROM y EEPROM deshabilitada
MC68HC11E9	-	12 K	512	512	\$0F	4 entradas de gran/captura RAM
MC68HC11E1	-	-	512	512	\$0D	E9 con ROM deshabilitada
MC68HC11E0	-	-	-	512	\$0C	E9 con ROM y EEPROM deshabilitada
MC68HC118E2	-	-	2 K	256	\$FF	No ROM para sistema expandido
MC68HC11F1	-	-	512	1 K	\$FF	Alta ejecución no multiplexada
MC68HC711K4	24 K	-	640	768	\$FF	k 4 de una sola programación
MC68HC11L6	-	16 K	512	512	\$0F	E9 con más ROM e I/O. 64/68
MC68HC711L6	16 K		512	512	\$0F	L4 de una sola programación

Tabla 3.1.2.a Miembros de la familia MC68HC11

El microcontrolador MC68HC11 incluye dentro del mismo integrado tres tipos de memorias: una memoria de acceso aleatorio (RAM), una de sólo lectura (ROM) y otra ROM programable y borrrable eléctricamente (EEPROM). En la tabla 3.1.2.a se muestra la relación de algunos tipos de integrados con sus memorias internas y sus capacidades.

La memoria RAM es una memoria de escritura y lectura estática para almacenamiento de información y variables temporales. En la figura 3.1.2.b se muestra un mapa de memoria típica de un MC68HC11.

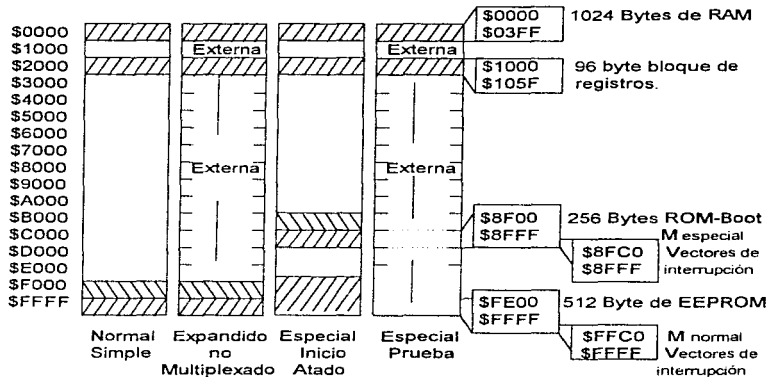


Figura 3.1.2.b Mapa de memoria típico de un MCU MC68HC11

La Memoria ROM.

EL uso principal de la memoria ROM es el de conservar las instrucciones del propio MCU. Las instrucciones son programadas en el microcontrolador durante el proceso de fabricación y no pueden ser cambiadas. Ciertamente el MCU tiene dos memorias ROM internas de forma separada. La más grande es la del usuario, disponible para la grabación de los programas del mismo, la otra tiene capacidad de 192 bytes y es llamada memoria de inicialización o arranque. Esta última sólo se llega a emplear cuando se trabaja al microcontrolador en modo de operación de inicialización.

El programa interno en la ROM puede ser deshabilitado por medio del bit de control basado en la EEPROM del registro de control de la configuración (CONFIG). Cuando el programa de la ROM

es deshabilitado, el espacio de la memoria de los 64 Kbytes de direccionamiento es empleado al máximo y debe ser enclaudrada una memoria externa para las instrucciones del programa.

La memoria RAM.

Esta memoria interna, que por lo regular es de 256 Kbytes, puede ser mapeada en el principio de cualquier bloque de 4 Kbytes en el espacio de direccionamiento de 64 Kbytes. De inicio, la memoria interna RAM está ubicada en las primeras 256 localidades (\$0000 - \$00FF) de los 64 Kbytes del mapa de memoria. En muchos casos (pero no en todos) es buena esa ubicación para la memoria interna RAM. Las primeras 256 localidades en memoria son accesibles utilizando el método de direccionamiento directo, el cual asume que el byte superior de la dirección de 16 bits es \$00.

La posición de la RAM en el espacio de direccionamiento de 64 K es controlado por el registro de RAM y el mapeo de I/O (registro INIT) vía software, lo que permite posicionar la RAM y/o los registros I/O a cualquier página de 4 K en el mapa de la memoria de 64 K.

7	6	5	4	3	2	1	0	
RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT \$103D

RAM3-RAM0. - Posiciona a la RAM en el mapa de memoria. Los cuatro dígitos especifican el dígito hexadecimal de la dirección de la RAM. Si todos son 0 la posición será \$0000 - \$00FF, si todos son 1 sería \$F000 - \$F0FF.

REG3-REG0. Da posición al bloque de registros de 64 bytes. Inicialmente el REG0 tiene un valor de 1 y REG3-REG1 de 0 para la posición inicial \$000-\$103F.

Existen varios propósitos para la función de RAM en espera. En sistemas operados por baterías, las función de RAM en espera proporciona una forma de conservar la limitada potencia de una batería durante tiempos en los cuales el MCU permanece inactivo, lo cual incrementa el tiempo efectivo que el sistema pueda funcionar sin recargar o cambiar la batería. Como se mencionó,

existen dos maneras de ahorrar energía vía software en el sistema, las cuales están ligadas a la función de RAM en espera, ya que minimizan la potencia suministrada al microcontrolador evitando que se pueda llegar a perder la información de la RAM.

La Memoria EEPROM.

Esta otra memoria interna del microcontrolador también varía su capacidad y su presencia, dependerá del modelo de MCU que se tenga. En la tabla 3.1 también se incluye cuáles son los modelos de microcontroladores que la tienen y cuál es su capacidad. El empleo de esta memoria interna es similar al que se le puede dar a la memoria ROM, pero posee algunas posibilidades interesantes que no están disponibles en las memorias ROM y RAM. Un simple ejemplo sería el de almacenar un número de serie único de algún producto terminado en la EEPROM. Una vez que la información ha sido grabada internamente en la EEPROM permanece inalterable siempre, aún y cuando la energía de V_{DD} del chip se interrumpa indefinidamente. A diferencia de la información almacenada en la ROM, la información almacenada en la EEPROM puede ser borrada o reprogramada por medio de un simple control de software. Debido a que las operaciones de programación y borrado utilizan una bomba de carga interna manejada por V_{DD} , no es necesario fuentes de poder adicionales.

El registro PROG controla la programación y el borrado de la memoria EEPROM interna y puede ser leído o sobrescrito en cualquier momento, pero las secuencias de programación y borrado son estrictamente controladas por la lógica para prevenir cambios accidentales en los datos de la EEPROM. A continuación se detallan cada uno de los bits de este registro.

7	6	5	4	3	2	1	0
ODD	EVEN	BYTE	ROW	ERASE	EELAT	EEPGM	PROG

\$103 B

ODD Programa los renglones impares de la matriz de la EEPROM

EVEN Programa los renglones pares de la matriz de la EEPROM

BYTE	Modo de borrado de la EEPROM.
ROW	Modo de borrado de la EEPROM

Estos últimos dos bits (byte y row) especifican el tipo de operación de borrado que se va a realizar y depende del estado del bit ERASE; que si es bajo, no tiene importancia.

BYTE 0	ERASE 1	borra los 512 bytes
BYTE 0	ERASE 0	borra un renglón de 16 bytes
BYTE 1	ERASE 0	borra un bytes
BYTE 1	ERASE 1	borra un bytes

ERASE 1 modo de borrado , 0 modo normal de lectura

EELAT. Controla el latch de la EEPROM

EEPGM. Activa el voltaje (Vpp) de programación de la EEPROM

Resumiendo, en la memoria interna EEPROM en el MCU, existe otro byte adicional de EEPROM (registro CONFIG), que controla algunas características básicas del funcionamiento del MCU. Aun cuando algunos MCU permiten el mapeo de su memoria EEPROM, cuando esta es fija se encuentra en las localidades de \$ B600 a \$ B7FF.

3.1.3 Puertos de Entradas/Salidas en Paralelo

El microcontrolador MC68HC11 tiene un total de 40 patas de entrada/salida (I/O). Todas esas patas son compartidas entre usos de I/O de propósito general y al menos una que otra función de periféricos internos del MCU. El sistema cuenta con un total de 5 puertos: el A, B, C, D y E.

Algunas de las funciones comparativas del puerto A incluyen I/O de propósito general, el sistema principal de temporización y el sistema de acumulador de pulsos. El puerto A tiene tres patas de dirección fija de salida. Cuatro patas de dirección fija de entrada y una pata bidireccional. El puerto A directamente lee desde y escribe sobre el registro PORTA. Los datos significativos

pueden ser leídos desde el puerto A aún cuando sus patas estén configuradas por una función de temporizador alternado o acumulador de pulso. Los datos escritos sobre el puerto A no afectan directamente las patas del puerto configurado para una función de salida de temporizador alternado, ya que los datos son mantenidos en un latch interno.

Las terminales de los puertos B y C, la señal A (STRA) y la señal B (STRB), deben de considerarse juntas debido a que sus funciones dependen básicamente del modo de operación del MC68HC11. Cuando el microcontrolador está operando en modo simple, estas 18 patas son usadas I/O de propósito general y para el sistema de sujeción de entradas-salidas. Cuando el microcontrolador esta operando en modo expandido, estas patas son usadas para el multiplexaje del bus de datos/direcciones. Las funciones de I/O de sujeción y propósito general, las cuales se pierden en el modo expandido, pueden ser recuperadas con el uso de dispositivos externos.

El puerto B es de propósito general de 8 bits con dirección fija de salida. Escrituras sobre el registro de puerto B (PORTB) genera datos que sostenidos y manejados hacia la salida por medio de las patas del puerto B. Lectura sobre el registro PORTB regresa el último dato que fue escrito sobre el puerto B. Cuando el subsistema de sujeción de I/O está operando en modo de generación simple, una escritura al registro PORTB automáticamente genera un pulso en la salida de la pata STRB.

El puerto C es bidireccional de I/O de 8 bits de propósito general. La dirección principal del flujo de datos en cada pata del puerto C es controlada independientemente por el bit correspondiente en el registro de control de dirección de datos del puerto C (DDRR). Además de las funciones normales de I/O del puerto C, hay un latch paralelo independiente de 8 bits que capta los datos por el puerto siempre que un flanco alto sea detectado en la pata de entrada del STRA. Las lecturas sobre PORTCL entregan el contenido del latch del puerto C, en cambio lecturas sobre PORTC entregan el valor corriente desde las patas del puerto C. Escrituras sobre los registros PORTC o PORTCL provocan una escritura de los datos a ser manejados hacia afuera por las patas del puerto C.

El puerto C puede configurarse como OR alambrada por medio de la puesta del bit de control de OR alambrada (CWOM) en el registro PIOC. Siempre que el subsistema de sujeción de I/O sea configurado por un modo completo de sujeción, el puerto C empleado para entrada o salida de datos en paralelos. La para STRA al momento de detectar un pulso provoca que el dato en el puerto C sea almacenado.

El puerto D es de datos bidireccionales de 6 bits de propósito general. Dos patas del puerto D son usadas de manera alterna por el subsistema de INTERFAZ de comunicación serial (SCI). Las restantes cuatro patas del puerto D son usadas de manera alterna por el subsistema de INTERFAZ periférica serial síncrona (SPI). La dirección principal del flujo de datos de cada pata del puerto D es seleccionada por su bit correspondiente en el registro de dirección para el puerto D (DDR D). El puerto D puede ser configurado para una operación OR alambrada.

El puerto E es de 8 bits de dirección fija de entrada. Alternamente sus patas funcionan como canal de entrada del convertidor analógico digital (A/D). Los buffers de entrada del puerto E están diseñados especialmente por lo que toman una excesiva corriente de la fuente cuando sus entradas están siendo manejadas por niveles intermedios.

La figura 3.1.3 muestra todos los registros y bits de control concernientes a las I/O paralelas tratadas. Los registros son mostrados en el orden en que ellos aparecen en el mapa de memoria del microcontrolador. La figura 3.1.3 muestra en número de registros y bits de control mencionados.

BIT 7	-	-	-	-	-	-	BIT 0	PORTA \$ 1000
STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INV B	PIOC \$ 1002
BIT 7	-	-	-	-	-	-	BIT 0	PORTC \$ 1003
BIT 7	-	-	-	-	-	-	BIT 0	PORTB \$ 1004
BIT 7	-	-	-	-	-	-	BIT 0	PORT CL \$ 1005

BIT 7	-	-	-	-	-	-	BIT 0	DDRC \$ 1007
0	0	BIT 5	-	-	-	-	BIT 0	PORTD \$ 1008
0	0	BIT 5	-	-	-	-	BIT 0	DDRD \$ 1009
BIT 7	-	-	-	-	-	-	BIT 0	PORTE \$ 100A
DDR 7	PAEN	PAMOD	PEDGE	0	0	RTRI	RTR 0	PACTL \$ 1026
SPIE	SPE	DWON	MSTR	CPOL	CPHA	SPRI	SPRO	SPCR \$ 1028

Figura 3.1.3 Registro I/O paralelos y sus bits de control

Estos bits de control son utilizados para habilitar otros dispositivos periféricos internos que utilizan las patas de I/O. Las posiciones de los bits etiquetadas con ceros en vez de nombres, indican bits que no han sido implantados y que su lectura siempre será cero.

3.1.4 Interfaz Periférica Serial Síncrona (SPI)

Las interfaces periféricas serial síncrona es uno de los subsistemas de comunicación serial independiente que incluye en el microcontrolador. Como su nombre lo implica el SPI es usado principalmente para permitir la comunicación del MCU con los dispositivos periféricos. El SPI también es capaz de efectuar comunicaciones interprocesador en un sistema maestro múltiple. La flexibilidad del SPI le permite trabajar con distintos tipos de dispositivos periféricos de diversos fabricantes. El subsistema puede configurarse como maestro o como esclavo. La velocidad de transmisión para el caso de configuración maestro alcanza 1 Mbits/s y para el caso de configuración esclavo la velocidad es de 2 Mbits/s.

Durante una transferencia, los datos son simultáneamente transmitidos (turnados serialmente hacia fuera) y recibidos (turnados serialmente hacia adentro). Una línea de reloj serial sincroniza el muestreo y el turno de información sobre las dos líneas de datos seriales. Una línea de selección

través de una pata de datos mientras que otro carácter de 8 bits es turnado hacia adentro por una de las patas de datos. Otra manera de ver esta transferencia es considerado un registro de corrimiento de 8 bits en un maestro y otro registro de 8 bits en un esclavo, están conectados como un registro de corrimiento circular de 16 bits.

El elemento central en el SPI es el bloque que contiene el registro de corrimiento (8-bits shift register) y el almacén temporal de lectura de datos (read data buffer). El sistema tiene un almacenamiento temporal simple en la transmisión y un temporal doble en la recepción de datos. Esta implica que un nuevo dato para transmisión no puede ser escrito en el registro hasta que la transacción previa sea completada; sin embargo los datos recibidos son transferidos al almacén temporal paralelo para dejar libre al primer registro para un segundo carácter serialmente.

El registro de control del SPI (SPCR), el registro de estado del SPI (SPSR) son registrados usados para configurar y operar el sistema SPI. Otro registro que influye en las actividades del SPI es el registro de control de dirección del puerto D (DDR D). A continuación se describen los dos primeros registros.

7	6	5	4	3	2	1	0	
SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPO0	SPCR \$1028

SPIE	Habilitación de interrupción del SPI. 0 interrupción deshabilitada.
SPE	Habilitación del SPI. 0 deshabilita al sistema 1 lo habilita.
DWOM	Selección de OR alambrada en el puerto D.
MSTR	Modo de selección de maestro/esclavo. = configura la SPI como esclavo, al configurar al SPI como maestro.
CPOL	Selecciona la polaridad del reloj. = selecciona reloj en nivel alto, 1 selecciona reloj en nivel bajo.
CPHA	Selecciona la fase del reloj.
SPR1, SPR0	Selecciona la velocidad del bit del SPI.

7	6	5	4	3	2	1	0	SPSR \$1029
SPIF	WCOL	-	MODF	-	-	-	-	

SPIF Bandera de transferencia completa de SPI.

WCOL Bandera de error de colisión en escritura.

MODF Bandera de configuración.

3.1.5 Interface Periférica Serial Asíncrona (SCI)

El transmisor receptor universal asíncrono (UART) es un tipo de sistema de interface de comunicación serial, el cual es otro subsistema independiente en el MC68HC11. El subsistema SPI de I/O proporciona una alta velocidad de comunicación sincrónica con unidades periféricas u otros microcontroladores, comúnmente en el mismo circuito impreso que el MCU. El sistema SCI se puede conectar a una terminal (CRT), a una computadora personal o a una de comunicación serial. El SCI es un sistema asíncrono tipo UART full dúplex, que emplea el formato de comunicación estándar de no retorno a cero (NRZ, un bit de comienzo, 8 ó 9 bits de datos y otro bit de paro). Tanto el transmisor como el receptor tiene una unidad doble de almacenamiento temporal de datos, lo que permite manejar con facilidad caracteres uno tras otro, aún cuando el CPU se retrase en la respuesta de cada carácter individual. Es importante mencionar que se tiene que proporcionar sistemas externos para hacer la transferencia de los niveles usados en RS232 o RS422 (típicamente ± 12 v) a los niveles lógicos de 0 a 5 V, manejandos por el controlador.

El Transmisor SCI

El transmisor del SCI (figura 3.1.5.a) utiliza un reloj interno de generación de velocidad de bit para evitar los datos por la pata TxD de forma serial. Una transmisión normal se inicia cuando se habilita el transmisor (poniendo en 1 a la bandera TE del registro de control SCCR2) y escribiendo los datos a ser transmitidos en el registro SCDR. Debido al almacén temporal de

datos doble que posee, un nuevo carácter puede ser escrito en el transmisor siempre que la bandera TDRE esté en 1.

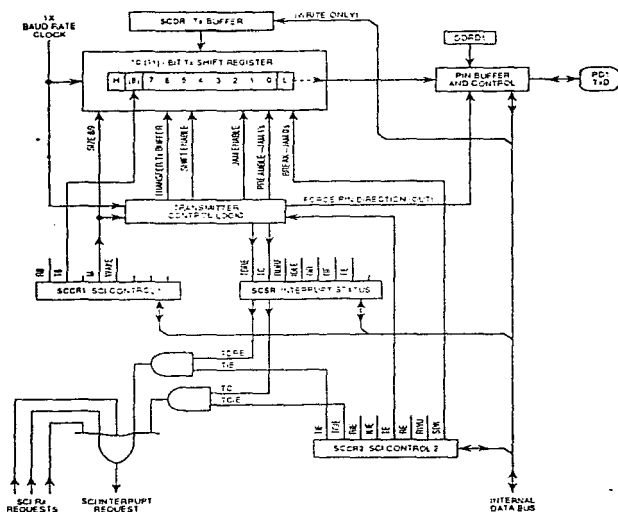


Figura 3.1.5.a Diagrama de bloques del transmisor del SCI

El corazón del transmisor es el registro de corrimiento serial. Comúnmente este registro toma el dato del almacén de transmisión de sólo lectura. Los datos entran en el almacén temporal de transmisión cuando el software escribe sobre el registro de datos del SCI.

El Receptor de SCI

Los datos recibidos por el SCI vienen de la pata RxD, y manejan el bloque de recuperación de datos, el cual es en sí, un registro de corrimiento de alta velocidad operado a 16 veces la velocidad de bit. El corazón del registro del receptor es el registro de corrimiento de recepción.

El receptor del sistema SCI es el responsable de la sincronización de la cadena de datos serial y de la recuperación de los caracteres de datos. Debido a que la cadena no contiene el reloj, la recuperación de los datos depende del dispositivo de transmisión y de que la velocidad de operación del receptor le sea muy cercana. El sistema SCI puede tolerar una cantidad moderada de ruido en el sistema sin perder nada de información. Las funciones de recepción del SCI son, en cierta medida, más complicadas que las transmisiones debido a la naturaleza asincrónica de los seriales ingresados.

Básicamente el sistema SCI está configurado y controlado por cinco registros (BAUD, SCCR1, SCCR2, SCSR y SCDR), además del registro del puerto D (DDR D) por la configuración que se asigna al puerto por medio de este último registro. El registro BAUD es usado para seleccionar la velocidad de operación del SCI. El registro SCCR1 incluye tres bits asociados con la operación del formato de datos opcional de 9 bits. En el caso del registro SCSR contiene dos banderas de estado de transmisión y 5 banderas relativas a la recepción. El registro SCDR es realmente dos registros separados, el TDR que es un registro de transmisión de datos de sólo escritura y el RDR que es un registro de recepción de datos de sólo lectura. El registro 2 de control del SCI (SCCR2) es el registro de control principal para el sistema SCI por lo que a continuación se detalla un poco más.

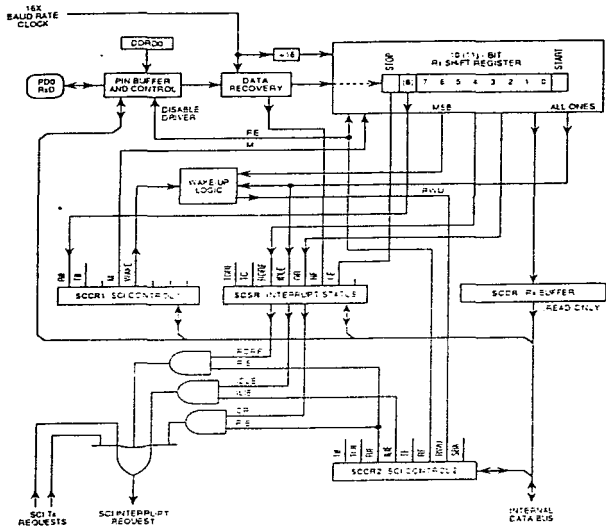


Figura 3.1.5 b Diagrama de bloques del receptor del SCI

7	6	5	4	3	2	1	0	
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2 \$102D

TIE	Habilitación de interrupción de transmisión. 0 Deshabilitación de interrupción.
TCIE	Habilitación de interrupción de transmisión completa o deshabilitación.
RIE	Habilitación de interrupción de recepción. 0 deshabilitación de interrupciones RDRF y OR1 interrupción de SCI requerida cuando RDRF u OR están en 1.
ILIE	Habilitación de interrupción de línea desocupada, 0 deshabilitación de interrupción IDLE, 1 si IDLE está en 1, una interrupción del SCI es requerida.
TE	Habilita transmisión. 0 deshabilitación, 1 habilitación.
RE	Habilita recuperación. 0 deshabilitación, 1 habilitación.
RWU	Característica Wake Up en el receptor.

3.1.6 Convertidor Analógico Digital

EL sistema de conversión analógico digital del MCU68HC11, utiliza la técnica de redistribución de cargas capacitivas para la conversión. El sistema de conversión A/D, es un convertidor de aproximaciones sucesivas de 8 canales de 8 bits, con una precisión de +/- 1/2 del bit menos significativo para todo el rango de temperatura. Debido al uso de la técnica de la redistribución de cargas, no es necesario de circuitos externos de muestreo y retención.

Técnica de Conversión de Redistribución de Cargas para Conversión A/D

La figura 3.1.6.a muestra el circuito simplificado para llevar a cabo una conversión A/D de 4 bits por aproximaciones sucesivas empleando la técnica de redistribución de cargas. El circuito que contiene el MC68HC11 contiene varios cambios con respecto al circuito mostrado, esto con el fin de obtener una mejor calidad y manufactura más simple. Puesto que la técnica de redistribución de cargas depende de la porción de la capacitancia, que de los valores absolutos de capacitancia, los capacitores de la figura 3.1.6.a están marcados en unidades.

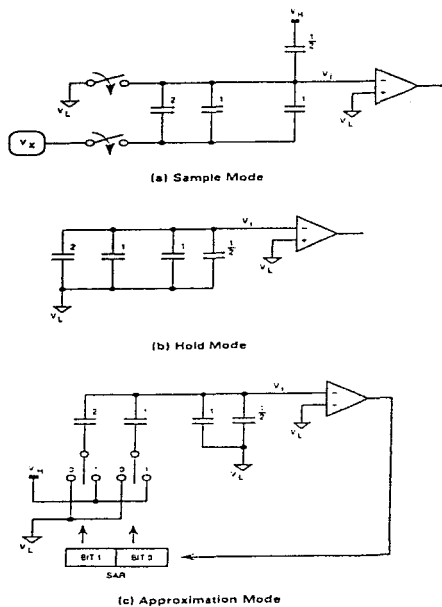


Figura 3.1.6.a Conversión A/D por redistribución de carga básica

Durante el tiempo de muestreo figura 3.1.6.a (a), la placa superior de los capacitores es puesta a un voltaje V_L (0.0 V) y las placas inferiores están conectadas a una señal

de entrada analógica desconocida, V_x . Usando una simple relación que Q_s es igual a CV , así entonces, el total de carga puede ser calculada por:

$$Q_s = 16(V_x - v_i)$$

A menos de que se establezca otra cosa, se asume que V_L es igual a 0.0 V ; por lo tanto:

$$Q_s = 16 V_x$$

Ahora el circuito cambia a el estado de retención por medio de interruptores analógicos controlados lógicamente (figura 3.1.6.a (b)). Para este caso, las placas superiores se desconectan de V_L , y las placas inferiores pasan de V_x a V_L . La carga se describe ahora como:

$$Q_s = (V_L - V_i)16$$

Puesto que $V_L = 0$

$$Q_L = - 16 V_i$$

Debido a que la carga se conserva, Q_s es igual a Q_L ; por lo tanto,

$$16 V_x = - 16 V_i$$

$$V_x = - V_i$$

$$V_i = - V_x$$

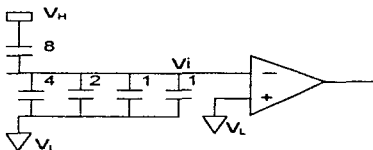
el cual es el voltaje inicial a la entrada del comparador. Finalmente en porción de conversión del proceso A/D, cada capacitor, comenzando por el más grande el cual le corresponde al del bit más significativo (MSB) del resultado digital, es puesto del voltaje V_L (0.0 V) a V_H . La salida del comparador después de que cada capacitor es cambiado, determina cual de las placas de los capacitores inferiores permanecerá a V_H o regresará a V_L antes de que el capacitor próximo sea cambiado.

A continuación se presenta un ejemplo de una secuencia de conversión para una entrada analógica, donde $V_x = 21/32 (V_H)$.

Durante el tiempo de muestreo, los capacitores alcanzan una carga total, dada por:

$$Q_s = 16 V_x = (16)21/32 (V_H) = 21/2 V_H$$

Durante la retención, la entrada negativa del comparador (V_i) pasa a V_x o $-21/32 V_H$. A continuación el capacitor de ocho unidades pasa por V_L a V_H , resultando el siguiente circuito:



La carga esta dada de la siguiente forma:

$$Q = 8(V_H - V_i) + 8(V_L - V_i)$$

pero como $V_L = 0$, entonces:

$$Q = 8 V_H - (V_i - 8 V_i)$$

$$Q = 8 V_H - 16 V_i$$

Por la conservación de carga, esta carga pasa a un valor igual de la carga original durante el tiempo de muestreo:

$$21.2 V_H = 8 V_H - 16 V_i$$

Resolviendo para V_i produce el siguiente resultado:

$$16 V_i = 8 V_H - 21/2 V_H$$

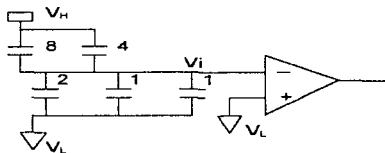
$$V_i = 1/2 V_H - 21/32 V_H$$

$$V_i = - 5/32 V_H$$

el cual es negativo.

Puesto que la salida del comparador es un 0 lógico, el capacitor de ocho unidades permanece conectado a V_H para el siguiente paso. También el bit 3 del registro de aproximaciones sucesivas (SAR) es puesto a uno analógico. Después de la secuencia de conversión, el registro de aproximaciones sucesivas tiene el equivalente digital de la entrada analógica original.

A continuación el capacitor de 4 unidades pasará de V_L a V_H y tendremos el siguiente circuito:



La carga está dada de la siguiente forma:

$$Q = 8(V_H - V_i) + 4(V_H - V_i) + 4 V_i$$

$$Q = 8 V_H - 8 V_i + 4 V_H - 4 V_i + 4 V_i$$

$$Q = 12 V_H - 16 V_i$$

Para la conservación de carga, esta carga es igual a la carga original:

$$21/2 V_H = 12 V_H - 16 V_i$$

Resolviendo para V_i , obtenemos el siguiente resultado:

$$16 V_i = 12 V_H - 21/2 V_H$$

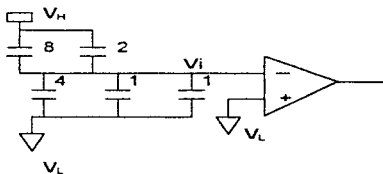
$$16 V_i = 3/2 V_H$$

$$V_i = 3/32 V_H$$

el cual es positivo.

La salida del comparador es un cero lógico; sin embargo, el capacitor de 4 unidades regresa a V_L antes del siguiente paso. También, el bit dos del registro de aproximaciones sucesivas pasa a ser cero.

A continuación, el segundo capacitor pasará a V_L a V_H dando por resultado el siguiente circuito:



La carga se escribe de la siguiente forma:

$$Q = 8(V_H - V_i) + 2(V_H - V_i) - 6 V_i$$

$$Q = 8 V_H - 8 V_i + 2 V_H - 2 V_i - 6 V_i$$

$$Q = 10 V_H - 16 V_i$$

Colocando esta carga igual a la carga original por conversión de carga tendremos el siguiente resultado:

$$21/2 V_H = 10 V_i - 16 V_i$$

$$16 V_i = 10 V_H - 21/2 V_H$$

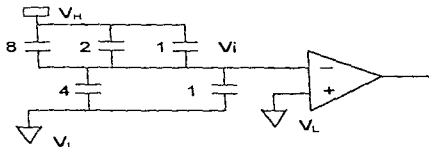
$$16 V_i = -1/2 V_H$$

$$V_i = 1/32 V_H$$

el cual es negativo.

Puesto que la salida del comparador es ahora un uno lógico, el capacitor de 2 unidades permanece conectado a V_H para el siguiente paso, y el bit uno del registro de aproximaciones sucesivas está ahora en uno.

En cuanto al último paso en la secuencia de conversión, el capacitor uno pasará de V_L a V_H . El capacitor de 2 unidades permanece conectado a V_L durante toda la secuencia de conversión. El siguiente circuito muestra el último paso de la conversión :



La carga esta dada como sigue:

$$Q = 8(V_H - V_i) + 2(V_H - V_i) + 1(V_H - V_i) - 5V_i$$

$$Q = 8 V_H + 2 V_H + 1 V_H - 8V_i - 2V_i - 1V_i - 5V_i$$

$$Q = 11 V_H - 16V_i$$

Por la conservación de cargas tenemos el siguiente resultado :

$$21/2 V_H = 11 V_H - 16V_i$$

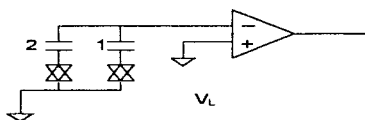
$$16V_i = 11 V_H - 21/2 V_H$$

$$16V_i = 1/2 V_H$$

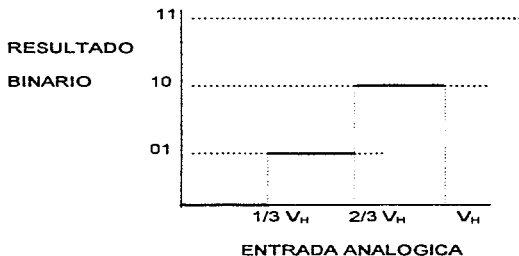
$$V_i = 1/32 V_H$$

El cual es positivo. Puesto que la salida del comparador es un cero lógico, el bit menos significativo del registro de aproximaciones sucesivas pasa a ser cero. Dado que la conversión esta completa, no es necesario regresar el capacitor uno a V_L . El resultado digital de ésta conversión es 1010_2 . En el ejemplo anterior, un voltaje de entrada analógico de $21/32 V_H$ produce un resultado de 1010_2 o $10/16 V_H$. Un error ocurre aún y cuando fueron asumidas componentes y

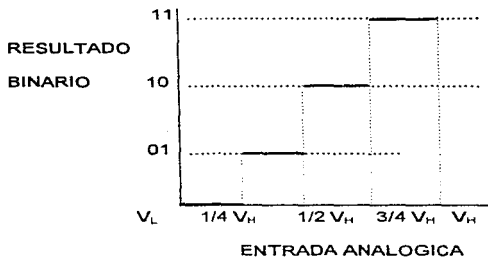
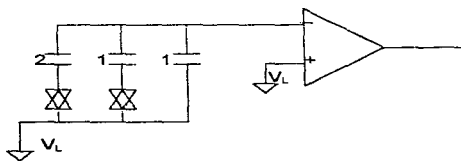
condiciones ideales. Un análisis para un convertidor A/D de 2 bits explica la razón. Si el capacitor de 2 unidades es omitido, tendremos el siguiente circuito:



La característica de transferencia de el circuito es la siguiente :



Sin el capacitor de una unidad extra, cada unidad de capacitancia corresponde a una tercera parte del total en vez de la correspondiente cuarta parte. Cuando es agregado el capacitor de 2 unidades, resulta el siguiente circuito con las características de transferencia:



Una entrada analógica de $1/4$ de V_H produce un resultado digital de 01₂ o $1/4 V_H$, pero una entrada analógica de $1/8$ de V_H produce un resultado digital de 00₂ o $0.0 V_H$, lo cual equivale a un error de $1/8$ de V_H o $1/2$ LSB.

Implementación del convertidor A/D en el MC68HC11.

El convertidor A/D en el MC68HC11 está compuesto por un convertidor A/D de redistribución de carga de aproximaciones sucesivas sencillo y por circuitería de control digital. La sección analógica es un poco más compleja que la descrita previamente pero los principios de

redistribución son idénticos. La sección digital contiene la lógica que hace trabajar al A/D como un sistema con el resto del MCU.

Convertidor A/D de aproximaciones sucesivas.

La figura 3.1.6.b muestra el convertidor A/D de aproximaciones sucesivas del MC68HC11 en el modo de muestreo. A diferencia de los ejemplos anteriores, el arreglo de capacitores tiene un capacitor en serie (C_s) separando la mitad de bajo orden de la mitad de alto orden. Este capacitor efectivamente divide el valor de los capacitores de bajo orden a la izquierda por 16, lo que simplifica el arreglo y asegura una mejor combinación de los tamaños de las capacitancias.

Excepto por los elementos de media y de 1.1 unidades, todos los capacitores están compuestos por grupos conectados de capacitores de una unidad, lo que minimiza errores causados por el dimensionamiento y procesamiento. En el MC68HC11, éstos capacitores consisten de placas de polisilicón separadas por un nivel de óxido, las placas inferiores están aisladas del sustrato por una segunda capa de óxido. La capacitancia de esta estructura no está sujeta a variaciones ocasionadas por voltajes en las placas relativas al sustrato.

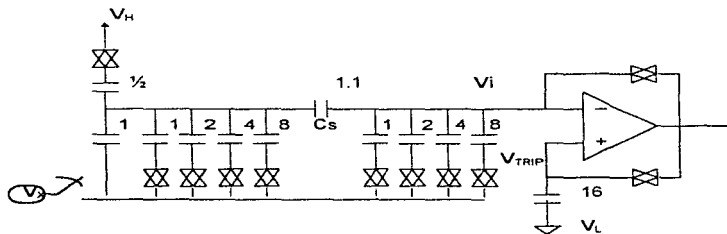


Figura 3.1.6.b Convertidor A/D de aproximaciones sucesivas del MC68HC11.

En el convertidor del MC68HC11, tanto las entradas como las salidas del comparador son puestas en corto durante el tiempo de muestreo, lo que causa que los tres puntos sean estables al voltaje de disparo del comparador; entonces, los offsets del comparador son cancelados efectivamente de los cálculos. Después del periodo de muestreo, las trayectorias en corto son desconectadas, y las secuencias de conversiones proceden como en los primeros ejemplos. El capacitor de 16 unidades de la entrada positiva del comparador a VL no es crítica en términos de dimensión porque es usada solo para mantener la entrada positiva al voltaje V_{TRIP} durante la conversión.

Bomba de Carga para el A/D y el Oscilador Resistor-Capacitor (RC).

La bomba de carga desarrolla aproximadamente de 7 a 8 volts, y este alto voltaje es usado para manejar las compuertas de los interruptores analógicos en el multiplexor y el arreglo capacitivo a la entrada. Este alto voltaje de compuerta asegura una fuente baja para drenar impedancia para señales analógicas con voltajes de V_{CC} (aproximadamente 6V), el convertidor permanecerá entregando buenos resultados.

La bomba de carga del convertidor A/D esta deshabilitada después del reset y es encendida estableciendo el bit de control de encendido (ADPU) en el registro de control OPTION antes que el sistema del convertidor A/D pueda ser usado. Después de encender el ADPU es requerido un retraso esto para permitir que la bomba de carga y los circuitos del comparador se establezcan antes de usar el sistema convertidor.

El proceso de conversión A/D por redistribución de carga es un proceso dinámico en el que la carga en el arreglo capacitivo se dispersará eventualmente. Este arreglo capacitivo es parte de un convertidor digital-analógico interno (DAC), lo cual indica que el proceso de conversión debe ser complementado dentro de un tiempo razonable después de que el tiempo de muestreo finalice. La otra circuitería del MCU es estática para permitir frecuencias de reloj muy bajas, de tal forma que ahorra energía. Para frecuencias de bus (E clock) por debajo de 750 KHZ, el reloj E deberá ser usado normalmente como el reloj de la conversión A/D porque hay un riesgo de error ocasionado por la dispersión de carga en temperaturas extremas. Pruebas de laboratorio indican un buen

desempeño para rangos del reloj E por debajo de los 10 KHz, pero esta especificación ha sido protegida contra variaciones en el proceso.

Un oscilador RC interno provee fuentes alternativas de reloj para el sistema A/D cuando el reloj E tiene un desempeño muy bajo y así se aseguran conversiones adecuadas. Esta fuente de reloj es seleccionada colocando un uno en el bit de control de selección de reloj (CSEL) en el resto de control OPTION. La frecuencia del oscilador RC varía con el procesamiento pero esta es típicamente de 2 MHz.

Cuando el reloj E está siendo usado como la fuente de reloj para el convertidor A/D, la secuencia de conversión está sincronizada inherentemente al reloj principal del CPU. El uso del reloj E, el cual es asíncrono al reloj del sistema tiene dos ventajas sobre el oscilador RC. La primera, la salida del comparador es muestreada en tiempos inactivos del ciclo de reloj del sistema, esto reduce efectos de ruido interno para el MCU. Cuando el oscilador RC está siendo usado, hay más error atribuido al ruido interno del reloj del sistema. Segunda, la actualización del registro de resultado ocurre automáticamente durante un intervalo del reloj del sistema, donde no se efectúan lecturas; así que, una actualización no interfiere con una lectura. Cuando el oscilador RC es usado, no existe conflicto entre lecturas y actualizaciones, pero existe un retraso adicional de sincronización impuesto al término de cada conversión de canal para permitir la sincronización con el reloj E del sistema.

En la siguiente ilustración se muestra el registro de control OPTION para referencia puesto que los bits de control ADPU y CSEL afectan al sistema de conversión A/D.



El bit de control CSEL también selecciona una fuente de reloj alternativa para la bomba de carga de la EEPROM en el circuito integrado. Esta bomba de carga está separada de la bomba de carga del convertidor A/D. CSEL requiere ser uno cuando el reloj E es muy lento para asegurar que la

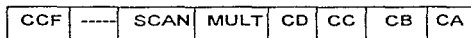
secuencia de aproximación sucesiva termine antes de que ocurra una pérdida significativa de carga. Cuando el reloj E está por encima de 2 MHz, CSEL deberá ser siempre cero; cuando el reloj E esté por debajo de 750 KHz, CSEL deberá ser siempre uno.

Para frecuencias del reloj E entre 750 KHz y 2 MHz, CSEL deberá ser uno para las operaciones de programación y borrado de la EEPROM de esta forma la bomba de carga de la EEPROM trabaja más eficazmente; de cualquier modo, CSEL deberá ser cero en las conversiones A/D para asegurar una alta precisión por la reducción de los efectos de ruido del circuito integrado. En la mayoría de las aplicaciones, no es necesario encender y apagar CSEL. En cambio, puede hacerse un intercambio sobre los requerimientos de la aplicación.

EL sistema A/D del MC68HC11 consiste de un convertidor A/D de aproximaciones sucesivas, un multiplexor a la entrada para seleccionar uno de los 16 canales (incluyendo los 8 canales del MCU), y sofisticada circuitería de control integrados para configurar y controlar las conversiones. Están incluidos cuatro registros de resultados dentro de la lógica de control para implementar las secuencias automáticas de conversión en un canal seleccionado durante cuatro tiempos o en cuatro canales (uno a la vez). Las secuencias de conversión están configuradas para repetirse continuamente o para hacer una de cuatro y después parar. Cuando el MCU trabaja a frecuencias muy bajas, un oscilador RC es seleccionado para continuar con la operación de conversión.

Registro control/estado del A/D

Todos los bits de este registro excepto el 7 pueden ser leídos y escritos, así que este es un indicador de estado de solo lectura, y el bit 6 el cual siempre se lee un cero. Sólo el bit 7 se borra con el reset, no así los otros que no los afecta.



U O U U U U U

ADCTL \$1030

RESET

Donde:

CCF-- Bandera de conversión completa. Este indicador de estado de sólo lectura se habilita cuando todos los registros de resultados contienen resultados válidos de conversiones. Cada vez que se escribe el registro ADCTL, este bit se borra automáticamente, y se inicia una nueva secuencia conversión inmediatamente. En el modo de exploración continuo, las conversiones continúan una tras la otra, y los registros son actualizados con datos actuales aún y cuando el bit CCF permanezca establecido.

Bit 6-- No está implementado, siempre es cero.

SCAN--Control de exploración continua. Cuando este bit es cero, las cuatro conversiones solicitadas se ejecutan, una a la vez, para llenar los 4 registros de resultados. Cuando este bit es uno, las conversiones continúan una tras la otra con los registros de resultados actualizados tan pronto se produce un nuevo dato.

MULT--Control de canal individual o múltiple. Cuando este bit es cero, el sistema A/D está configurado para efectuar cuatro conversiones consecutivas en un canal especificado por los cuatro bits de selección de canal (CD - CA del registro ADCTL). Cuando este bit es uno, el sistema A/D está configurado para efectuar conversiones en cada canal en el grupo de cuatro canales especificados por los bits de selección de canal CD y CC. En este modo de canal múltiple, cada canal es asociado con un registro de resultado específico.

Selección de canal CD, CC, CA

Estos cuatro bits de selección de canal, son usados para especificar el canal o los canales que serán operados en un operación de conversión A/D.

CD	CC	CB	CA	Señal de Canal	Resultado en ADRx si MULT = 1
0	0	0	0	PE0	ADR1
0	0	0	1	PE1	ADR2
0	0	1	0	PE2	ADR3
0	0	1	1	PE3	ADR4
0	1	0	0	PE4	ADR1
0	1	0	1	PE5	ADR2
0	1	1	0	PE6	ADR3
0	1	1	1	PE7	ADR4
1	0	0	0	Reservado	ADR1
1	0	0	1	Reservado	ADR2
1	0	1	0	Reservado	ADR3
1	0	1	1	Reservado	ADR4
1	1	0	0	VH	ADR1
1	1	0	1	VL	ADR2
1	1	1	0	½ HV	ADR3
1	1	1	1	Reservado	ADR4

Tabla 3.1.6.c Asignación de canales para el A/D

La tabla 3.1.6.c muestra la relación entre los bits CD - CA y el canal o canales a estar en operación. Cuando el modo de canal múltiples es seleccionado (MULT = 1), la selección de CB y CA no tiene efecto, y el grupo de 4 canales afectados son seleccionados por CD y CC.

Registros de resultados (ADR4 - ADR1) para el A/D.

El registro de resultados en el A/D está compuesto por registros de sólo lectura y son usados para retener un resultado de conversión de 8 bits. Después de que éstos han sido llenados con datos válidos de una secuencia de conversión, el estado del registro CCF es establecido para indicar que

los resultados son válidos. Son entonces calculados resultados de conversión nuevos en la lógica del A/D y son transferidos dentro de los registros de resultados en una parte del ciclo de reloj en donde lecturas no se efectúan. Sin embargo, no ocurre interferencia alguna entre lecturas de software y actualizaciones de resultados.

3.1.7 Reloj Principal e Interrupción de Tiempo Real.

En esta sección se describirá el reloj principal del sistema del MC68HC11. Todas las principales divisiones del reloj en el MCU están enlazadas y se derivan del oscilador a los generadores de baud-rate, lo cual ayuda y a su vez enlaza el contador de reloj al resto del sistema.

Descripción general

Este sistema de reloj está basado en un contador de 16 bits de carrera libre con un preescalador programable de cuatro etapas. Una función de sobreflujo del reloj permite por medio de software extender la capacidad de reloj del sistema más allá de los 16 bits del contador. Tres funciones independientes de captura de entrada para grabar automáticamente el tiempo cuando una transición seleccionada es detectada en una terminal de entrada del reloj. Cinco funciones de comparación de salida están incluidas para generar señales de salida o para detectar retrasos de tiempo de software.

Un circuito de interrupción periódica programable llamado interrupción en tiempo real (RTI) se deriva del contador del reloj principal de 16 bits. Por medio de software se selecciona uno de los cuatro valores para el RTI, el cual es más comúnmente usado para marcar el paso en las ejecuciones de rutinas de software.

La función COP (Computer Operating Properly) está poco relacionada a el reloj principal, porque la entrada de reloj para el COP esta derivada de un conjunto de contadores de carrera libre.

comparación de salida tiene su propio registro de comparación de 16 bits. Todas las funciones de reloj, incluyendo el sobreflujo para el reloj y el RTI tienen sus controles de interrupción propios y vectores de interrupción por separado. Bits de control adicionales permiten por medio de software el control de los flancos de disparo de cada función de captura de entrada y las acciones automáticas que resultan de las funciones de comparación de salida.

Diagrama general de bloques del reloj.

En la figura 3.1.7.a se muestra un diagrama de bloques del reloj principal. El bloque de control de las terminales del puerto A incluye lógica para funciones de reloj y funciones de E/S de propósito general. En el bloque que forman las terminales PA0, PA1 y PA2, está contenida la lógica de detección de flanco y la de control que permiten al usuario seleccionar los flancos que dispararán una captura de entrada. El nivel digital de estas terminales puede ser leído en cualquier momento aún si las terminales están siendo usadas en la función de captura de entrada. Las terminales PA6 - PA3 son usadas como salidas de propósito general o como terminales de comparación de salidas.

Cuando una de estas terminales esta siendo usada para una función de comparación de salida, no se puede escribir en ella directamente como en una salida de propósito general. Cada una de las funciones de comparación de salida (OC5-OC2) está relacionada con una de las terminales de salida del puerto A. Una terminal de comparación de salida tiene una lógica adicional, que permite controlar las combinaciones de las terminales PA3-PA7. PA7 es usada como E/S de propósito general, como entrada al acumulador de pulsos o como salida de comparación OC.

Concepto de captura de entrada.

La función de captura de entrada es un elemento fundamental de la arquitectura del reloj del MCU. Para el MCU, el tiempo físico está representado por la cuenta en el contador de carrera libre de 16 bits. Este contador es el elemento central en el sistema de reloj principal. Las funciones de captura de entrada, usadas para grabar el tiempo en el cual ocurre un evento externo, son

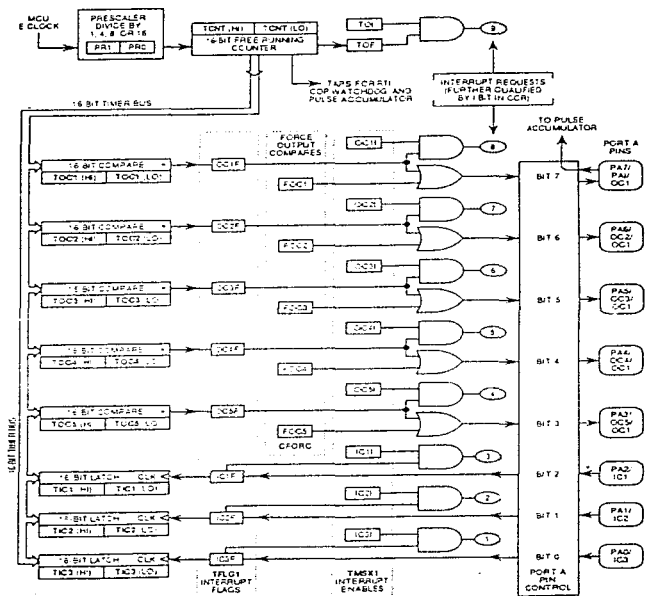


Figura 3 1.7.a Diagrama de bloques del reloj principal.

realizadas por el almacenamiento de los contenidos del contador de carrera libre cuando un flanco seleccionado es detectado en una terminal de entrada de reloj.

El tiempo para el cual ocurrió el evento es salvado en el registro de captura; por lo tanto, aunque puede tomar un tiempo indeterminado para responder al evento, el software puede informar cuando se llevo a cabo el evento. Por medio de los registros de los tiempos para los flancos sucesivos o señales de entrada, el software puede determinar el período y/o el ancho de pulso de la señal. Para medir su periodo se requiere, cuando menos, que dos flancos sucesivos de la misma polaridad sean capturados. Para la medición de el ancho de un pulso, se requiere sean capturados dos flancos de polaridad alterna.

Otro importante uso de las funciones de captura de entrada es la de establecer una referencia de tiempo. Para este caso, una función de captura de entrada es usada junto con una función de comparación de entrada para realizar la tarea. Por ejemplo, si el usuario desea activar una señal de salida cierto número de ciclos de reloj después de que se detectó un evento de entrada (un flanco), la función de captura de entrada deberá ser usada para grabar el tiempo al cual ocurrió el flanco. La cantidad en número del retraso deseado, deberá ser sumado al tiempo capturado y entonces será guardado en un registro de comparación de salida. Debido a que ambas señales de (captura de entrada y comparación de salida) están referenciadas al contador de 16 bits, el retraso puede ser controlado por la resolución del contador de carrera libre independientemente del software.

Concepto de comparación de entrada.

La función de comparación de entrada es también un elemento fundamental de la arquitectura del sistema de reloj del microcontrolador MCU; para éste, el tiempo físico es representado por la cuenta de un contador de 16 bits de carrera libre. Este contador es el elemento central en el sistema de reloj principal.

Las funciones de comparación de salida son usadas para programar una acción que ocurre a un tiempo especificado. Para cada una de las funciones de comparación de salida hay un registro de comparación de 16 bits por separado y también un comparador dedicado de 16 bits. El valor del registro de comparación es comparado con el valor del contador de carrera libre en cada ciclo de bus. Cuando el valor del registro de comparación iguala al valor del contador, se genera una salida la cual establece una bandera de estado de comparación de salida e inicializa las acciones automáticas para esa función de comparación de salida. Acciones automáticas opcionales pueden ser iniciadas por una comparación de salida tales como requerimientos de interrupción por hardware y cambios de estado en las terminales de salida asociadas al reloj.

Una de las aplicaciones más sencillas de la función de comparación de salida es la de producir un pulso de duración especificada. Primero un valor correspondiente a el flanco de entrada del pulso es escrito en el registro de comparación de salida. La comparación de salida se configura para establecer automáticamente la correspondiente salida baja o alta dependiendo de la polaridad del pulso a ser producido. Después de que ocurre esta comparación, la comparación de salida es reprogramada para automáticamente regresar la terminal de salida a su estado de nivel inactivo para la próxima comparación. Un valor que corresponde al ancho del pulso se suma al valor original del registro de comparación de salida. Debido a que los cambios de estado de las terminales ocurren automáticamente a valores específicos del contador, el ancho del pulso puede ser controlado con precisión a la resolución del contador.

Contador de carrera libre y el pre-escalador.

Este contador comienza su cuenta a partir de \$0000, después del reset al MCU y continua contando ascendente continuamente. Cuando alcanza la cuenta máxima de \$FFFF, el contador regresa a la cuenta a partir de \$0000, establece una bandera de sobreflujo y continúa su cuenta ascendente. Mientras el MCU trabaje en modo normal, no hay medio alguno para reinicializar, cambiar o interrumpir el conteo del contador. El contador puede ser leído a cualquier tiempo para conocer que tiempo se lleva. Todas las actividades del reloj principal del sistema

están referenciadas a este contador; por lo tanto, todas las funciones de reloj están relacionadas unas con otras.

El registro de conteo TCNT del reloj se lee usando una instrucción de lectura de doble byte como Load (LDD) o Load X (LDX). La mitad baja del contador pasa a través de un buffer hacia el registro TCNT. Cuando la mitad baja del contador es leída usando una instrucción de lectura de un byte, el valor que regresa es sencillamente el valor de los 8 bits de menor orden del contador de reloj principal. Cuando el byte más significativo del registro TCNT es leído, se inhibe el buffer de el byte de menor orden del registro TCNT, esto sucede por un ciclo de bus. A continuación se muestra el registro TCNT.



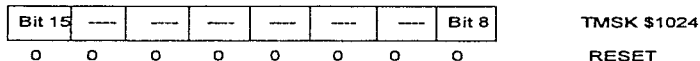
El pre-escalador.

Un pre-escalador programable permite al usuario seleccionar una de 4 opciones de reloj para manejar el contador de reloj principal de 16 bits. Esta selección permite al usuario un intercambio entre la resolución de reloj y el rango del mismo. Las razones a escoger dan un rango de 500 ns a 8 ms en la resolución y 32.77 ms a 524 ms entre sobreflujos, (para E de 2 MHz). El rango del cronómetro es importante porque el software requerido para funciones de tiempo es más complejo si se tienen que considerar los sobreflujos. Cuando los sobreflujos tienen que ser considerados los cálculos tienden a ser mucho más complicados.

Otro factor a considerar en la selección del rango del pre-escalador es el consumo de energía. Debido a que el consumo de energía en un dispositivo CMOS es directamente proporcional a la frecuencia de operación. Así que, al reducir la frecuencia se ahorrará energía. El usuario deberá considerar la resolución requerida y el tamaño del periodo de tiempo necesario.

Cuando se tienen que manejar sobreflujos por software, la cadencia y el tiempo de ejecución de la rutina de servicio del sobreflujo se pueden convertir en un factor para la precisión de las funciones de salida del reloj.

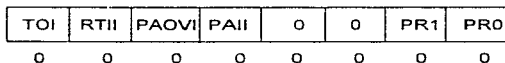
A continuación se muestra y se explica los bits de selección del pre-escalador, PRI y PRO, los cuales están en el registro enmascarable del reloj (TMSK2).



Sobreflujo.

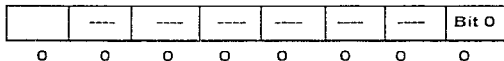
Para el caso en donde periodos de tiempo mayores que el rango del contador tengan que ser medidos o producidos, el sobreflujo del reloj debe ser usado. Esto equivale a medir tiempos más grandes que 60 segundos para el uso de los segundos para un display de un reloj digital. El despliegue de los minutos y las horas pueden ser ideados como contadores por software, lo cual extiende el rango de el contador de segundos. Cada vez que el contador de segundos tiene un sobreflujo, (va de 59 a 0), el contador de minutos es incrementado. Si se desea un periodo menor a 60 s, se deberá sumar el número de segundos deseados al inicio del conteo para así obtener el tiempo deseado.

En el MC68HC11, el bit de estado de la bandera de sobreflujo del reloj (TOF) se establece cada vez que el contador tiene un paso de \$FFFF a \$0000. Este bit tiene la opción de generar automáticamente una interrupción de petición cada vez que ocurra un sobreflujo, estableciendo el bit de habilitación de interrupción por sobreflujo (TOI) en el registro de bandera de reloj 2 (TFLG). El software debe reconocer que ha visto la condición de sobreflujo por medio del borrado del indicador de estado TOF. El contador de carrera libre continua su conteo aún y cuando el indicador de estado TOF no haya sido borrado. A continuación se describen los registros y las funciones del bit de estado TOF y de la habilitación de interrupción TOI. Los bits restantes de estos registros no están asociados con el sobreflujo del reloj.



TMSK \$1024

RESET



TFLG \$1025

RESET

TOI, TOF – Habilidad de interrupción de sobreflujo y bandera de sobreflujo del reloj.

El bit de estado TOF se establece automáticamente cada vez que el contador pasa de \$FFFF a \$0000. Este bit se borra escribiendo un uno en el registro TFLG2 en la posición del bit 7. El bit de control TOI permite al usuario configurar el sobreflujo del reloj para operación poleada o manejada por interrupciones pero no afecta el establecimiento o borrado del bit TOF. Cuando el bit TOI es cero, las interrupciones de sobreflujo de reloj son inhibidas, y el sobreflujo del reloj está trabajando en modo poleado. En este modo, el bit TOF deber ser poleado (leído) por medio del software del usuario para determinar cuando ha ocurrido un sobreflujo. Cuando el bit de control TOI es uno, una petición de interrupción de hardware es generada en el momento en que el bit TOF es establecido. Antes de dejar la rutina de servicio de interrupción, se debe borrar el bit TOF por medio de la escritura a el registro TFLG2.

PR1, PR0 – Selección del pre-escalador del reloj.

Estos dos bits seleccionan la razón para el reloj principal. La tabla 3.1.7.b, muestra las razones entre el factor del pre-escalamiento y el valor de los bits de control. En los modos normales de operación solamente se puede cambiar la razón del pre-escalador en los primeros 64 ciclos después de un reset, y la elección se mantendrá hasta la siguiente reinicialización.

PR1	PR0	Factor preescalab	Frecuencia del Cristal		
			2 ²³ Hz	8 Mhz	4MHz
			Una cuenta (resolución)/sobreflujo (rango)		
0	0	1	477 ns/32.77ms	500 ns/32.77ms	1 µs/65.54 ms
1	0	4	191 µs/125 ms	2 µs/131.1 ms	4 µs/262.1 ms
0	1	8	3.81 µs/250 ms	4 µs/262.1 ms	8 µs/524.3 ms
1	1	16	7.631 µs/0.5 s	8 µs/524.3 ms	16 µs/1.049 s
			2.1 MHz	2 MHz	1 MHz
			Frecuencia del bus (reloj E)		

Tabla 3.1.7.b Relaciones para el pre-escalamiento.

Derivación de contador.

En modos especiales de operación se cuenta con una función de bypass para simplificar los tiempos de prueba de las funciones del reloj principal. La función es activada mediante la escritura de un uno en un bit de control de bypass (CBYP) en el registro de control TEST1, que solamente se puede escribir en modos especiales. Cuando el CBYP es uno, el contador del reloj principal es

reconfigurado para que el pre-escalador se pase por alto y las mitades alta y baja del contador de 16 bits sea simultáneamente manejada por el reloj PH2.

Ya que esta función es para modos especiales, no interfiere en ninguna aplicación en los modos normales.

Función de interrupción de tiempo real (RTI).

La función RTI puede ser empleada para generar interrupciones de hardware a intervalos de tiempos fijos. Comúnmente se organizan las rutinas que conforman una aplicación en secuencias de llamadas mayores de subrutinas.

La longitud de tiempo necesario para completar todas las rutinas es variable y depende de cuanto tiene que hacer cada rutina, pero el tiempo para el peor de los casos es en el que se ejecutan todas las rutinas debe ser conocido. Después de haber pasado por todas las rutinas, el software entra a un modo de retardo hasta que una señal de tiempo de referencia es detectada. Al detectar la señal, se ejecuta un salto al inicio de la secuencia y se vuelven a llevar a cabo todas las rutinas en secuencia. Sabiendo el tiempo entre señales de tiempo de referencia sucesivas, una rutina puede medir el tiempo real tomado en cuenta el número de veces que es ejecutado y multiplicado por el tiempo entre señales de referencia de tiempo sucesivas. En este caso, el período RTI.

En el MC68HC11, el sistema RTI puede ser empleado para proporcionar esta señal de referencia de tiempo. Para adecuarse a las necesidades para aplicaciones variadas, son posibles 4 diferentes rangos para la señal RTI. Estos rangos están en función de la frecuencia del oscilador del MCU y de el valor de dos bits de control por software que son : (RTR1 y RTR0). No obstante que la velocidad puede ser cambiada en cualquier momento, típicamente se establece después de inicializar y se conserva.

La fuente de reloj para la función RTI es un reloj de carrera libre que no puede ser detenido o interrumpido. Este reloj causa que los tiempos fuera sucesivos del RTI sea una constante, lo cual

es independiente de la cadencia del software asociado con el servicio y borrado de bandera. Así que un período de RTI comienza con el tiempo fuera previo del RTI y no con el que es borrado.

El problema más común con el que se enfrentan los usuarios en el subsistema RTI es que se olvida borrar el RTIF una vez que se ha reconocido. Si la bandera no es borrada por un software específico en el TFLG2, seguirá como "pendiente" la siguiente vez que sea chequeado. Si el sistema se está trabajando en modo interrupción, la interrupción solicitada será atendida inmediatamente después del regreso de la RTI al final de la rutina. Esto provoca que el sistema se quede "atrapado" en un ciclo de servicio de peticiones de interrupción y la única salida es un reset al sistema. Para el caso en que el sistema opere en modo poleado, las rutinas se ejecutan correctamente la primera vez pero las secuencias principales se ejecutan demasiado rápido la segunda ocasión ya que el software "piensa" que el período del RTI ya ha concluido.

A continuación se presentan los registros y una breve explicación de lo anterior.

TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2 \$1024
TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2 \$1025

RTII, RTIF – Habilitación de interrupción de tiempo real, bandera de interrupción de tiempo real.

El bit de estado RTIF es automáticamente establecido al final de cada período de RTI. Este bit es borrado por la escritura de un uno en la posición 6 del registro TFLG2. El bit de control RTII permite que el usuario configure el sistema RTI para operar en modo poleado o de interrupción pero no afecta el establecimiento o borrado de RTIF. Cuando el RTII es cero, el sistema RTI está

en modo poleado y debe hacer un muestreo son software al RTIF para saber cuando un periodo del RTI ha transcurrido. Por el contrario. Si el RTI es igual con uno, se genera una petición de interrupción de hardware cada vez que el RTIF es uno. Antes de dejar la rutina de servicio, se debe borrar, vía software, el RTIF con la escritura en el registro TFLG2.

A continuación se explican los bits de selección de la opción del RTI localizados en el registro PACTL o registro de control del acumulador de pulsos.

DDRA7	PAEN	PAMOD	PEDGE	0	0	RTR1	RTR0	TFLG2	\$1026
0	0	0	0	0	0	0	0	RESET	

RTR1, RTR0 – Selección de razones de interrupción de tiempo real.

RTR1	RTR0	E + 12 ¹³	Frecuencia del Cristal		
			2 ²³ Hz	8 MHz	4MHz
			Razón nominal del RTI		
0	0	1	3.91 ms	4.10 ms	8.19 ms
0	1	2	7.81 ms	8.19 ms	16.38 ms
1	0	4	15.62 ms	16.38 ms	32.77 ms
1	1	8	31.25 ms	32.77 ms	65.64 ms
			Frecuencia del bus (reloj E)		
			2.1 MHz	2 MHz	1 MHz

Tabla 3 1.7.c Velocidades RTI contra RTR1, RTR0 para varias frecuencias de cristal.

Estos dos bits determinan la razón a la cual las interrupciones serán solicitadas por el sistema RTI. El RTI es manejado por un reloj E dividido entre 2^{13} compensado para que sea independiente del pre-escalador del reloj. Estos dos bits seleccionan un factor adicional de división.

En la tabla 3.1.7.c se muestran las razones de RTI que resultan de varias combinaciones de frecuencias de cristal y de valores de los bits de control del RTI. El valor del RTI se establece a su máximo al salir del reset y puede ser cambiado en cualquier momento.

Función de supervisión (COP).

La función de seguridad COP está relacionada con el sistema de reloj principal. La cadena de reloj para esta función se deriva de la cadena del reloj principal.

Los estados del contador hasta la derivación de E dividido por 2^{15} no tienen entrada de reset, los estados del divisor por encima de este son restablecidos cada vez que se ejecuta la secuencia de borrado del COP. Esta estructura determina la incertidumbre del periodo del COP porque el software no tiene forma alguna de saber cuando aparecerá el primer flanco de reloj en la derivación de E dividido por 2^{15} . Esta tolerancia o incertidumbre depende de la frecuencia del bus (E) pero no varía con respecto a la selección de la razón CR1 y CR0. Los bits que se muestran a continuación (CR1 y CR0) son los de selección de la velocidad de reloj del COP. Los demás no tienen relación con este sistema.

ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0
0	0	0	0	0	0	0	0

OPTION \$1039

RESET

CR1, CR0 – Bits para la selección de la velocidad de reloj del COP.

El reloj interno E del MCU es dividido por 2^{15} antes de entrar en el sistema de COP. Los bits de control CR1 y CR0 regulan factores posteriores de escala para el reloj del sistema de supervisión como se muestra en la tabla 3.1.7.d.

CR1	CR0	E + 12 ¹⁵ DIVIDIDO POR	Frecuencia del Cristal		
			2 ²³ Hz	8 Mhz	4MHz
			Tiempo nominal de salida		
0	0	1	15.625 ms	16.384 ms	32.768 ms
0	1	2	62.5 ms	65.536 ms	131.07 ms
1	0	4	250 ms	262.14 ms	524.29 ms
1	1	8	1 ms	1.049 s	2.1 s
			2.1 MHz	2 MHz	1 MHz
			Frecuencia del bus (reloj E)		

Tabla 3.1.7.d Factores de escala para el sistema de supervisión.

La columna derecha de la tabla muestra los periodos del sistema de supervisión resultantes para tres frecuencias típicas de oscilación. En el reset, se configura el periodo más corto. En operación normal, éstos bits pueden ser escritos una sola vez, en los primeros 64 bits después del reset.

Tips para el borrado de banderas del reloj.

Para borrar el estado de un bit de bandera en el registro de bandera de reloj es cargar un acumulador con una mascara que tenga un uno o unos en los bits a ser cambiados; entonces se escribe este valor en los registros TFLG1 o TFLG2.

La instrucción para borrar un bit puede ser usada también para borrar una bandera en los registros TFLG1 o TFLG2. La máscara que será usada con la instrucción de BCLR, deberá tener ceros en las posiciones de los bits correspondientes a las banderas a ser borradas y unos en los bits

restantes. Para el borrado de la bandera TOF, se ejecutan BCLR TFLG2 con una máscara %01111111, será suficiente.

No es apropiado usar la instrucción establecer bit (BSET) para el borrado de banderas en los registros de banderas de reloj, ya que esto podría inadvertidamente borrar una o más banderas en el registro.

Existen varias secuencias de instrucciones que pueden ser usadas para el borrado de banderas de reloj. En general, cada secuencia toma diferente número de bytes para el código objeto y un número diferente de ciclos para el tiempo de ejecución. La mejor secuencia depende de varios aspectos, incluyendo el espacio de memoria que el usuario desee que tome dicha ejecución.

Secuencia de instrucción	Código de operación	operandos	Modo de direccionamiento	Bytes	Ciclos	Secuencia total	
1	LDAA	#S80	(IMM)	2	2	4	5
	STAA	< TFLG2	(DIR)	2	3		
2	BCLR	< TFLG2\$7F	(DIR)	3	6	3	6
3	LDAA	#S80	(IMM)	2	2	5	6
	STAA	TFLG2	(EXT)	3	4		
4	LDAA	#S80	(IMM)	2	2	4	6
	STAA	TFLG2,X	(IND,X)	2	4		
5	BCLR	TFLG2,X \$7F	(IND,X)	3	7	3	7
6	LDAA	#S80	(IMM)	2	2	5	7
	STAA	TFLG2,Y	(IND,Y)	3	5		
7	BCLR	TFLG2,Y \$7F	(IND,Y)	4	8	4	8

Tabla 3.1.7.e. Instrucciones que pueden ser usadas para borrar el bit de estado TOF en el registro TFLG2.

Para algunos casos, la diferencia entre el tamaño del programa y el tiempo de ejecución son importantes. Algunas de las secuencias requieren que los registros en modo de direccionamiento directo se encuentren en el espacio de memoria (\$0000 - \$00FF), y esto no es muy práctico para muchas aplicaciones.

A partir de que otras secuencias usan el modo de direccionamiento indexado, su uso eficiente dependerá de si el registro indexado apunta o no al espacio en donde se encuentra dicho registro. La tabla 3.1.7.e nos muestra siete secuencias de instrucciones diferentes que podrían ser usadas para borrar el bit de estado TOF en el registro TFLG2.

3.1.8 Función de captura de entrada.

Cada función de captura de entrada incluye un registro de retención (latch) de 16 bits, en su lógica de detección de flancos y en la lógica de generación de interrupciones. El latch captura el valor del curso del contador cuando un flanco elegido es detectado en la terminal de entrada de el reloj correspondiente. La lógica de detección de flancos cuenta con bits de control para que con el uso de software, el usuario pueda determinar la polaridad del flanco que será reconocido. Cada una de las tres funciones de captura de entrada pueden ser configuradas independientemente para detectar flancos de subida solamente, únicamente flancos de bajada, o cualquier flanco.

La lógica de generación de interrupción incluye una bandera de estado que indica que un flanco ha sido detectado, y un bit de habilitación de interrupción, el cual determina si la función de captura de entrada correspondiente generará una petición de interrupción por hardware. Si la petición de interrupción es inhibida, la captura de entrada está trabajando en modo poleado y el software debe de leer la bandera de estado para reconocer que un flanco fue detectado.

Los flancos de captura de entrada generalmente no tienen sincronía con el contador interno del reloj que tiene relación con el reloj PH2. Estas peticiones de captura asincrónica son entonces sincronizadas con PH2 para que el almacenamiento contra el medio ciclo opuesto de PH2 desde el cual el contador del reloj está siendo incrementado. Este proceso de sincronización introduce un

retardo entre cuando el flanco sucede y cuando es almacenado. De cualquier manera, en la mayoría de los casos este retraso debe ser ignorado ya que este retardo es compensado con otros.

El elemento central de cada función de captura de entrada es el latch de captura de entrada, el cual puede ser leído por el software como un par de registros de 8 bits. Los registros TICx no son afectados por el reset y no pueden ser escritos por software. Cuando un flanco ha sido detectado y sincronizado, el valor del contador de 16 bits es transferido al par de registros de captura de entrada como una sola transferencia de 16 bits. La captura de valores del contador del reloj y el incremento del mismo, ocurren en ciclos opuestos del reloj PHI2 de manera que el valor de la cuenta esté estable siempre que una captura ocurra. Las funciones de captura de entrada operan independientemente de las otras, y las tres funciones pueden capturar la misma cuenta de 16 bits si los flancos de entrada son todos detectados dentro del mismo ciclo de la cuenta del reloj.

Una lectura de bit de orden superior de un par de registro de captura de entrada inhibe una nueva transferencia de captura por un ciclo del bus. Mientras se utilice una instrucción de doble byte para leer valores de captura de entrada, el usuario está seguro de que los dos bytes corresponden el uno con el otro. Si una nueva captura de entrada ocurre de manera que una transferencia hubiera ocurrido inmediatamente después de la lectura del byte superior, será retrasada por un ciclo más no se perderá.

La acción de captura de una función de entrada ocurre cada vez que un flanco seleccionado en la terminal correspondiente de entrada es detectado. Esto significa que el valor leído de captura de entrada corresponde al más reciente flanco de la terminal, el cual puede no ser el flanco que causó que se estableciera al bandera de captura.

La acción de captura de una función de entrada ocurre cada vez que un flanco seleccionado en la terminal correspondiente de entrada es detectado. Esto significa que el valor leído del registro de captura de entrada corresponde al más reciente flanco de la terminal, el cual puede no ser el flanco que causó que se estableciera la bandera de captura de entrada. En algunas aplicaciones puede haber un número de flancos brevemente espaciados.

En los casos en que estas capturas extras son indeseables, el software puede controlar los bits de selección de flanco para inhibir capturas subsiguientes hasta que la presente captura haya sido manipulada.

A continuación se presentan los registros TMSK1 y TFLG1 para estas funciones.

OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I	TMSK1 \$1022
------	------	------	------	------	------	------	------	--------------

OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F	TFLG1 \$1023
------	------	------	------	------	------	------	------	--------------

ICxI, ICxF – Habilitación de interrupción de captura de entrada y de bandera de captura.

El bit de estado ICxF automáticamente es uno, cada vez que un flanco seleccionado es detectado en la terminal de captura correspondiente.

Este bit de estado es borrado por medio de la escritura del registro TFLG1 con un uno en la posición del bit correspondiente. El bit de control ICxI permite al usuario configurar cada función de captura de entrada para operación poleada o manejada por interrupción pero no afecta al establecimiento o borrado del bit correspondiente, y la captura de entrada es operada en modo de poleo. En este modo, el bit ICxF debe ser leído por el software del usuario para determinar cuando ha sido detectado un flanco. Cuando el bit de control ICxI es uno, una petición de interrupción de hardware es generada cuando el correspondiente bit de ICxF se establece como uno. Antes de dejar la rutina de servicio de interrupción, se debe limpiar el bit ICxF por medio de la escritura en el registro TFLG1.

Opciones Programables

El usuario puede programar cada función de captura de entrada para detectar una polaridad particular del flanco en la terminal correspondiente de entrada. Un par de bits de control (EDGxA y EDGxB) en el registro 2 de control de número (TCTL2) es usado para seleccionar el o los flancos detectados por cada una de las funciones de captura de entrada.

0	0	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	OPTION \$1039 RESET
0	0	0	0	0	0	0	0	

EDGxB, EDGxA: Control de Flancos de Captura de Entrada.

Estos pares de bits determinan a que flancos serán sensibles las funciones captura de entrada y codificados de acuerdo a la figura 3.1.8.

EDGxB	EDGxA	Configuración
0	0	Captura deshabilitada
0	1	Captura en flancos de subida solamente
1	0	Captura en flancos de bajada solamente
1	1	Captura en cualquier flanco

Tabla 3.1.8. Sensibilidad en la captura de entrada.

3.1.9 Funciones de Comparación de Salida

En vez de ser solamente configuradas para llevar a cabo una sola función, las funciones de

comparación de salida son configuradas y controladas por registros de control accesibles por software y por bits de manera que ejecuten una amplia variedad de tareas.

Existen 5 funciones de comparación de salida en MC68HC11. Cada comparación de salida tiene un registro de comparación de 16 bits y un comparador dedicado de 16 bits. El comparador lleva a cabo una comparación del valor del reloj de carrera libre contra el registro de comparación de 16 bits durante cada cuenta de reloj cuando se detecta una concordancia, una bandera de estado es establecida (OCxF): una interrupción es generada opcionalmente y las terminales de salida de reloj son combinadas automáticamente de acuerdo a los bits de control. Ya que cada una de las cinco interrupciones son enmascarables separadamente con un bit local de control que habilita la interrupción y ya que cada uno tiene su propio vector de interrupción no hay necesidad de efectuar alguna lectura de software para determinar la causa de la interrupción.

Cuatro de las funciones de comparación de salida operan de manera muy similar a las de otras familias de microcontroladores, sin embargo, los MCU's de la familia HC11 tienen más canales de comparación de salida y un mayor control sobre las terminales de salida de reloj. La quinta comparación de salida (OC1) puede controlar cualquier combinación de las cinco terminales de salida de reloj aun cuando otra comparación de salida ya esté controlando las terminales. Para OC5-OC2, un par de bits en el control del registro 1 de control del reloj (TCLTL1) que controla la acción automática que ocurrirá en la terminal respectiva de salida de reloj cuando una comparación de salida suceda. Los pares de bits de control (omx, O1x) son codificados para permitir las cuatro siguientes posibilidades:

1. La comparación de salida de reloj no causa cambios en la terminal.
2. Interrumpir la terminal en cada comparación exitosa.
3. Forzar la terminal a cero en cada comparación exitosa.
4. Forzar la terminal a uno en una terminal de salida.

Cada una de las comparaciones de salida es asociada a una terminal del puerto A, y las acciones automáticas de la terminal para cada comparación de salida son controladas independientemente.

Para OC1, las acciones automáticas de la terminal son controladas por los registros OC1 mascarado (OC1M) y el registro de datos OC1 mascarado (OC1D). El registro OC1M determina que terminales del puerto A serán afectadas por OC1. EL registro OC1D especifica los datos que serán enviados a las terminales afectadas del puerto A cuando haya una concordancia exitosa de OC1. Si OC1 y otra comparación de salida están controlando la misma terminal, y si las dos tratan de cambiar la terminal simultáneamente, OC1 tendrá la prioridad.

Las terminales de comparación de salida pueden ser usadas como salidas de propósito general sin tener nada que ver con el reloj o como salidas de reloj directamente controladas por el mismo. Cuando una de las terminales se utiliza para el sistema del reloj no puede ser escrita como salida de propósito general. Si el usuario requiere cambiar el estado de una terminal sin esperar la comparación de salida, puede utilizar la función de "forzar" la comparación de salida. Otra forma de hacerlo, es desenganchando el reloj temporalmente cambiando los bits de control.

El registro de 16 bits de comparación de salida para cada función de comparación de salida puede ser leído o escrito por el software como un par de registros de 8 bits. los registros TOCx se forzan a SFFFF durante el reset.

Una escritura al byte de orden superior del par de registros de comparación de salida inhibe la función de comparación de salida por un ciclo de bus. Con esto se asegura que el registro de 16 bits de comparación cuyo byte superior acaba de ser escrito, y el inferior contiene todavía datos del valor anterior, sea leído en forma correcta. Siempre que el usuario utilice una instrucción de doble byte (como STD), para la actualización de registros de salida, este mecanismo de inhibición de un ciclo de bus previene que se efectúen comparaciones no intencionales después de la escritura de la mitad de orden alto pero antes de la mitad de orden bajo.

Mientras que una función de comparación de salida sea configurada para cambiar el estado de una terminal o para generar una interrupción, la acción ocurre cada vez que la cuenta de reloj concuerde con el registro de comparación. Para generar una sola interrupción después de un

retraso de tiempo, léase el registro TCNT, súmese el valor correspondiente al retardo deseado, escribáse el valor del registro de comparación de salida y escribáse los controles apropiados para habilitar la interrupción. Cuando ocurra la interrupción, deshabilítase la interrupción para prevenir que se presente otra al tenerse una cuneta igual de reloj.

Los siguientes registros explican las banderas de estado de comparación de salida

OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F	TFLG1 \$1023
0	0	0	0	0	0	0	0	RESET

OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I	TMSK \$1022
0	0	0	0	0	0	0	0	RESET

El bit de estado OCxI es automáticamente establecido como uno cada vez que el registro de comparación correspondiente concuerda con el reloj. Este bit de estado es borrado por medio de la escritura de un uno en la posición correspondiente del bit del registro TFLG1. El bit de control OCxI permite al usuario configurar cada función de comparación de salida para operación poleada o manejada por interrupción pero no afecta al establecimiento o borrado del bit OCxI correspondiente. Cuando OCxI es cero, la interrupción de comparación de salida es operada en modo poleado. Cuando OCxI es uno, una petición de interrupción de hardware es generada siempre que el bit correspondiente OCxI tenga un uno. Antes de dejar la rutina de servicio de interrupción, el software deberá limpiar el bit OCxI escribiendo en el registro TFLG1.

3.1.10 El Acumulador de Pulsos

El acumulador de pulsos que es similar a los relojes en los MCU's de las primeras familias del 6805, es un sistema mucho más sencillo que el reloj principal del sistema, visto anteriormente. Este sistema está basado en un contador de 8 bits y puede ser configurado para operar como un

simple contador de eventos o para acumulación de tiempo. A diferencia del reloj principal el contador acumulador de pulsos de 8 bits puede ser leído o escrito en cualquier momento (en el contador de 16 bits del reloj principal no se puede escribir). Los bits de control le permiten al usuario configurar y controlar el subsistema de acumulador de pulsos. Dos interrupciones enmascarables están asociadas con el sistema, cada una teniendo sus propios controles y vector de interrupción.

La terminal 7 de E/S del puerto A (PA/PAI/OC1) asociada a el acumulador de pulsos puede ser configurada para operar como un reloj o como una señal de puerta para habilitar un reloj E dividido entre 64 que alimenta al contador de 8 bits. las funciones alternativas de las terminales de entrada del acumulador de pulsos, presentan otras aplicaciones interesantes.

Descripción General

El acumulador de pulsos es un sistema reloj/contador, que puede ser configurado para operar en cualquiera de dos modos básicos.

En el modo de conteo de eventos, en este modo el contador de 8 bits está amarrado a los flancos activos de entrada de la terminal PAI incrementándose por cada uno. Para el modo de acumulación de tiempo, el contador de 8 bits está amarrado al reloj E dividido entre 64 y sujeto al estado de la terminal PAI. La figura 3.1.11 muestra un diagrama de bloque simplificado del acumulador de pulsos para cada uno de sus dos modos de operación.

La tabla 3.1.10.a resume los periodos de tiempo importantes para el acumulador de pulsos (cuando opera en el modo de acumulación de tiempo) para varias velocidades de cristal. Las fórmulas en la parte inferior de la tabla pueden ser usadas para diferentes frecuencias de cristal de las mostradas en la tabla.

E	Frec. Cristal	Periodo E	Resolución	Sobreflujo
2.1 MHz	2^{23} Hz	477 ns	30.52 μ s	7.81 ms
2 MHz	6 MHz	500 ns	32 μ s	8.19 ms
1 MHz	4 MHz	1 μ s	64 μ s	16.38 ms
Fórmula:			64(E)	16.384 (E)

Tabla 3 1.10 a Periodos de tiempo del acumulado de pulsos contra velocidades de cristal.

El reloj E dividido entre 64 es una derivación que se obtiene del sistema de reloj principal. En general, cualquier señal de entrada para la terminal PAI es asíncrono a este reloj E dividido entre 64, sin embargo la primera cuenta podría ocurrir entre cualquiera de los 64 ciclos de E después de que la terminal PAI esté en el nivel elegido.

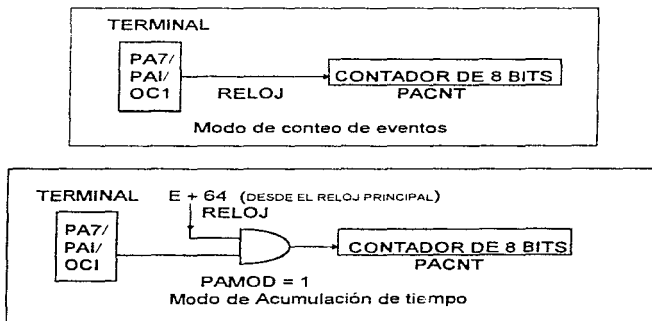


Figura 3 1.10 b. Modos de operación del acumulador de pulsos

El bit de selección de flanco del acumulador de pulsos (PEDGE) controla la polaridad de las señales en PAI que serán reconocidas por el sistema del acumulador de pulsos. En modo de conteo, el reloj es la salida del detector de flanco de la terminal PAI. En el modo de acumulación de tiempo, es un reloj interno de E dividido por 64 con la activación por medio de la terminal PAI.

El bit de habilitación de interrupción por sobreflujo del acumulador de pulsos (PAOVI), determina en todo caso si la bandera de interrupción por sobreflujo del acumulador de pulsos generará una petición de interrupción por hardware. El bit de habilitación de interrupción de flanco de entrada del acumulador de pulsos (PAII) determina si los flancos detectados en la terminal PAI establecerán la bandera de entrada del acumulador de pulsos (PAIF) o no. Además de las habilitaciones de interrupción local PAII y PAOVI, estas interrupciones están sujetas a un enmascaramiento por el bit 1 en el registro de código de condición en la CPU.

El buffer de entrada en la terminal PAI siempre se encuentra conectado desde la terminal hasta el acumulador de pulsos y la lógica de lectura del puerto A, pero, el buffer de salida está habilitado o deshabilitado por el bit de control de dirección de datos (DDRA7) en el registro de control del acumulador de pulsos (PACTL). Normalmente, cuando se comienza a trabajar con el acumulador de pulsos, la terminal PAI está configurada como una entrada en estado de alta impedancia (DDRA7=0), pero es posible por medio de software o por el reloj principal tener el control directo del acumulador de pulsos al establecer en el registro DDRA7 un uno.

Registros de Estado de Control del Acumulador de Pulsos

Todos los registros de estado y control que relacionan al acumulador de pulsos son:

TOI	RTII	PAOVI	PAII	0	I / O	PRI	PRO
0	0	0	0	0	0	0	0

TMSK2 \$1024

RESET

TOF	RTIF	PACVF	PAIF	0	0	0	0
0	0	0	0	0	0	0	0

TFLG2 \$1022
RESET

DDRA	PAEN	PAMO	PEDG	0	0	RTRI	RTRO
0	0	0	0	0	0	0	0

TFLG2 \$1026
RESET

BIT 7	—	—	—	—	—	—	BIT 0
0	0	0	0	0	0	0	0

PACNT \$1026
RESET

DDRA7 Control de dirección de datos para el puerto A bit 7

1 = puerto A bit 7 configurado para salida de datos.

0 = puerto A bit 7 configurado para entrada de datos.

Normalmente cuando el acumulador de pulsos está en uso, la terminal PAI estará configurada como entrada. En casos raros, la terminal PA7/PAI/OCI puede ser configurada como salida para permitir, a OCI o a una salida de software, el manejo del sistema de acumulador de pulsos. Ya que el buffer de entrada está siempre conectado a la terminal, cualquier función de salida que esté controlando la terminal PA7 también manejará el acumulador de pulsos.

PAEN Habilitación del acumulador de pulsos.

0 = Acumulador de pulsos deshabilitado.

1 = Acumulador de pulsos habilitado.

Cuando el acumulador de pulsos está deshabilitado, el contador de 8 bits deja de contar, y las interrupciones del acumulador de pulsos son inhibidas. Así, las banderas no pueden ser

establecidas, permanecerán establecidas las que hayan estado en el momento en que el acumulador de pulsos fue deshabilitado.

PAMOD Selección de modo del acumulador de pulsos.
0 = Modo de conteo de eventos externos.
1 = Modo de acumulación de tiempo con barrera.

En este modo (PAMOD = 1), el bit PEDGE tiene valor adicional. Además de especificar la polaridad del flanco que causa que el bit PAIF sea establecido, PEDGE, también controla el nivel de inhibición de la barrera. Las interrupciones de PAIF ocurren en el borde de salida de una señal de habilitación de barrera o puerta; seleccionar los flancos de bajada entonces, provocan que el reloj de carrera libre E dividido por 64 sea deshabilitado mientras la terminal PAI este baja.

PAOVI, PAOF Bandera y habilitación de interrupción de sobreflujo del acumulador de pulsos.

El bit de estado PAOVF es automáticamente establecido como un uno cada vez que el contador del acumulador de pulsos pasa de SFF a S00. Este bit de estado es borrado mediante la escritura del registro TFLG2 con un uno en el bit 4. El bit de control PAOVI permite al usuario el configurar el sobreflujo del acumulador de pulsos para operación poleada o manejada por interrupción pero no afecta la lectura o escritura de PAOVF. Cuando PAOVI es cero, las interrupciones de sobreflujo del acumulador de pulsos son inhibidas, y el sistema opera en modo poleado. En este modo, PAOVF debe ser muestreado por software del usuario para determinar cuando ha ocurrido un sobreflujo. Cuando el bit de control PAOVI es uno, una petición de interrupción es generada cada vez que PAOVF se establece como uno.

PAII, PAIF Bandera de habilitación de interrupción de flanco de entrada del acumulador de pulsos.

El bit de estado PAIF es automáticamente uno cada vez que un flanco seleccionado es detectado en la terminal PA7/PAI/OCI. Este bit de estado es borrado por la escritura de un uno en la

posición del bit 5 del registro TFLG2. El bit de control PAII faculta al usuario a configurar la detección de flanco de entrada del acumulador de pulsos para operar en forma poleada o manejado por interrupciones. Cuando PAII es cero, las interrupciones de entrada al acumulador de pulsos son inhibidos, y el sistema trabaja en modo poleado. En este modo el bit PAIF debe ser muestreado para determinar si ha ocurrido un flanco. Cuando el bit de control PAII es uno, una petición de interrupción por hardware es generada cada vez que PAIF es uno.

3.2 Configuración y Modos de Operación

En esta sección se tratan los mecanismos que permiten al MC68HC11 conformar una amplia variedad de aplicaciones. Estos mecanismos incluyen circuitería de modo de selección, un registro de configuración no-volátil basado en memoria del tipo EEPROM y un bit de registro de control protegido. La mayoría de los bits de control del MCU son accedidos vía software en cualquier tiempo. Muy pocas funciones del MCU son influenciadas por el modo de operación. Por ejemplo, todas las funciones de los temporizadores, del convertidor analógico digital y las funciones de entrada/salida (I/O) serial trabajan de la misma manera en los modos de operación expandido como en el modo simple.

3.2.1 Configuración del Sistema

Existen solo dos modos fundamentales de operación para el MCU. El de modo simple o el de modo expandido.

Cada modo tiene una variación normal y una variación especial. Estas cuatro variaciones de modo se seleccionan por los niveles de voltaje aplicados a las patas de modo A (MODA) y de modo B (MODB) durante el restablecimiento del MCU. La variación especial del modo simple es llamado modo especial de arranque; la variación especial del modo expandido es llamado modo especial de prueba. El modo especial de arranque permite que programas sean descargados a través de la interfaz de comunicaciones serial interna (SCI) dentro de la memoria RAM para que sean ejecutados. El modo de operación de arranque es un modo especial de usuario, no un modo de prueba de fábrica. El modo especial de prueba, el cual es propuesto originalmente para pruebas de fábrica, pocas ocasiones es utilizado por el usuario, excepto para emulación, desarrollo u otras circunstancias especiales. En los modos especiales de operación del MCU, algunas funciones especiales de prueba se vuelven más accesibles.

Configuración por Hardware

El mecanismo de selección de modo por hardware comienza con los niveles lógicos aplicados a las terminales correspondientes a MODA y MODB cuando el MCU está en el modo de restablecimiento. Los niveles capturados determinan el estado lógico de los bits de control del modo especial (SMOD) y de selección del modo A (MDA) en el registro de interrupción de la más alta prioridad. Ciertamente estos dos bits controlan los circuitos lógicos involucrados en la selección de modo por hardware. La figura 3.2 presenta un resumen de la operación de las terminales de y los bits de control de modo.

Entradas		Descripción del Modo	Bits de control en HPRIO			
MODB	MODA		RBOOT	SMOD	MDA	IRV
1	0	Modo normal simple	0	0	0	0
1	1	Modo normal expandido	0	0	1	0
0	0	Modo especial arranque	1	1	0	1
0	1	Modo especial de prueba	0	1	1	1

Figura 3.2.1.a Configuración de los modos de operación del MC68HC11.

Registro CONFIG basado en EEPROM

El registro de configuración (CONFIG) no volátil, permite una flexibilidad adicional en el MCU, que de otra manera podría proporcionarse por una estructura más compleja de selección de modo por hardware. Las funciones controladas por este registro son características que deben ser conocidas inherentemente por el sistema del MCU. Los bits de control accedidos por software ordinariamente no podrían regular de una manera estos controles.

La propagación y borrado de este registro de configuración emplea la misma lógica empleada para

la programación y borrado de la matriz de memoria interna EEPROM. Durante cualquier restablecimiento del sistema, los contenidos del byte de EEPROM son transferidos al registro estático en funcionamiento sobre el bus de datos. Debido a este mecanismo, los cambios de la localidad del CONFIG de EEPROM no son visibles y no altera la operación del MCU mientras para el subsecuente restablecimiento.

El registro CONFIG es un registro de control poco común usado para habilitar o deshabilitar las memorias internas ROM, EEPROM, el sistema de vigilancia de operación adecuada y, opcionalmente, la característica de seguridad de la EEPROM en el MCU. A diferencia de cualquier otro registro, el CONFIG mantiene su información aun cuando el MCU se encuentra sin alimentación de energía.

Bits del Registro de Control Protegidos

En el MC68HC11, algunos registros y bits de control están protegidos contra escritura excepto bajo circunstancias especiales. El mecanismo de protección incluye la habilidad de escribir estos bits solo dentro de los primeros 64 ciclos de bus, después de cada restablecimiento y/o la habilidad de escribirlos solamente una vez después de cada restablecimiento. Estos bits controlan la configuración básica del MCU en donde una escritura accidental podría causar serios problemas de sistema.

3.2.2 Los Modos de Operación

Al principio de esta sección se mencionó de los dos modos de operación que posee el microcontrolador y sus dos variantes. A continuación se dará una explicación más detallada.

Modo de Operación Normal del MCU

El vector de restablecimiento es buscado desde la dirección \$FFFE, FFFF y la ejecución del programa comienza desde la dirección indicada por este vector. En el modo normal simple la

memoria interna programable de 8 Kbytes es habilitada en este espacio de memoria de modo que el vector de restablecimiento es buscado desde esta memoria ROM interna.

En modo normal expandido la memoria ROM interna puede o no estar habilitada, dependiendo del bit ROMON en el registro CONFIG. Si la memoria interna está activa, el vector se busca desde dentro de esta ROM, de lo contrario desde una memoria externa con dirección \$FFFF, FFFE.

Modo Normal Simple

Debido a que este modo simple no requiere ninguna función externa de dirección o de bus de datos, el puerto B, el puerto C, la señal A (STRA) y la señal B (STRB) pueden ser empleados como entradas/salidas (I/O) paralelo de propósito general. En este modo, todo el software necesario para controlar al MCU está contenido en las memorias internas. El bit de control ROMON en el registro CONFIG deberá ser sobre manejado para forzar a la memoria ROM de 8 Kbytes a activarse.

Modo Normal Expandido

Este modo de operación permite que, memorias externas y dispositivos periféricos sean accedidos por un bus de datos /direcciones multiplexados. Por la multiplexación de los 8 bits de dirección de menor orden así como datos en las patas del puerto C, solamente 18 patas son necesarias para proporcionar un bus de datos de 8 bits, un bus de direcciones de 16 bits y dos líneas de control de bus, a través de los puertos C.

Modo de Operación Especial del MCU

En las variaciones de modo especial, los vectores de restablecimiento e interrupción están localizados en \$BFC0 - \$BFFF, y el software tiene acceso a las características especiales de prueba. Una de estas características especiales de prueba (la deshabilitación del bit de control de

restablecimiento (DSR) en el registro de control TEST1) deshabilita temporalmente el COP y funciones de restablecimiento de reloj del monitor. Todas las funciones especiales y privilegios están disponibles tanto en el modo especial de prueba y modo de arranque.

Modo Especial de Prueba

Este modo de operación fue originalmente propuesto para las pruebas internas de producción de Motorola, sin embargo existen pocos casos en donde el usuario puede utilizar este modo de operación. Estos casos especiales incluyen la programación del registro CONFIG, la programación de calibración de datos dentro de la EEPROM y desarrollo de situaciones como emulación. Debido a que este modo de operación permite sobrescribir en los bits de control, es posible el anular el restablecimiento para verificar el contenido del registro de CONFIG para luego regresar al modo normal y habilitar de nueva cuenta las protecciones.

Modo Especial de Arranque (bootstrap)

Cuando el MCU es puesto en modo especial de arranque, una pequeña memoria ROM interna es habilitada en las direcciones \$BF40 - \$BFFF. El vector de restablecimiento es buscado en esta ROM de arranque, y el MCU procede a ejecutar el inicio del programa desde esta ROM. El programa en esta ROM inicializa el sistema de la SCI interna, verifica la opción de seguridad y acepta un programa de 256 bytes a través del SCI y luego se salta a la carga del sistema en la dirección \$0000 en la memoria RAM.

3.3 Modelo de Programación e Instrucciones

Además de ejecutar todas las instrucciones de los microprocesadores M6800 y M6801, el conjunto de instrucciones del microcontrolador incluye 91 nuevos códigos de operación. La nomenclatura MCC68xx es usada en conjunción con una arquitectura específica de UCP y un conjunto de instrucciones que se opone a la nomenclatura del MC68HC11xx, la cual es una referencia de un miembro específico de la familia de microcontroladores MC68HC11.

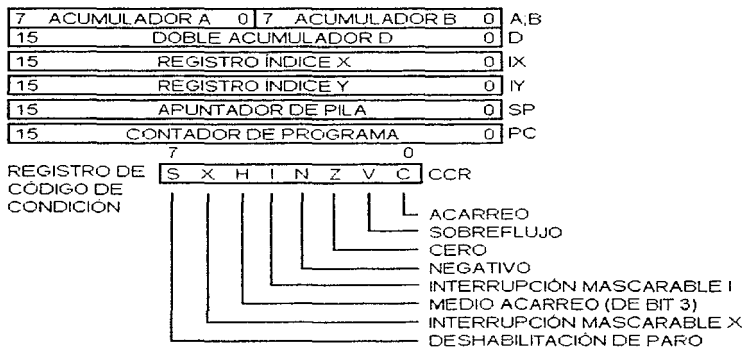


Figura 3.3.a. Modelo de Programación del MC68HC11.

La figura 3.3.a muestra los siete registros de UCP disponibles al programador. Los dos acumuladores de 8 bits (A y B) pueden ser usados por algunas instrucciones como un solo acumulador de 16 bits llamado registro D, el cual permite ejecutar un grupo de instrucciones de 16 bits aun cuando el microcontrolador es de 8 bits.

El grupo más grande de instrucciones adicionales involucran al nuevo registro de índice Y. También fueron sumadas 12 instrucciones de manipulación de bit que pueden operar cualquier localidad de memoria o registro.

3.3.1 Modos de Direccionamiento

El CPU cuenta con seis modos de direccionamiento para el acceso a la memoria. estos modos son los siguientes: de acceso inmediato, acceso indirecto, extendido, indexado, inherente y relativo.

- Direccionamiento Inmediato (IMM)

El dato actual está contenido en el byte inmediatamente después de la instrucción. Este tipo de operación requiere de dos a cuatro bytes. La dirección efectiva de la instrucción está especificada por el carácter # y los puntos implícitos del byte siguiente al código de operación. Con el siguiente ejemplo se muestra el movimiento de la operación:

LDAA#\$AA

Entonces tendremos en el acumulador A el siguiente movimiento:

Antes:

0000	0000
------	------

Después:

1010	1010
------	------

Esto nos indica que en el acumulador A, se tendrá el número hexadecimal AA pero convertido en su correspondiente valor binario. Donde el carácter # se emplea para indicar que es un direccionamiento inmediato.

- **Direccionamiento Directo (DIR)**

En el modo de direccionamiento directo, el byte menos significativo de la dirección efectiva de la instrucción aparece en el byte siguiente al código de operación. El byte más significativo de la instrucción se asume que es \$00 y no se incluye en la instrucción. Este hecho limita para operandos el uso del direccionamiento directo sólo al área de memoria de \$0000 a \$00FF.

Para este direccionamiento tenemos el siguiente ejemplo:

ADDA \$32

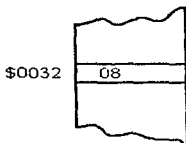
Antes acumulador A:

0000 0000

Después acumulador A:

0000 1000

Mapa de Memoria



Esto nos dice que el valor que se encuentra en la dirección \$0032, le será sumado al acumulador A.

- **Direccionamiento Extendido (EXT)**

En el modo de direccionamiento extendido, la dirección efectiva de la instrucción aparece explícita en los dos bytes siguientes al código de operación. Sin embargo la mayoría de las instrucciones usadas para este tipo de direccionamiento emplean tres bytes: uno para el código de operación y dos para la dirección.

Como ejemplo tenemos lo siguiente, con el acumulador $A = 0$:

ADDA \$2354

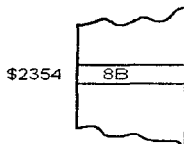
Acumulador antes:

0000 0000

Acumulador después:

0101 1011

Mapa de Memoria



Para este caso, el dato que se encuentra en la localidad \$2354 será sumado al valor que tenga el acumulador A.

- **Direccionamiento Indexado (INDX, INDY)**

En el modo de direccionamiento indexado, los registros X o Y son usados para calcular la dirección efectiva. Para este caso la dirección efectiva es variable y depende del contenido actual de los registros X o Y a los que se les suma un desplazamiento de 8 bits, el cual está contenido en la instrucción. Así que tenemos el siguiente ejemplo:

Con el acumulador D igual con cero, le será sumado un dato el cual está ubicado en la localidad del registro X y el resultado de la suma del desplazamiento, para este caso \$02

ADDA 4, X

Antes

Acumulador D:

0000 0000 0000 0000

Registro X:

0010 1111 1010 0011

Después

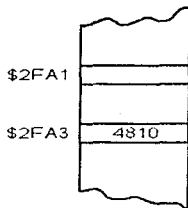
Acumulador D:

0100 1000 0001 0000

Registro X:

0010 1111 1010 0011

Mapa de Memoria

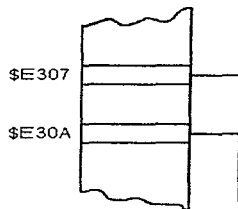


- **Direccionamiento Relativo (REL)**

El direccionamiento relativo es un modo de direccionamiento especializado que se aplica a las instrucciones de salto llamados saltos relativos. Este tipo de direccionamiento genera dos bytes de código de máquina: uno para el código de operación y el otro es un número en complemento a dos el cual representa un desplazamiento desde alguna dirección. El rango de este desplazamiento está comprendido de -128 a +127 bytes. A continuación se ejemplifica esta instrucción:

Código de Máquina	Operación	Operando
E3 00	ADDD	0, X
E3 04	ADDD	4, X
E3 07	BEQ	S3
E3 0A	ABA	

Mapa de Memoria



Para este ejemplo, si el resultado de la operación anterior es cero, dará un salto tres localidades de memoria adelante y continuará con la siguiente instrucción de esa localidad.

- **Direccionamiento Inherente (INH)**

En este modo de direccionamiento, todo lo que se necesita para ejecutarse la instrucción es conocido por el CPU. Los operandos (si existen) son registros del CPU y no necesitan ser traídos de la memoria, ejemplo:

ABA

Acumulador A antes:

0000 0000

Acumulador B antes:

1101 1000

Acumulador A después:

1101 1000

La operación efectuada fue la suma del acumulador B a el acumulador A, almacenando el resultado en el acumulador A.

3.3.2 El Conjunto de Instrucciones

En esta sección se explicaran las capacidades y organización del conjunto de instrucciones del MCU, por lo que se dividirán en grupos funcionales de instrucciones. Por esto, algunas instrucciones podrían aparecer en más de un grupo funcional. Para expandir el número de instrucciones usadas en el MC68HC11, se ha adicionado un mecanismo de pre-byte. Las

instrucciones que no requieren del pre-byte están en la página 1 del mapa de códigos de operación. Las instrucciones que lo requieren se hayan en las páginas 2, 3, y 4. Un código de pre-byte se aplica solo al código de operación inmediato a él. Esto es, se asume que todas las instrucciones son de un solo byte, a menos que el primer byte de la instrucción pertenezca a uno de los tres códigos de pre-byte, que son \$18 para la página 2, \$1 para la página 3 y SCD para la página 4.

3.3.2.1 Instrucciones de Acumulador y Memoria

La mayoría de estas instrucciones usan dos operandos, uno de ellos es un acumulador o reg. de índice y el otro se obtiene de la memoria por un modo de direccionamiento. Estas instrucciones de memoria acumulador se pueden dividir en seis subgrupos:

Función	Mnemónico	IMM	DIR	EXT	INDX	INDY	INH
Borra el byte de memoria	CLR			X	X	X	
Borra acumulador A	CLRA						X
Borra acumulador B	CLRB						X
Carga acumulador A	LDDA	X	X	X	X	X	
Carga acumulador B	LDAB	X	X	X	X	X	
Carga doble acumulador D	LDD	X	X	X	X	X	
Extrae A de la pila	PULA						X
Extrae B de la pila	PULB						X
Coloca A en la pila	PSHA						X
Coloca B en la pila	PSHB						X
Almacena acumulador A	STAA	X	X	X	X	X	
Almacena acumulador B	STAB	X	X	X	X	X	
Almacena doble acumulador D	STD	X	X	X	X	X	
Transfiere A hacia B	TAB						X
Transfiere A hacia CCR	TAP						X
Transfiere B hacia A	TBA						X
Transfiere CCR hacia A	TPA						X
Intercambia D con X	XGDX						X
Intercambia D con Y	EGDY						X

Tabla 3.3.2.1.a Instrucciones de carga, almacenamiento y transferencia.

1. Cargas, almacenamientos y transferencias
2. Operaciones aritméticas
3. Multiplicación y división
4. Operaciones lógicas
5. Prueba de datos y manipulación de bits
6. Rotaciones y corrimientos

Función	Mnemónico	IMM	DIR	EXT	INDX	INDY	INH
Suma acumuladores	ABA						X
Suma acumulador B a X	ABX						X
Suma acumulador B a Y	ABY						X
Suma con acarreo a A	ADCA	X	X	X	X	X	
Suma con acarreo a B	ADCB	X	X	X	X	X	
Suma memoria a A	ADDA	X	X	X	X	X	
Suma memoria a B	ADDB	X	X	X	X	X	
Suma memoria D (16 bits)	ADDD	X	X	X	X	X	
Compara A con B	CBA						X
Compara A con memoria	CMPA	X	X	X	X	X	
Compara B con memoria	CMPB	X	X	X	X	X	
Compara D con mem. (16 bits)	CPD	X	X	X	X	X	
Ajuste decimal de A (BCD)	DAA						X
Decrementa el byte de memoria	DEC			X	X	X	
Decrementa el acumulador A	DECA						X
Decrementa el acumulador B	DECB						X
Incrementa byte de memoria	INC			X	X	X	
Incrementa el acumulador A	INCA						X
Incrementa el acumulador B	INCB						X
Complementa a 2 byte de mem.	NEG			X	X	X	
Complementa a 2 acumulador A	NEGA						X
Complementa a 2 acumulador B	NEGB						X
Sustraer con acarreo de A	SBCA	X	X	X	X	X	
Sustraer con acarreo de B	SBCB	X	X	X	X	X	
Sustraer memoria de A	SUBA	X	X	X	X	X	
Sustraer memoria de B	SUBB	X	X	X	X	X	
Sustraer memoria de D (16 bits)	SUBD	X	X	X	X	X	
Prueba para cero o menor	TST			X	X	X	
Prueba para cero o menor de A	TSTA						X
Prueba para cero o menor de B	TSTB						X

Tabla 3.3.2.1 b Instrucciones de operaciones aritméticas

1. **Cargas, almacenamientos y transferencias.**- Todas las actividades del MCU involucran al menos una transferencia desde la memoria o dispositivos periféricos a el CPU y viceversa. Las instrucciones de carga, almacenamiento o transferencia se resumen en la tabla 3.3.2.1.a. indicando también sus posibles modos de direccionamiento.

Operaciones Aritméticas.- Este grupo de instrucciones puede ejecutar operaciones aritméticas sobre una variedad de operandos. La tabla 3.3.2.1.b presenta este tipo de instrucciones. Este grupo de instrucciones puede realizar operaciones directas de 8 o 16 bits y fácilmente puede extenderse para operandos multipalabra puede también ejecutar directamente operaciones de complemento a dos, así como binarias. Las instrucciones de comparación realizan una sustracción dentro del CPU para actualizar los bits de código de condición sin alterar ningún operando.

3. **Multiplicación y División.**- Una operación de multiplicación y dos de división son proporcionadas. Una multiplicación de 8 bits por 8 bits da un resultado de 16 bits. La división integral de 16 bits por 16 bits da un resultado de 16 bits y un residuo de otros 16 bits. La división de fracciones (FDIV) divide un numerador de 16 bits entre un largo denominador de 16 bits, produciendo un resultado de 16 bits (una fracción binaria entre 0 y 0.99998). La tabla 3.3.2.1.c presenta estos operandos

Función	Mnemónico	INH
Multiplica ($A \times B$ i D)	MUL	X
División factorial (D/X i X ; r i D)	FDIV	X
División integral (D/X i X ; r i D)	IDIV	X

Tabla 3.3.2.1.c. Instrucciones de multiplicación y división.

4. **Operaciones lógicas.**- Este grupo de instrucciones se emplea para realizar las operaciones de lógica booleana AND, OR, OR exclusiva y complemento a uno. El resumen de este conjunto de instrucciones se presenta en la tabla 3.3.2.1.d.

Función	Mnemónico	IMM	DIR	EXT	INDX	INDY	INH
AND A con memoria	ANDA	X	X	X	X	X	
AND B con memoria	ANDB	X	X	X	X	X	
Prueba bit(s) A con memoria	BITA	X	X	X	X	X	
Prueba bit(s) B con memoria	BITB	X	X	X	X	X	
Complem. A uno byte de mem.	COM	X	X	X	X	X	
Complementa a uno A	COMA						X
Complementa a uno B	COMB						X
XOR A con memoria	EORA	X	X	X	X	X	
XOR B con memoria	EORB	X	X	X	X	X	
OR A con memoria	ORAA	X	X	X	X	X	
OR B con memoria	ORAB	X	X	X	X	X	

Tabla 3.3.2.1.d Instrucciones de operaciones lógicas

5. Prueba de datos y manipulación de bit.- Estas instrucciones se utilizan para trabajar con operandos tan pequeños como un bit, pero también pueden operar sobre cualquier localidad de 8 bits en el espacio de memoria de 64 Kbytes. La instrucción de prueba de bit (BITA o BITB) ejecuta instrucciones de AND dentro del CPU para actualizar los bits del registro de código de condición sin alterar ningún operando. La tabla 3.3.2.1.e presenta este conjunto de instrucciones.

Función	Mnemónico	IMM	DIR	EXT	INDX	INDY	INH
Prueba bit(s) A con memoria	BITA	X	X	X	X	X	X
Prueba bit(s) B con memoria	BITB	X	X	X	X	X	X
Borra bit(s) en memoria	BCLR		X			X	X
Pone bit(s) en memoria	BSET		X			X	X
Salto si se borran bit(s)	BRCLR		X			X	X
salto si se colocan bit(s)	BRSET		X			X	X

Tabla 3.3.2.1.e Instrucciones de prueba de dato y manipulación de bit.

6. Corrimientos y rotaciones.- Todas las funciones de corrimiento y rotación en el MCU involucran el bit de acarreo en el registro CCR además de un operando de 8 o 16 bits en la instrucción, el cual permite extenderse fácilmente a operaciones multipalabra. La instrucción de

corrimiento derecho (ASR) mantiene el valor original del bit más significativo del operando, con lo cual se facilita la manipulación de números en complemento a dos. Estas instrucciones se presentan en la tabla 3.3.2.1.f.

Función	Mnemónico	IMM	DIR	EXT	INDX	INDY	INH
Corrimiento aritmético izq. mem.	ASL			X	X	X	
Corrimiento aritmético izq. A	ASLA						X
Corrimiento aritmético izq. B	ASLB						X
Corrimiento aritmético izq. D	ASLD						X
Corrimiento aritmético der. mem.	ASR			X	X	X	
Corrimiento aritmético der. A	ASRA						X
Corrimiento aritmético der. B	ASRB						X
Corrimiento lógico izq. mem.	(LSL)			X	X	X	
Corrimiento lógico izq. A	(LSLA)						X
Corrimiento lógico izq. B	(LSLB)						X
Corrimiento lógico izq. D	(LSLD)						X
Corrimiento lógico der. mem.	LSR			X	X	X	
Corrimiento lógico der. A	LSRA						X
Corrimiento lógico der. B	LSRB						X
Corrimiento lógico der. D	LSRD						X
Rotación izquierda memoria	ROL			X	X	X	
Rotación izquierda A	ROLA						X
Rotación izquierda B	ROLB						X
Rotación derecha A	RORA						X
Rotación derecha B	RORB						X

Tabla 3.3.2.1.f Corrimientos y rotaciones.

7. Instrucciones de pila y registro índice.- El apuntador de pila siempre apunta a la siguiente localidad libre en la pila, a reserva de casos exclusivos. La razón común para transferir un valor desde el apuntador de pila al de un registro índice es el de permitir al direccionamiento indexado acceder la información que fue formalmente empujada sobre la pila. La tabla 3.3.2.1.g, resume todas las instrucciones de pila y de registro índice.

Función	Mnemónico	INIM	DIR	EXT	INDX	INDY	INH
Suma acumulador B a X	ABX						X
Suma acumulador B a Y	ABY						X
Compara X con mem. (16 bits)	CPX	X	X	X	X	X	
Compara Y con mem. (16 bits)	CPY	X	X	X	X	X	
Decrementa el apunt. de pila	DES						X
Decrementa el registro indice X	DEX						X
Decrementa el registro indice Y	DEY						X
Incrementa el apuntador de pila	INS						X
Incrementa el registro indice X	INX						X
Incrementa el registro indice Y	INY						X
Carga el registro indice X	LDX	X	X	X	X	X	
Carga el registro indice Y	LDY	X	X	X	X	X	
Carga el apuntador de pila	LDS	X	X	X	X	X	
Extrae X de la pila	PULX						X
Coloca X en la pila	PSHX						X
Coloca Y en la pila	PSHY						X
Almacena el registro indice X	STX	X	X	X	X	X	
Almacena el registro indice Y	STY	X	X	X	X	X	
Almacena el apuntador de pila	STS	X	X	X	X	X	
Transfiere apuntador de pila a X	TSX						X
Transfiere apuntador de pila a Y	TSY						X
Transfiere X al apunt. de pila	TXS						X
Transfiere Y al apunt. de pila	TYS						X
Intercambia D con X	XGDX						X
Intercambia D con Y	XGDY						X

Tabla 3.3.2.1 g Instrucciones de pila y del registro de indice.

3.3.2.2 Instrucciones de Registro de Código de Condición

Estas instrucciones permiten manipular los bits en el registro CCR. Inicialmente parecería que son instrucciones de puesta y borrado para cada uno de los 8 bits del CCR; sin embargo, estas instrucciones solo se presentan para tres de los ocho bits (C, I, y V).

Función	Mnemónico	INH
Borra bit de acarreo	CLC	X
Borra bit de interrupción mascarable	CLI	X
Borra bit de sobreflujo	CLV	X
Pone bit de acarreo	SEC	X
Pone bit de interrupción mascarable	SEI	X
Pone bit de sobreflujo	SEV	X
Transfiere A a CCR	TAP	X
Transfiere CCR a A	TPA	X

Tabla 3.3.2.2 Instrucciones del registro de código de condición.

3.3.2.3 Instrucciones de Control de Programa

Este grupo de instrucciones, el cual se utiliza mas para controlar el flujo del programa que para datos manipulados, se ha dividido en cinco grupos:

1. Saltos relativos
 2. Saltos absolutos
 3. Llamada de subrutina y retornos
 4. Interrupciones
 5. Varios
1. Saltos relativos.- Estas instrucciones permiten al CPU tomar decisiones basadas en los bits del registro de condición. El rango de saltos (localidades de -128 / +127). Todos los bloques de decisión en un diagrama de flujo deberán corresponder a la instrucción condicional de salto relativo que se resumen en la tabla 3.3.2.3.a.

Función	Mnemónico	REL	DIR	INDX	INDY	INH
Salto si borra acarreo	BCC	X				C=0 ?
Salto si coloca acarreo	BCS	X				C=1 ?
Salto si es igual a cero	BEQ	X				Z=1 ?
Salto si igual o mayor que	BGE	X				≥ Signado
Salto si mayor que	BGT	X				> Signado
Salto si mayor	BHI	X				> Sin signo
Salto si menor que	BLT	X				< Signado
Salto si menos	BMI	X				N=1 ?
Salto si diferente	BNE	X				Z=0 ?
Salto si mas	BPL	X				N=0 ?
Salto si borran bits en byte de memoria	BRCLR		X	X	X	Manip de bit
Salto nunca	BRN	X				NOP 3 ciclos
Salto si ponen bit en byte de memoria	BRSET		X	X	X	Manip de bit

3.3.2.3.a Instrucciones de salto relativo

2. Instrucciones de Salto Absoluto.- La instrucción salto absoluto permite pasar el control a cualquier dirección de los 64 Kbytes del mapa de memoria. Ver tabla 3.3.2.3.b

Función	Mnemónico	DIR	EXT	INDX	INDY	INH
Salto	JMP	X	X	X	X	

3.3.2.3.b Instrucciones de salto absoluto.

Función	Mnemónico	REL	DIR	EXT	INDX	INDY	INH
Salto a subrutina	BSR	X					
Salto absoluto a subrutina	JSR		X		X	X	X
Regreso de subrutina	RTS						X

Tabla 3.3.2.3.c Instrucciones de subrutina y retornos.

3. Instrucciones de rutina y de Retorno.- Estas instrucciones permiten una fácil manera de dividir las tareas de programación en bloques más manejables llamados subrutinas. La UCP automatiza el proceso de recordar la dirección del programa principal donde el procesador

debe regresar al terminar la subrutina. En la siguiente tabla se dan estas instrucciones.

4. Instrucciones de interrupciones.- Estas interrupciones se refieren a las operaciones de interrupción del CPU. En la tabla 3.3.2.3.d., se presentan estas tres interrupciones.

Función	Mnemónico	INH
Regreso de interrupción	RTI	X
Interrupción por software	SWI	X
Espera por interrupción	WAI	X

3.3.2.3.d Instrucciones de interrupciones.

5. Instrucciones varias.- La instrucción NOP puede ofrecer retrasos de tiempos, la de STOP va a detener por completo el reloj del CPU y la de TEST es una que solo se usa en la fabricación del microcontrolador.

Función	Mnemónico	INH
No operación (retraso de dos ciclos)	NOP	X
Paro de reloj	STOP	X
Prueba	TEST	X

3.3.2.3.e Instrucciones varias.

CAPITULO IV

SISTEMA DE COMUNICACIONES,

REGISTRO Y MANEJO DE DATOS.

4.1 Canales de comunicación para microcomputadoras.

4.1.1 Conceptos Básicos.

El nombre de "interfaz" es un término general para especificar la frontera o punto de contacto entre dos partes de un sistema. En sistemas digitales usualmente con él se hace referencia al conjunto de puntos de conexión de señales que el sistema o cualquiera de sus componentes presenta al exterior. El verbo "interconectar" o la frase "realizar una interfaz" significa enlazar dos o más componentes o sistemas a través de sus respectivos puntos de interfaz, de forma tal que entre ellos pueda transferirse información.

En un sistema con microprocesadores (microcomputadoras) hay principalmente dos tipos de interfaces: la interfaz microprocesador o microcomputadora que, como se muestra en la figura 4.1.1.a, corresponde al bus del sistema; y las interfaces presentadas por los dispositivos de E/S con que están asociadas, y varían grandemente en complejidad. Para conectar un dispositivo de E/S a un microprocesador, normalmente entre el dispositivo y el bus del sistema se incluye un circuito de interfaz de E/S. Este circuito sirve para acoplar los formatos de las señales y las características de temporización de la interfaz del microprocesador con la interfaz del circuito de E/S. La tarea global de enlazar un microprocesador y dispositivos de E/S en la forma que ilustra la figura 4.1.1.a. se denomina "interconexión" o realización de una interfaz.

El problema de interconexión de un microprocesador tiene dos aspectos principales (hardware y software). Los problemas relacionados con la interfaz hardware consisten en relacionar los circuitos de interfaz más adecuados, interconectar los hilos adecuados y asegurar que todas las señales de la interfaz tienen las características eléctricas pertinentes. El aspecto software de la interfaz implica la redacción de programas de E/S, que controlan la transferencia de información hacia y desde los dispositivos de E/S. Este flujo de información puede ser entre los dispositivos de E/S y la CPU (microprocesador). En la mayoría de los casos, sin embargo, las operaciones de E/S llevan consigo la transferencia de datos entre los dispositivos de E/S y la memoria principal. Los pasos individuales de una operación de E/S usual (por ejemplo, entrada o salida de una palabra

de datos) son bastante sencillos, y pueden fácilmente ser especificados usando el repertorio usual de instrucciones del microprocesador.

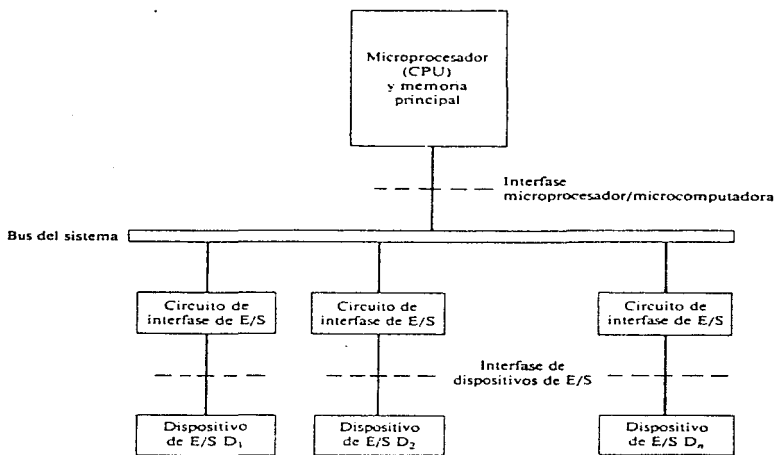


Figura 4.1.1.a Interfaces principales en una microcomputadora

Las características de interfaz de un dispositivos de E/S, tal como la longitud de palabra usada para la transferencia de datos al exterior y la máxima velocidad de transferencia de datos, con frecuencia son significativamente diferentes de las del microprocesador al que es conectado. Las

consideraciones anteriores implican que los circuitos de interfaz de E/S deben realizar las siguientes funciones :

- Conversión de datos.
- Sincronización .
- Selección del dispositivo.

La conversión de datos se refiere al acoplo de las características físicas y lógicas de las señales de datos empleadas por el dispositivo de E/S a las empleadas por el bus del sistema. Esto incluye una conversión de señales entre las formas analógica y digital y la conversión entre el formato de transmisión de datos serie (bit a bit) utilizado por algunos dispositivos de E/S y los formatos paralelo (palabra a palabra) utilizados por la mayoría de los microprocesadores. La sincronización se necesita para acomodar las diferentes velocidades operativas de la CPU, memoria principal y dispositivos de E/S. Esto usualmente requiere la inclusión en el circuito de interfaz de una o más palabras de memoria temporal ó intermedia.

Los dispositivos de E/S y la CPU funcionan independientemente en el sentido de que sus relojes internos no están sincronizados uno con otro. Por ello deben intercambiarse señales de control (listo, petición, reconocimiento, etcétera.) de conformidad a través de los circuitos de interfaz para iniciar o terminar las operaciones de E/S. La selección del dispositivo también implica el intercambio de señales de control.

La CPU sitúa una palabra de dirección asociada con el dispositivo en cuestión en el bus de direcciones del sistema, y activa las adecuadas líneas de control de disponible para entrada (leer) o salida (escribir).

Un microprocesador se utiliza frecuentemente para captar o medir variables no eléctricas tales como temperatura, posición, velocidad e intensidad de luz. Para este objetivo se utiliza un transductor o sensor para detectar la variable física en cuestión y generar una señal eléctrica proporcional. Un transductor típico produce una señal de datos que es analógica en vez de digital,

ésta señal debe convertirse en digital antes de poder enviarse al procesador. Por lo tanto, asociados con la mayoría de los transductores , debe de darse un proceso de conversión analógica a digital.

La tecnología LSI/VLSI ha posibilitado integrar en tan solo un **CI (circuito Integrado)** todos los circuitos de interfaz para dispositivos de E/S. Claramente estos circuitos de uso específico pueden simplificar satisfactoriamente el problema de la interfaz. También hay circuitos de uso general cuyas características pueden alterarse por software para adaptarse a las especificaciones de muy diversos tipos de dispositivos de E/S. Entonces como en muchos aspectos del diseño de sistemas digitales , las interfaces de E/S involucran al hardware y al software pudiendo afectar substancialmente a los costos de desarrollo y producción del sistema global. La mayoría de las familias de microprocesadores disponen de diversos circuitos de interfaz de E/S de uso específico y uso general. Un importante paso en el diseño de microcomputadoras es seleccionar los circuitos de cuyo costo-rendimiento sean los más adecuados para una aplicación dada.

4.1.2 Tipos de Transmisión

Una de las operaciones más comunes que se presentan en cualquier sistema digital es la transmisión de información de un lugar a otro. La información puede transmitirse a una distancia mínima de algunos milímetros sobre la misma tarjeta de circuito o a varios kilómetros cuando el operador de una terminal de computadora se comunica con una computadora que está en otra ciudad. La información que se transmite se encuentra en forma binaria y, por lo general, está representada por los voltajes que aparecen en las salidas del circuito de transmisión que se encuentran conectadas a las entradas del circuito de recepción. Dada estas características, la transmisión de datos puede clasificarse en:

- Transmisión serial
- Transmisión paralela

La figura 4.1.2.a ilustra la forma en que se transmite un número binario del circuito A al B utilizando la transmisión paralela. Cada bit del número binario está representado por una de las

salidas del circuito A, donde la salida A4 es el bit más significativo (MSB) y la A0 es el bit menos significativo (LSB).

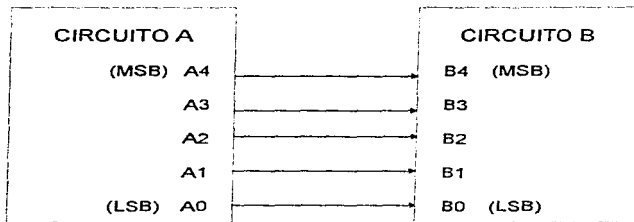


Figura 4.1.2.a Transmisión Paralela

Cada salida del circuito A está conectada a la correspondiente entrada del circuito B de manera que los cinco bits de información se transmiten en forma simultánea (paralela).

La figura 4.1.2.b muestra una conexión serial entre los circuitos A y B, en este caso existe una sola conexión. La salida del circuito A produce una señal digital cuyo nivel de voltaje cambia a intervalos regulares, de acuerdo con el número binario que se está transmitiendo. De esta forma la información se transmite en serie (un bit a la vez) sobre la línea de señal.

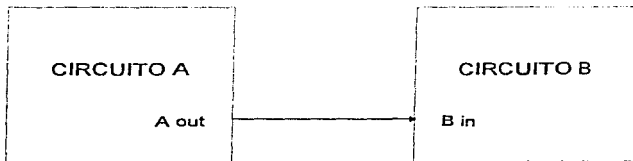


Figura 4.1.2.b Transmisión Serial

La diferencia principal entre estos dos tipos de transmisión es la velocidad. La transmisión de la información binaria de una parte a otra de un sistema digital se puede realizar más rápidamente mediante el uso de la forma de transmisión paralela, ya que todos los bits se transmiten en forma simultánea, mientras que la transmisión serial envía un bit a la vez. Sin embargo la transmisión paralela necesita más líneas de señal conectada entre emisor y receptor, es decir, emplea más hardware y por lo tanto es más costosa y se incrementa con la longitud del bus. La transmisión de datos en forma paralela a grandes distancias (20 metros o más) es también complicada por un fenómeno llamado deslizamiento de la señal. Este se refiere al hecho de que un conjunto de señales emitidas simultáneamente por una fuente de señal a través de hilos distintos no siempre llegan simultáneamente al elemento de destino de datos; en su lugar llegan en instantes de tiempo ligeramente diferentes, o "deslizadas". El deslizamiento es originado por pequeñas diferencias en las resistencias y capacidades eléctricas que provocan retardos en la propagación de las señales de línea. Esto se corrige con un "antideslizamiento", que consiste en retardar el proceso de transmisión de la señal hasta que la señal más lenta llegue a su destino. Esto puede reducir significativamente la velocidad máxima del bus paralelo. Por ésta razón los buces serie, son utilizados para transmisión de datos a través de largas distancias.

Por convección, el envío de datos en la transmisión serial generalmente inicia con el bit menos significativo. Se utiliza una señal de reloj para diferenciar entre varios bits. La entrada-salida (E/S) serial de datos es más común que la paralela. La velocidad de los datos seriales es conocida como velocidad de transmisión (baud rate) y está definida como el número de bits transmitidos en un segundo. La velocidad mínima más común es de 300 bps (bits por segundo). Los submúltiplos de ésta son: 50, 100 y 150 bps, los múltiplos son: 600, 1200, 2400, 4800, 9600 hasta 19200 bps.

A su vez, la transmisión serial de datos se divide en:

- Sincrona
- Asíncrona

- Transmisión Serial Síncrona

La característica principal de la transmisión serial síncrona, es que los datos son enviados o recibidos basándose en una señal de reloj. Después de decidir la velocidad de transmisión (bps), el dispositivo transmisor envía un bit del dato en cada pulso de reloj.

Para poder interpretar correctamente los datos, el dispositivo receptor debe conocer el inicio y el fin de cada unidad de datos, es por eso que el dispositivo receptor debe conocer el número de unidades de datos a transmitirse. Los datos son divididos en pequeños bloques de bits, a cada uno de estos bloques se le llama **palabra (word)**, y puede consistir de 5 a 8 bits. La longitud de palabra comúnmente utilizada es de siete y ocho bits debido a que las palabras formadas se asemejan a la descripción de caracteres del código ASCII.

Si en una palabra se utilizan menos de 8 bits para un dato, el resto será ignorado. Cada palabra puede contener un bit de paridad (par o impar). Una palabra de 9 bits (8 bits de datos y 1 de paridad) puede ser representada como:

XXXX	XXXX	P
8 BITS DE DATOS		1 DE PARIDAD

El receptor síncrono espera en un modo de "caza" los datos. Tan pronto como el receptor reconoce uno o dos bits de sincronía, según el número de bits de sincronía empleados, empieza a interpretar los datos. En transmisión serial síncrona, el transmisor requiere enviar continuamente los datos al receptor, si los datos no están listos para ser transmitidos, el transmisor enviará caracteres de sincronía hasta que los datos se encuentren listos.

- Transmisión Serial Asíncrona

En este tipo de transferencia de datos, el dispositivo transmisor no requiere estar sincronizado con el receptor. El transmisor puede enviar una o más palabras, cuando tiene datos listos para ser

enviados. Cada palabra puede tener un formato que indique el inicio y el fin de cada palabra, es decir, bits de inicio y de paro (bits de start y stop) con un pulso de doble longitud. De esta forma entre el último bit de la palabra y el bit de paro se tendrá un bit de paridad que se utiliza para checar la integridad de los datos. Los bits de datos, de start, stop y el de paridad forman un frame de datos.

El formato para datos seriales asíncronos contiene la siguiente información:

- uno, uno y medio o dos bits de start con nivel bajo
- de 5 a 8 bits de datos
- un bit de paridad opcional, ya sea paridad par, impar, paridad por marca (bit encendido), paridad espacio (bit apagado) o no paridad.
- uno, uno y medio o dos bits de stop con nivel alto (Considérese que 1.5 bit corresponde a 1.5 veces el intervalo del bit).

Para este tipo de transmisión se requiere de un circuito integrado de interfaz entre la microcomputadora y el dispositivo de Entrada/Salida, que tenga las siguientes funciones:

- Convertir una unidad de datos de 8 bits paralelos de la microcomputadora a datos seriales para su transmisión al dispositivo serial.
- Convertir los datos seriales del dispositivo de Entrada/Salida a una unidad de datos de 8 bits paralelos para transmitir la unidad a la microcomputadora.

Comúnmente para efectuar la interfaz de un bus serie con el bus del sistema se utiliza un receptor-transmisor universal asíncrono o UART (de "Universal Asynchronous Receiver-Transmitter")

El UART transforma dentro de una computadora las señales paralelas (generalmente 8 líneas) en un tren de pulsos seriales. Este circuito acepta 8 líneas de datos (paralelas) de entrada y/o salida

efectuando una conversión a datos seriales, además de que puede trabajar en ambos sentidos, por lo que puede convertir señales seriales en paralelo que pueda reconocer la PC.

4.1.3 Estándares de Comunicación

4.1.3.1 Conectores e Interfaces.

Los interfaces de nivel físico se utilizan para conectar dispositivos de usuario al circuito de comunicaciones. Para llevar a cabo esta importante función en la mayoría de las especificaciones relativas a interfaces de nivel físico describen tres características principales.

1. **Características Eléctricas.** Estas son las que determinan los niveles de voltaje (o corriente) y la temporización de los cambios eléctricos que representan los unos y los ceros. Muchos de los protocolos de nivel físico clasifican estas funciones en cuatro grupos : control, sincronismo, datos y masa.
2. **Características Mecánicas.** Describen los conectores y los hilos del interfaz. Por lo general, todas las líneas de datos, de señalización y de control están incluidas en un mismo cable, y se conectan a enchufes terminadores situados en ambos extremos del cable.
3. **Características Funcionales.** Describen lo que deben hacer los conectores, y la secuencia de eventos necesaria para llevar a cabo la transferencia efectiva de datos a través de la interfaz.

Diferentes asociaciones , organismos y fabricantes están tratando continuamente de definir una serie de especificaciones, de tal manera, que estando normalizadas las diferentes interfaces y protocolos, sean compatibles unos equipos con otros. Entre éstos destacan el **CCITT (Comité Consultivo Internacional de Telecomunicaciones y Telefonía)**, el **ANSI (Instituto Nacional Americano de Normalización)**, la **ISO (Organización Internacional de Estándares)** y la **EIA (Asociación de Industrias Electrónicas)**.

La EIA enfoca su actividad principalmente en el campo de la normalización eléctrica, y entre sus logros más destacados cabe mencionar las interfaces RS-232C y RS-449, convertidos en estándares y usados por la mayoría de los fabricantes de equipos de cómputo, manteniendo una política abierta a otros campos diferentes a los propios, facilitando al usuario la elección de los sistemas más adecuados a sus necesidades y ampliando la gama de equipos periféricos que éste pueda elegir, sin problema alguno de compatibilidad. Podemos ver a una interfaz como una frontera compartida que está determinada con una serie de características eléctricas y funcionales perfectamente definidas, dicho de otra manera, por unas características físicas de interconexión comunes (ver Figura 4.1.3.1.a),

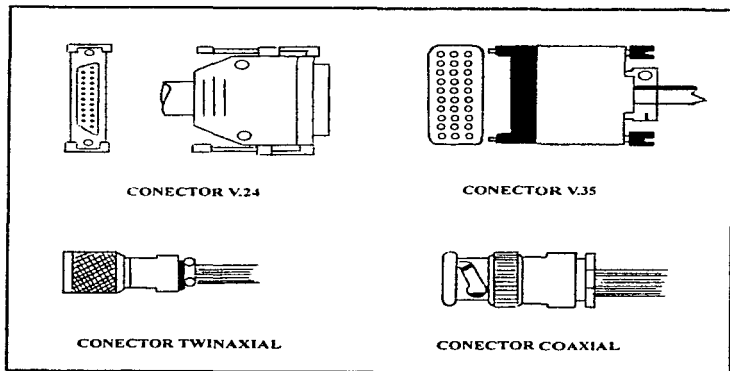


Figura 4.1.3.1.a Diferentes tipos de conectores

Características de la señal y significado de las mismas, regulando al mismo tiempo el intercambio de ellas, de tal forma que podamos establecer y mantener una conexión entre dos terminales de datos.

4.1.3.2 Interfaz RS-232C

En la actualidad, la gama de protocolos y estándares de comunicación existentes es muy amplia, sin embargo, en la práctica se ha extendido el uso de la bien conocida interfaz serie, RS-232C.

1) Características mecánicas.

El interfaz RS-232C, se encuentra entre el equipo terminal de datos (DTE) y el equipo de comunicación de datos (DCE), utilizando intercambio de datos binarios en serie. Al tratar de establecer la conexión de un equipo con otro, lo primero que se plantea es la compatibilidad física de los conectores que se utilizan. La interfaz RS-232C define un conector de 25 terminales o pines, el conector macho está asociado del lado del DTE y el conector hembra del lado del DCE. Típicamente se usa el tipo DB-25, aunque como este no está definido en el estándar, otros fabricantes utilizan uno diferente. La asignación de señales a los contactos del conector, así como la longitud máxima recomendada, determinada por la capacidad del mismo se muestra en la tabla 4.1.3.2.a.

2) Características eléctricas.

La parte fundamental dentro de la recomendación RS-232C es la definición de las características de las señales que por él transitan, el nivel eléctrico cubre el margen de tensiones y corrientes de cada pin, estando limitado el voltaje entre ± 3 y ± 25 volts, y la intensidad a 3 miliamperes, de tal manera que aún en el caso de corto circuito entre pines no se cause daño alguno a los diversos componentes, ver figura 4.1.3.2.b.

PIN	NOMBRE	DTE DCE	DESCRIPCION	CCITT	EIA
1	FG		TIERRA DE PROTECCION (Frame Ground)	101	AA
2	TD	→	TRANSMISION DE DATOS (Transmitter data)	103	BA
3	RD	←	RECEPCION DE DATOS (Received Data)	104	BB
4	RST	→	PETICION DE EMISION (Request to Send)	105	CA
5	CTS	←	PREPARADO PARA TRANSMITIR (Clear to send)	106	CB
6	DSR	←	MODEM PREPARADO (Data Set Ready)	107	CC
7	SG		TIERRA DE REFERENCIA (Signal Ground)	102	AB
8	DCD	←	DETECTOR DE PORTADORA (Data Carrier Detected)	109	CF
9		←	VOLTAJE POSITIVO DE TEST (Positive DC test voltage)		
10		←	VOLTAJE NEGATIVO DE TEST (Negative DC voltage)		
11	QM	←	SELECTOR DEL CANAL DE TRANSMISION (Equalizer mode)	BELL	208A
12	SDCD	←	DETECTOR DE PORTADORA SECUNDARIO	122	SCF
13	SCTS	←	PREPARADO PARA TRANSMITIR SECUNDARIO	121	SCB
14	STD	→	TRANSMISION DE DATOS SECUNDARIO	118	SCA
	NS	→	SINCRONISMO NUEVO (New Sync)	BELL	208A
15	TC	←	RELOJ DE TRANSMISION (Transmitter Clock)	114	DB
16	SRD	←	RECEPCION DE DATOS SECUNDARIO	119	SBB
	DCT	←	RELOJ DE TRANSMISION DIVIDIDO (Divided Clock Transmitter)	BELL	208A
17	RC	←	RELOJ DE RECEPCION (Received Clock)	115	DD
18	DCR	←	RELOJ DE RECEPCION DIVIDIDO (Divided Clock Received)	BELL	208A
19	SRTS	→	PETICION DE EMISION SECUNDARIO	120	SCA
20	DTR	→	TERMINAL DE DATOS PREPARADA (Data Terminal Ready)	108 2	CD
21	SQ	←	DETECTOR DE CALIDAD DE SEÑAL (Signal Quality Detect)	110	CG
22	RI	←	INDICADOR DE LLAMADA (Ring Indicator)	125	CB
23		→	SELECTOR DE VELOCIDAD (Data Rate Selector)	111	CH
		←	SELECTOR DE VELOCIDAD (Data Rate Selector)	112	CI
24	TC	→	RELOJ DE TRANSMISION EXTERNO (Ext. Transmitter Clock)	113	DA
25		→	OCUPADO (Busy)	BELL	113B

Tabla 4.1.3.2.a. Interfaz EIA - CCITT

Dentro de estas características eléctricas debe destacar la impedancia de la interfaz, con una resistencia de carga comprendida entre 3000 y 7000 ohms, y la capacidad del mismo con un valor máximo de 2500 picofarads (la capacidad por metro de los cables utilizados se considero del orden de 160 picofarads/metro). La distancia máxima de los cables debe ser de 15 metros para poder garantizar la correcta transmisión de la señal digital sin distorsiones apreciables.

De igual manera, estas características para la recomendación V.28 del CCITT se aplican a todos los circuitos binarios de enlace con velocidades inferiores a los 20,000 bps, siendo los valores más comunes los de 300, 600, 1200, 2400, 4800, 9600 y 19200 bits por segundo (bps).

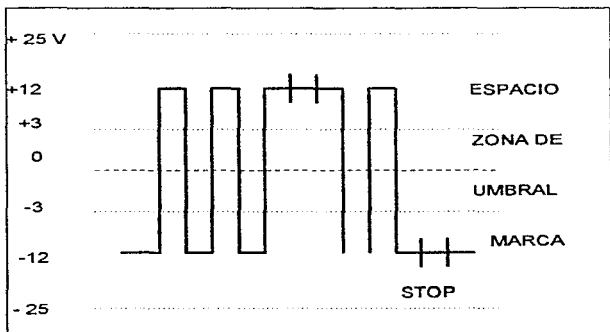


Figura 4.1.3.2 b. Representación de un carácter ASCII

3) Características funcionales.

Son estas las más importantes desde el punto de vista del usuario, pues son las que debe conocer para aplicarlas correctamente. Dentro de éstas destacan las siguientes:

- **Transferencia de datos a través de la interfaz**
- **Control de las diversas señales de interfaz**
- **Proporcionar las diversas señales de sincronización que regulan la transferencia de bits**
- **Referencia de la señal eléctrica**

A continuación se define la descripción funcional de los circuitos de intercambio, teniendo en consideración que pueden existir diversas opciones. Asimismo, la recomendación RS-232C contempla dos canales independientes, principal y secundario, coexistiendo en la misma interfaz, se mencionará uno solo, ya que son similares.

Circuitos de Intercambio de Datos

- **TD (Transmitted Data)**
- **RD (Received Data)**

Estos son los más importantes debido a que representan la información a transmitir. Todos los demás circuitos están destinados a garantizar esta correcta transmisión.

Circuitos de Control de intercambio

- **RTS (Request to send)**
- **CTS (Clear to Send)**
- **DSR (Data Set Ready)**
- **DCD (Data Carrier Detect)**
- **DTR (Data Terminal Ready)**
- **RI (Ring Indicator)**

Estos circuitos, entre otros, permiten establecer y controlar el enlace y la transferencia de información entre un equipo DTE y un equipo DCE, estableciendo la llamada y el control de la señal portadora en el medio de enlace.

Circuitos de Sincronización

En una transmisión asíncrona, es necesario el uso de unas señales que regulen la cadencia de la información, estas son las señales de reloj.

- TC (Transmitter Clock)
- RC (Receiver Clock)
- ETC (External Transmitter Clock)

Estas señales se encargan de mantener perfectamente sincronizados al transmisor y al receptor, permitiendo el correcto muestreo de la señal y evitando errores, además de temporizar las señales de datos.

Circuitos de Referencia

- GND (Protective Ground)
- SG (Signal Ground)

Todas las señales tienen como referencia el pin 7 del conector, por lo tanto es fundamental ya que su ausencia impedirá el correcto funcionamiento de las demás al alterar su significado. El pin 1 está conectado al chasis del equipo y sirve de protección tanto eléctrica como magnéticamente. Usualmente se unen ambos circuitos, aunque es necesario tomar las debidas precauciones.

4.1.3.3 Interface RS-449

La norma RS-232C es y ha sido la más utilizada por el conjunto de fabricantes de equipos de cómputo, pero presenta una serie de limitaciones que hoy en día han sido ampliamente superadas y recogidas en otras normativas, tal como es el caso del RS-449, que presenta un significativo avance frente a su antecesor.

Por ejemplo, la norma RS-232C está limitada a velocidades de 20,000 bps para una separación máxima entre DTE y DCE de 15 metros, con una señal poco inmune a las interferencias exteriores

y un número limitado de circuitos, aparte de que el conector utilizable no está mecánicamente definido, lo que puede presentar incompatibilidades entre equipos. La norma RS-449 trata de resolver estos inconvenientes, proporcionando 37 circuitos básicos, más otros 10 adicionales, permitiendo velocidades de hasta 2 Mbits/segundo y distancia de 600 metros; en caso de utilizar circuitos balanceados, una gran inmunidad frente al ruido. Además es compatible con las recientes recomendaciones del CCITT.

Por tanto, esta nueva norma está destinada, a reemplazar a la anterior. Las características eléctricas de las mismas se especifican en las normas RS-422 (para circuitos equilibrados) y la RS-423 (para circuitos no equilibrados), compartiendo o no una tierra común. Otra característica es la separación de los canales principal y secundario o de retorno.

La norma RS-449 aporta, además de mayor velocidad y alcance, una serie de nuevas funciones, orientadas al servicio, previendo una mayor cantidad de pruebas y recuperación bajo el control del equipo terminal de datos (DTE).

Otra diferencia es que al estar el margen de voltajes situado entre ± 0.2 y ± 6 volts, la región umbral, en la que las señales permanecen indefinidas, se reduce de 6 volts, en la RS-232C, a tan solo 0.4 volts, siendo además las exigencias de la toma de tierras mucho menos críticas.

Recientemente el CCITT ha establecido la recomendación X.21, que incluye un protocolo para enviar y recibir llamadas, así como los datos mediante transmisión sincrónica duplex, utilizando solamente 6 señales y con una velocidad mínima de 64000 bps, con posibilidad de establecer una conexión directa a una red telefónica digital.

Para utilizar ésta recomendación se requiere una mayor inteligencia, tanto por parte del TDE como del DCE, no siendo adecuada para la utilización en línea punto a analógicas, líneas conmutadas ó líneas semi-duplex, que hoy en día constituyen la mayoría de las existentes.

RS-449		RS-232C		RECOMENDACIÓN V.24 CCITT	
SG	Tierra de señalización	AB	Tierra de señales	102	Tierra de señales
SC	Emisión común			102 a	DTE común
RC	Recepción común			102 b	DCE común
IS	Terminal en servicio				
IC	Llamada entrante	CE	Indicador señal llamada	125	Indicador señal llamada
TR	Terminal preparada	DC	Terminal datos prep.	108 2	Terminal datos preparado
DM	Modo de datos	CC	Conjunto datos prep.	107	Conjunto datos preparado
SD	Emisión de datos	BA	Transmisión datos	103	Transmisión datos
RD	Recepción de datos	BB	Recepción datos	104	Recepción datos
TT	Sincronismo de terminal	DA	Sincronía en trans.	113	Sincronía en transmisión
ST	Sincronización emisión	DB	Sincronía en trans.	114	Sincronía en transmisión
RT	Sincronización recep	DD	Sincronía en recep.	115	Sincronía en recepción
RS	Peticion de emision	CA	Peticion emision	105	Peticion emision
CS	Preparado transmitir	CB	Prep para transmitir	106	Prep. Para transmitir
RK	Receptor preparado	CF	Detector señal lin. Rec.	109	Detector de señal de lin. Rec.
SQ	Calidad de señal	CG	Detector calidad señal	110	Detector de calidad de señal
NS	Señal nueva				
SF	Frecuencia seleccionada			126	Frec. De transmisión selec.
SR	Selector Vel. Binaria	CH	Selector Vel. Binaria	111	Selector de vel. Binaria
SI	Indicador Vel. Binaria	CI	Selector Vel. Binaria	112	Selector de vel. Binaria
SSD	Emisión datos sec.	SBA	Transmisión datos sec.	118	Trans. Dat. Canal de vuelta
SRD	Recepción datos sec.	SBB	Recepción datos sec.	119	Recep. Dat. Canal de vuelta
SRS	Peticion de emision sec.	SCA	Peticion emision sec.	120	Trans. De señal lin. X canal
SCS	Preparado Trans. Sec.	SCB	Prep. Transmitir sec.	121	Canal de vuelta preparado
SRR	Receptor sec. Prep.	SCF	Receptor Prep. Sec.	122	Detector de lin. Rec. X canal
LL	Loopback local			141	Loopback local
RL	Loopback remoto			140	Loopback remoto
TM	Modo Test			142	Indicador de test.
SS	Espera Selectiva			116	Espera selectiva
SB	Inicador de espera			117	Indicador de espera

Tabla 4.1 3.4.a Tabla de equivalencias RS-449, RS-232C y V.24 CCITT

La mayor ventaja X.21 sobre RS-232C y RS-449 radica en el hecho de que las señales van codificadas en forma digital serie, lo que facilita el acceso a muchas de las ventajas que se dan al utilizar la técnica de conmutación de paquetes; siendo éste el nivel más bajo (físico) del protocolo X.25, definido como la interface entre el DTE y el DCE para terminales que funcionan en modo paquete en redes públicas de datos. Precisamente, por el hecho de tener las señales codificadas es por lo que solo una de las líneas de la señal son capaces de desempeñar todas las funciones.

4.1.3.4 Normas Equivalentes.

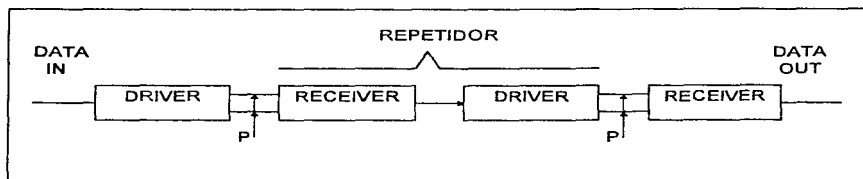
Puesto que la EIA es un organismo estadounidense y el CCITT uno europeo, los fabricantes que descan estar en un mercado internacional deberán cumplir las recomendaciones de cada uno de ellos; esto es más sencillo de lo que se piensa, pues cada norma de EIA suele tener un equivalente en el CCITT; así las recomendaciones equivalentes a las normas RS-449, RS-422 y RS-423 son las V-24, V-11 y V-10 respectivamente.

En la tabla 4.1.3.4.a se muestran las equivalencias de estas normas. Aunque los nombres dados a cada circuito pueden ser totalmente diferentes, esos son funcionalmente idénticos lo que permite la interconexión de equipos independientemente del lugar de procedencia.

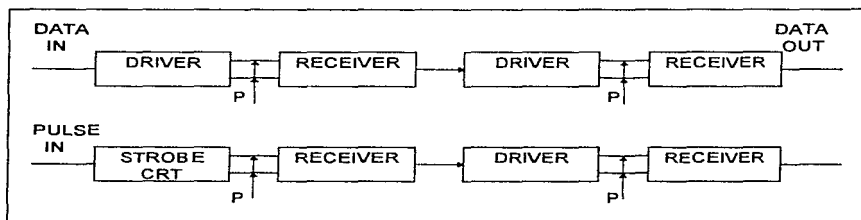
4.1.3.5 Repetidores para largas distancias.

En algunos casos, las líneas de comunicación pueden ser muy largas de tal forma que el nivel de ruido permisible alcance su límite máximo o que la atenuación de la señal no permita obtener una buena recepción. En tal caso, es recomendable la utilización de un pequeño dispositivo que amplifique y restablezca la señal, de tal forma que el receptor detecte la señal con un buen nivel, a estos dispositivos se les conoce como drivers. En la figura 4.1.3.5.a se muestra el esquema básico de un driver y receptor como repetidor.

Si se desea efectuar una transmisión de multicanal, se recomienda utilizar repetidores tanto para la señal como para el reloj de sincronía o de pulsos utilizando cable de par trenzado, este esquema también se muestra en la figura 4.1.3.5.a.



a) Línea de canal sencillo (Single channel line)



b) Línea Multicanal con strobe

Figura 4.1.3.5.a. Repetidores Driver-Receiver.

Actualmente existe un gran número de drivers y receptores para utilizarse en la interfaz RS-232C. Los requerimientos básicos del RS-232C para drivers y receptores se muestran en las tablas 4.1.3.5.b y 4.1.3.5.c.

Corto circuito de salida	Debe proveer tierra de protección Corriente de salida < 500 mA
Resistencia de salida (off)	> 300 Ω
Voltaje de salida en circuito abierto	0 a ± 25 V
Voltaje de salida del drive con carga de 3 K Ω a 5 K Ω	> 5 V y < 15 V
Tiempo de levantamiento y caída en los límites de transición de 3 V a -3 V	1 ms ó $\leq 4\%$ de duración nominal de la señal (la que sea menor)
Velocidad de transmisión máxima	20.000 bits por segundo

Tabla 4.1.3.5.b. Requerimientos para el driver con interfaz a RS-232C

Resistencia de entrada	> 3 K Ω y < 7K Ω
Carga capacitiva de entrada	< 2500 pF
Límites del voltaje de entrada	± 25 V
Voltaje de entrada a circuito abierto	< 2.0 V
Velocidad de transmisión máxima	20.000 bits por segundo

Tabla 4.1.3.5.c. Requerimientos para el receptor con interfaz a RS-232C

Voltaje de salida del drive a circuito abierto en la salida (V _o)	± 4 V a ± 6 V
Voltaje de salida del drive con carga terminal de 450 Ω (V _t)	≥ 0.9 V _o
Corriente de salida a corto circuito (I _s)	< 150 mA
Corriente de escape a la salida (off) (I _N)	< 100 μ s voltaje de salida de -6 V a 6 V
Slew rate de salida (S _r)	No debe exceder 15 V/ μ s en cualquier punto durante el periodo de transición.
Voltaje de salida	Entre 0.1 y 0.9 V _{ss}
Tiempo de transición (t _r)	Para pulsos de 1 ms o mayores, el tiempo de transición medido entre 0.1 y 0.9 V _{ss} debe estar entre 100 μ s y 300 μ s.

Tabla 4.1.3.6.a. Requerimientos para el driver con interfaz a RS-423-A

Corriente de entrada con un voltaje de alimentación entre -10V y 10V (I _{in})	La corriente de entrada debe caer dentro del área sombreada de la figura 4.1.3.6.c
Resistencia de entrada (R _{in})	4 k Ω bajo condiciones de encendido ó apagado.
Sensibilidad de entrada con un voltaje de entrada de -7V a 7 V (V _{th})	La entrada diferencial requerida para asegurar la recepción de la señal es de 200 mV. El receptor debe mantener una operación correcta a voltajes de entrada diferencial desde 200 mV a 6 Volts.
Voltaje máximo de entrada diferencial (V _{id})	El voltaje máximo de entrada sin dañar el receptor es de 12 V
Balance de entrada con cualquier voltaje de modo común desde -7 V a 7 V.	Las características del voltaje de entrada balanceado debe ser tal que el receptor permanezca en su estado binario con un voltaje diferencial de 400 mV aplicando una resistencia de 500 Ω en cada entrada.
Receptores múltiples y carga total	Pueden ser conectados hasta 10 receptores en cada línea. La carga total de los múltiples receptores y la circuitería de seguridad debe tener una resistencia mayor ó igual a 400 Ω
Polarización a tierra	El cable de señal a tierra debe estar solo en el lado del driver al final de la línea.

Tabla 4.1.3.6.b. Requerimientos para el receptor con interfaz RS-423-A

4.1.3.6. Estándar RS-423-A

Este estándar especifica las características de circuitos digitales para intercambio de señales binarias en serie con voltajes desbalanceados entre el DTE y DCE. Estos dispositivos son utilizados en comunicación de datos a baja velocidad o en funciones de control. El RS-423-A permite utilizar un driver y hasta 10 receptores en una sola línea de datos. Los requerimientos básicos para drivers y receptores de éste estándar se muestran en las tablas 4.1.3.6.a. y 4.1.3.6.b.

4.1.3.7. Estándar RS-422-A

Las transmisiones a gran velocidad entre una computadora y sus componentes periféricos a través de grandes distancias, bajo condiciones de nivel de ruido altos, pueden efectuarse con un poco de dificultad con drivers y receptores de tipo single-ended.

PARÁMETRO	RS-232-C	RS-423-A	RS-422-A
Modo de operación	Single-ended	Single-ended	Diferencial
Número de Drivers y Receptores	1 driver 1 receptor	1 driver 10 receptores	1 driver 10 receptores
Longitud máxima del cable (ft)	50	4000	4000
Velocidad de transmisión máx. (bps)	20 K	100 K	10 M
Voltaje máximo en modo común	± 25 V	± 6 V	6 V -0.25 V
Salida del driver	± 5 V min ± 15 V máx	± 3.6 V min ± 6 V máx	± 2 V min
Carga en el driver	3 k Ω a 7 k Ω	450 Ω min	100 Ω min
Slew rate del driver	30 V/ μ s max	Control ext.	NA
Corriente de corto circuito a la salida del drive	500 mA a V_c ó a tierra	150 mA a tierra	150 mA a tierra
Resistencia de salida en el drive (On) (Estado Z) (Off)	NA 300 Ω	NA 60 K Ω	NA 60 K Ω
Resistencia de entrada al receptor	3 K Ω a 7 K Ω	4 K Ω	4 K Ω
Sensibilidad del receptor	± 3 V	± 200 mV	± 200 mV

Tabla 4.1.3.7.a. Tabla comparativa de los estándares RS-232C, RS-423-A y RS-422-A

Este estándar permite efectuar interfaces a distancias largas con voltaje balanceado entre receptor y receptor. En la tabla 4.1.3.7.a. se muestra la comparación entre RS-232C, RS-423-A y RS-422-A.

El circuito digital de interfaz con voltaje balanceado se utiliza normalmente para transmisión de datos, reloj o líneas de control en donde la velocidad de señalización va desde 100 kbps hasta 10 Mbps. Las especificaciones del estándar RS-422-A no tienen restricciones en frecuencias de

operación. La transmisión diferencial en líneas balanceadas son recomendables en las siguientes condiciones :

- Para interconectar líneas muy largas de operación desbalanceada.
- Líneas de transmisión expuestas a altos niveles de ruido o niveles electromagnéticos.

El RS-422-A tiene un voltaje de salida (diferencial) con una impedancia de 100 ohms o menos. Su voltaje de salida diferencial está en el rango de 2 V a 6 V máximo.

El voltaje de salida balanceado se define como sigue : El voltaje de salida diferencial no debe ser menor a 2.0 V con dos resistencias de 50 ohms ($\pm 1\%$) conectadas en serie entre las salidas. La diferencia entre los voltajes de salida deben ser menores a 0.4 V. El voltaje de offset (Vos) a la salida del driver debe ser medido desde la unión de las dos resistencias de 50 ohms y la tierra del driver, no debe exceder 3.0 V y la corriente debe ser menor a 150 mA.

4.2 Comunicación entre el Microcontrolador y la PC.

4.2.1 Selección del canal de comunicación entre la PC y el MCU.

El objetivo de este punto es seleccionar una interface que se ajuste a las necesidades de comunicación entre el MCU y la PC.

En los puntos anteriores fueron vistas algunas interfaces que podrían ser opciones de solución para el sistema. Para la selección de la interfaz a ser empleada en este sistema, es necesario considerar las características físicas, eléctricas, ambientales, que el sistema observará, tales como :

- Localización de la interfaz, esto es, el lugar físico en donde se ubicará el MCU y la PC.
- Distancia entre el MCU y la PC.
- Tipo de cableado a utilizar.
- Características eléctricas del MCU y la interface a utilizar.

Para nuestro caso, la PC y el MCU se encontrarán ubicados uno junto al otro en el momento de la transferencia de información, por lo tanto, esto no implicará mayores dificultades en cuanto a la comunicación y conexión de ambos dispositivos, respecto a la selección de la interfaz, el problema de la distancia no se presenta para este caso, ya que la distancia sería mínima.

Otro aspecto a considerar son las condiciones ambientales en que se encontrarán los dispositivos. Las condiciones más extremas se presentarán en el momento en que el MCU realice el registro de las variables físicas durante el recorrido del vehículo (velocidad, temperatura, distancia, etc). Dada la característica del sistema de poderse montar y desmontar fácilmente, suponemos que al momento de la transferencia de información, tanto la PC como el MCU se encontrarán en alguna oficina o lugar en donde las condiciones son muy favorables, por lo que el ruido o las variaciones de temperatura no son factores de consideración en el canal de comunicación; el blindaje y los componentes del circuito del MCU para las condiciones durante el recorrido, son tratados en otro punto.

Así, que dadas las características que la interfaz debe observar, tales como : máxima longitud de cable, compatibilidad con el MCU, costos, etc.; se decidió por seleccionar la interfaz RS-232C, ya que tiene un uso muy difundido para éste tipo de aplicaciones actualmente.

Esta interfaz tiene la ventaja de que se acopla perfectamente al MCU y en cuanto al cableado, este es de muy bajo costo.

4.3 Características y Selección de la Base de Datos Relacional.

Para el almacenamiento de los datos adquiridos por el microcontrolador se necesitará de una base de datos que nos permita no sólo guardar los datos sensados, sino que tenga la flexibilidad y manejo de información, sin caer en la duplicidad. Por lo que se pretende utilizar una base de datos relacional que además de permitir una mayor explotación de la información, evita la duplicidad de ésta.

4.3.1 Características.

4.3.1.1 Borland Paradox 5.0

- **Características Estándar**

- ⇒ Precio : Introductorio \$495 Dólares, actualización \$200 Dólares
- ⇒ Registro máximo por archivos de datos : Limitado al espacio en disco
- ⇒ Número de tipos de datos : 17
- ⇒ Capacidades relacionales : Si tiene
- ⇒ Número de tablas que pueden ser ligadas : Ilimitadas

- **Herramientas de entrada de datos**

- ⇒ Prueba patrón : Si
- ⇒ Verificación basado en normas : Si
- ⇒ Fuerza al uso de mayúsculas : Si
- ⇒ Campos incrementados automáticamente : Si

- **Formatos de base de datos importados y exportados**

- ⇒ ASCII delimitado por coma : Si
- ⇒ Paradox : Si
- ⇒ dBase : Si

- ⇒ 1-2-3 : Sí
- ⇒ Excel : Sí
- ⇒ EL usuario puede añadir y borrar índices : Sí

- **Facilidades de Consulta**

- ⇒ Consulta por ejemplo : Sí
- ⇒ Herramienta de ayuda para consulta : Sí
- ⇒ Salva consultas : Sí
- ⇒ Criterio de ordenación máximo : 255
- ⇒ Herramientas de trazado y gráficos : Sí

- **Formas**

- ⇒ Número de patrones suministrados : 8⁴
- ⇒ Reportes tabulado-transversal : Sí
- ⇒ Proporciona conexión a bases de datos SQL : Sí
- ⇒ Lenguaje de Programación : Sí

- **Servicio y Soporte**

- ⇒ Soporte telefónico diario libre de impuesto : 11 horas
- ⇒ Duración de soporte técnico gratis : Ilimitado únicamente para instalación y configuración.
- ⇒ BBS : Sí
- ⇒ Foros en línea : Sí
- ⇒ Fax : Sí

- **Interfaz/tutorial**

- ⇒ Menús lógicos : Malos
- ⇒ Controles de Mouse : Bueno
- ⇒ Controles de teclado : Bueno

- ⇒ Documentación : Mala
- ⇒ Tutorial : Bueno

- **Diseño de Base de Datos**
 - ⇒ Concepto y Planeación : Malo
 - ⇒ Patrones : Pésimos
 - ⇒ Creación de scripts/macros : Característica no disponible, valor = 0

- **Entrada de datos**
 - ⇒ Verificación de datos : Excelente
 - ⇒ Importación de datos : Buena
 - ⇒ Variedad de tipos de datos : Excelente

- **Alteración de una base de datos existente**
 - ⇒ Edición de datos : Regular
 - ⇒ Edición de scripts/macros : Característica no disponible
 - ⇒ Edición de formas : Regular

- **Creación y ejecución de una consulta**
 - ⇒ Reutilización : Excelente

- **Creación y ejecución de un reporte**
 - ⇒ Reutilización : Excelente

- **Análisis de datos**
 - ⇒ Gráficas : Bueno
 - ⇒ Tabulaciones-cruzadas : Buena
 - ⇒ Soporte para características relacionales : Excelente

- **Administración de formas**

- ⇒ Creación : Regular
- ⇒ Uso : Excelente

4.3.1.2 Claris FileMaker Pro 2.1

- **Características Estándar**

- ⇒ Precio : \$399 Dólares
- ⇒ Registro máximo por archivos de datos : **Limitado a 32MB**
- ⇒ Número de tipos de datos : 7
- ⇒ Capacidades relacionales : Si tiene
- ⇒ Número de tablas que pueden ser ligadas : 16

- **Herramientas de entrada de datos**

- ⇒ Prueba patrón : No
- ⇒ Verificación basado en normas : Si
- ⇒ Fuerza al uso de mayúsculas : No
- ⇒ Campos incrementados automáticamente : Si

- **Formatos de base de datos importados y exportados**

- ⇒ ASCII delimitado por coma : Si
- ⇒ Paradox : No
- ⇒ dBase : Si
- ⇒ 1-2-3 : Si
- ⇒ Excel : No
- ⇒ EL usuario puede añadir y borrar índices : **No**

- **Facilidades de Consulta**

- ⇒ Consulta por forma : Si

- ⇒ Herramienta de ayuda para consulta :No tiene
- ⇒ Salva consultas : Si
- ⇒ Criterio de ordenación máximo : 10
- ⇒ Herramientas de trazado y gráficos : No tiene

- **Formas**

- ⇒ Número de patrones suministrados : 7⁴
- ⇒ Reportes tabulado-transversal : No
- ⇒ Proporciona conexión a bases de datos SQL : No
- ⇒ Lenguaje de Programación : No

- **Servicio y Soporte**

- ⇒ Soporte telefónico diario libre de impuesto : 12 horas
- ⇒ Duración de soporte técnico gratis : 90 días.
- ⇒ BBS : No
- ⇒ Foros en línea :Si
- ⇒ Fax : Si

- **Interfaz/tutorial**

- ⇒ Menús lógicos : Bueno
- ⇒ Controles de Mouse : Excelente
- ⇒ Controles de teclado : Regular
- ⇒ Documentación : Regular
- ⇒ Tutorial : Regular

- **Diseño de Base de Datos**

- ⇒ Concepto y Planeación : Regular
- ⇒ Patrones : Regular
- ⇒ Ayuda de diseño/asistente : Característica no disponible

⇒ Creación de scripts/macros : Regular

• **Entrada de datos**

- ⇒ Verificación de datos : Buena
- ⇒ Importación de datos : Regular
- ⇒ Variedad de tipos de datos : Regular

• **Alteración de una base de datos existente**

- ⇒ Edición de datos : Buena
- ⇒ Edición de scripts/macros : Regular
- ⇒ Edición de formas : Excelente

• **Creación y ejecución de una consulta**

- ⇒ Reutilización : Regular

• **Creación y ejecución de un reporte**

- ⇒ Reutilización : Regular

• **Administración de formas**

- ⇒ Creación : Excelente
- ⇒ Uso : Bueno

• **Análisis de datos**

- ⇒ Gráficas : Característica no disponible
- ⇒ Tabulaciones-cruzadas : Característica no disponible
- ⇒ Soporte para características relacionales : Pésimo

4.3.1.3 Lotus Approach 3.0

- **Características Estándar**
 - ⇒ Precio : \$495 Dólares
 - ⇒ Registro máximo por archivos de datos : Varios, más de 1 billón
 - ⇒ Número de tipos de datos : 8
 - ⇒ Capacidades relacionales : Si tiene
 - ⇒ Número de tablas que pueden ser ligadas : Varias

- **Herramientas de entrada de datos**
 - ⇒ Prueba patrón : Sí
 - ⇒ Verificación basado en normas : Sí
 - ⇒ Fuerza al uso de mayúsculas : Sí
 - ⇒ Campos incrementados automáticamente : Si

- **Formatos de base de datos importados y exportados**
 - ⇒ ASCII delimitado por coma : Sí
 - ⇒ Paradox : Sí
 - ⇒ dBase : Sí
 - ⇒ 1-2-3 : Sí
 - ⇒ Excel : Si
 - ⇒ EL usuario puede añadir y borrar índices : Si

- **Facilidades de Consulta**
 - ⇒ Consulta por forma : Sí
 - ⇒ Consulta por ejemplo : Si
 - ⇒ Herramienta de ayuda para consulta : No tiene
 - ⇒ Salva consultas : Sí
 - ⇒ Criterio de ordenación máximo : 255

⇒ Herramientas de trazado y gráficos : Si tiene

- **Formas**

- ⇒ Número de patrones suministrados : 51⁵
- ⇒ Reportes tabulado-transversal : Si
- ⇒ Proporciona conexión a bases de datos SQL : Si
- ⇒ Lenguaje de Programación : No

- **Servicio y Soporte**

- ⇒ Soporte telefónico diario libre de impuesto : 11.5 horas
- ⇒ Duración de soporte técnico gratis : 90 días.
- ⇒ BBS : Si
- ⇒ Foros en línea :Si
- ⇒ Fax : Si

- **Interfaz/tutorial**

- ⇒ Menús lógicos : Malos
- ⇒ Controles de Mouse : Bueno
- ⇒ Controles de teclado : Regular
- ⇒ Documentación : Mala
- ⇒ Tutorial : Regular

- **Entrada de datos**

- ⇒ Verificación de datos : Buena
- ⇒ Importación de datos : Excelente
- ⇒ Variedad de tipos de datos : Regular

- **Alteración de una base de datos existente**

- ⇒ Edición de datos : Buena

- ⇒ Edición de scripts/macros : Regular
- ⇒ Edición de formas : Buena

- **Diseño de Base de Datos**
 - ⇒ Concepto y Planeación : Regular
 - ⇒ Patrones : Regular
 - ⇒ Ayuda de diseño/asistente : Malo
 - ⇒ Creación de scripts/macros : Regular

- **Creación y ejecución de una consulta**
 - ⇒ Reutilización : Regular

- **Creación y ejecución de un reporte**
 - ⇒ Reutilización : Buena

- **Administración de formas**
 - ⇒ Creación : Excelente
 - ⇒ Uso : Bueno

- **Análisis de datos**
 - ⇒ Gráficas : Buena
 - ⇒ Tabulaciones-cruzadas : Buena
 - ⇒ Soporte para características relacionales : Buena

4.3.1.4 Microsoft Access

- **Características Estándar**
 - ⇒ Precio : \$495 Dólares
 - ⇒ Registro máximo por archivos de datos : Limitado al espacio en disco

- ⇒ Número de tipos de datos : 8
- ⇒ Capacidades relacionales : Si tiene
- ⇒ Número de tablas que pueden ser ligadas : 32

- **Herramientas de entrada de datos**

- ⇒ Prueba patrón : Si
- ⇒ Verificación basado en normas : Si
- ⇒ Fuerza al uso de mayúsculas : Si
- ⇒ Campos incrementados automáticamente : Si

- **Formatos de base de datos importados y exportados**

- ⇒ ASCII delimitado por coma : Si
- ⇒ Paradox : Si
- ⇒ dBase : Si
- ⇒ 1-2-3 : Si
- ⇒ Excel : Si
- ⇒ EL usuario puede añadir y borrar índices : Si

- **Facilidades de Consulta**

- ⇒ Consulta por forma : No
- ⇒ Consulta por ejemplo : Si
- ⇒ Herramienta de ayuda para consulta : Si tiene
- ⇒ Salva consultas : Si
- ⇒ Criterio de ordenación máximo : 255
- ⇒ Herramientas de trazado y gráficos : Si tiene

- **Formas**

- ⇒ Número de patrones suministrados : 45
- ⇒ Reportes tabulado-transversal : Si

- ⇒ Proporciona conexión a bases de datos SQL : Si
- ⇒ Lenguaje de Programación : Si

- **Servicio y Soporte**
 - ⇒ Soporte telefónico diario libre de impuesto : 12 horas
 - ⇒ Duración de soporte técnico gratis : Ilimitado
 - ⇒ BBS : Si
 - ⇒ Foros en línea : Si
 - ⇒ Fax : Sí

- **Interfaz/tutorial**
 - ⇒ Menús lógicos : Regular
 - ⇒ Controles de Mouse : Regular
 - ⇒ Controles de teclado : Regular
 - ⇒ Documentación : Mala
 - ⇒ Tutorial : Buena

- **Entrada de datos**
 - ⇒ Verificación de datos : Excelente
 - ⇒ Importación de datos : Regular
 - ⇒ Variedad de tipos de datos : Regular

- **Alteración de una base de datos existente**
 - ⇒ Edición de datos : Buena
 - ⇒ Edición de scripts/macros : Buena
 - ⇒ Edición de formas : Regular

- **Creación y ejecución de una consulta**
 - ⇒ Reutilización : Excelente

- **Creación y ejecución de un reporte**
 - ⇒ Reutilización : Buena

- **Diseño de Base de Datos**
 - ⇒ Concepto y Planeación : Mala
 - ⇒ Patrones : Regular
 - ⇒ Ayuda de diseño/asistente : Regular
 - ⇒ Creación de scripts/macros : Regular

- **Administración de formas**
 - ⇒ Creación : Regular
 - ⇒ Uso : Bueno

- **Análisis de datos**
 - ⇒ Gráficas : Regular
 - ⇒ Tabulaciones-cruzadas : Buena
 - ⇒ Soporte para características relacionales : Buena

4.3.1.5 Symantec Q&A 4.0 for Windows

- **Herramientas de entrada de datos**
 - ⇒ Prueba patrón : Si
 - ⇒ Verificación basado en normas : Si
 - ⇒ Fuerza al uso de mayúsculas : Si
 - ⇒ Campos incrementados automáticamente : Si

- **Formatos de base de datos importados y exportados**
 - ⇒ ASCII delimitado por coma : Si

- ⇒ Paradox : Sí
- ⇒ dBase : Sí
- ⇒ 1-2-3 : Sí
- ⇒ Excel : Sí
- ⇒ EL usuario puede añadir y borrar índices : No

- **Características Estándar**

- ⇒ Precio : \$249.95 Dólares
- ⇒ Registro máximo por archivos de datos : 16 millones
- ⇒ Número de tipos de datos : 7
- ⇒ Capacidades relacionales : No tiene
- ⇒ Número de tablas que pueden ser ligadas : 1

- **Facilidades de Consulta**

- ⇒ Consulta por forma : Sí
- ⇒ Consulta por ejemplo : Sí
- ⇒ Herramienta de ayuda para consulta : Sí tiene
- ⇒ Salva consultas : Sí
- ⇒ Criterio de ordenación máximo : 50
- ⇒ Herramientas de trazado y gráficos : No tiene

- **Formas**

- ⇒ Número de patrones suministrados : 7⁴
- ⇒ Reportes tabulado-transversal : No
- ⇒ Proporciona conexión a bases de datos SQL : No
- ⇒ Lenguaje de Programación : Sí

- **Servicio y Soporte**

- ⇒ Soporte telefónico diario libre de impuesto : 9 horas

- ⇒ Duración de soporte técnico gratis : 90 días
- ⇒ BBS : Sí
- ⇒ Foros en línea : Sí
- ⇒ Fax : Sí

- **Interfaz/tutorial**

- ⇒ Menús lógicos : Regular
- ⇒ Controles de Mouse : Bueno
- ⇒ Controles de teclado : Regular
- ⇒ Documentación : Regular
- ⇒ Tutorial : Buena

- **Diseño de Base de Datos**

- ⇒ Concepto y Planeación : Malo
- ⇒ Patrones : Pésimo
- ⇒ Ayuda de diseño/asistente : Malo
- ⇒ Creación de scripts/macros : Regular

- **Entrada de datos**

- ⇒ Verificación de datos : Buena
- ⇒ Importación de datos : Buena
- ⇒ Variedad de tipos de datos : Regular

- **Creación y ejecución de una consulta**

- ⇒ Reutilización : Excelente

- **Creación y ejecución de un reporte**

- ⇒ Reutilización : Excelente

- **Alteración de una base de datos existente**
 - ⇒ Edición de datos : Regular
 - ⇒ Edición de scripts/macros : Buena
 - ⇒ Edición de formas : Buena

- **Administración de formas**
 - ⇒ Creación : Regular
 - ⇒ Uso : Regular

- **Análisis de datos**
 - ⇒ Gráficas : Característica no disponible
 - ⇒ Tabulaciones-cruzadas : Pésimo
 - ⇒ Soporte para características relacionales : Pésimo

4.3.2 Selección de la base de datos.

De acuerdo a las características expuestas de las diferentes bases de datos, se vio que Borland Paradox 5.0 tiene la capacidad de ser una base de datos relacional y permite la importación y exportación de datos desde otras tipos de bases y de archivos de tipo texto, pero no cuenta con una ayuda en línea y menús lógicos y tiene una documentación mala que no la hace sencilla de manejar y aprender en un corto tiempo.

En cuanto a Claris File-Maker Pro 2.1 es una base de datos relacional que posee buenas características de diseño y verificación de datos, pero en cuanto a su capacidad de exportación e importación de datos no permite la utilización de Excel, es decir, que es mala en el soporte de OLE (Object Linking and Embedding) "Ligado e Inclusión de Objetos" y en DDE (Dynamic Data Exchange) "Intercambio Dinámico de Datos", por lo que File-Maker no es lo suficientemente flexible para el caso en que se requiriera usar alguna de estas características, pues el proyecto pretende mostrar gráficas estadísticas que permitan en un momento dado analizar el comportamiento y funcionamiento del automóvil.

Lotus Approach 3.0 es una base de datos relacional que posee verificación de datos e incremento automático de campos además de una excelente importación y exportación de datos de varios formatos de bases de datos, sin embargo tiene una lógica de menús y documentación mala, así como no cuenta con una ayuda en línea. Symantec Q&A 4.0 for Windows no posee características relacionales, aunque sus herramientas de entrada de datos son buenas, así como su importación y exportación de los mismos; en cuanto al diseño de bases de datos con lo que respecta a su concepto y planeación es mala y en el ligado de bases y el soporte de OLE y DDE también lo es.

Microsoft Access, es una base de datos relacional que tiene buenas herramientas de entrada de datos, pues verifica e incrementa automáticamente los campos, así como facilidades de consulta de información y de contar con un buen tutorial y ayuda en línea que agiliza su uso y aprendizaje. Su ligado de bases de datos es bueno así como un excelente soporte de OLE/DDE y en la integridad de datos y archivos.

Por lo que se selecciona a Microsoft Access como la base de datos a utilizar, pues otra característica que nos inclina a elegirla es ser la base nativa de Visual Basic, herramienta que anteriormente fue analizada y seleccionada como aquella que va a establecer la interfaz con el usuario.

Access cuenta además de diferentes Asistentes (Bases de datos, búsquedas, importación y exportación y consultas sencillas), son sólo algunos de los asistentes y generadores que posee la nueva versión de MS-Access, y con ellos sólo necesita saber lo que desea hacer con sus datos, pues realizar casi toda la creación y mantenimiento de objetos del sistema queda en manos de estas herramientas.

Access sacrifica velocidad por seguridad, pues no le permite modificar la estructura de una tabla si una forma usa esa tabla.

Access posee una herramienta llamada "macros" que le permite automatizar tareas dentro del ambiente de una base de datos, y además son la columna vertebral sobre la cual descansan los "módulos", que son programas escritos en "Visual Basic edición para Aplicaciones", y éste es el lenguaje de programación sobre el que Microsoft fundamenta todas sus herramientas de oficina.

Características	Borland Paradox 5.0	Claris File- Maker	Lotus Approach	Symantec Q&A 4.0	MS Access 2.0
Precio	\$495 dls.	\$399 dls.	\$495	\$249.95	\$495
Capacidad Relacional	■	■	■	□	■
Incremento automático de campos	■	■	■	■	■
Importado y Exportado de diferentes formatos	■	regular	■	■	■
Facilidades de Consulta	■	regular	■	■	■
Lenguaje de Programación	■	□	□	■	■
Ayuda en línea (aprendizaje)	□	□	□	□	■
Verificación de datos	■	■	■	■	■
Ligado de Bases	■	■	■	■	■
Soporte de OLE/DDE	■	malo	■	malo	■

Nota: ■ = Excelente, ■ = Bueno, □ = No tiene

Tabla 4.3.2.a Comparación de diferentes manejadores de bases de datos.



Figura 4.3.2.b Pantalla principal de Microsoft Access 2.0

4.3.2.1 Características y funcionamiento del manejador de Base de datos seleccionado.



Figura 4 3 2 1 Microsoft Access con la base de datos "Compviaj"

Microsoft Access, es un sistema de administración de bases de datos relacionales para Microsoft Windows. El diseño de Microsoft Access está orientado a ofrecer una insuperable potencia de acceso a los datos, que se combina con la extrema facilidad de uso que permite Windows. Access cuenta con sencillas herramientas, que proporcionan la potencia de una base de datos de características completas.

Para poder sacar el máximo provecho de las herramientas con que cuenta Access, resulta fundamental saber diseñar correctamente una base de datos. La clave para comprender el funcionamiento de Access estriba en tener bases sólidas acerca de la organización de las bases de

datos.

4.3.2.1.1 Diseño de una base de datos

Para poder obtener el máximo de provecho de las herramientas con que cuenta Access, es importante saber diseñar correctamente una base de datos.

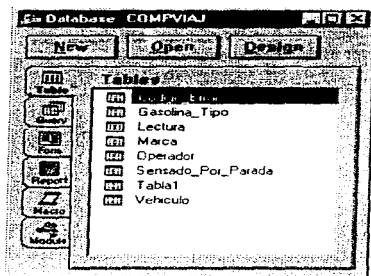


Figura 4.3.2.1.1 Ventana de Diseño de Bases de Datos

Los pasos para diseñar una base de datos son :

- Determinar el propósito de la base de datos
- Determinar las tablas
- Determinar los campos
- Determinar las relaciones entre tablas
- Depurar el diseño

El poder de manejo de una base de datos relacional como Access incide en su rápida búsqueda y extracción de información almacenada en tablas separadas. Para que Access trabaje eficientemente, cada tabla en la base de datos debe incluir un campo o conjunto de campos que

identifiquen únicamente cada renglón o registro almacenado en la tabla.

4.3.2.1.2 Características de las Tablas

Antes de crear cualquier otro objeto de una base de datos como lo son consultas, formas o reportes, se deben crear primero las tablas.

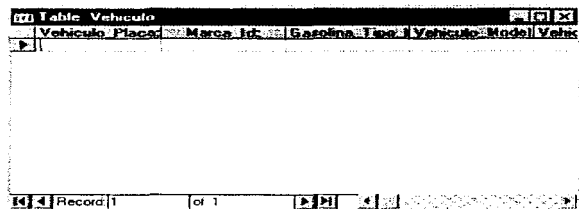


Figura 4.3.2.1.2.a Creación de una tabla.

Una tabla es una colección de datos acerca de un tema en particular. Los datos en una tabla son presentados en un formato matricial con columnas llamadas campos y renglones llamados registros.

En una tabla, un campo es una categoría de información, esto puede ser los nombres de varias categorías de productos, descripción de la categoría, en tanto que un registro es una colección de información acerca de una persona, producto, etc.

Con Microsoft Access, se puede usar la ventana tabla para crear y ver las tablas. Puede abrir la ventana tabla en cualquiera de las presentaciones existentes: Diseño y Hoja de datos.

Field Name	Data Type
Employee ID	Counter
Last Name	Text
First Name	Text
Title	Text
Birth Date	Date/Time
Hire Date	Date/Time
Address	Text
City	Text
Region	Text
Postal Code	Text
Country	Text
Home Phone	Text
Extension	Text
Photo	OLE Object
Notes	Memo
Reports To	Number

4.3.2.1.2.b Campos de una tabla

En el modo de presentación diseño de una tabla siempre aparece una ventana, donde la parte superior se utiliza para la declaración de campos y la parte inferior para especificar las

Field Name	Data Type	Description
ProductID	Counter	
ProductName	Text	
UnitPrice	Currency	

Field Properties	
Field Size	50
Format	
Input Mask	
Caption	Product Name
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	Yes (Duplicate: OK)

A field name can be up to 64 characters long, including spaces. Press F3 for help on field names.

4.3.2.1.2.c Declaración de campos.

propiedades correspondientes a cada campo. La declaración de los campos consiste en determinar el nombre y el tipo de datos correspondiente. Cuando se declaran campos es necesario especificar :

- Nombre del campo
- Tipo de datos
- Descripción (es opcional)

Después de teclear el nombre del campo, se debe escoger el tipo de dato que tendrá el campo.

Los tipos de Datos que maneja Access son :

Tipo de dato	Almacenamiento	Tamaño
Texto	Caracteres alfanuméricos	1 byte por carácter. Acepta hasta 255
Memo	Caracteres alfanuméricos (usualmente grandes párrafos)	Acepta hasta 64,000 bytes.
Numerico	Valores numéricos (enteros o valores fraccionarios)	1, 2, 4 u 8 bytes
Fecha/Hora	Fechas y horarios	8 bytes
Moneda	Valores monetarios	4 bytes
Contador	Valor que aumenta progresivamente	4 bytes
Si/No	Valores Booleanos	1 bit
Objeto OLE	Objetos OLE, gráfico u otra información binaria	Acepta hasta 1 GB (limitado por espacio en disco duro)

Las propiedades de un Campo, cuando se abre una tabla en la parte del diseño, Access despliega las propiedades del campo que se está abriendo en la parte inferior de la ventana. Las propiedades se especifican una por una para cada campo.

Propiedades para los diferentes tipos de datos :

- Tamaño de Campo
- Máscara de entrada
- Título
- Regla de validación
- Requerido
- Indexado
- Formato
- Lugares decimales
- Valor predeterminado
- Texto de validación
- Permitir longitud cero

Estableciendo las propiedades de los campos en el modo de presentación, diseño de una tabla, es posible controlar la apariencia de los datos, evitar su incorrecta introducción, especificar valores predeterminados y acelerar la búsqueda y la ordenación en la tabla.

Por ejemplo puede dar formato a los números para facilitar su lectura o establecer reglas que los datos deben cumplir para que sea posible introducirlos en un campo.

Microsoft Access, cuenta con el asistente para tablas, en donde puede elegir entre docenas de ejemplos : tablas de ejemplos y cientos de campos prácticos así como dejar que el asistente cree las tablas automáticamente.

Microsoft Access, puede filtrar los registros de una tabla para ver únicamente los registros deseados. Además puede presentar y editar simultáneamente todos los índices de una tabla, en la ventana índices.

4.3.2.1.3 Características de las relaciones entre tablas

Cuando se crea una tabla, ésta no está relacionada con las demás tablas de la Base de datos. Para crear una relación, se debe abrir la ventana de relaciones y añadir las tablas o consultas que se deseen relacionar. Microsoft Access, cuenta con una ventana gráfica llamada Relaciones, que

permite realizar de forma visual un boceto del diseño de la base de datos y observar simultáneamente todas las relaciones de la base de datos.

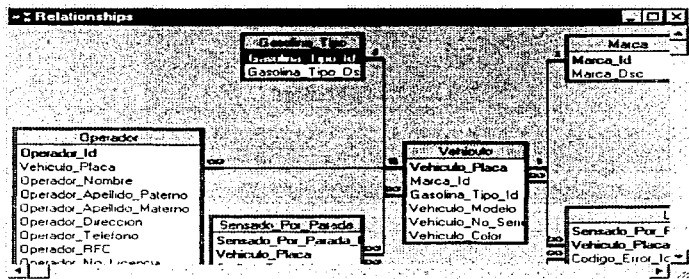


Figura 4.3.2.1.3 Ejemplo de Entidad-Relación

Gracias a las actualizaciones y eliminaciones en cascada, ahora puede actualizar un campo de una tabla y hacer que Microsoft Access actualice automáticamente los campos relacionados de las tablas relacionadas.

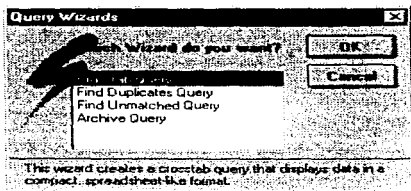
También puede eliminar un registro y hacer que Microsoft Access elimine automáticamente los registros de las tablas relacionadas.

4.3.2.1.4 Características en consultas

Una consulta es la respuesta a una serie de preguntas acerca de datos almacenados en tablas o incluso en consultas. La forma en que diseñe la consulta, indicará con exactitud a Microsoft Access que datos debe recuperar.

Microsoft Access, cuenta con el asistente para consultas que le ayudará a crear consultas complejas para tareas frecuentes de administración de datos, tales como ver una presentación en

tabla de referencias cruzadas de sus datos, buscar todos los registros duplicados de una tabla o consulta, o buscar todos los registros que sean distintos de los registros de otra tabla.



4.3.2.1.4 Asistente para la creación de consultas

Aunque no se hayan establecido relaciones entre tablas, Microsoft Access crea automáticamente combinaciones en una consulta entre tablas que contienen campos coincidentes.

Cuando una consulta incluye campos de varias tablas relacionadas, se pueden actualizar los datos a ambos lados de la combinación. Puede definir una consulta de selección para limitar el número de registros devueltos.

4.3.2.1.5 Modos de presentación de las consultas

Utilice el modo de presentación Diseño para crear una nueva consulta o modificar el diseño de una ya existente. En el modo presentación Diseño podrá emplear herramientas gráficas para la creación de consultas. Utilice el modo de presentación de Hoja de Datos para mostrar los datos recuperados por la consulta. Utilice el modo de presentación SQL cuando desee crear o modificar una consulta mediante inscripciones SQL. Microsoft Access dentro de la ventana de presentación SQL puede introducir cualquier instrucción del SQL. Mientras edita la instrucción del SQL también dispone de acceso a los comandos del menú y a otras ventanas de Microsoft Access.

4.3.2.1.6 Tipos de consultas

Las consultas sirven para :

- Hacer cambios a información contenida en tablas
- Como recurso para elaborar formas, reportes o incluso otras consultas
- Obtener información de diferentes tablas, que a su vez están relacionadas
- Realizar cálculos

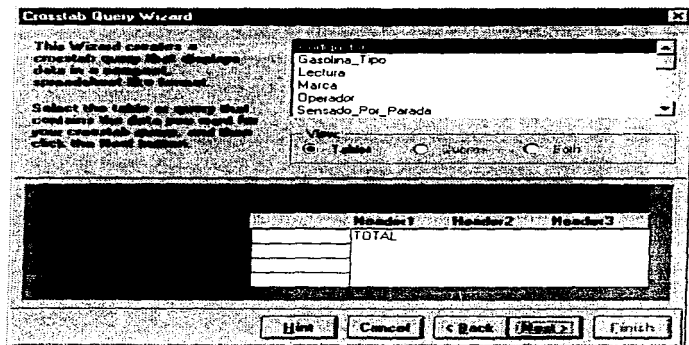


Figura 4.3.2.1.6 Ejemplo de la creación de una consulta

- **Selección**

Son los más comunes, aquí se plantean preguntas sobre datos almacenados en las tablas y la respuesta es una hoja de datos, en donde se puede ver y cambiar los datos de las tablas subyacentes.

- **Referencias Cruzadas**

Presenta los datos con títulos en las filas y columnas, como en una hoja de cálculo. Con una

consulta de tabla de referencias cruzadas se puede reducir gran cantidad de información en un formato de fácil lectura.

- **Consultas de Unión**

Combina campos coincidentes de dos o más tablas.

- **Consultas de Definición de Datos**

Crea, modifica o elimina tablas de una base de datos de Microsoft Access utilizando instrucciones de SQL.

- **Acción**

Modifican muchos registros en una sola operación. Se utiliza para crear una nueva tabla, elimina registros de una tabla, añadir nuevos registros a una tabla o modificarlos. A partir de la selección actualizan, agregan y eliminan datos, Access provee cuatro tipos de consultas de este tipo que son:

⇒ Consultas de Creación de Tablas

⇒ Consultas de Eliminación

⇒ Consulta de Datos Añadidos

⇒ Consulta de Actualización

¿Por qué se utilizan consultas?

Microsoft Access ofrece una gran flexibilidad, a la hora de diseñar sus consultas podrá:

- **Elegir campos.** No es necesario incluir en la consulta todos los campos de una tabla.
- **Elegir registros.** Puede especificar los criterios que deben cumplir los registros para que se incluyan en la hoja de respuesta dinámica de la consulta.
- **Ordenar registros.** Puede ver los registros en un orden determinado. Para mostrar los datos

ordenados en un formulario, puede crear una consulta que ordene los registros y a continuación utilizarla como origen de los datos para el formulario.

- **Formular preguntas sobre datos de varias tablas.** Puede utilizar una consulta para responder a una pregunta sobre datos procedentes de más de una tabla y ver los resultados en una sola hoja de datos.
- **Realizar cálculos.** Puede crear nuevos campos llamados campos calculados, que contengan el resultado de un cálculo. Para mostrar campos calculados en formularios o informes, puede crear una consulta que contenga el campo calculado y basar en ella el formulario o informe.
- **Usar una consulta como origen de los datos para formularios, informes y otras consultas.**

Para seleccionar solo los datos que deben aparecer en un formulario o informe, puede crear una consulta de selección y utilizarla como origen de los datos del formulario o informe. Utilizando la consulta para incluir datos de varias tablas y establecer criterios para mostrar solo un conjunto limitado de datos. Cada vez que abra el formulario o imprima el informe, la consulta recuperará información actualizada de las tablas. También puede introducir nuevos datos o modificar los existentes directamente en la hoja de datos de la consulta, o bien en un formulario que este basado en esta consulta.

- **Modificar los datos de las tablas.** Con las consultas de acciones, puede actualizar, eliminar o añadir un grupo de registros simultáneamente. También puede utilizarlas para crear una nueva tabla que contenga registros de otra tabla o de un grupo de tablas existentes.

Saber como especificar criterios en una consulta es lo más importante para diseñarla, únicamente con ellos podrá extraer exactamente la información que requiere.

Los criterios indican a Access la información que presentará en la hoja de respuesta dinámica.

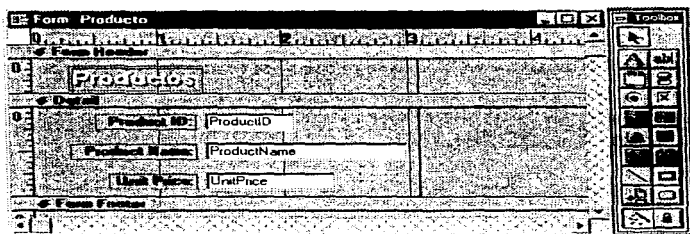
Entre los criterios que pueden incluirse en una consulta están :

- Selección de un rango de registros
- Patrón de caracteres
- Registros que no tienen un valor asociado
- Registros que tengan múltiples criterios
- Registros que contengan o no valores en determinados campos
- Registros para una fecha en específico
- Registros basados en cálculos
- Criterios para grupos de campos
- Criterios para todos los campos
- Desplegar parte de un campo texto
- Desplegar parte de un campo fecha
- Selección de valores únicos

4.3.2.1.7 Creación de formularios

El corazón de toda aplicación para una base de datos es su interfaz, definida por sus formularios de pantalla. Mediante el uso de Wizard de Formularios, es bastante simple lograr un conjunto de formularios básicos partiendo de los cuatro tipos disponibles. Una columna, Tabulado, Gráfico y Principal/Subformulario y asignarle uno de los cinco estilos posibles. Además, existen un nuevo formulario Autoform que crea los mejores formularios posibles basándose en la tabla o la consulta seleccionada. También es posible generar formularios que usan las capacidades de macro completa que ofrece Access para organizar su aplicación y procesamiento de control, eliminando gran parte de la codificación que normalmente se requiere para tal fin.

Una forma es una herramienta de Access que permite introducir, cambiar, ver o imprimir información . Ver figura 4.3.2.1.7.a



4.3.2.1.7.a Presenta el Diseño de un Formulario de Access

Una forma ayuda a ver fácilmente información, en tanto se ahorra tiempo y previene errores al introducir datos. Toda la información en una forma está contenida en controles, objetos que despliegan la información, optimizan acciones o decoran la forma.

Con Access podrá diseñar formularios fáciles de utilizar y que presenten la información de la forma deseada podrá utilizar diversos elementos de diseño (texto, datos, imágenes, líneas y color para crear exactamente el formulario deseado, se debe de elegir los elementos que desea utilizar y determina la forma que desea disponerlos en el formulario.

Los tipos de formas con que cuenta Access son :

- **Columna simple.** Despliega los valores de un registro en una columna, además de que cada valor es separado por una línea. Muestra solo un registro.
- **Tabular.** Muestra todos los valores de un registro en un renglón, haciendo así un formato de renglones y columnas. Especifica varios registros a la vez.
- **Principal/subforma.** Una subforma dentro de una forma enseña la relación uno a muchos entre la información de la forma principal y la información de la subforma. La forma principal

exhibe la información en un formato de hoja de datos.

- **Gráfico.** Una forma gráfica muestra información con un formato de gráfica, ya sea de barras, de columna o de pastel.

Los modos de presentación de una Forma son :

- **Diseño.** Facilita la creación o modificación del diseño de una forma.
- **Formulario.** Permite introducir, cambiar y visualizar la información. En este modo de presentación, usualmente se ven todos los campos de un registro a la vez.
- **Hoja de datos.** Donde introduce, cambia y visualiza la información. En este modo de presentación se pueden ver muchos registros al mismo tiempo, no así todos los campos de un registro sin que haga el scroll, además no despliega imágenes, objetos OLE o botones como lo muestra el modo de presentación formulario.
- **Presentación Preliminar.** Indica el aspecto que tendrá una forma al ser impresa.

4.3.2.1.8 Características de los informes

Un informe es un conjunto de información que usted organiza según criterios y al que aplica formato de acuerdo con sus especificaciones.

Un informe constituye una forma de recuperar y presentar los datos como información significativa que se puede usar y distribuir.

Con Access podrá diseñar informes que presenten la información de la forma deseada. Podrá utilizar numerosos elementos de diseño (texto, datos, líneas, cuadros y gráficos) para crear exactamente el informe que necesitará, es necesario elegir los elementos que desea utilizar y

determinar la forma en que desea disponer en el informe.

Crear un informe es un método muy eficaz de presentar los datos en forma de documento impreso. Si bien es posible imprimir los formularios y las hojas de datos, los informes brindan mayor flexibilidad a la hora de presentar información de resumen.

En Access podrá crear informes para :

- Organizar y presentar los datos en grupo
- Calcular totales y parciales, totales de grupo, sumas de totales y porcentaje de totales
- Incluir subformularios, subinformes y gráficos
- Presentar los datos en un formato atractivo, con imágenes, líneas y fuentes especiales

4.3.2.1.9 Controles

Access ofrece una variedad de controles que se pueden agregar a la forma o informe. Para la creación de controles se utiliza la caja de herramientas, la cual se despliega automáticamente cuando se abre una forma o informe en el modo de presentación diseño.

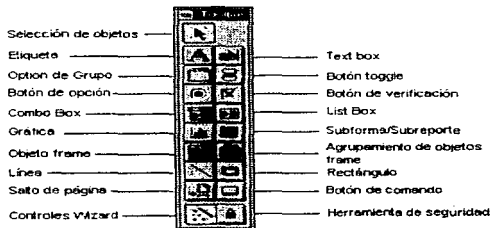


Figura 4.3.2.1.9 Caja de Herramientas

Los controles que proporciona Access son dependientes, pues la información del control proviene

de tablas de consultas. Se usan para desplegar, introducir y actualizar valores de los campos de una base de datos. Los valores pueden ser texto, número, imágenes, gráficas, etcétera. El text box (caja de texto) es el control dependiente más común.

4.3.2.2.0 Características de las macros

Una macro realiza automáticamente una tarea o una serie de tareas. Cada tarea que usted desee realizar con Access se denomina acción.

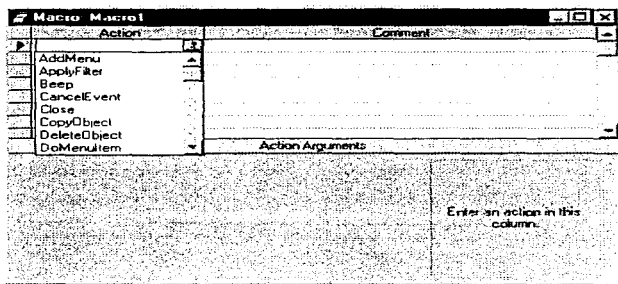


Figura 4.3.2.2.0 Ventana para la creación de macros.

Access ofrece una lista de acciones, entre las que se selecciona una o más para crear una macro. Cuando se ejecute la macro, Microsoft Access realizará las acciones en el mismo orden en que figuran dentro de la macro, utilizando los objetos o los datos especificados. Mediante uso de macros puede conseguir que los formularios, los informes y los demás objetos de la base de datos funcionen conjuntamente de forma aun más inteligente.

Las macros también son útiles para automatizar tareas rutinarias o repetitivas, como la impresión de informes semanales. Automatizando sus tareas rutinarias conseguirá mayor eficacia y precisión

en la base de datos, ya que una macro siempre realizará la tarea de la misma forma.

4.3.2.2.1 Características de los módulos

Los módulos almacenan código de Access Basic que pueden escribirse para personalizar, mejorar y extender las posibilidades de la base de datos.

El contenido de la información aquí presentada, sólo fue un panorama general de todo lo que posee Microsoft Access, para obtener mayor información, consultar el manual de Usuario de Microsoft Access, que constituye una completa guía para crear y trabajar con una base de datos de Access. Además del manual de Usuario, existe la ayuda en pantalla, que proporciona información de referencia e instrucciones prácticas para todas las tareas que pueda realizar Access.

Además existen las Fichas-Guía que son instrucciones en pantalla que proporcionan instrucciones paso a paso para ayudarle a conocer Microsoft Access mientras crea y utiliza su propia base de datos.

4.4 Características y Selección de la Herramienta Visual.

Los lenguajes son sistemas de comunicación. Un lenguaje de programación incluye todos los símbolos, caracteres y reglas de uso que permiten la comunicación del usuario con la máquina. Estos son creados para una aplicación especial (tal como controlar un robot), mientras que otras son herramientas de uso general, más flexibles y apropiadas para muchos tipos de aplicaciones. En cualquier caso, todos deben tener instrucciones que pertenecen a las categorías de entrada/salida, cálculo/manipulación de textos, lógicas/comparación y almacenamiento/recuperación.

4.4.1 Características.

4.4.1.1 Lenguaje C.

El mundo de la computación ha sufrido una revolución desde la publicación, en 1978, de el lenguaje de programación C. Las grandes computadoras son ahora mucho más grandes, y las computadoras personales tienen capacidades que rivalizan con los mainframes de hace una década. También el lenguaje C ha cambiado en ese tiempo, aunque sólo en forma modesta, y se ha extendido más allá de lo que fueron sus orígenes como el lenguaje del sistema operativo UNIX.

La creciente popularidad de C, los cambios en el lenguaje a lo largo de los años, y la creación de compiladores por grupos no involucrados en su diseño, se combinaron para demostrar la necesidad de una definición del lenguaje más precisa y contemporánea. En 1983, el American National Standards Institute (ANSI) (Instituto Nacional de Estándares Americanos) estableció un comité cuyos propósitos eran producir "una definición no ambigua del lenguaje C e, independiente de la máquina", cuidando la conservación de su espíritu. El resultado es el estándar ANSI para el lenguaje C.

C es un lenguaje de programación de propósito general que ha sido estrechamente asociado con el sistema UNIX en donde fue desarrollado puesto que tanto el sistema como los programas que corren en él están escritos en lenguaje C. Sin embargo, este lenguaje no está ligado a ningún

sistema operativo ni a ninguna máquina, y aunque se le llama "lenguaje de programación de sistemas" debido a su utilidad para escribir compiladores y sistemas operativos, se utiliza con igual eficacia para escribir importantes programas en diversas disciplinas.

Muchas de las ideas importantes de C provienen del lenguaje BCPL, desarrollado por Martin Richards. La influencia de BCPL sobre C se continuó indirectamente a través del lenguaje B, el cual fue escrito por Ken Thompson en 1970 para el primer sistema UNIX de la DEC PDP-7.

BCPL y B son lenguajes "carentes de tipos". En contraste, C proporciona una variedad de tipos de datos. Los tipos fundamentales son caracteres, enteros y números de punto flotante de varios tamaños. Además, existe una jerarquía de tipos de datos derivados, creados con apuntadores, arreglos, estructuras y uniones. Las expresiones se forman a partir de operadores y operandos; cualquier expresión, incluyendo una asignación o una llamada a función, puede ser una proposición. Los apuntadores proporcionan una aritmética de direcciones independiente de la máquina.

Otra característica significativa del estándar es la definición de una biblioteca que acompañe a C. Esta especifica funciones para tener acceso al sistema operativo (por ejemplo, leer de archivos y escribir en ellos), entrada y salida con formato, asignación de memoria, manipulación de cadenas y otras actividades semejantes. Una colección de **headers** (encabezadores) estándar proporcionan un acceso uniforme a las declaraciones de funciones y tipos de datos. Los programas que utilizan esta biblioteca para interactuar con un sistema anfitrión están asegurados de un comportamiento compatible. La mayor parte de la biblioteca está estrechamente modelada con base en la "biblioteca E/S estándar" del sistema UNIX.

Aunque C coincide con las capacidades de muchas computadoras, es independiente de cualquier arquitectura. Con un poco de cuidado es fácil escribir programas portátiles, esto es, programas que puedan correr sin cambios en una variedad de máquinas. El estándar explica los problemas de la transportabilidad, y prescribe un conjunto de constantes que caracterizan a la máquina en la que se ejecuta el programa.

4.4.1.2 Visual C++.

Como sucede en otros lenguajes, el C++ está compuesto por declaraciones y enunciados en los que se especifica las instrucciones exactas que deben seguirse cuando se ejecute el programa.

El C++ fue creado por Bjarne Stroustrup en los laboratorios Bell. Se pretende que C++ sea el sucesor del popular lenguaje C y que lo supere principalmente mediante el añadido de extensiones de lenguaje orientadas a objetos. Un lenguaje orientado a objetos representa los atributos y operaciones de los objetos.

Además, el C++ proporciona varias mejoras al C que no está orientado a objetos. Por lo tanto, el aprendizaje del C++ le ofrece la ventaja de familiarizarse con el C. A diferencia del C, que ya ha sido estandarizado, el C++ se encuentra todavía en el proceso de estandarización.

La programación en C++ requiere que usted tenga en mente las bibliotecas de soporte que ejecutan varias tareas, como la entrada, la salida, el manejo de texto, las operaciones matemáticas, la E/S (entrada/salida) de archivos, etc. En lenguajes como el BASIC, el soporte para este tipo de operaciones se manifiesta la trasluz de los programas debido a que se dispone de él automáticamente. En consecuencia, muchos programas se presentan como componentes individuales que son independientes de cualquier otro componente de programación. Por el contrario, la programación en C++ le hace más consciente de la dependencia de diversas bibliotecas. La ventaja de esta característica del lenguaje es que se puede seleccionar entre bibliotecas similares, incluidas las que uno mismo desarrolle. Por lo tanto, los programas de C++, son modulares. Los compiladores de C++, entre los que se puede contar al Visual C++, emplean archivos de proyecto y archivos de programa. El banco de trabajo del Visual C++ se vale de archivos de proyectos para administrar la creación y actualización de un programa.

Los archivos de proyecto especifican la biblioteca. Los archivos de programa crean una aplicación.

El banco de trabajo del Visual C++ es la interfaz visual del compilador, el enlazador y el depurador, así como de otras herramientas del C++ utilizadas para crear, administrar y mantener programas en C++. Se puede cargar el banco de trabajo con sólo hacer clic en el icono de Visual C++ o doble clic en el programa MSVC20.EXE desde el File Manager (administrador de archivos).

El banco de trabajo de Visual C++ no utiliza interruptores u opciones de línea de comando para afinar su carga y operación. Se vale en cambio de una de sus opciones de menú.

El banco de trabajo de Visual C++ es el ambiente empleado para desarrollar aplicaciones de C y C++. El banco de trabajo es una herramienta versátil que facilita tanto la creación de aplicaciones para MS-DOS y Windows como la de bibliotecas. El banco de trabajo del Visual C++ tiene opciones de menú, una barra de herramientas, una barra de estado y un área de despliegue. El editor integrado se vale de esta área para desplegar y editar archivos en ventanas MDI (interfaz de documentos múltiples).

Una de las valiosas características del banco de trabajo es su amplia ayuda en línea. Con la opción Help, se puede hacer consultas sobre casi todos los aspectos del banco de trabajo y del lenguaje C, así como acerca de otros aspectos del desarrollo de programas. El banco de trabajo del Visual C++ contiene un menú con nueve opciones principales. Cada opción contiene una familia de selecciones que ejecutan alguna tarea o ajustan alguna opción.

Las nuevas herramientas visuales, App Wizard, Class Wizard y el editor de recursos App Studio, de diseño excelente, están disponibles desde dentro del Visual Workbench (VWB) (Banco de trabajo de Visual). El editor incorporado y el trazador de errores los tomaron prestados del Quick C para Windows. El trazador de errores integrado es exclusivamente para programas de Windows, pero desde el VWB se pueden lanzar versiones de Code View (Vista de código), para DOS o para Windows.

Una opción bienvenida de VWB es la compilación en segundo plano. Los tiempos de

compilación, aunque se han mejorado, todavía son relativamente lentos, pero al menos puede hacer otra cosa mientras espera.

4.4.1.3 Visual Basic.

Es una herramienta gráfica que permite crear de una manera fácil y rápida aplicaciones para el ambiente Windows. La programación en Visual Basic le permite explotar completamente la interfaz gráfica de usuario (GUI) y además tener una mayor producción en la generación de sus aplicaciones por el tipo de comandos implementados.

Las características más importantes de Visual Basic son las siguientes :

- **Diseño de Aplicaciones.** Se basa en la creación de formas y controles por medio de su ambiente de programación amigable. Cada forma tiene asociado un segmento de código que establece su funcionalidad ; esto es, el script (guión) de la forma.
- **Programación orientada a objetos.** Visual Basic permite hacer uso de objetos predeterminados mediante las propiedades y métodos asociados a cada objeto pero no permite la creación de nuevos objetos por lo que no puede ser catalogado completamente como un lenguaje orientado a objetos.
- **Limitantes.** En comparación con la programación en C y C++ para Windows. Visual Basic está limitado en velocidad y en el conjunto de funciones propias. Sin embargo, cubre estas limitantes otorgando la flexibilidad de acceder las APIs (**Application Programming Interface**) "Interfaz de Programación de Aplicaciones", de Windows. Controles externos desarrollados por terceros y librerías dinámicas que permiten aumentar el rango de funciones de Visual Basic y la velocidad de respuesta de una aplicación.
- **Alcance.** En la estrategia de Microsoft, Visual Basic se encuentra como el producto ideal para el desarrollo de aplicaciones comerciales y corporativas, dejando a las herramientas MS Access y MS FoxPro para el usuario final por su facilidad de uso y a MS Visual C++ para desarrolladores de aplicaciones más complejas como lo constituye la creación de un driver para un dispositivo, un driver de acceso a una base de datos o los mismos controles que se

usan en Visual Basic.

Visual Basic tiene las características necesarias para el ambiente de desarrollo como son una rápida edición, verificación e incluye herramientas de depuración muy extensas. Visual Basic es un ambiente de desarrollo que está basado en el lenguaje BASIC. Su aprendizaje requiere de poco tiempo, pues uno de los lenguajes más conocidos hoy en día. La alta productividad de Visual Basic, se debe a las herramientas de ayuda que posee y reducen el tiempo de desarrollo. Puede rápidamente dibujar una interfaz o prototipo para su aplicación y escribir el código que responde a eventos o acciones ejecutados con la interfaz de esos elementos. Por consiguiente, todas las aplicaciones de Microsoft Office soportarán aplicaciones de Visual Basic, pues el mismo lenguaje de desarrollo es utilizado.

Visual Basic se puede combinar con **OLE (Object Linking and Embedding)** "Ligado e Inclusión de Objetos", es una tecnología que habilita al programador de aplicaciones basadas en Windows en crear una aplicación que pueda desplegar datos desde muchas aplicaciones diferentes y posibilita al usuario a editar esos datos desde adentro de la aplicación en la que fue creado. En algunos casos el usuario puede editar esos datos desde la aplicación de Visual Basic.

En el manejo de archivos se cuenta con la posibilidad de crear, guardar y eliminar archivos. El acceso a dichos archivos puede ser de una manera secuencial (para archivos de tipo texto), de forma aleatoria (para archivos con registros de tamaño idéntico) o con un acceso binario (para archivos con registros de diferente tamaño).

El número límite de colores que se pueden utilizar en Visual Basic, está determinado por el tipo de monitor que se maneje en la computadora.

Visual Basic es capaz de generar un archivo ejecutable, lo que proporciona la facilidad de ser portable, además puede soportar múltiples plataformas. Se pueden escribir aplicaciones de 16-bit para Windows 3.1x o aplicaciones de 32-bit para Windows 95 y Windows NT.

4.4.2 Selección del lenguaje.

Para desarrollar la interfaz entre el microcontrolador y el dispositivo de visualización y/o simulación, se deberá utilizar un lenguaje que sea sencillo de manejar y programar proporcionando facilidades tales como objetos ya creados a los cuales se asignen procedimientos que se ejecutarán cuando ocurra el evento programado.

A la vez, deberá permitir el manejo de gráficos para facilitar el diseño de una pantalla amigable de comunicación con el usuario que también sirva de visor para la simulación ; este lenguaje también deberá ser capaz de establecer comunicaciones via puerto serial, para enviar instrucciones y monitorear al microcontrolador.

Anteriormente se habían mencionado los lenguajes C, Visual C++ y Visual Basic para Windows, de los cuales se descartan C y Visual C++, pues el lenguaje C, a pesar de su portabilidad, la creación de gráficos no es muy rápida y sencilla. Por otra parte, Visual C++ viene con algunas herramientas de programación visual impresionantes, pero todavía tiene que saber programación de Windows.

Si bien es fácil generar la estructura de una aplicación, personalizarla requiere un amplio conocimiento del lenguaje C++. Los programas Wizard son herramientas buenas para el aprendizaje y pueden aumentar considerablemente la productividad de aquellos que tienen una buena base de C++, pero no protegen al usuario de los peligros internos de Windows. Estos programas no harán que los novatos salgan programando, como lo hace el Visual Basic.

Otra desventaja de Visual C++, es que ni siquiera se piense ejecutarlo con menos de 8MB de memoria. Si bien su caja dice que 4MB es el mínimo, en realidad es el mínimo que se necesita para compilar algo y no el mínimo par usar cualquiera de las opciones. Para lograr una operación satisfactoria, se requieren 12MB de memoria.

Pero hay que tener cuidado si se está usando una computadora con un bus ISA, un controlador

SCSI con un manejador ASPI4-DOS de Adaptec cargado, y más de 16MB de RAM. El uso de memoria tan exigente de Visual C++ pondrá de manifiesto enseguida que el manejador puede tener acceso a direcciones de no más de 16MB ; y el sistema se trabará cuando trate de compilar.

Tendrá que quitar el disco (y sacrificar el uso de las unidades de cinta o CD-ROM) o quitar la memoria por encima de 16MB. Los tres lenguajes permiten el acceso al puerto serial de la computadora.

Visual Basic es el único que permite un excelente desarrollo de programas para Windows, además de ser el ambiente de trabajo más utilizado actualmente. No obstante, los tres lenguajes son utilizados para el desarrollo profesional de aplicaciones.

Por otro lado, Visual Basic posee una amplia barra de herramientas y ventanas con los diferentes objetos y eventos utilizados en la programación, lo cual lo hace , el más sencillo de utilizar.

El tamaño de los archivos de programas resultantes no es significativo. Además, cuenta con una ayuda en línea lo que facilita su uso y, en un momento dado, su aprendizaje.

Dadas todas las características de este lenguaje, los pasos a seguir para desarrollar un programa en él son :

- Definir el problema.
- Dibujar la interfaz, es decir, crear las ventanas y/o cuadros de diálogo (formas) y agregar los controles o herramientas (botones, textos, menús, listas, etc.).
- Designar las propiedades.
- Escribir el código para los eventos.

Las siguientes gráficas muestran la comparación de las características antes mencionadas.

Interfaz de diseño amigable**Manejo de Gráficos****Desarrollo para Windows****Facilidad de Uso**

Tamaño de los archivos**Comunicaciones Seriales****Uso de Memoria**

Mediante el análisis de los puntos anteriores, se ha decidido que la interfaz de trabajo con el microcontrolador se desarrolle en el lenguaje Visual Basic para Windows por todas las facilidades que éste ofrece para el desarrollo de programas y la posibilidad de acceso a él.

CAPITULO V

DISEÑO DEL SISTEMA

5.1 Desarrollo e implantación a nivel hardware.

En este capítulo se considerarán detalles específicos del diseño e implantación del sistema a nivel hardware.

El hardware del sistema va a estar formado básicamente por tres partes principales:

- **Módulo de recolección de datos.-** Este estará compuesto por los diferentes sensores (velocidad, nivel y temperatura) que se encargan del registro de las variables físicas durante el recorrido del vehículo.
- **El Sistema Mínimo.-** Está compuesto por el microcontrolador MC68HC11 y los diferentes dispositivos (memorias, compuertas, teclado, display, etc) que permiten su funcionamiento, en el modo de operación expandido-multiplexado.
- **Módulo de comunicaciones.-** Este módulo está integrado por todos los elementos que hacen posible la comunicación entre el MCU y la computadora personal (P.C.). Este módulo solo entrará en operación al momento de realizar la transferencia de la información hacia la P.C.

Debido a la facilidad que ofrecen los puertos del microcontrolador para la recolección de datos, así como la versatilidad del circuito integrado MAX-232 para poder realizar las transferencias de información, el módulo de recolección de datos y el de comunicaciones se contemplan dentro del diseño del sistema mínimo.

A continuación se presenta un diagrama de bloques general de las partes fundamentales que conforman el hardware del sistema.

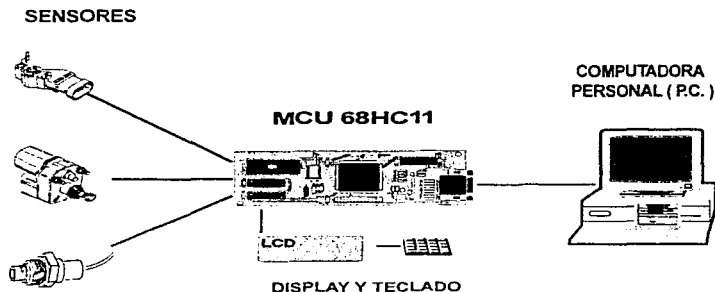


Figura 5.1.a Diagrama general del hardware del sistema

5.1.1 Diseño del sistema mínimo del MC68HC11

Para la realización del proyecto, se consideró un sistema mínimo basado en el MC68HC11E9, el cual trabaja en el modo de operación expandido-multiplexado. Se optó por trabajar en este modo de operación con el objeto de explotar las ventajas que ofrece el MCU de poder acceder dispositivos externos tales como memorias (RAM, ROM y EEPROM) así como también un display de cristal líquido (LCD) y un teclado.

Como ya se estudió anteriormente, para poder trabajar en el modo de operación expandido-multiplexado, es necesario que las terminales MODA (pin 3) y MODB (pin 2) presenten las siguientes condiciones :

MODA = 5 V (1 lógico)

MODB = 5 V (1 lógico)

En este modo de operación tienen vital importancia las funciones de los puertos B, C y D, ya que a través de ellos se proporcionan las señales de dirección, datos y control (AS y R/W) que permiten al MCU el direccionamiento de los 64 Kbytes de espacio posible.

Para realizar la selección de cualquier dispositivo externo, es necesario, contar con una dirección compuesta de 16 bits (A0-A15) que conforma el bus de direcciones, y que apunta a la ubicación de un dispositivo específico dentro del mapa de memoria del MCU. Más adelante, se tratarán los detalles del mapa de memoria del MC68HC11 que se consideró en el desarrollo del presente trabajo.

Por considerarlo de suma importancia, se describen a continuación las funciones de cada uno de los puertos B, C y D dentro del modo de operación expandido-multiplexado.

- Puerto B.- Todas las terminales del puerto B, representan la parte más significativa de la dirección (bus de direcciones), para el acceso a los dispositivos externos. Durante cada ciclo del MCU, los bits A8 - A15 del bus de direcciones son las líneas de salida PB0 - PB7 del puerto B respectivamente.
- Puerto C.- Todos los pines del puerto C, son configurados como líneas de dirección/datos multiplexados. Durante la porción del ciclo del MCU para direccionamiento (señal E en bajo) los bits A0 a A7 del bus de direcciones corresponden a las líneas PC0 a PC7 del puerto C respectivamente. Durante la porción del ciclo del MCU para datos (señal E en alto), los bits 0 a 7 son pines de datos bidireccionales controlados por la señal R/W.
- Puerto D.- Los bits 0 a 5 del puerto D, son usados como líneas de entrada/salida ó con los subsistemas de comunicación serial (SCI) e Interfaz Serial de Periféricos (SPI). El pin 6 del puerto D proporciona una señal de control (AS) la cuál es empleada para demultiplexar las

señales de dirección y datos del puerto C. El pin 7 del puerto D (PD7) proporciona la señal de control (R/W) que es empleada para el control de la dirección de las transferencias en el bus de datos externo.

La circuitería externa al MCU, que conforma el sistema mínimo, se puede dividir en los siguientes módulos o circuitos :

- Módulo de alimentación
- Circuito de reset
- Selector de dispositivos
- Display
- Circuito oscilador
- Módulo de comunicaciones
- Memorias
- Teclado

A continuación se presenta una descripción de cada uno de los módulos listados anteriormente (para una mejor claridad, solo se representan las terminales más relevantes de los dispositivos) :

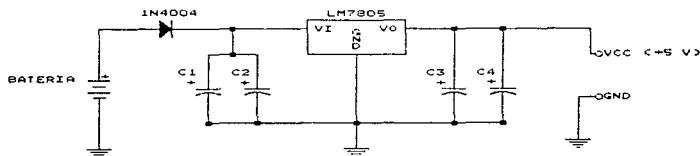
Módulo de alimentación.

Para que el circuito pueda operar, se requiere que sea alimentado con 5 Volts, ésta alimentación puede proporcionarse a través de una fuente de D.C., basada en la batería del vehículo, sin embargo, si éste fuera el caso tendríamos algunos inconvenientes tales como :

- Elevación en el costo del sistema
- Instalación más compleja
- Posibilidad de fallas como consecuencia de otros dispositivos como alarmas, luces, sistemas eléctricos en puertas, vidrios, etc.
- Posibilidad de sabotaje del sistema, al provocar un corto circuito ó desconexión de la alimentación.

Por lo anterior y gracias a la característica de bajo consumo de potencia (familia CMOS) de los componentes electrónicos empleados, se optó por una alimentación independiente a través de una batería comercial de D.C.

La figura 5.1.1.a muestra el diagrama electrónico del módulo de alimentación. Podemos observar que éste circuito está basado en un regulador de voltaje (LM7805CTB), que permite un voltaje de entrada en el rango de 7 V a 35 V (ésto siempre y cuando se respeten las condiciones del dispositivo), proporcionando una salida regulada de $5\text{ V} \pm 5\%$, con una corriente de 1 A y una disipación de potencia de 15 Watts.



Nota: Todos los capacitores son de 0.1 μF a 16 Voltios

Figura 5.1.1.a Módulo de alimentación

Circuito Oscilador

Este circuito está formado por un cristal que trabaja a una frecuencia de 8 Mhz; esta frecuencia es necesaria para que el sistema pueda operar con una velocidad de bus interno (señal E) de aproximadamente 2 Mhz, así mismo permite establecer la velocidad de transmisión de 9600 bauds que se requiere para la comunicación del puerto serial D con la computadora personal. Los

valores de los componentes que integran el circuito oscilador son los sugeridos por el fabricante para el caso en que el MCU trabaje a frecuencias altas.

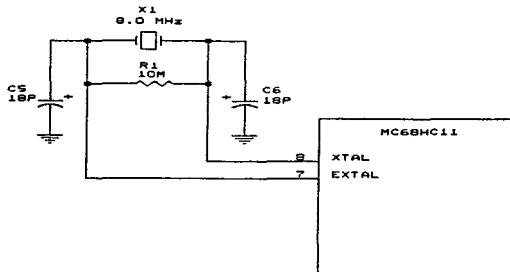


Figura 5.1.1.b Circuito Oscilador

Circuito de Reset.

Este circuito es particularmente importante si llegarán a presentarse situaciones indeseadas como fallas internas ó "caídas" del sistema.

Como puede observarse en la figura 5.1.1.c, éste circuito está conformado por un interruptor manual (SR) normalmente abierto y un conjunto de inversores (SN74HC04N), estos elementos conforman un reset manual (interrupción por hardware) al sistema. Cuando el interruptor SR es accionado, se genera una señal de reset (pulso bajo) a la entrada el MCU.

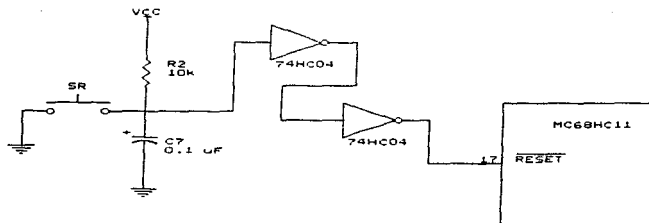


Figura 5.1.1.c Circuito de Reset

El conjunto RC que se encuentra acoplado al interruptor, permite obtener una señal con un periodo de duración de aproximadamente 1 ms, lo cual es suficiente para que el MCU pueda detectar la interrupción. El conjunto de inversores se emplean para poder acondicionar la señal a la entrada del microcontrolador.

Módulo de comunicaciones.

Los detalles de la comunicación entre el MCU y la PC, se han tratado anteriormente (Cap. IV Sistema de comunicaciones, registro y manejo de datos), de acuerdo con esto, el canal de comunicación óptimo resultó ser la interfaz RS-232C: ya que el MCU a través del puerto de comunicaciones D, puede acoplarse directamente al puerto serial de la PC, conectando los pines TxD y RxD del puerto D con sus respectivos pines del puerto serial de la PC, con un conector DB9 ó DB25, según sea el caso.

Para llevar a cabo la tarea de la comunicación, el sistema está provisto de un circuito integrado denominado MAX-232. Este circuito tiene la capacidad de proporcionar la amplitud necesaria

para la comunicación, esto con el fin de evitar pérdidas en la línea de comunicación si la distancia que separa a la PC del MCU fuera considerable.

Otro aspecto importante es el que se refiere al manejo de los niveles de voltaje entre el MCU y la PC, dado que dichos niveles son diferentes entre el puerto D del MCU y el puerto serial de la PC: es necesario trasladarlos y de ésta forma obtener un funcionamiento adecuado del circuito.

El MAX-232 puede generar dos fuentes de voltaje, una positiva y otra negativa. La fuente positiva se genera de la fuente de alimentación de +5 V, la cuál mediante la conexión de un capacitor externo en las terminales 1 y 3 del C.I., produce una salida de +10 V en la terminal 2 del mismo C.I. La fuente negativa se obtiene a partir de la fuente de +10 V, mediante la conexión de otro capacitor externo entre las terminales 4 y 5 del C.I., con esto se obtiene un voltaje de -10V. Estas fuentes son las que entregan la amplitud necesaria para las transmisiones del RS-232.

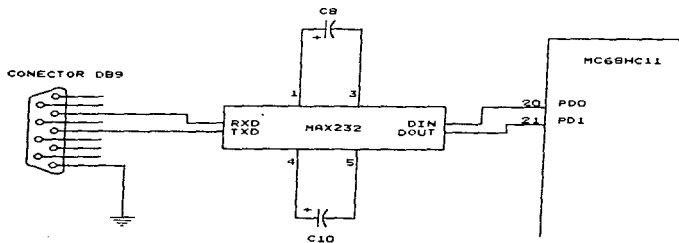


Figura 5.1.1.d Módulo de Comunicaciones

Con dos manejadores y dos receptores, el MAX-232 es el adecuado para la interfaz RS-232, en donde las conexiones DTE (Data Terminal Equipment) y la DCE (Data Communications Equipment) usan una línea de datos y una de control.

El MAX-232 se conecta a través de sus terminales 9 y 10 al puerto D del MCU, esto es a PDO (Entrada para recepción de datos RxD de la interfaz de comunicación serial SCI) y a PDI (salida para transmisión de datos TxD del SCI).

Selector de Dispositivos

Para una mejor comprensión del funcionamiento de este módulo, es necesario conocer primeramente la definición del mapa de memoria del MCU, que se realizó para el presente trabajo.

En la tabla 5.1.1.e se muestra la definición del mapa de memoria del MCU. Como ya se ha mencionado el MC68HC11 maneja un bus de direcciones de 16 líneas (puertos B y C del MCU), esto permite un manejo de 64 Kbytes de espacio direccionable. Dentro de éste espacio, existen áreas que no es posible modificar puesto que son definidas por el fabricante (áreas sombreadas en la tabla), por ejemplo las memorias internas y el área de registros que es donde se programan todos los recursos internos del MCU.

Las áreas restantes se definieron a conveniencia y de acuerdo a las necesidades que se presentaron. Así, por ejemplo la EEPROM Externa quedó definida en la localidad \$4000, el teclado en la \$6000, etc.

Un área de especial interés es la \$E000-\$FFFF, ya que aquí se encuentra el vector de interrupción (\$FFFE-\$FFFF) que indica la localidad de memoria donde se encuentra el programa a ejecutar, una vez que el sistema es arrancado.

DIRECCION INICIAL	DIRECCION FINAL	DISPOSITIVO ASOCIADO
\$0000	\$00FF	256 Bytes de RAM Interna
\$1000	\$103F	64 Bytes del Bloque de Registros
\$2000	\$3FFF	8 Kbytes de RAM Externa
\$4000	\$5FFF	8 Kbytes de EEPROM Externa
\$6000	\$6000	Teclado
\$8000	\$8000	Display
\$B600	\$B7FF	512 Bytes de EEPROM Interna
\$E000	\$FFFF	8 Kbytes de ROM Externa

Tabla 5.1.1 e Definición del mapa de Memoria del MC68HC11 para el sistema

La memoria ROM externa se ha definido en esta área a propósito, puesto que de esta forma es posible grabar aquí el programa principal y ejecutarlo automáticamente cada vez que se inicializa el sistema.

La forma de acceder los diferentes dispositivos, es a través de operaciones de lectura y/o escritura a las diferentes localidades de memoria en donde éstos se encuentran asociados, por ejemplo, si se desea escribir en el display, se debe escribir en la localidad \$8000, si deseamos leer la EEPROM externa, debemos leer la localidad \$6000, etc.

Para poder llevar a cabo la tarea de selección de los diferentes dispositivos periféricos, el sistema cuenta con un decodificador (74HC138) de 3 a 8 líneas, el cuál provee al sistema de las señales para habilitar la lectura y/o escritura de dichos dispositivos.

Las líneas de selección del decodificador corresponden a las líneas A15, A14 y A13 (los bits más significativos) del bus de direcciones. La temporización para las diferentes señales de salida del

decodificador son controladas por la señal de reloj E (pin 5) que se encuentra conectada a la entrada del decodificador.

Observando la tabla 5.1.1.f. se puede comprender mejor el mecanismo de decodificación del bus de direcciones, en ésta tabla se muestran algunos de los estados del bus de direcciones y la localidad (dispositivo periférico) que se asocia. Las líneas A15, A14 y A13 (columnas sombreadas) representan las señales que alimentan al decodificador.

BUS DE DIRECCIONES																LOCALIDAD ASOCIADA	
PUERTO B (DIRECCION)								PUERTO C (DIRECCION/DATOS)									
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0		
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	\$2000	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	\$4000	
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	\$6000	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	\$8000	
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	\$E000	

Tabla 5.1.1.f Estados del bus de direcciones y los localidades asociadas

Las líneas que representan la parte menos significativa (puerto C) del bus de direcciones son separadas de las líneas de datos, a través de un latch (74HC373) que es controlado a través de la señal AS (address strobe). De ésta manera dependiendo del ciclo del MCU, las terminales de este circuito representan a las líneas A7-A0 del bus de direcciones ó bien conforman el bus de datos del sistema.

La figura 5.1.1.g representa el diagrama electrónico del circuito selector de dispositivos.

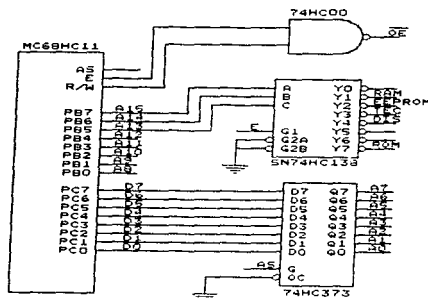


Figura 5.1.1.g Circuito selector de dispositivos

Memorias

Una de las grandes ventajas del uso de un sistema en el modo de operación expandido-multiplexado, es el aprovechamiento al máximo de la capacidad del MC68HC11 para acceder y operar diferentes dispositivos externos. Uno de estos dispositivos es la memoria.

Dadas las características del problema a solucionar y por otro lado la cantidad de datos que deben de registrarse, es mucho más ventajoso el uso de memoria externa, puesto que internamente el MCU MC68HC11 solo cuenta con 512 bytes de EEPROM y 256 bytes de RAM, que resultan insuficientes para nuestro propósito en el presente trabajo.

En contraste, en el modo de operación que se ha seleccionado, es posible emplear 8 Kbytes de ROM para alojar al programa principal, 8 Kbytes de EEPROM para el registro de datos y 8 Kbytes de RAM adicionales para cálculos y almacenamiento temporal.

A continuación se describen las características principales de las memorias con que cuenta el sistema y su función principal :

- Memoria ROM de 8K (27C64). En realidad es una memoria del tipo UV EPROM (borrable a través de luz ultravioleta). En esta memoria se pueden almacenar programas y/o datos que se utilicen frecuentemente, ya que permanecen almacenados indefinidamente. Para nuestro caso, esta memoria se emplea para alojar el programa principal de registro de datos y de comunicación. Además dado que está asociada a las localidades \$E000-\$FFFF, se tiene acceso al vector de interrupción de arranque (\$FFFE-\$FFFF), lo cual permite ejecutar el programa principal cada vez que se arranca el sistema.
- Memoria RAM de 8K (6264). Este tipo de memoria posee la característica de ser volátil, es decir, su contenido se elimina una vez que la alimentación al sistema es suspendida. Debido a esto, esta memoria solo se emplea para el manejo de programas y/o datos en forma temporal. El tamaño de esta memoria es muy conveniente para el desarrollo de futuros programas de mantenimiento ó diagnóstico al sistema, ya que pueden ser ejecutadas con el MCU trabajando en modo stand-alone y sin la necesidad de reemplazar el programa alojado en la ROM.
- Memoria EEPROM de 8K (2864 A). Esta es una de las memorias más funcionales, ya que su característica de borrado eléctrico, permite el almacenamiento de programas y/o datos y conservarlos aún si la alimentación es suspendida. Para nuestro trabajo, se ha elegido esta memoria para albergar los valores de las variables físicas que se vayan registrando durante el recorrido del vehículo, ya que sus características permiten limpiar su contenido y reutilizarla cada vez que se inicie un nuevo trayecto. Como se estudiará más adelante, el tamaño de esta memoria se definió a partir del cálculo del número de variables posibles a registrar durante el recorrido más largo posible (72 hrs continuas).

En la figura 5.1.1.h se muestra el diagrama electrónico de la conexión de las diferentes memorias. Como puede observarse, todas las memorias comparten el mismo bus de direcciones y de datos, las señales de habilitación para cada una provienen del decodificador de 3 a 8 del módulo de selección de dispositivos.

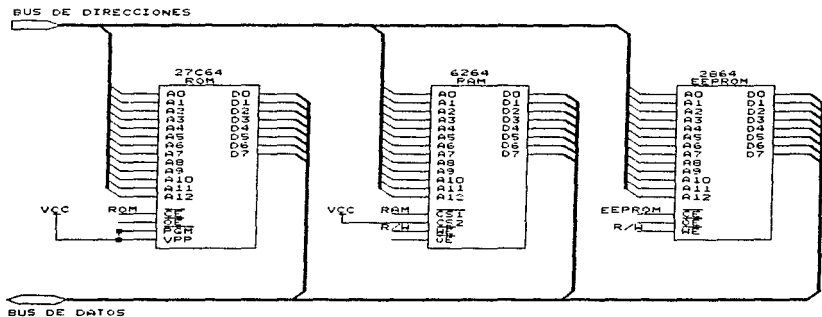


Figura 5.1.1.h Conjunto de memorias del sistema

Display

El circuito está provisto con un display de cristal líquido (LCD) del tipo AND-491. Este display está compuesto por dos líneas de 16 caracteres cada una. Así mismo puede observarse que las líneas de datos de éste dispositivo están conectadas directamente al bus de datos del sistema.

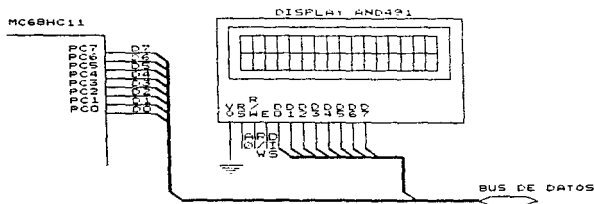


Figura 5.1.1. Circuito electrónico del display AND-491 del sistema

Teclado

El teclado es del tipo matricial y está integrado por veintitrés teclas. Dieciséis de ellas representan los caracteres del código hexadecimal (0 al 9 y A a la F) propios para poder realizar la programación del sistema directamente. Se dispone del hardware suficiente para definir un mayor número de teclas (hasta 32 posibles), de las cuales algunas pueden ser definidas para funciones especiales como la edición e inspección directamente de las localidades de memoria del sistema a través de un programa de mantenimiento o diagnóstico.

Se cuenta también con un grupo de flip-flops tipo D (SN741HC273N) y un buffer de ocho líneas (SN74HC244N), ambos dispositivos permiten realizar la decodificación del teclado matricial, al mismo tiempo que acondicionan las señales, evitando el efecto de "rebote" producto de los interruptores mecánicos.

La integración del display y el teclado facilita al operador la puesta en marcha del sistema y la operación del mismo, ya que es posible programar rutinas interactivas con menús y mensajes.

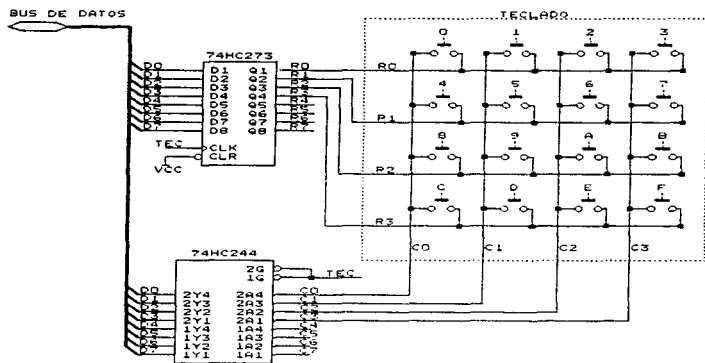
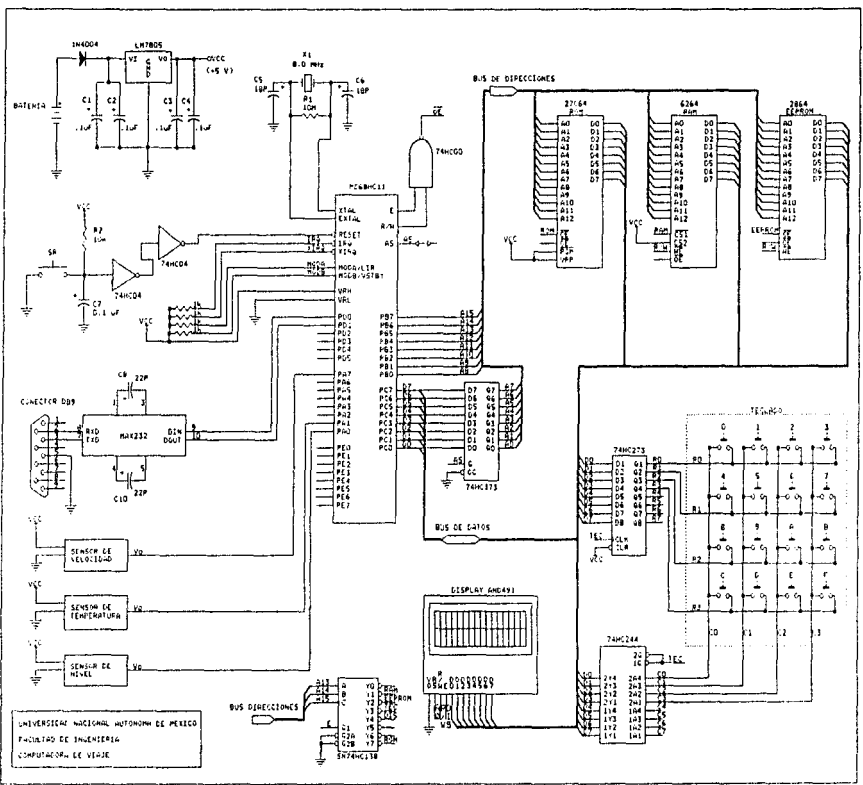


Figura 5 1.1 j Circuito electrónico del teclado del sistema

A continuación se muestra el diagrama electrónico completo del sistema mínimo:



5.1.2 Módulo de Recolección de Datos

El módulo de recolección de datos, está conformado por el siguiente grupo de sensores :

1. sensor de nivel de combustible
2. sensor de temperatura
3. sensor de velocidad

Estos sensores, son los encargados de registrar las variables físicas involucradas en el trayecto del vehículo y de enviar las señales correspondientes al microcontrolador. Como ya se estudió anteriormente, todos los sensores empleados en el presente trabajo, son de uso comercial en la industria automotriz, por lo que sus características eléctricas, permiten un acoplamiento directo entre estos y el sistema mínimo.

En el capítulo correspondiente a sensores se mencionaron sus principales características y los lugares óptimos de trabajo, por lo que en el presente subcapítulo sólo será necesario resaltar algunos puntos importantes para comprender la manera en que la computadora de viaje realiza la recolección de los datos .

El sensor de nivel.

Como ya se estudió, este sensor se emplea para registrar el nivel de gasolina contenido en el tanque en diferentes intervalos de tiempo durante el recorrido, está compuesto por un flotador que se mueve dependiendo del nivel de llenado del tanque, al moverse el flotador se hace variar una resistencia entre 0-277 Ω , este dispositivo se alimentado con 5 volts y se encuentra aterrizado al chasis del vehículo. El flotador se encuentra generalmente junto con la bomba de gasolina ambos colocados dentro del tanque de combustible.

De lo anterior se desprende que este sensor a la salida entrega un voltaje entre 0 y 5 volts analógicos dependiendo de la caída de potencial en la resistencia, de esta manera es posible

realizar una relación proporcional entre voltaje resultante y nivel de gasolina contenido en el tanque.

El microcontrolador MC68HC11 registra voltajes digitales de 0 ó 5 volts por lo que es necesario conectar la salida del sensor a la entrada del convertidor analógico digital del microcontrolador para que convierta este voltaje analógico en digital y se pueda llevar a cabo la medición.

La siguiente tabla, muestra algunas de las relaciones que guardan el nivel de voltaje enviado por el sensor y el porcentaje (%) de combustible en el tanque del vehículo.

Voltaje [V]	Nivel de gasolina [% del contenido del tanque]
0.02	100
2.5	50
5	0

Tabla 5.1.2 a Relación voltaje-combustible del sensor de nivel

Sensor de Temperatura.

Este sensor será de gran utilidad para verificar periódicamente la temperatura del motor durante el recorrido, como ya se estudió en el capítulo II, este dispositivo es un termistor que cambia su resistencia en función inversa con la temperatura, así a temperaturas normales de operación (90 °C a 112 °C), el voltaje debe ser entre 1.5 y 2.0 volts.

Temperatura (°C)	Voltaje (V)
-44	5
90	2
112	1.5
135	0

Tabla 5.1.2 b Algunos de los valores característicos del sensor de temperatura

La salida del sensor debe ir conectada al convertidor analógico-digital del MC68HC11. La tabla 5.1.2.b muestra algunos de los valores característicos que maneja este sensor.

Sensor de velocidad (VSS)

Este dispositivo es uno de los más importantes, puesto que a partir de éste se miden directa ó indirectamente muchas de las variables físicas que caracterizan un recorrido como velocidad promedio, velocidad máxima, aceleración, número de paradas, tiempos de cada parada, kilómetros recorridos, etc. La forma de calcular cada una de estas variables se tratará más adelante.

El sensor de velocidad es un generador mecánico de pulsos, que utiliza un micro-interruptor normalmente abierto formado de dos platinos y montado cerca del eje rotatorio. El voltaje digital que entrega esta entre 0 y 5 volts. El número de pulsos se incrementa proporcionalmente la velocidad del vehículo y varía dependiendo del tipo de vehículo y de sensor que se trate, para nuestro caso la señal enviada por este sensor es de aproximadamente 2486 pulsos/Km.

5.1.3 Registro de las variables físicas

Dentro de nuestro proyecto se instalaron tres sensores dentro vehículo que se va a estudiar, estos son el sensor de velocidad, sensor de temperatura y sensor de nivel de combustible o también llamado gasómetro.

El sensor de velocidad leerá la información y la transmitirá por medio de pulsos, midiendo previamente cada pulso que equivale a una determinada distancia se podrá saber con una mayor exactitud en un determinado tiempo la distancia que se recorrió, el tiempo que transecurrido y la velocidad.

Básicamente la información será registrada en los siguientes campos:

PW	NSER	CODRE	FECHA	COMBI	CE
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
HINI	HFIN	CODOP1	CODOP2	COMBF	CE
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Donde:

- PW: Password del supervisor
- NSER: Número de serie del circuito
- FECHA: Fecha de inicio del recorrido
- HINI: Hora inicio del recorrido
- HFIN: Hora de terminación del recorrido
- CODOP1 y CODOP2 : Código del operador
- CODRE: Código del recorrido
- COMBI: Combustible inicial
- COMBF: Combustible final
- CE: Código de error del sensor

La información será registrada en dos tablas diferentes la primera tabla guardará los datos que se recopilen cada 5 minutos. En ella se grabará el número de pulsos y la temperatura.

Nº DE PULSOS	CE	TEMPERATURA	CE

Tabla 5.3.a Registro de información cada 5 minutos

en la segunda tabla se guardará el número de parada (NP), hora de arribo (HA), hora de partida (HP), combustible de arribo (CA) y combustible de partida (CP), cada uno con su respectivo código de error en caso de que lo tenga.

HA	HP	CA	CE	CP	CE

Tabla 5.3 b Registro de información en cada parada

Donde

HA: Hora de arribo

HP: Hora de partida

CA: Combustible de arribo

CP: Combustible de partida

CE: Código de error

Como se mencionó anteriormente se va a realizar una serie de muestras de las siguientes variables:

- Velocidad
- Temperatura
- Nivel de combustible

Obteniendo estas variables se podrán obtener la velocidad media, velocidad instantánea, velocidad máxima, distancia recorrida, número de paradas, tiempo por paradas, entre otras.

En la figura 5.3.c se muestra el diagrama general del sistema de recopilación de datos durante el recorrido.

DIAGRAMA GENERAL DEL RECORRIDO

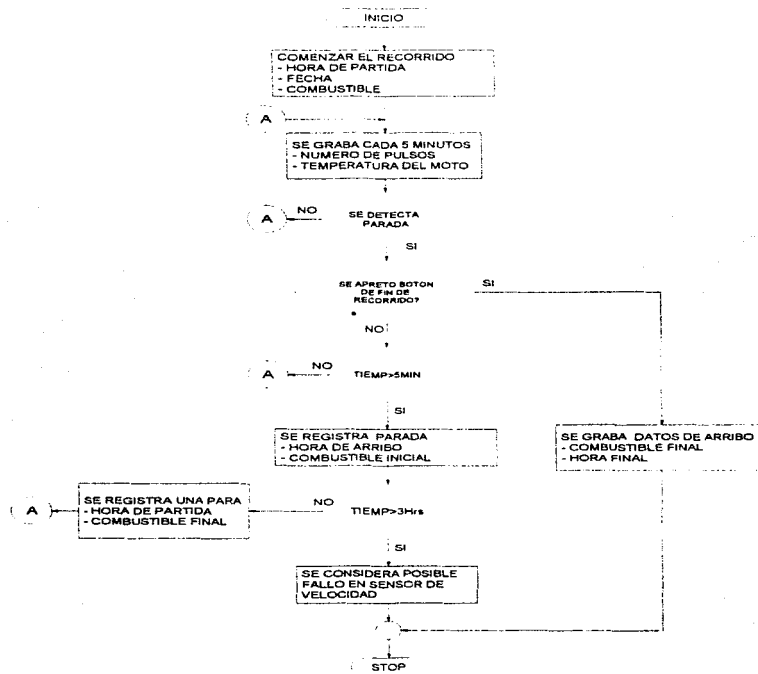


Figura 5.3.c Diagrama general del recorrido del vehiculo

Una vez montada la computadora de viaje un supervisor procederá a dar un reset a la tarjeta para inicializarla. Primeramente se pedirá el password al supervisor y se le mandara a un menú:

"Menú"

1. Registro del recorrido
2. Transferencia de información.

Al seleccionar la opción 1 mandara a la secuencia de recorrido donde todas las variables se inicializaran, esto es los contadores y banderas se pondrán con valor de cero.

Se tendrá que teclear posteriormente la fecha, código del o de los operadores en caso que sean dos (CODOP1 y/o CODOP2), código de recorrido (CODRE), y la hora en que se inicia (HINI). se pregunta si los datos son correctos al estar bien se realizará una rutina del sensado de combustible guardándolo en un registro llamado COMB, si el tanque esta vacío o se detecta que no esta registrando ninguna información se encenderá una bandera grabando la información en C.E. de COMB. donde se registrara el código de error y de todas formas se registrara el valor del combustible.

Una vez que el autobús este en movimiento se irá registrando el número de pulsos y la temperatura del motor (tabla 5.3.a), si el sistema llegase a detectar una parada encenderá un cronometro que contara hasta 5 minutos donde la computadora considerará que es una parada, si el autobús en menos de ese tiempo se mueve no registrará datos del arribo y continuará grabando la información cada 5 minutos si el autobús sigue en movimiento. Si la computadora de viaje detecto una parada grabará la hora de arribo (HA) y el combustible de arribo (CA). No debe de ser mayor el tiempo de 3 horas de estar detenido el autobús pues la computadora considerará que el sensor de velocidad se daño, dado que sería ilógico que se detenga a comer o cargar gasolina en un tiempo mayor a las 3 hrs. El sistema procederá a detenerse pues el sensor de velocidad es el dispositivo más importante con el que se podrán obtener datos de velocidad distancia entre

otras variables. Una vez que el autobús estuvo detenido un tiempo menor de las 3 horas y comience a moverse se grabarán el combustible de partida (CP) y la hora de partida (HP).

El sistema cuenta con un sistema para detectar si se apretó un botón, el cual nos indicará que el autobús ya llegó a su punto de arribo y ese botón será apretado por el supervisor, la computadora considerará que el trayecto ya terminó y grabará el combustible final (COMBF) y la hora final (HFIN).

5.1.4 Transmisión y recepción de datos.

En los capítulos anteriores, se han estudiado las características del hardware que conforma el módulo de comunicaciones, así mismo en el punto 5.2 Diseño del Software, se tratarán los detalles de los programas tanto del microcontrolador como de la PC. El objetivo de este punto es presentar el procedimiento general que debe realizarse para establecer la comunicación entre el MCU y la PC y poder transmitir la información.

El propósito del procedimiento de comunicación es verificar que el enlace entre el sistema mínimo (a través del C.I. MAX-232) y el puerto serial de la computadora (COM1 ó COM2) está funcionando adecuadamente para realizar la transferencia de información.

Una vez que concluya la transmisión de la información del MCU hacia la PC, se procederá a realizar su interpretación y análisis.

La función que debe realizar el MCU durante la inicialización es ejecutar una rutina de comunicación que se encarga de recibir la información de la PC y regresarla simulando el efecto de ECO, con esto la computadora determinará si el sistema está listo para transmitir la información.

Los parámetros de comunicación que deben de ser considerados para efectuar el enlace del MCU con la computadora son los siguientes :

- Tipo de Interfaz : RS-232C
- Velocidad : 9600 bauds
- Bits de datos : 8
- Paridad : Ninguna
- Bits de stop : 1

La selección de los parámetros anteriores se debió a que pueden ser programados tanto del lado de la computadora como del MCU, en otras palabras, permiten la compatibilidad entre estos dos elementos para poder efectuar un enlace de comunicación, así como por ser parámetros estándar para puertos seriales. No obstante, estos parámetros pueden ser adaptados de acuerdo a las necesidades de volumen y velocidad de transferencia de información.

Los parámetros de comunicación deben ser definidos en el registro de configuración para la transmisión del MCU y en el control de comunicaciones que se utiliza en el programa de interfaz en la computadora.

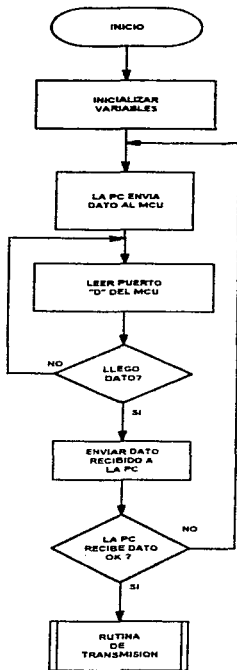
Este control de comunicación permite efectuar el enlace del puerto serial de la PC con el microcontrolador. Por el lado del MCU, la comunicación se efectúa utilizando el subsistema de comunicación SCI (Serial Communication Interface) que posee los elementos necesarios para realizar enlaces de forma serial por medio del puerto D del MCU. Como ya se mencionó, los detalles del programa en ensamblador que efectuará la rutina de inicialización y comunicación, se tratan más adelante.

Por el lado de la computadora se tiene un programa desarrollado en el software llamado Visual Basic, el cuál es un paquete de computación para desarrollo de aplicaciones gráficas. Este software, basa la transmisión y recepción de la información en un control de comunicaciones seriales a través de los puertos de la computadora de acuerdo a los parámetros de comunicación definidos para efectuar éste enlace. Los detalles del programa de control de comunicaciones para el caso de la PC, también se tratan más adelante en el punto 5.2 Diseño del Software.

El programa desarrollado en Visual Basic para ejecutar las rutinas de comunicación en la PC y que efectúan el enlace con el MCU durante la inicialización, tiene como objeto principalmente el mandar desde la PC cierta secuencia de bytes y que el microcontrolador los reciba y retransmita simulando el ECO, de esta forma se puede determinar si el subsistema de comunicaciones está funcionando correctamente, lo cual permitirá inicializar la transferencia de información proveniente del MCU.

El diagrama de flujo del procedimiento general, que se deberá realizar para establecer la comunicación entre la PC y el MCU se muestra a continuación.

**DIAGRAMA DE FLUJO DEL PROCEDIMIENTO
GENERAL DE LA COMUNICACION DEL MCU CON LA PC**



INICIALIZA VARIABLES PARA DEFINIR VELOCIDAD DE TRANSMISION, REGISTRO DE CONTROL Y VARIABLES DE TRABAJO. TANTO DEL LADO DE LA PC COMO DEL MCU.

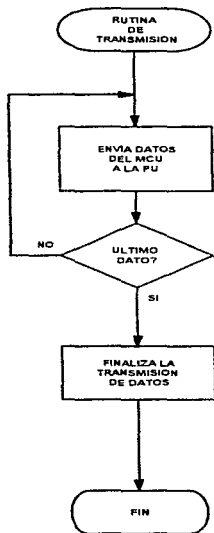
LA PC ENVIA DATO HACIA EL MCU PARA VERIFICAR QUE EL ENLACE ESTE OK.

EL MCU LEE EL PUERTO "D" HASTA RECIBIR DATO DE VERIFICACION PROVENIENTE DE LA PC.

TRANSMITE EL DATO RECIBIDO PARA SIMULAR EL ECO Y VERIFICAR EL ENLACE

LA PC VERIFICA QUE LOS DATOS QUE RECIBE EL MCU SON OK. SI ES ASI, LA COMUNICACION SE HA ESTABLECIDO CORRECTAMENTE. EN CASO CONTRARIO CONTINUA ENVIANDO DATOS DE PRUEBA.

SI EL ENLACE SE ESTABLECE CON EXITO, SE PROCEDE EJECUTAR LA TRANSMISION DE LOS DATOS QUE EL MCU TIENE REGISTRADOS EN SU MEMORIA.



EL MCU SE ENCARGA DE TRANSMITIR LOS DATOS QUE TIENE REGISTRADOS EN MEMORIA HACIA LA PC.

SE VERIFICA QUE SE HA TRANSMITIDO TODOS LOS DATOS. SI ES ASI SE FINALIZA LA RUTINA DE TRANSMISION. EN CASO CONTRARIO SE CONTINUA CON EL ENVIO DE DATOS.

5.1.5 Acoplamiento entre módulos

Como ya se mencionó anteriormente, el sistema está compuesto por tres módulos principales, que son :

- Módulo de recolección de datos.
- Sistema Mínimo.
- Módulo de comunicaciones.

El módulo de comunicaciones está compuesto por el grupo de sensores a utilizar (velocidad, nivel de combustible y temperatura). De acuerdo a lo estudiado anteriormente, la selección de este grupo de sensores se realizó tomando como base el mercado automotriz actual. Una de las ventajas principales que se tienen, es que la mayoría de estos dispositivos, están diseñados para trabajar con microprocesadores (los que controlan el funcionamiento del motor), por lo que los niveles de voltaje y corriente que manejan son los adecuados para nuestro sistema.

Además, gracias a sus características, cada uno de los sensores seleccionados tiene una gran inmunidad al ruido y pueden responder a diferentes niveles de voltaje de alimentación. Esto es muy importante, ya que permite que las señales que se reciben de estos, se alimenten directamente al sistema sin la necesidad de un circuito de acondicionamiento de señales adicional. Por otro lado, el Módulo de comunicaciones está compuesto básicamente por el C.I. MAX-232 y se encuentra integrado al sistema mínimo. Esto es sumamente ventajoso, ya que todos los dispositivos de acoplamiento ya se encuentran en el mismo circuito, además de que el MAX-232 se encarga directamente de acondicionar los niveles de voltaje entre el MCU y la PC para la transferencia de información. De esta forma únicamente es necesario contar con una interfaz del tipo RS-232C para conectar el sistema a la PC y llevar a cabo la transferencia de la información.

En resumen, podemos mencionar que gracias a las características en el diseño de los sensores, así como a las capacidades del C.I. MAX-232, es posible manejar señales compatibles entre los diferentes módulos, sin la necesidad de emplear circuitos adicionales de acondicionamiento, basta con definir los diferentes niveles de voltaje en los rangos de operación adecuados.

5.2 Diseño del Software.

En este tema se definen las rutinas de software que deben ser ejecutadas para la operación del sistema. El diseño del software engloba los programas de comunicación que establecen en forma serial la comunicación entre la computadora y el MC68HC11. También se crearán los programas que efectuarán el enlace y la decodificación de la información proveniente del MCU, además de construir la interface gráfica por medio de Visual Basic para la navegación del sistema y de la creación de la base de datos por medio de MS Access, para el almacenamiento de la información del MC68HC11 y otros datos que serán generados por cálculos internos de los programas de Visual Basic.

5.2.1 Programación del Microcontrolador.

En el apéndice A se presenta el programa que se encuentra alojado en la memoria del Microcontrolador y que permitirá el registro de las variables físicas en el recorrido del vehículo, así como la transferencia de dichos datos hacia la computadora personal.

5.2.2 Descripción del control de comunicaciones de Visual Basic 4.0.

El control de comunicaciones provee la comunicación serial para aplicaciones que permiten la transmisión y recepción de datos a través del puerto serial.



Nombre del archivo : MSCOMM16.OCX, MSCOMM32.OCX

Nombre de la clase : MSComm

5.2.2.1 Características

Este control de comunicaciones proporciona dos alternativas para efectuar la comunicación :

- **Comunicación por Evento.** Es un método muy poderoso para manejar la interacción con el puerto serial. En algunas situaciones es necesario notificar el momento en que se lleva a cabo algún evento, esto es, cuando hay cambio en la línea de Detección de Portadora (CD) o en la línea de Petición para Envío (RTS). En estos casos es conveniente utilizar el evento OnComm para atrapar y manejar estos eventos. El evento OnComm también permite manejar y detectar los errores de comunicación.
- **Comunicación por poleo.** Con este método es necesario verificar el valor de la propiedad CommEvent para detectar la comunicación o algún error, este poleo debe efectuarse cada vez que se efectúe una función crítica dentro del programa. Este método se recomienda para aplicaciones pequeñas en donde no se realizan transmisiones que involucren el manejo de los diferentes eventos de comunicación.

Debe utilizarse un Control de comunicación por cada puerto que se requiera utilizar. La dirección del puerto y de la interrupción deben ser definidas en el Panel de Control de Windows. Algunas

de sus propiedades son :

- **Break**

- ⇒ Descripción : Indica o limpia la señal de break. Esta no esta disponible durante el diseño.
- ⇒ Sintaxis : {form}.JMSComm.Break[= {True | False}]
- ⇒ Tipo de dato : Integer (boolean)
- ⇒ Comentarios : En la siguiente tabla se explican los valores de esta señal.

Valor	Descripción
True	Indica la señal de break
False	Limpia la señal de break

Cuando la propiedad tiene el valor True manda la señal de break, esta señal suspende la transmisión hasta que se cambie el valor de la propiedad a False.

- **CDTimeout**

- ⇒ Descripción : Define el tiempo en milisegundos que el control espera por la señal Carrier Detect (CD). Esta señal indica que ha transcurrido el tiempo definido prentiendo la propiedad CommEvent con comCDTO (Carrier Detect Timeout error) y genera el evento OnComm.
- ⇒ Sintaxis : {form}.JMSComm.CDTimeout[= milliseconds&]
- ⇒ Tipo de dato : Long
- ⇒ Comentarios : Cuando la línea CD permanece en estado bajo (CDHolding = False y ha transcurrido el número de milisegundos definido en la propiedad CDTimeout, el control de comunicaciones prende la propiedad de CommEvent con comCDTO (Carrier Detect Timeout Error) y genera el evento OnComm.

- **CommEvent**

- ⇒ Descripción : Regresa el ultimo evento de error o de comunicación. Esta propiedad no esta disponible durante el diseño.

- ⇒ Sintaxis : [form.]MSCComm.CommEvent
- ⇒ Tipo de dato : Integer
- ⇒ Comentarios : La propiedad CommEvent contiene el número del código de error o del evento. Los valores de código que puede tomar esta propiedad se muestran en las siguientes tablas.

Códigos de error de la propiedad CommEvent :

Valor	No. de Error	Descripción
comBreak	1001	Indica que se recibió la señal de break
comCTSTO	1002	Clear to Send Timeout
comDSRTO	1003	Data Set Ready Timeout
comFrame	1004	El hardware detectó un error en el frame
comOverrun	1006	Overrun en el puerto. No se leyó un carácter antes de que llegará el siguiente.
ComCDTO	1007	Carrier Detect Timeout
comRxOver	1008	Overflow en el buffer de recepción
comRxParity	1009	Error de paridad
comTxFull	1010	El buffer de transmisión está lleno

Códigos de comunicación de la propiedad CommEvent :

Valor	Código	Descripción
comEvSend	1	Se envían los caracteres contenidos en el buffer de transmisión
comEvReceive	2	Se recibió el número de caracteres definidos por Rthreshold.
ComEvCTS	3	Cambio detectado en la línea Clear To Send
comEvDSR	4	Cambio detectado en la línea Data Set Ready. Este evento ocurre cuando DSR cambia de -1 a 0.
comEvCD	5	Cambio detectado en la línea Carrier Detect
comEvRing	6	Se detectó la campana (RING). Algunos UART's no soportan esta característica
comEvEOF	7	Carácter de fin archivo recibido (ASCII26)

- **InBufferCount**

- ⇒ Descripción : Determina el número de caracteres que están contenidos en el buffer de recepción. Esta propiedad no esta disponible durante el diseño.
- ⇒ Sintaxis : [form.]MScComm.InBufferCount[= count%]
- ⇒ Tipo de dato : Integer
- ⇒ Comentarios : Esta propiedad se refiere al número de caracteres recibidos que están contenidos en el buffer de recepción. Para limpiar este buffer se debe aplicar el valor de cero a InBufferCount.

- **InBufferSize**

- ⇒ Descripción : Determina o indica el tamaño del buffer de recepción en bytes
- ⇒ Sintaxis : [form.]MScComm.InBufferSize[= numBytes%]
- ⇒ Tipo de dato : Integer
- ⇒ Comentarios : Esta propiedad se refiere al tamaño total del buffer de recepción. El tamaño de default es de 1024 bytes. Entre mayor sea el tamaño de este buffer, se deja menor memoria para la aplicación. Sin embargo, el contar con un buffer pequeño se corre el riesgo de tener overflow. Como una regla general se recomienda empezar con un buffer de 1024 bytes. Si se presenta el error de overflow, incremente el tamaño del buffer a la velocidad de transmisión de la aplicación.

- **OutBufferSize**

- ⇒ Descripción : Determina o indica el tamaño del buffer de transmisión en bytes.
- ⇒ Sintaxis : [form.]MScComm.OutBufferSize[= NumBytes%]
- ⇒ Tipo de dato : Integer
- ⇒ Comentarios : Esta propiedad se refiere al tamaño total del buffer de transmisión. El tamaño de default es de 512 bytes. Entre mayor sea el tamaño de este buffer, se deja menor memoria para la aplicación. Sin embargo, el contar con un buffer pequeño se corre el riesgo de tener overflow. Como una regla general se recomienda empezar con

un buffer de 512 bytes. Si se presenta el error de overflow, incrementar el tamaño del buffer a la velocidad de transmisión de la aplicación.

• **PortOpen**

- ⇒ Descripción : Determina o indica el estado del puerto de comunicación (abierto o cerrado). Esta propiedad no esta disponible durante el diseño.
- ⇒ Sintaxis : [form.]MSComm.PortOpen[= {True | False}]
- ⇒ Tipo de dato : Integer (boolean)
- ⇒ Comentarios : La siguiente tabla muestra los valores que puede tomar esta propiedad.

Valor	Descripción
True	El puerto de comunicación esta abierto
False	El puerto de comunicación esta cerrado

Al indicar un valor de True a la propiedad PortOpen se abre el puerto, y con el valor False se cierra el puerto limpiando los buffers de transmisión y recepción. Al terminar la aplicación, el puerto de comunicación cierra automáticamente el puerto serial.

Si la propiedad CommPort tiene un número de puerto inválido, el control de comunicaciones genera el error 68 (Dispositivo no disponible) si el dispositivo no existe cuando se intenta abrir con la propiedad PortOpen.

El puerto serial debe soportar también los valores de la propiedad Settings que se están definiendo, en caso contrario se pueden tener problemas de hardware y la aplicación no funcionará correctamente.

Así también, si alguna de las propiedades DTEnable y RTSEnable son cambiadas con un valor True antes de que sea abierto el puerto y cambian a False cuando se cierra el puerto. En otro caso, DTR y RTS permanecen en su estado previo.

5.2.3 Descripción del front-end (Interface gráfica con el usuario).

La herramienta que posee la característica de un ambiente de trabajo bajo Windows, y nos ofrece como producto final, sistemas con una interface amigable y además familiar para el usuario es la herramienta de desarrollo de Visual Basic.

Las principales ventajas que proporciona este software son :

- Contiene controles que facilitan el manejo de gráficos, interacción con el usuario, así como la comunicación externa por los puertos seriales.
- Posee su propio compilador para generar programas ejecutables, así como un debugger que facilita la detección de errores.
- Facilita la creación de programas para la instalación de la aplicación desarrollada.

5.2.3.1 Características de Microsoft Visual Basic 4.0 Profesional.

Los elementos básicos para el diseño de una aplicación gráfica en Visual Basic son :

- Proyecto
- Formas
- Módulos
- Lenguaje
- Menús
- Caja de herramientas

Proyecto

Un Proyecto es un grupo de formas, módulos y controles diseñados por el desarrollador, los cuales están relacionados internamente de acuerdo al objetivo de la aplicación.

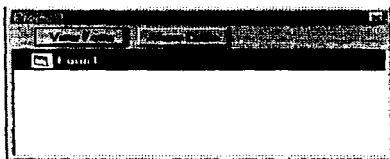


Figura 5.2.3.1 a Ventana de Proyecto.

Un proyecto está formado por los siguientes archivos:

Tipo de archivo	Extensión	Descripción
Forma	.FRM .FRX	Estos archivos incluyen la forma, los objetos en la forma y el código que corre cuando se ejecuta la forma.
Módulos estándar de Visual Basic	.BAS	Estos módulos contienen procedimientos Sub y Function que pueden ser llamados de cualquier forma u objeto.
Custom Controls	.OCX .VBX	Estos controles proporcionados por Microsoft en adición a los controles estándar o por desarrolladores.
Módulos Clase de Visual Basic	.CLS	Estos módulos (opcionales) contienen la definición de clase, métodos y propiedades.
Archivos de Recurso	.RES	Estos archivos contienen información binaria utilizada por la aplicación. Estos archivos son usualmente usados cuando se crean programas para múltiples lenguajes.

Forma

Una Forma contiene los controles y código asociados a ésta, sin embargo, éste código puede ser utilizado por cualquier otra forma cuando se incluye dentro de un módulo.

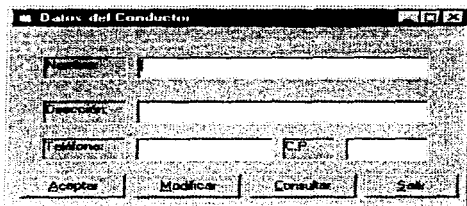


Figura 5.2.3.1 b Ejemplo de una forma y sus controles.

Las formas son creadas para proporcionar la interface de la aplicación con el usuario. Cada ventana es una forma que despliega controles, gráficas u otras formas. Estas formas pueden ser utilizadas de diferentes maneras :

- Para mostrar imágenes en una aplicación.
- Como una ventana de dialogo.
- Como un documento en una aplicación.

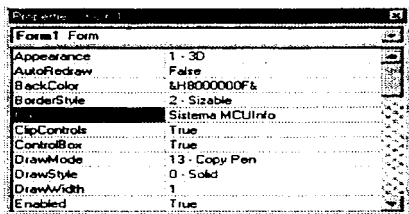


Figura 5.2.3.1.c Propiedades de una forma.

Los Controles son herramientas tales como cajas de texto, botones de opciones, etiquetas para proporcionar o pedir información al usuario. Conjuntamente son los que proporcionan la apariencia visual a la aplicación. Cada uno de estos elementos tiene propiedades específicas de tamaño, color, posición, nombre, etc.

Módulos

Un módulo únicamente contiene procedimientos, tipos y declaraciones de datos y definiciones.

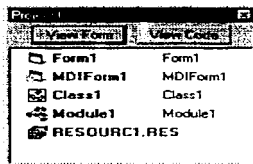


Figura 5.2.3.1.d Ventana de Proyecto, donde se muestra listado un módulo.

Lenguaje

Visual Basic incluye un Lenguaje de programación sencillo (easy to use) que es compatible con Qbasic y Basic Compiler 7.1, así como con otros productos de Microsoft. Las principales características de este lenguaje son :

- Maneja siete tipos de datos (Integer, Long Integer, Single, Double, Currency, String, Variant, User defined).
- Funciones matemáticas.
- Funciones para manejo de cadenas de caracteres (strings).
- Posee las estructuras IF...then...else, While, Case.
- Contiene un depurador (debugger) que chequea automáticamente la sintaxis, permite la ejecución de instrucción por instrucción, así como una ventana para verificar variables.

Menú

Los Menús son muy útiles para aplicaciones con muchas opciones y comandos, la utilización de menús facilitan el uso de la aplicación.

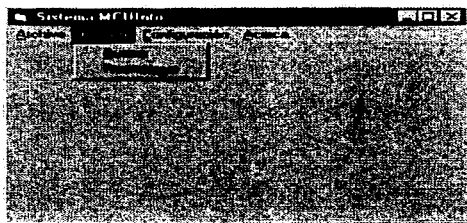


Figura 5.2.3.1 e Pantalla que muestra el despliegado de un menú.

Caja de herramientas

Contiene todos los objetos y controles que se pueden añadir a las formas para crear sus aplicaciones. Se pueden agregar controles a la Caja de herramientas, utilizando el comando Custom Controls en el menú de Herramientas.

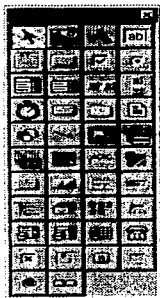


Figura 5.2.3.1.f Caja de Herramientas.

Descripción de cada control (de izquierda a derecha).

- **Pointer.** Se utiliza para cambiar el tamaño de un control, moverlo y seleccionar conjunto de controles.



- **Picture Box.** Despliega imágenes con formato BMP o PCX.



- **Label.** Sirve para colocar letreros que no requieran ser cambiados.



- **Text Box.** Se utiliza para desplegar o capturar texto.



- **Frame.** Se utiliza para crear gráficas o agrupar conjuntos de controles dentro de este control.



- **Command Button.** Crea un botón que controla alguna acción.



- **Check Box.** Se utiliza para crear cajas de selección de opciones, en la cual se necesita seleccionar múltiples opciones.



- **Option Button.** Se utiliza para crear grupos de opciones en la cual se necesita seleccionar una y solo una de ellas.



- **Combo Box.** Es una combinación de un "list box" y un "text box". El usuario puede realizar una selección de una lista o capturar un texto.



- **List Box.** Se utiliza para desplegar una lista de artículos en la cual el usuario puede seleccionar uno.



- **Horizontal Scroll Bar y Vertical Scroll Bar.** Se utilizan para navegar a través de una lista muy larga de información que no puede ser desplegada en una sola pantalla.



- **Timer.** Se utiliza en aplicaciones que requieran eventos de tiempo.



- **Drive List Box.** Se utiliza para desplegar las unidades de disco validas.



- **Directory List Box.** Despliega el directorio de alguna unidad seleccionada, incluyendo sus subdirectorios.



- **File List Box.** Despliega la lista de archivos que pueden ser cargados o salvados, de acuerdo a un conjunto de opciones.



- **Shape.** Se utiliza para dibujar rectángulos, triángulos o círculos.



- **Line.** Se utiliza para dibujar líneas con ciertas variedades de estilo.



- **Image.** Despliega imágenes adicionando cierto control sobre ella.



- **Data Control.** Permite el uso de bases de datos en la aplicación.



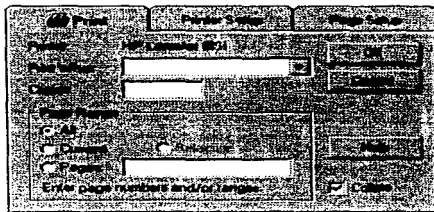
- **OLE Control.** Permite el uso de OLE en la aplicación.



- **Common Dialog.** Se usa para los diálogos en windows por ejemplo para seleccionar un font o cambiar colores.



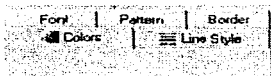
- **SSTab.** Se utiliza para presentar de una manera sencilla diálogos o pantallas de información en una sola forma, utilizando la misma interface vista en muchas aplicaciones de Microsoft.



- **Animatted Button Control.** Permite el control de iconos o gráficas para realizar un efecto de animación en un botón.



- **TabStrip.** Se usa como una división en un cuaderno o etiqueta de un grupo de archivos, folders.



- **Toolbar.** Contiene una coleccion de botones utilizados para crear una caja de herramientas que está asociada con una aplicación.



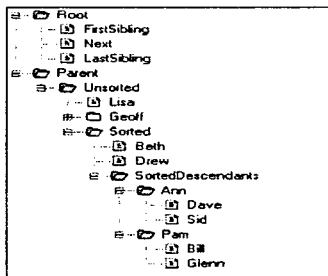
- **Statusbar.** Nos proporciona una ventana que generalmente se encuentra en la parte de abajo de una forma padre, a través de la cual se pueden desplegar varias clases del estado de los datos. El Statusbar puede ser dividido hasta un máximo de 16 paneles.



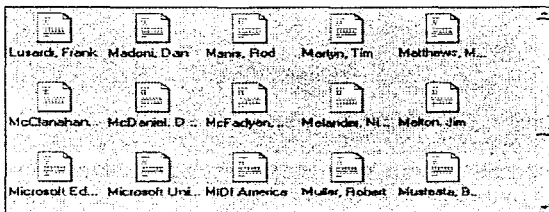
- **ProgressBar.** Muestra el progreso de una operación a través de un rectángulo que se va llenando conforme avanza el proceso.



- **TreeView.** Este control despliega una lista jerárquica de objetos "Nodo", donde cada uno de ellos consiste de una etiqueta y un bitmap opcional. Un Treeview es utilizado típicamente para desplegar los encabezados en los documentos, los artículos en un índice, los archivos y directorios en un disco o cualquier otra clase de información que es propicia de ser vista de forma jerárquica.



- **ListView.** Despliega artículos utilizando una de cuatro formas diferentes de vistas. Puede arreglar artículos dentro de columnas con o sin encabezados como también desplegar iconos acompañados de texto.



- **ImageList.** Contiene una colección de objetos ListImage, cada uno de los cuales puede ser referenciado por su índice o llave.



- **Slider.** Es una ventana que contiene un deslizador y marcas de los segundos (son opcionales).



- **DBList.** Llena automáticamente una list box de datos con un campo de un control de Datos y opcionalmente pasa un campo seleccionado a un segundo control de Datos.



- **DBCombo.** Llena automáticamente una list box de datos con un campo de un control de Datos y opcionalmente pasa un campo seleccionado a un segundo control de Datos. El campo

seleccionado es copiado a un text box combinado con el control que puede ser utilizado para editar el campo seleccionado.



- **DBGrid.** Despliega y habilita la manipulación de datos de una serie de renglones y columnas representando registros y campos de un objeto Recordset.



- **MSComm.** Proporciona comunicación serial para su aplicación, permitiendo la transmisión y recepción de datos a través del puerto serial.



- **SSCheck.** Permite alinear el texto en tercera dimensión de el check box hacia la derecha o hacia la izquierda.



- **SSFrame.** Determina si el frame aparece añadido o en relieve.



- **SSCommand.** Es un botón de comando en tercera dimensión. Puede desplegar una ilustración así como un bitmap o un icono. Un bisel variable permite que el botón aparezca en relieve en la pantalla.



- **SSPanel.** Se utiliza para desplegar un plano o texto en tercera dimensión en un fondo con la misma característica. También para agrupar otros controles en un fondo de tercera dimensión como una alternativa a un control de frame o el dar una apariencia de tercera dimensión a controles estándar como son: list box, combo box, scroll bar entre otros.



- **SSOption.** Permite alinear el texto en tercera dimensión de el option button hacia la derecha o hacia la izquierda.



- **SSRibbon.** Es un push button que puede ser utilizado en grupos para emular la funcionalidad de la barra de herramientas en una hoja de cálculo de Microsoft Excel. Este control tiene una propiedad para que un bitmap gráfico pueda ser añadido.



Pasos para crear una aplicación en Visual Basic.

El proceso para crear una aplicación en Visual Basic puede ser dividida en siete pasos:

1. Crear la interface de Usuario.

El crear una interface es dibujar controles y objetos en una forma. El orden para hacer más sencillo de leer y depurar el código, es asignar nombres a los objetos utilizando las convenciones estándar para nombrarlos.

2. Poner las propiedades a los objetos de la interface.

Después de añadir objetos a un forma, se colocan las propiedades de esos objetos. Se pueden inicializar valores usando la ventana de Propiedades en el tiempo de diseño o usar código para modificarlas en tiempo de ejecución.

3. Escribir el código para eventos.

Después de colocar las propiedades iniciales a la forma y cada objeto, se añade el código que ejecutará un determinado evento. Los eventos ocurren cuando diferentes acciones suceden en un control o un objeto. Por ejemplo, Click es un evento que puede ocurrir para un botón de Comando.

4. Salvar el proyecto.

Cuando se crea un proyecto, hay que asegurarse de darle un nombre y después guardando con el comando de Salvar Proyecto Como del menú Archivo.

5. Probar y depurar la aplicación.

Cuando añadimos código a el proyecto, podemos usar la herramienta Run en la barra de herramientas para correr nuestra aplicación y ver su desempeño. Se puede usar también la herramienta de depurar para verificar errores y modificar el código.

6. Hacer una archivo ejecutable.

Una vez que se ha completado el proyecto, crear su ejecutable por medio del comando Hacer Archivo EXE en menú de Archivo.

7. Crear un setup a la aplicación.

Esto se realiza debido a que el archivo ejecutable es dependiente de otros archivos, como es una dll (vb40032.dll), controles, archivos OCX que son utilizados en el proyecto, por lo que se debe crear un programa de setup usando el Asistente de Setup que le proporcionará al programa todos los archivos que serán requeridos.

Nuevas características de Visual Basic 4.0

- **Colecciones y reutilización de objetos.** Se pueden crear objetos reutilizables con sus propias características y métodos y ensamblarlos en un modelo objeto. Estos objetos son llamados

Clases, y están contenidos en un nuevo módulo de Visual Basic llamado Módulo de Clases. Y se pueden crear sus propias colecciones por medio de un objeto llamado Colección.

- **Jet 2.5 y el 3.0** El Jet engine 2.5 (16 bits) y el jet 3.0 (32 bits). Estos motores de bases de datos proporcionan varias características, incluyendo un nuevo modelo de programación de Objetos de Acceso de Datos (DAO). El Jet 2.5 y 3.0 son funcionalmente compatibles, pues ambos pueden crear aplicaciones de bases de datos desde el mismo código, simplificando el intercambio y la recompilación.
- **Nueva extensión para el archivo de Proyecto.** Para distinguir los archivos de proyecto de Visual Basic de otras herramientas de desarrollo, se cambió la extensión de .MAK a .VBP.
- **Controles para Windows 95.** Estos controles (Custom Controls) nos permiten crear aplicaciones con la interface de Windows 95 de **look and feel** (ver y sentir). Por ejemplo, las aplicaciones pueden incorporar vistas de listas, árboles, una barra de estado, barras de herramientas y cajas de diálogo con etiquetas. El control RichTextBox permite al usuario la entrada y edición de texto mientras también proporciona un avanzado formato con características que sobrepasan al control TextBox convencional, pues posee la capacidad de **drag and drop** (arrastrar y dejar). El uso del Control Slider (deslizador) que permite a los usuarios una nueva forma de entrada de datos.
- **Edición Empresa.** Tiene nuevas e interesantes características como las siguientes :
 - ⇒ Construcción de aplicaciones cliente/servidor para la administración de información crítica del negocio.
 - ⇒ Automatización remota, por ejemplo, permite crear componentes reutilizables y compartidos que usan la interface de automatización de OLE para la comunicación de red.

- ⇒ Cuenta con un juego herramientas diseñadas específicamente para el desarrollo cliente/servidor, incluyendo utilerías para catálogo, instalación y configuración de componentes y la administración del acceso de código fuente.

5.2.4 Diseño y Construcción de la Base de Datos.

Un sistema de Base de Datos es un sistema computarizado de información para el manejo de datos por medio de paquetes de software llamados Sistemas de Manejo de Bases de Datos (DBMS), los tres componentes principales de un sistema de Base de datos : el hardware, el software DBMS y los datos por manejar.

Una Base de Datos es un conjunto de información que tiene relación entre sí y es lo suficientemente amplia para satisfacer las necesidades de los usuarios de dicha información.

El usuario podrá recobrar datos de varias partes de la Base de Datos ya que los archivos ahí almacenados están conectados directa o indirectamente.

El proceso de diseño involucra los tres procesos siguientes :

1. Agrupamiento de los datos globales de una Base de Datos en uno o más registros de tipo conceptual (Entidades).
2. Definición de las relaciones conectando entidades relacionadas en un modelo conceptual de Base de Datos.
3. Transformación del modelo conceptual de la Base de Datos en un esquema mejorado que se puede describir mediante un DBMS específico para la implantación de la Base de Datos.

5.2.4.1 Diccionario de datos.

El diccionario de datos es un listado organizado de todos los datos pertinentes al Sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas y componentes de almacenamiento.

El empleo del Diccionario de Datos :

- Reduce ambigüedades
- Especifica detalles

La siguiente figura muestra las tablas con sus campos, (así como sus características) que conforman la Base de Datos del sistema.

Tabla de Catálogo de líneas(autobuses).

Campo	Tipo	Longitud	Llave	Descripción
Linea_autobus_Id	Contador	8	Si	Clave de la línea de autobuses
Nombre_linea	Texto	50	No	Nombre de la línea de autobuses

Tabla de Código de error.

Campo	Tipo	Longitud	Llave	Descripción
Código_Error_Id	Numerico	Double	Si	Clave del código de error
Código_Error_Dsc	Texto	18	No	Descripción del código de error

Tabla de Tipo de gasolina.

Campo	Tipo	Longitud	Llave	Descripción
Gasolina_Tipo_Id	Contador	8	Si	Clave del tipo de gasolina
Gasolina_Tipo_Dsc	Texto	18	No	Descripción del tipo de gasolina

Tabla de Lectura.

Campo	Tipo	Longitud	Llave	Descripción
Sensado_Por_Periodo_Id	Número	Entero largo	Primaria	Conteo del número de sensores
Vehiculo_Placa	Número	Entero largo	Si	Placa del vehículo
Código_Error_Id	Número	Double	Si	Clave del código de error
Sensado_Por_Periodo_Di stancia	Número	Double	No	Sensado por periodo de distancia
Sensado_Por_Periodo_V elocidad	Número	Double	No	Sensado por periodo de velocidad
Sensado_Por_Periodo_Ti empo	Número	Double	No	Sensado por periodo de tiempo

Tabla de Marca.

Campo	Tipo	Longitud	Llave	Descripción
Marca_Id	Contador	-	Si	Clave de la marca del autobús
Linea_autobus_Id	Contador	8	Foránea	Clave de la línea de autobuses
Marca_Dsc	Texto	18	No	Descripción de la marca de autobús
Modelo_Dsc	Texto	25	No	Descripción del modelo
Modelo_Año	Número	Double	No	Año del modelo de autobús

Tabla del Operador.

Campo	Tipo	Longitud	Llave	Descripción
Operador_Id	Contador	-	Si	Clave de la marca del autobús
Vehiculo_Placa	Número	Double	No	Placas del autobús
Operador_Nombre	Char	20	No	Nombre del operador
Operador_Apellido_Paterno	Char	20	No	Apellido paterno del operador
Operador_Apellido_Materno	Char	20	No	Apellido materno del operador
Operador_Direccion	Char	40	No	Dirección del operador
Operador_Telefono	Número	10	No	Teléfono del operador
Operador_RFC	Char	10	No	RFC del operador
Operador_No_Licencia	Número	10	No	Número de licencia del operador
Operador_Sexo	Texto	1	No	Sexo del operador
Linea_autobus_Id	Contador	8	Si	Clave de la línea de autobuses

Tabla de Sensado por parada.

Campo	Tipo	Longitud	Llave	Descripción
Sensado_Por_Parada_Id	Númeroico	Entero largo	Primaria	Conteo por paradas
Vehiculo_Placa	Numerico	Double	Foranea	Placas del autobus
Codigo_Error_Id	Númeroico	Double	Si	Clave del codigo de error
Hora_de_Arribo	Númeroico	Double	No	Hora de arribo del autobus
Hora_de_Partida	Numerico	Double	No	Hora de partida del autobus
Combustible_de_Arribo	Numerico	Double	No	Contenido de combustible con el que arriba el autobús

5.2.4.2 Diagrama entidad relación.

El diagrama de entidad-relación de la base de datos para la computadora de viaje es el siguiente :

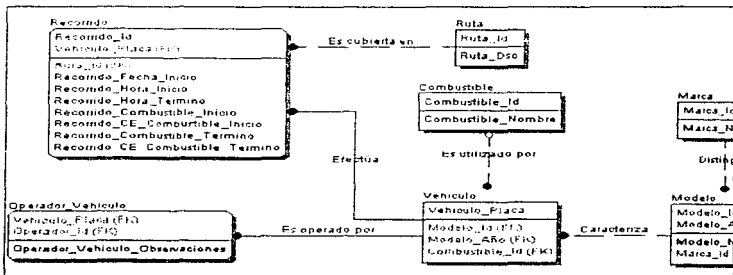


Figura 5.2.4.2. Diagrama entidad-relación

5.2.4.3 Desarrollo de la Base de Datos.

La implantación de las tablas que forman la base de datos que se creó en Microsoft Access se llevó de la siguiente manera :

- A. Crear una base de datos en Microsoft Access
- B. Crear las tablas que componen a la base de datos
- C. Hacer las relaciones necesarias

Primeramente al entrar a Access aparece un menú como el que se muestra en la figura 5.2.4.3.a, seleccionamos la opción de Archivos y elegimos crear una nueva base de datos.



Figura 5.2.4.3.a Menú inicial de Microsoft Access

Esta opción nos va a pedir el nombre que le vamos a dar a la base de datos, el cual puede ser de 8 caracteres como máximo. Una vez éste, damos aceptar. (Observar la figura 5.2.4.3.b).

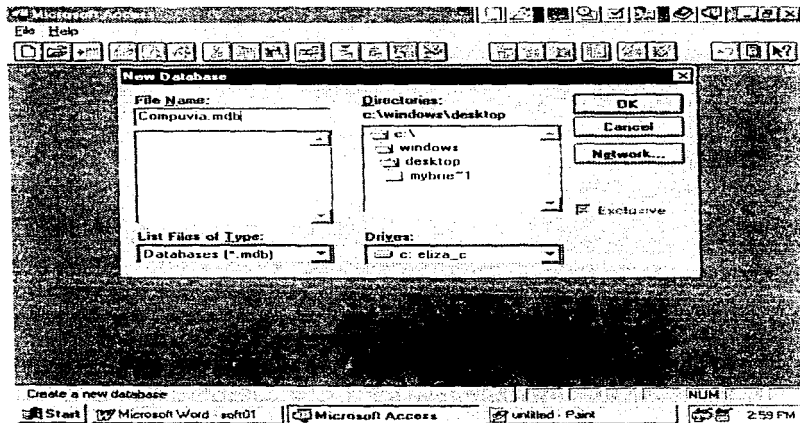


Figura 5.2.4.3 b Generación de una nueva Base de Datos

En la figura 5.2.4.3.c se muestra la ventana de la Base de Datos, con los respectivos elementos que la forman :

Microsoft Access nos presenta una ventana en la que nos muestra diferentes opciones : tablas, consultas, formas, reportes macros y módulos. Todas las opciones nos permiten crear un nuevo objeto o abrir uno ya existente.

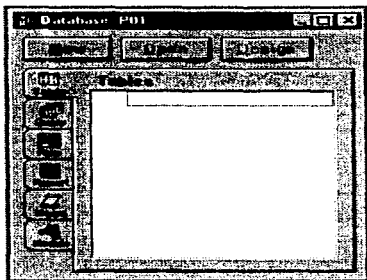


Figura 5.2.4.3.c Ventana de Base de Datos

Para utilizar una de las opciones de Microsoft Access se debe :

1. Elegir la opción deseada.
2. Seleccionar uno de los tres botones siguientes :



para generar un nuevo objeto de la opción seleccionada



para ver la información de un objeto



para editar el objeto

B. Creación de la Tabla.

Para poder crear la tabla, seleccionamos la opción de tablas y damos clic en la opción de nueva opción, como las demás, nos permite elegir si deseamos que el asistente nos auxilie, para crear el objeto con cierta estructura o no. Al seleccionar la opción de tabla en blanco, nos aparece la figura 5.2.4.3.d.

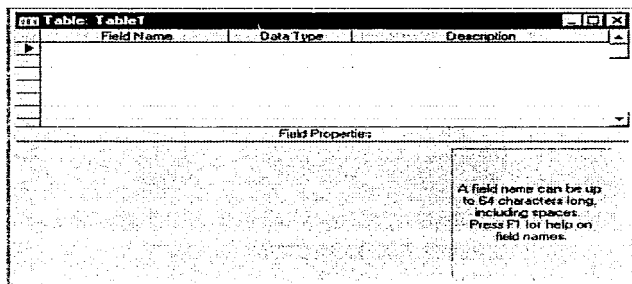


Figura 5.2.4.3 d Diseño de una tabla

En la parte superior se tiene la sección de declaración de campos que esta compuesta por tres columnas, la primera para darle un nombre al campo, la segunda para darle el tipo de datos que deseamos y la tercera para dar el descriptivo de lo que se va a introducir en el campo.

En la parte inferior tenemos la sección de propiedades de los campos que nos permite adecuar las propiedades a nuestras necesidades.

Al entrar en esta opción también contamos con una barra de herramientas de la cual, utilizaremos principalmente el icono de especificar la llave primaria que debemos presionar después de seleccionar el o los campos que deseamos sea la llave de nuestra tabla. Asimismo, vamos a utilizar el icono de guardar el objeto activo, con el cual salvaremos con nombre nuestra tabla o bien las modificaciones realizadas a una ya existente.

La figura 5.2.4.3.e muestra como creamos la tabla de líneas que forma parte de nuestro sistema. En esta tabla se muestran los campos que la forman con sus respectivas características.

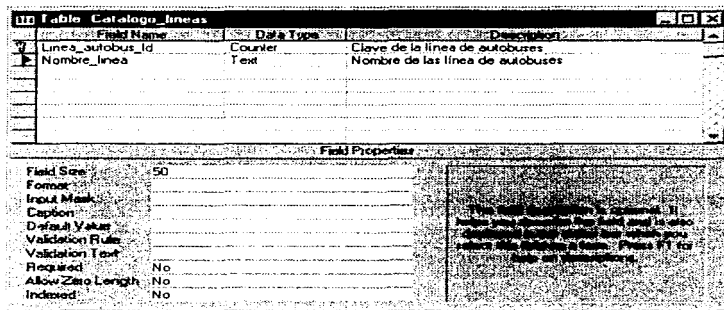


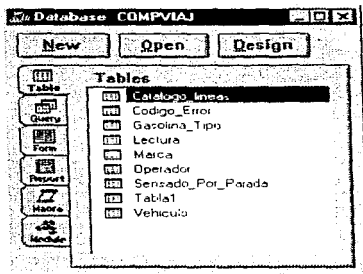
Figura 5.2.4.3.e Tabla de Catálogo de líneas de la base de datos.

Una vez introducidos los campos de la tabla y validados todos los diferentes tipos de datos, se procede a cerrar la tabla en la cual aparece la pantalla de la figura 5.2.4.3.f en donde se proporciona el nombre con el cual se quiere guardar.



Figura 5.2.4.3.f Ventana que muestra Access al salvar una tabla.

Una vez explicada la manera en como se creó la tabla de Catalogo_lineas se procedió a realizar de manera similar las otras tablas que conforman a la base de datos, en la figura 5.2.4.3.g se muestra el nombre de todas las tablas creadas.



5.2.4.3.g Ventana que muestra todas las tablas de la Base de Datos.

C. Creación de las relaciones necesarias

En el menú de Microsoft Access se tiene la opción de relaciones, al presionar este comando nos aparece en la pantalla que se muestra en figura 5.2.3.3.h, en el cual existe la posibilidad de ir agregando las tablas que sean necesarias para determinar las relaciones que deseamos.

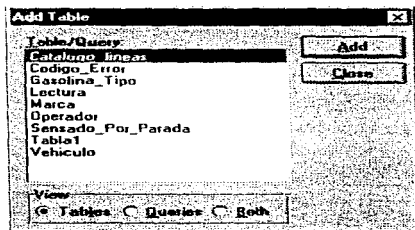
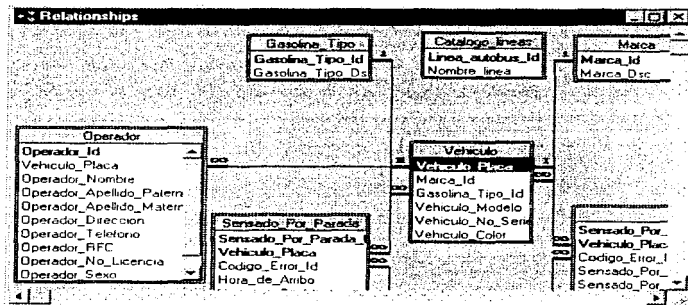


Figura 5.2.4.3.h Ventana con la que se agregan las tablas.

Las tablas relacionadas se representan mediante una línea que conecta los campos coincidentes. Microsoft Access utiliza las relaciones para crear automáticamente combinaciones en las nuevas consultas y para relacionar los registros mostrados en los subformularios. Las tablas deben estar relacionadas por campos equivalentes (llaves). No es necesario que los campos equivalentes tengan el mismo nombre, pero si deben contener el mismo tipo de datos.



y así consecutivamente obtenemos:



5.2.4.3 i Construcción de las relaciones de las bases de datos.

Normalmente, las tablas se relacionan vinculando el campo de clave principal de una tabla con un campo coincidente de otra tabla. Las relaciones de las tablas de nuestra base de datos se muestra en la figura 5.2.4.3 i.

5.2.5 Diseño y construcción del Front-End

Como ya se ha mencionado el Front-End para el "Diseño de una computadora de viaje para camiones, autobuses y vehículos de mensajería", se desarrolló en Visual Basic v. 4.0. y es precisamente en este tema donde se realizará y explicará el procedimiento de desarrollo del mismo.

5.2.5.1 Diseño del Front-End del Sistema

Partiendo de que debe ser visualmente agradable y de fácil manejo para el usuario, se determinó desarrollar un diseño modular en el que la presentación de los datos quedara agrupada conforme a características similares para su rápido acceso.

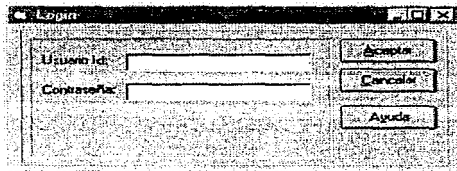
Por otra parte, es en esta etapa donde se realiza un análisis de color, para determinar las propiedades de todos los controles involucrados en la pantalla y de la forma misma; concluyendo que los colores más adecuados para la presentación del sistema son: la combinación de grises, blancos, pasteles y los colores primarios para el despliegue de gráficos, esto conforme a los estándares de desarrollo de Microsoft.

De lo antes mencionado, se procede a la implantación de las pantallas.

5.2.5.1.1 Pantalla de login del sistema

Para la construcción de la pantalla se seleccionó un control de forma de la caja de herramientas, la dimensión de ésta se hizo automáticamente con el arrastre del mouse y se definieron las siguientes propiedades :

- BackColor = &H8000000F&
- BorderStyle = Fixed Single
- Caption = Login
- ClipControls = True
- Font = MS Sans Serif
- MinButton = False
- MaxButton = False
- WindowState = 0 Normal



5.2.5.1.1 Pantalla de Login.

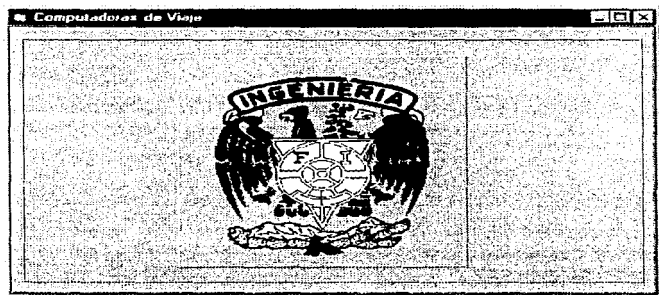
Después se seleccionó y arrastró un Caja de Texto con máscara o en su defecto una Caja de Texto y definió sus propiedades :

- PasswordChar = ""
- BackColor = &H80000005&

Por último se seleccionó un control de etiqueta de tercera dimensión (Panel) y se definió como :

- Visible = True
- ForeColor = &H80000012&
- Autosize = True

5.2.5.1.2 Pantallas de Presentación.



5 2 5 1 2.a Pantallas de Presentación

Para el diseño de estas pantallas solo se necesitaron 3 objetos : la misma forma, panel de 3D, un timer y un cuadro de imagen ; estos objetos se toman del cuadro de herramientas.

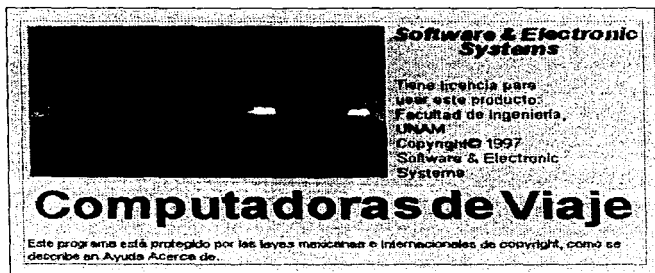
- **Cuadro de imagen.** Puede ser un icono u otro elemento gráfico, como un bitmap o metafile creado en otra aplicación. En nuestro sistema se cargo el archivo escudo.bmp que contiene el escudo de la UNAM en tercera dimensión.
- **Timer.** Controla el tiempo que permanece desplegada la pantalla, después de un tiempo determinado desaparece.
- **Panel de 3D.** Este control permite dar una sensación de realce a la pantalla, sus distintas propiedades se explican adelante.

En Visual Basic, cada objeto tiene propiedades, es decir, a cada objeto le damos una serie de características como son : color, tamaño, leyenda, etc.; para proporcionarles éstas, solo presionamos F4 previamente seleccionando el objeto, y así escogemos las características deseadas del cuadro de propiedades.

OBJETO	PROPIEDAD	VALOR
Forma 1	Name WindowState	Presentación 2
Panel	Name	Panel3D
Timer	Time	2000
Caja Imagen	Name Picture	Picture 1 Escudo.bmp

Figura 5.2.5.1.2.b Propiedades de los objetos de la forma 1.

5.2.5.1.3 Pantalla de inicio del sistema.



Al inicio de la ejecución del programa, se despliega la pantalla con el nombre del sistema y los derechos reservados con los que cuenta. Dicho formato se diseñó de tal manera, pues se trata de dar un enfoque de acuerdo a las normas y estándares que rigen el desarrollo de sistemas de software en la actualidad.

Para el desarrollo de esta pantalla se usaron los controles de panel y frame en 3D e inserción de un bitmap, así como el control de etiquetas.

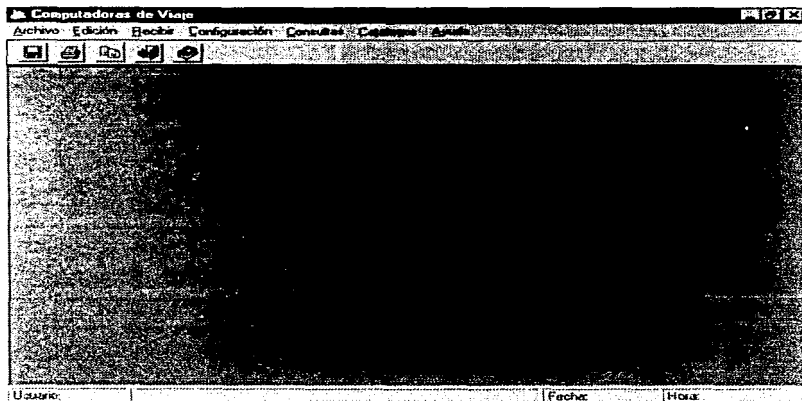
Las propiedades del Panel son las siguientes :

- Autotamaño : Panel hijo
- Alineación : Centrada
- Tipo de letra : MS-Sans Serif

El control de Frame de 3D, tiene las siguientes propiedades :

- Alineación : Justificada a la izquierda
- Estilo de sombra : En relieve
- Tipo de letra : MS-Sans Serif
- Picture : Bitmap

5.2.5.1.4 Pantalla del Menú Principal



En esta pantalla se engloban todos los submódulos en los que está construido el sistema.

Se conforma de una MIDI, con diferentes menús y botones que hacen referencia a cada uno de estas opciones (menús). Esto es, con la finalidad de hacer más rápido el acceso a las distintas partes del sistema. La función que realiza cada uno de estas opciones es la siguiente :

A la posición que ocupan estos botones dentro de la pantalla se le llama Barra de Herramientas. Note que en la parte superior izquierda de la pantalla se tiene un letrero que dice Archivo, éste forma parte de lo que se conoce como Barra de Menúes, que es donde se definen las opciones alternas o adicionales para éste tipo de aplicaciones.

Para accionar la opción llamada Archivo, basta con hacer click sobre este u oprimir las teclas [Alt][a] y entonces se despliega un pequeño menú (este tipo de menús se les conoce como de Pull-Down), en el cual se tiene las mismas opciones de los botones que se presentan en la barra de herramientas.

Como se mencionó anteriormente, la forma que se utilizó para actuar como fondo de la aplicación es la llamada MDI_principal, la cual contiene también otros elementos que son los siguientes:

- **A**rchivo

Objeto : Menú

Nombre : archivo

Caption : &Archivo

Enabled : True

Descripción : A este elemento se le llama control de menú y como su nombre lo indica se utiliza para crear Menúes. En este caso forma parte del menú principal del sistema. En este tipo de controles se puede utilizar el símbolo "&" antes de la letra que se utiliza para seleccionar la opción.

⇒ **A**brir

Objeto : Menú

Nombre : abrir

Caption : Abrir

Enabled : True

Descripción : Forma parte del menú de Archivo del menú principal del sistema.

⇒ Guardar

Objeto : Menú

Nombre : guardar

Caption : Guardar

Enabled : True

Descripción : Forma parte del menú de Archivo del menú principal del sistema.

• Edición

Objeto : Menú

Nombre : edición

Caption : &Edición

Enabled : True

Descripción : Es otro control de menú y forma parte del menú principal del sistema.

⇒ Buscar

Objeto : Menú

Nombre : buscar

Caption : Buscar

Enabled : True

Descripción : Es submenú de la opción de Edición.

• Recibir

Objeto : Menú

Nombre : recibir

Caption : &Recibir

Enabled : True

Descripción : Es otro control de menú y forma parte del menú principal del sistema. Aquí es donde se establece la comunicación de la PC con el microcontrolador.

Transferencia de Datos

De manera sencilla, el proceso efectuado para transferir datos del sistema electrónico al sistema de software, es el siguiente:

En primer lugar, el microcontrolador envía una serie de datos a la computadora via el puerto serial. El puerto serial seleccionado es abierto desde el programa. En caso de que el puerto serial no este abierto, o no ha sido posible abrirlo, el sistema generará un mensaje de error indicando el problema.

Los datos enviados por el microcontrolador se encuentran contenidos dentro de una cadena, es decir todos los datos serán enviados de una vez a la PC.

Mediante instrucciones de Visual Basic, es leída y almacenada esta cadena en un arreglo de longitud suficiente para contener esta. Enseguida esta cadena debe ser separada en cada uno de los componentes elementales que la conforman (datos).

A continuación cada uno de estos datos debe ser decodificado a su valor de código de carácter, puesto que estos son recibidos por la PC, en forma de caracteres ASCII.

Cada dato en forma de código ASCII, debe ser ahora convertido a su correspondiente valor en hexadecimal, que el microcontrolador en realidad envía los datos en código hexadecimal. De esta manera se garantiza que cada dato enviado por el sistema electrónico, es recibido por la PC de manera correcta. Por último lo único que resta hacer con cada uno de los datos en forma hexadecimal, es interpretarlos para poder efectuar con ellos cualquier proceso dentro del sistema de software.

- **Consultas**

Objeto : Menú

Nombre : consulta

Caption : &Consultas

Enabled : True

Descripción : Es otro control de menú y forma parte del menú principal del sistema. Se diseñó el conjunto de consultas en una sola pantalla, distribuyéndolas por medio del control SSTab. La siguiente figura muestra la pantalla de consultas.

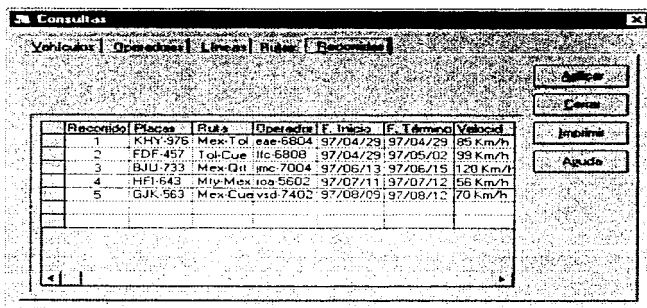


Figura 5.2.5.1.4 a Pantalla de Consultas

Esta ventana como ya se mencionó consta principalmente del control SSTab, de paneles en 3D, CommandButton y DBGrid. Las propiedades de los controles utilizados para el diseño de esta ventana son las siguientes :

⇒ Forma

Objeto : Forma

Nombre : frmConsulta

Caption : Consultas

Enabled : True

⇒ SSTab

Objeto : SSTab

TabCount : 4

TabCaption : 0 &Vehiculos

1 &Operadores

2 &Lineas

3 &Rutas

4 &Recorridos

Fonts : MS Sans Serif

⇒ Paneles

Objeto : SSPanel

Alineación : Centrado

Enabled : True

Font : MS Sans Serif

⇒ CommandButtons

Objeto : CommandButton

Nombre : cmdAplicar, cmdCerrar, cmdImprimir, cmdAyuda

Caption : &Aplicar, &Cerrar, &Imprimir y &Ayuda

Enabled : True

⇒ DBGrid

Objeto : DBGrid

Caption : ""

Scrollbar : Automático

Columnas : 0 Placas

1 Marca

2 Modelo

3 Descripción

4 No. de Serie

- Catálogos

Objeto : Menú

Nombre : catálogos

Caption : &Catálogo

Enabled : True

Descripción : Es un control de menú y forma parte del menú principal. Al hacer click en esta opción se despliega la siguiente pantalla :

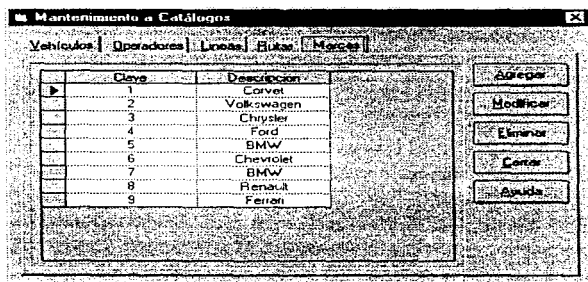


Figura 5.2.1.4.b Pantalla de Catálogos

Esta ventana sigue la misma pauta de diseño que la de Consultas y para su construcción se utilizaron los siguientes controles :

⇒ Forma

Objeto : Forma

Nombre : frmCatálogo

Caption : Mantenimiento de Catálogos

Enabled : True

⇒ SSTab

Objeto : SSTab

TabCount : 5

TabCaption : 0 &Vehiculos

1 &Operadores

2 &Lineas

3 &Rutas

4 &Marcas

Fonts : MS Sans Serif

⇒ Paneles

Objeto : SSPanel

Alineación : Centrado

Enabled : True

Font : MS Sans Serif

⇒ CommandButtons

Objeto : CommandButton

Nombre : cmdAgregar, cmdModificar, cmdEliminar, cmdCerrar, cmdAyuda

Caption : &Agregar, &Modificar, &Eliminar &Cerrar y &Ayuda

Enabled : True

⇒ DBGrid

Objeto : DBGrid

Caption : ""

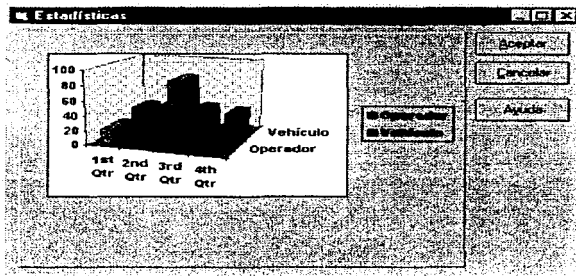
Scrollbar : Automático

Columnas : 0 Placas

- 1 Marca
- 2 Modelo
- 3 Descripción
- 4 No. de Serie

• Estadísticas

Otra ventana que nos proporciona información a través de gráficas es la opción de Estadísticas, la cual nos da una idea del comportamiento del vehículo con sólo ver los datos que arroja.



⇒ Paneles

Objeto : SSPanel

Alineación : Centrado

Enabled : True

Font : MS Sans Serif

⇒ CommandButtons

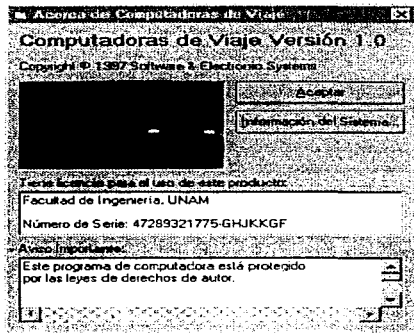
Objeto : CommandButton

Nombre : cmdAceptar, cmdCancelar, cmdAyuda

Caption : &Agregar, &Modificar, &Eliminar &Cerrar y &Ayuda

Enabled : True

• Ayuda



En esta opción del sistema contamos con la pantalla de "Acerca de", la cual no muestra datos del sistema como son sus licencias, la versión y por quién fue desarrollado.

⇒ **Forma**

Objeto : Forma

Nombre : frmAcerca

Caption : Acerca de Computadoras de Viaje

Enabled : True

WindowState : 0 Normal

⇒ **Etiqueta**

Objeto : Etiqueta

Nombre : lblSistema

Caption : Computadora de viaje

Enabled : True

FontName : MS Sans Serif

⇒ **Botón de comandos**

Objeto : CommandButton

Nombre : cmdAceptar

Caption : Aceptar

Enabled : True

FontName : MS Sans Serif

⇒ **Botón de comandos**

Objeto : CommandButton

Nombre : cmdInformación

Caption : Información del Sistema

Enabled : True

FontName : MS Sans Serif

- **Configuración**

En esta pantalla se despliegan los parámetros de comunicación, se tiene la opción de cambiar alguno de estos parámetros, para lo cual es necesario accionar el botón que los define, entonces el sistema despliega la ventana mostrada :

⇒ Forma

Objeto : Forma

Nombre : frmConfiguración

Caption : Configuración de Parámetros de Configuración

Enabled : True

FontName : MS Sans Serif

⇒ Frame

Objeto : Frame

Caption : &Velocidad (Bauds)

Enabled : True

FontSize : 8.25

FontName : MS Sans Serif

⇒ **Frame**

Objeto : Frame

Caption : &Bits de datos

Enabled : True

FontSize : 8.25

FontName : MS Sans Serif

⇒ **Frame**

Objeto : Frame

Caption : &Bits de Parada

Enabled : True

FontSize : 8.25

FontName : MS Sans Serif

⇒ **Frame**

Objeto : Frame

Caption : &Paridad

Enabled : True

FontSize : 8.25

FontName : MS Sans Serif

⇒ **Frame**

Objeto : Frame

Caption : &Puerto COM

Enabled : True

FontSize : 8.25

FontName : MS Sans Serif

En el apéndice A se presenta el código complemento del software del sistema.

5.3 Areas de aplicación

Como se mencionó anteriormente las computadoras de viaje cada vez van aumentando su campo de aplicación dentro de los diversos medios de transporte de pasajeros y mercancías.

Existen dentro del mercado una gran variedad de computadoras de viaje que pueden ser montadas en el automóvil o vienen ya incorporadas desde el momento de fabricación.

Uno de los objetivos principales del funcionamiento de esta computadora, es cubrir como mínimo los requerimientos del mercado actual, utilizando además una base de datos relacional para recopilar información de los recorridos y generar estadísticas de los mismos.

Este proyecto puede ser aplicado con gran éxito en empresas que manejan una cantidad considerable de camiones de pasajeros, camiones de carga o vehículos de mensajería, permitiendo a los empresarios llevar un control administrativo más eficiente.

CONCLUSIONES

La solución de cualquier problema de Ingeniería, precisa conocer a fondo todos los aspectos involucrados para obtener la solución más óptima. En nuestro caso, fue indispensable realizar una ardua tarea de investigación de las necesidades y la forma de operación de las empresas del transporte en México, así como de los últimos avances tecnológicos en la industria automotriz.

En el mercado actual ya existen las computadoras de viaje aunque su uso se extiende más en los automóviles particulares, y si bien es cierto que en la industria del transporte se han realizado algunas innovaciones, estas aún no cubren totalmente las expectativas.

La reducción de los costos de operación, el aumento de la seguridad y la optimización de los tiempos de recorrido, son algunas de las necesidades de la industria del transporte actual. Es por ello que nuestro diseño se enfocó en un aumento de las capacidades de las computadoras de viaje actuales, incorporando el uso de una interfaz gráfica con el usuario, parte innovadora del proyecto.

En la actualidad el sensado de los datos es reflejado en un disco de cartón (tacógrafo) que es rotulado por diagramas que indican el comportamiento del vehículo, con lo que la explotación de dicha información es limitada, por lo cual se decidió el uso de herramientas de cómputo que permiten una relación más amigable con el usuario, además de almacenar el historial del comportamiento de cada vehículo y así poder disponer de información confiable en todo momento y tener la posibilidad de realizar proyecciones de recursos, obteniendo una visión global del funcionamiento de la empresa para llevar a cabo una administración más eficiente.

En el sistema propuesto se contemplaron sólo las variables más importantes que intervienen en el recorrido de un vehículo, sin embargo, la versatilidad del dispositivo facilita su adaptación para el manejo de un gran número de condiciones o variables físicas, ello

dependerá del grado de exactitud y seguridad que la empresa requiera. En todo momento se emplearon dispositivos de bajo costo y de fácil adquisición obteniendo un diseño confiable, rentable y de alta calidad.

Todavía no hemos llegado a formarnos una idea clara de las capacidades y aplicaciones de este dispositivo en el futuro, con suerte nuestra investigación servirá como un firme fundamento para estudios mas avanzados en esta importante área.

BIBLIOGRAFIA

- BLACK UYLESS, Computer Networks. Protocols, Standards and Interfaces, Prentice-Hall, 1987.
- CREUS SOLE ANTONIO, Instrumentación Industrial, Publicaciones Marcombo, México-Barcelona, 1981.
- DE CASTRO MIGUEL, Inyección de Gasolina, Ediciones Ceac, Barcelona, 1989.
- DE CASTRO MIGUEL, Ordenadores de abordó, Ediciones Ceac, Barcelona 1990.
- DEPARTAMENTO DEL DISTRITO FEDERAL, Transportes-México-Historia, Anuario, México 1992.
- DOEBELIN ERNEST O. Diseño y Aplicación de Sistemas de Medición, Diana, México, 1980.
- GUILLEN STEFANO, Dispositivos Electrónicos en el automóvil, Ediciones Ceac, Barcelona España, 1989.
- HAYES P. JOHN, Digital System Design and Microprocessors, McGraw-Hill, Michigan U.S.A., 1988.
- HOLMAN P. JACK, Métodos Experimentales para Ingenieros, McGraw-Hill, 1990.
- INTERNATIONAL CONGRESS AND EXPOSITION, Engine and Driveline Control Systems, Detroit Michigan, Febrero / Marzo de 1988.
- KAUFMAN MILTON & SEIDMAN ARTHUR, Handbook of Electronics Calculations for Engineers and Technicians, McGraw-Hill, 1992.
- KUO, B.C. Automatic Control Systems, Prentice-Hall, New Jersey, 1975.
- MARROQUIN ZALETJA JAIME M., Estudios del derecho del Transporte, IPN-UPHCSA, México 1979.
- MICROSOFT, Visual Basic 4.0, Fundamentals & Programming Manuals, New Jersey USA, 1996.
- MICROSOFT, SQL Server 6.5 Implementation manual, New Jersey USA, 1996.

-
- **MOMPIN POBLET JOSE**, Transductores y medidores electrónicos, Marcombo, Barcelona-México, 1983.
 - **MORRIS MANO M**, Lógica Digital y Diseño de Computadores, Prentice-Hall, USA, 1982.
 - **MOTOROLA INC.**, MC68HC11 Reference Manual, Phoenix Arizona, 1991.
 - **NATIONAL SEMICONDUCTOR CO.**, Santa Clara CA, 1993.
 - **PALLETT**, Control Automático de vuelo, Magallanes, Madrid España, 1982.
 - **PALLETT**, Microelectrónica en sistemas de aeronaves, Magallanes, Madrid España, 1988.
 - **PURDY ELECTRONICS CO.**, AND Liquid Crystal Display Catalog , Sunnyvale , CA. 1996.
 - **SECRETARIA DE COMUNICACIONES Y TRANSPORTES**, Reglamento de Tránsito en Carreteras Federales, SCT, México 1990.
 - **TOCCI J. RONALD**, Digital Systems / Principles and Applications, Prentice Hall, USA 1985.
 - **VIESCAS L. JOHIN**, Guía completa de Access 2 para Windows, McGraw-Hill, Madrid España, 1994.

APENDICE A
CODIGOS DE PROGRAMACION

PROGRAMA DEL RECORRIDO EN EL MICROCONTROLADOR

M68HC11 Absolute Assembler Version 2.70C:promicte.ASC

```

1 A
2 A
3 A
4 A
5 A
6 A
7 A          0000          ORG          50000
8 A 0000 8E3000          LDS          #S3000
9 A
10 A
11 A
12 A
13 A
14 A
15 A
16 A
17 A
18 A 0003 BD02E0          JSR          BORMEM
19 A 0006 CE4000          INI          LDX          #S4000
20 A 0009 861F           LDAA         #S1F
21 A 000B 97D0           STAA        SDO
22 A 000D BD0273          JSR          DISLET
23 A 0010 C60A           LDAB        #S0A
24 A 0012 BD029B          JSR          TIEMPO
25 A 0015 BD0264          JSR          LIMDIS
26 A 0018 CE4020          LDX         #S4020
27 A 001B 860F           LDAA        #S0F
28 A 001D 97D0           STAA        SDO
29 A 001F BD0273          JSR          DISLET
30 A 0022 C60A           LDAB        #S0A
31 A 0024 BD029B          JSR          TIEMPO
32 A 0027 BD0264          JSR          LIMDIS
33 A 002A CE4030          LDX         #S4030
34 A 002D 861F           LDAA        #S1F
35 A 002F 97D0           STAA        SDO
36 A 0031 BD0273          JSR          DISLET
37 A 0034 C60A           LDAB        #S0A
38 A 0036 BD029B          JSR          TIEMPO
39 A
40 A
41 A
42 A
43 A
44 A
45 A 0039 BD0264          INIPA        JSR          LIMDIS
46 A 003C CE4050          LDX         #S4050
47 A 003F 860A           LDAA        #S0A

```

```

.....
* UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
* FACULTAD DE INGENIERIA
* PROGRAMA DE RECORRIDO PARA
* EL MICROCONTROLADOR MC68HC11
.....

```

```

.....
* RUTINA INICIAL :
* LIMPIA MEMORIA Y VARIABLES TEMPORALES
* DESPLIEGA MENSAJES DE INICIO DEL RECORRIDO
* INICIALIZA EL CONTADOR DE PULSOS Y EL
* CONVERTIDOR ANALOGICO-DIGITAL DEL HC11
.....

```

```

.....
* SOLICITA, RECIBE Y VALIDA EL
* PASSWORD DE ENTRADA AL SISTEMA
.....

```


48	A	0041	97D0		STAA	\$D0
49	A	0043	BD0273		JSR	DISLET
50	A	0046	BD02D7		JSR	DPAR
51	A	0049	8604		LDAA	#304
52	A	004B	97D0		STAA	\$D0
53	A	004D	CE00D1		LDX	#300D1
54	A	0050	BD0325	RECPASS	JSR	SENTEC
55	A	0053	2700		STAB	\$00,X
56	A	0055	862A		LDAA	#32A
57	A	0057	B78001		STAA	\$8001
58	A	005A	BD026D		JSR	VD
59	A	005D	C601		LDAB	#501
60	A	005F	BD029B		JSR	TIEMPO
61	A	0062	08		INX	
62	A	0063	7A00D0		DEC	\$00D0
63	A	0066	26E8		BNE	RECPASS
64	A	0068	C601		LDAB	#501
65	A	006A	BD029B		JSR	TIEMPO
66	A	006D	36D1		LDAA	SD1
67	A	006F	8136		CMPA	#336
68	A	0071	2615		BNE	MALPASS
69	A	0073	36D2		LDAA	SD2
70	A	0075	8138		CMPA	#338
71	A	0077	260F		BNE	MALPASS
72	A	0079	96D3		LDAA	SD3
73	A	007B	8131		CMPA	#331
74	A	007D	2609		BNE	MALPASS
75	A	007F	96D4		LDAA	SD4
76	A	0081	8131		CMPA	#331
77	A	0083	2603		BNE	MALPASS
78	A	0085	7E008E		JMP	MENU
79	A	0088	BD0264	MALPASS	JSR	LIMDIS
80	A	008B	7E028D		JMP	MP
81	A					
82	A					
83	A					
84	A					
85	A					
86	A					
87	A					
88	A					
89	A	008E	BD0264	MENU	JSR	LIMDIS
90	A	0091	CE407A		LDX	#3407A
91	A	0094	861F		LDAA	#31F
92	A	0096	97D0		STAA	\$D0
93	A	0098	BD0273		JSR	DISLET
94	A	009B	BD02D7	SMENU	JSR	DPAR
95	A	009E	BD0325		JSR	SENTEC
96	A	00A1	C131		CMPB	#331
97	A	00A3	270A		BEQ	FECHA
98	A	00A5	C132		CMPB	#332
99	A	00A7	2703		BEQ	TRANS
100	A	00A9	7E009B		JMP	SMENU
101	A	00AC	7E0413	TRANS	JMP	COMUNIC

RECPASS

MALPASS

 * DESPLIEGA EL MENU PRINCIPAL SOLICITANDO
 * LA TAREA A REALIZAR (REGISTRO DEL RECORRIDO
 * O TRANSFERENCIA DE INFORMACION) Y ENVIANDO EL
 * CONTROL A LA Rutina CORRESPONDIENTE.

MENU

SMENU

TRANS

```

102 A 00AF
103 A
104 A
105 A
106 A
107 A
108 A 00AF C601
109 A 00B1 BD029B
110 A 00B4 BD0264
111 A 00B7 CE409A
112 A 00BA 8607
113 A 00BC 97D0
114 A 00BE BD0273
115 A 00C1 BD02D7
116 A 00C4 8606
117 A 00C6 97D0
118 A 00C8 CE2100
119 A 00CB BD0325
120 A 00CE E700
121 A 00D0 F78001
122 A 00D3 BD026D
123 A 00D6 C501
124 A 00D8 BD029B
125 A 00DB 08
126 A 00DC 7A00D0
127 A 00DF 2706
128 A 00E1 BD02A7
129 A 00E4 7E00CB
130 A 00E7 BD03BB
131 A 00EA B62100
132 A 00ED 8100
133 A 00EF 27BE
134 A 00F1 8132
135 A 00F3 24BA
136 A 00F5 B62101
137 A 00F8 8100
138 A 00FA 27B3
139 A 00FC 3113
140 A 00FE 24AF
141 A 0100 BD03F6
142 A 0103 C131
143 A 0105 2703
144 A 0107 7E00AF
145 A 010A CEB600
146 A 010D 8603
147 A 010F B721D0
148 A 0112 BD02FA
149 A 0115 C601
150 A 0117 BD029B
151 A 011A BD0264
152 A 011D CE40A1
153 A 0120 8606
154 A 0122 97D0
155 A 0124 BD0273

```

* Rutina para recibir, validar y grabar la
* fecha, hora y codigos de operador y recorrido

FECHA	LDAB	#S01
	JSR	TIEMPO
	JSR	LIMDIS
	LDX	#S409A
	LDA	#S07
	STAA	SDO
	JSR	DISLET
	JSR	DPAR
	LDA	#S06
	STAA	SDO
	LDX	#S2100
REC FEC	JSR	SENTEC
	STAB	S00.X
	STAB	S001
	JSR	VD
	LDAB	#S01
	JSR	TIEMPO
	INX	
	DEC	S00D0
	BEQ	SF
	JSR	IMP
	JMP	REC FEC
SF	JSR	PDT
	LDA	S2100
	CMFA	#S00
	BEQ	FECHA
	CMFA	#S12
	BHS	FECHA
	LDA	S2101
	CMFA	#S0
	BEQ	FECHA
	CMFA	#S13
	BHS	FECHA
	JSR	DOK
	CMFB	#S31
	BEQ	GRAFEC
	JMP	FECHA
GRAFEC	LDX	#S8600
	LDA	#S03
	STAA	S21D0
	JSR	GRADAT
HORA	LDAB	#S01
	JSR	TIEMPO
	JSR	LIMDIS
	LDX	#S40A1
	LDA	#S06
	STAA	SDO
	JSR	DISLET

156 A 0127 BD02D7		JSR	DPAR
157 A 012A 8604		LDAA	#204
158 A 012C 97D0		STAA	SD0
159 A 012E CE2100		LDX	#S2100
160 A 0131 BD0325	RECHOR	JSR	SENTEC
161 A 0134 E700		STAB	\$00.X
162 A 0136 F78001		STAB	\$8001
163 A 0139 BD026D		JSR	VD
164 A 013C C601		LDAB	#S01
165 A 013E BD029B		JSR	TIEMPO
166 A 0141 08		INX	
167 A 0142 7A00D0		DEC	\$00D0
168 A 0145 2711		BEQ	SH
169 A 0147 D6D0		LDAB	\$00D0
170 A 0149 C102		CMPB	#S02
171 A 014B 26E4		BNE	RECHOR
172 A 014D 863A		LDAA	#S3A
173 A 014F B78001		STAA	\$8001
174 A 0152 BD026D		JSR	VD
175 A 0155 7E0131		JMP	RECHOR
176 A 0158 BD03BB	SH	JSR	PDT
177 A 015B B62100		LDAA	\$2100
178 A 015E 8125		CMPA	#S25
179 A 0160 24B3		BHS	HORA
180 A 0162 B62101		LDAA	\$2101
181 A 0165 816C		CMPA	#S60
182 A 0167 24AC		BHS	HORA
183 A 0169 BD03F6		JSR	DOK
184 A 016C C131		CMPB	#S31
185 A 016E 2703		BEQ	GRAHOR
186 A 0170 7E0115		JMP	HORA
187 A 0173 CE8609	GRAHOR	LDX	#SB609
188 A 0176 8602		LDAA	#S02
189 A 0178 B721D0		STAA	\$21D0
190 A 017B BD02FA		JSR	GRADAT
191 A 017E C601	OPER1	LDAB	#S01
192 A 0180 BD029B		JSR	TIEMPO
193 A 0183 BD0264		JSR	LIMDIS
194 A 0186 CE40A7		LDX	#S40A7
195 A 0189 86CB		LDAA	#S0B
196 A 018B 97D0		STAA	SD0
197 A 018D BD0273		JSR	DISLET
198 A 0190 BD02D7		JSR	DPAR
199 A 0193 8604		LDAA	#S04
200 A 0195 97D0		STAA	SD0
201 A 0197 CE2100		LDX	#S2100
202 A 019A BD0325	RECOOPER1	JSR	SENTEC
203 A 019D E700		STAB	\$00.X
204 A 019F F78001		STAB	\$8001
205 A 01A2 BD026D		JSR	VD
206 A 01A5 C5C1		LDAB	#S01
207 A 01A7 BD029B		JSR	TIEMPO
208 A 01AA 08		INX	
209 A 01AB 7A00D0		DEC	\$00D0

210	A	01AE	26FA	BNE	RECOVER1
211	A	01B0	BD03BB	JSR	PDT
212	A	01B3	BD03F6	JSR	DOK
213	A	01B6	C131	CMPB	#531
214	A	01B8	2703	BEQ	GRAOPER1
215	A	01BA	7EG17E	JMP	OPER1
216	A	01BD	CEB603	LDX	#5B603
217	A	01C0	8602	LDAA	#502
218	A	01C2	B721D0	STAA	\$21D0
219	A	01C5	BD02FA	JSR	GRADAT
220	A	01C8	C601	LDAB	#501
221	A	01CA	BD029B	JSR	TIEMPO
222	A	01CD	BD0264	JSR	LIMDIS
223	A	01D0	CE40B2	LDX	#540B2
224	A	01D3	960B	LDAA	#50B
225	A	01D5	97D0	STAA	SD0
226	A	01D7	BD0273	JSR	DISLET
227	A	01DA	BD02D7	JSR	DPAR
228	A	01DD	9604	LDAA	#504
229	A	01DF	97D0	STAA	SD0
230	A	01E1	CE2100	LDX	#52100
231	A	01E4	BD0325	JSR	SENTEC
232	A	01E7	E700	STAB	SOO,X
233	A	01E9	F78001	STAB	\$8001
234	A	01EC	BD026D	JSR	VD
235	A	01EF	C601	LDAB	#501
236	A	01F1	BD029B	JSR	TIEMPO
237	A	01F4	08	INX	
238	A	01F5	7A00D0	DEC	SO0D0
239	A	01F8	26EA	BNE	RECOVER2
240	A	01FA	BD03BB	JSR	PDT
241	A	01FD	BD03F6	JSR	DOK
242	A	0200	C131	CMPB	#531
243	A	0202	2703	BEQ	GRAOPER2
244	A	0204	7E01C8	JMP	OPER2
245	A	0207	CEB605	LDX	#5B605
246	A	020A	8602	LDAA	#502
247	A	020C	B721D0	STAA	\$21D0
248	A	020F	BD02FA	JSR	GRADAT
249	A	0212	C601	LDAB	#501
250	A	0214	BD029B	JSR	TIEMPO
251	A	0217	BD0264	JSR	LIMDIS
252	A	021A	CE40BD	LDX	#540BD
253	A	021D	860B	LDAA	#50B
254	A	021F	97D0	STAA	SD0
255	A	0221	BD0273	JSR	DISLET
256	A	0224	BD02D7	JSR	DPAR
257	A	0227	8604	LDAA	#504
258	A	0229	97D0	STAA	SD0
259	A	022B	CE2100	LDX	#52100
260	A	022E	BD0325	JSR	SENTEC
261	A	0231	E700	STAB	SOO,X
262	A	0233	F78001	STAB	\$8001
263	A	0236	BD026D	JSR	VD

264	A	0239	C601	LDAB	#S01	
265	A	023B	BD029B	JSR	TIEMPO	
266	A	023E	08	INX		
267	A	023F	7A00D0	DEC	S00D0	
268	A	0242	26EA	BNE	RECREC	
269	A	0244	BD03BB	JSR	PDT	
270	A	0247	BD03F6	JSR	DOK	
271	A	024A	C131	CMPB	#531	
272	A	024C	2703	BEQ	GRAREC	
273	A	024E	7E0212	JMP	RECORR	
274	A	0251	CEB607	LDX	#SB607	
275	A	0254	9602	LDAA	#S02	
276	A	0256	B721D0	STAA	F21D0	
277	A	0259	BD02FA	JSR	GRADAT	
278	A	025C	C601	LDAB	#S01	
279	A	025E	BD029B	JSR	TIEMPO	
280	A	0261	7E0478	JMP	RECORRIDO	
281	A					
282	A					
283	A					
284	A					
285	A					
286	A	0264	9601	LIMDIS	LDAA	#S01
287	A	0266	B78001		STAA	\$8001
288	A	0269	BD026D		JSR	VD
289	A	026C	39		RTS	
290	A					
291	A	026D	F68000	VD	LDAB	\$8000
292	A	0270	2BFB		BMI	VD
293	A	0272	39		RTS	
294	A					
295	A	0273	A600	DISLET	LDAA	S00.X
296	A	0275	B78001		STAA	\$8001
297	A	0278	BD024D		JSR	VD
298	A	027B	08		INX	
299	A	027C	7A00D0		DEC	S00D0
300	A	027F	2711		BEQ	REGRESA
301	A	0281	96D0		LDAA	S00
302	A	0283	810F		CMPA	#S0F
303	A	0285	26EC		BNE	DISLET
304	A	0287	86C0		LDAA	#SC0
305	A	0289	B78000		STAA	\$8000
306	A	028C	BD026D		JSR	VD
307	A	028F	7E0273		JMP	DISLET
308	A	0292	86CC	REGRESA	LDAA	#SCC
309	A	0294	B78000		STAA	\$8000
310	A	0297	BD026D		JSR	VD
311	A	029A	39		RTS	
312	A					
313	A	029B	3C	TIEMPO	FSHX	
314	A	029C	CEFFFF		LOOP11	#SFFFF
315	A	029F	09		LOOP10	
316	A	02A0	26FD		DEX	
317	A	02A2	5A		BNE	LCCP10
					DECB	

.....
 * RUTINAS DE VERIFICACION Y MANEJO DEL DISPLAY

318	A	02A3	26F7		BNE	LOOP11
319	A	02A5	38		PULX	
320	A	02A6	39		RTS	
321	A					
322	A	02A7	D6D0	IMP	LDAB	S00D0
323	A	02A9	C104		CMPB	#S04
324	A	02AB	2705		BEQ	IMP1
325	A	02AD	C102		CMPB	#S02
326	A	02AF	2701		BEQ	IMP1
327	A	02B1	39		RTS	
328	A	02B2	862F	REGRESA1	LDAA	#S2F
329	A	02B4	B78001	IMP1	STAA	\$B001
330	A	02B7	BD026D		JSR	VD
331	A	02BA	7E02B1		JMP	REGRESA1
332	A					
333	A	02BD	CE405A	MP	LDX	#S405A
334	A	02C0	861F		LDAA	#S1F
335	A	02C2	97D0		STAA	\$D0
336	A	02C4	BD0273		JSR	DISLET
337	A	02C7	860C		LDAA	#S0C
338	A	02C9	B78000		STAA	\$B000
339	A	02CC	BD026D		JSR	VD
340	A	02CF	C6FF		LDAB	#SFF
341	A	02D1	BD029B		JSR	TIEMPO
342	A	02D4	7E0039		JMP	INIPA
343	A					
344	A	02D7	86D0	DPAR	LDAA	\$SD0
345	A	02D9	B78000		STAA	\$B000
346	A	02DC	BD026D		JSR	VD
347	A	02DF	39		RTS	
348	A					
349	A	02E0	C605	BORMEM	LDAB	#S06
350	A	02E2	F7103B		STAB	\$103B
351	A	02E5	F7B600		STAB	\$B600
352	A	02E8	C607		LDAB	#S07
353	A	02EA	F7103B		STAB	\$103B
354	A	02ED	CE0D05		LDX	#S0D05
355	A	02F0	09	LOOP	DEX	
356	A	02F1	26FD		BNE	LOOP
357	A	02F3	7F103B		CLR	\$103B
358	A	02F6	BD0264		JSR	LIMDIS
359	A	02F9	39		RTS	
360	A					
361	A					
362	A					
363	A					
364	A					
365	A	02FA	18CE2100	GRADAT	LDY	#S2100
366	A	02FE	18A500	GD	LDAA	\$00.Y
367	A	0301	C602		LDAB	#S02
368	A	0303	F7103B		STAB	\$103B
369	A	0306	A700		STAA	\$00.X
370	A	0308	C603		LDAB	#S03
371	A	030A	F7103B		STAB	\$103B

.....
 * RUTINA PARA GRABAR DATOS EN MEMORIA DEL MICRO

372	A	030D	183C		PSHY	
373	A	030F	18CE0BB8		LDY	#50BB8
374	A	0313	1809	LOOPA	DEY	
375	A	0315	26FC		BNE	LOOPA
376	A	0317	7F103B		CLR	\$103B
377	A	031A	1838		FULY	
378	A	031C	08		INX	
379	A	031D	1808		INY	
380	A	031F	7A21D0		DEC	\$21D0
381	A	0322	26DA		BNE	GD
382	A	0324	39		RTS	
383	A					
384	A					
385	A					
386	A					
387	A					

* RUTINA PARA SENSAR Y DECODIFICAR EL TECLADO						

388	A	0325	8600	SENTEC	LDAA	#500
389	A	0327	97A0		STAA	SA0
390	A	0329	860E		LDAA	#50E
391	A	032B	BD0348		JSR	REN
392	A	032E	C100		CMPB	#500
393	A	0330	2615		BNE	REGRE1
394	A	0332	860D		LDAA	#50D
395	A	0334	BD0348		JSR	REN
396	A	0337	C100		CMPB	#500
397	A	0339	260C		BNE	REGRE1
398	A	033B	860B		LDAA	#50B
399	A	033D	BD0348		JSR	REN
400	A	0340	C100		CMPB	#500
401	A	0342	2603		BNE	REGRE1
402	A	0344	7E0325		JMP	SENTEC
403	A	0347	39			
404	A	0348	7C00A0	REGRE1	RTS	
405	A	034B	B76000	REN	INC	500A0
406	A	034E	B65000		STAA	56000
407	A	0351	D5A0		LDAA	56000
408	A	0353	C101		LDAB	SA0
409	A	0355	270B		CMPB	#501
410	A	0357	C102		BEQ	REN1
411	A	0359	2716		CMPB	#502
412	A	035B	C103		BEQ	REN2
413	A	035D	2721		CMPB	#503
414	A	035F	C600		BEQ	REN3
415	A	0361	39	REGRE2	LDAB	#500
416	A	0362		REGRE3	RTS	
417	A	0362	44			
418	A	0363	2424	REN1	LSRA	
419	A	0365	44		BCC	CERO
420	A	0366	2426		LSRA	
421	A	0368	44		BCC	UNO
422	A	0369	2428		LSRA	
423	A	036B	44		BCC	DOS
424	A	036C	242A		LSRA	
425	A	036E	7E035F		BCC	TRES
					JMP	REGRE2

426 A 0371 44	REN2	LSRA	
427 A 0372 2429		BCC	CUATRO
428 A 0374 44		LSRA	
429 A 0375 242B		BCC	CINCO
430 A 0377 44		LSRA	
431 A 0378 242D		BCC	SEIS
432 A 037A 44		LSRA	
433 A 037B 242F		BCC	SIETE
434 A 037D 7E035F		JMP	REGRE2
435 A 0380 44	REN3	LSRA	
436 A 0381 242E		BCC	OCHO
437 A 0393 44		LSRA	
438 A 0384 2430		BCC	NUEVE
439 A 0386 7E035F		JMP	REGRE2
440 A 0389 C630	CERO	LDAB	#30
441 A 038B 7E0361		JMP	REGRE3
442 A 038E C631	UNO	LDAB	#31
443 A 0390 7E0361		JMP	REGRE3
444 A 0393 C632	DOS	LDAB	#32
445 A 0395 7E0361		JMP	REGRE3
446 A 0398 C633	TRES	LDAB	#33
447 A 039A 7E0361		JMP	REGRE3
448 A 039D C634	CUATRO	LDAB	#34
449 A 039F 7E0361		JMP	REGRE3
450 A 03A2 C635	CINCO	LDAB	#35
451 A 03A4 7E0361		JMP	REGRE3
452 A 03A7 C636	SEIS	LDAB	#36
453 A 03A9 7E0361		JMP	REGRE3
454 A 03AC C637	SIETE	LDAB	#37
455 A 03AE 7E0361		JMP	REGRE3
456 A 03B1 C638	OCHO	LDAB	#38
457 A 03B3 7E0361		JMP	REGRE3
458 A 03B6 C639	NUEVE	LDAB	#39
459 A 03B8 7E0361		JMP	REGRE3
460 A 03BB CE2100	PDT	LDX	#2100
461 A 03BE 6F00	SQ30	CLR	\$00.X
462 A 03C0 08		INX	
463 A 03C1 8C2106		CPX	#2106
464 A 03C4 26F8		BNE	SQ30
465 A 03C6 CE2100		LDX	#2100
466 A 03C9 A600	SCF	LDAA	\$00.X
467 A 03CB 48		ASLA	
468 A 03CC 48		ASLA	
469 A 03CD 48		ASLA	
470 A 03CE 48		ASLA	
471 A 03CF A700		STAA	\$00.X
472 A 03D1 08		INX	
473 A 03D2 08		INX	
474 A 03D3 8C2106		CPX	#2106
475 A 03D6 26F1		BNE	SCF
476 A 03D8 CE2100		LDX	#2100
477 A 03DB A600	SSUM	LDAA	\$00.X
478 A 03DD AB01		ADDA	\$01.X
479 A 03DF A700		STAA	\$00.X

480 A 03E1 08		INX	
481 A 03E2 08		INX	
482 A 03E3 8C2106		CPX	#52106
483 A 03E6 26F3		BNE	SSUM
484 A 03E8 CE2100		LDX	#52100
485 A 03E8 A602		LDAA	\$02.X
486 A 03ED B72101		STAA	\$2101
487 A 03F0 A604		LDAA	\$04.X
488 A 03F2 B72102		STAA	\$2102
489 A 03F5 39		RTS	
490 A			
491 A 03F6 86C0	DOK	LDAA	#\$C0
492 A 03F8 B78000		STAA	\$8000
493 A 03FB BD026D		JSR	VD
494 A 03FE CE40CB		LDX	#\$40CB
495 A 0401 860F		LDAA	#\$0F
496 A 0403 97D0		STAA	\$D0
497 A 0405 BD0273		JSR	DISLET
498 A 0408 BD02D7		JSR	DPAR
499 A 040B BD0325		JSR	SENTEC
500 A 040E C132		CMPB	#\$32
501 A 0410 24E4		BHS	DOK
502 A 0412 39		RTS	
503 A			
504 A			
505 A			
506 A			
507 A			
508 A			
.....			
* RUTINA DE TRANSFERENCIA DE INFORMACION			
* DEL MICRO A LA P.C.			
.....			
509 A 0413 BD0264	COMUNIC	JSR	LIMDIS
510 A 0416 BD026D		JSR	VD
511 A 0419 CE40D7		LDX	#\$40D7
512 A 041C 860F		LDAA	#\$0F
513 A 041E 97D0		STAA	\$D0
514 A 0420 BD0273		JSR	DISLET
515 A 0423 C60F		LDAB	#\$0F
516 A 0425 BD029B		JSR	TIEMPO
517 A 0428 8610		LDAA	#\$10
518 A 042A B71028		STAA	\$1028
519 A 042D 8630		LDAA	#\$30
520 A 042F B7102B		STAA	\$102B
521 A 0432 7F102C		CLR	\$102C
522 A 0435 8608		LDAA	#\$08
523 A 0437 B7102D		STAA	\$102D
524 A 043A 86C0		LDAA	#\$C0
525 A 043C B7102E		STAA	\$102E
526 A 043F CEB600		LDX	#\$B600
527 A 0442 B6102E	ETD	LDAA	\$102E
528 A 0445 48		LSLA	
529 A 0446 24FA		BCC	ETD
530 A 0448 A600		LDAA	\$00.X
531 A 044A B7102F		STAA	\$102F
532 A 044D 18CE0FFF		LDY	#\$0FFF
533 A 0451 1909	RETY	LDY	

534 A	0453	2EFC	BGT	RETY
535 A	0455	08	INX	
536 A	0456	8CB800	CPX	#\$B800
537 A	0459	26E7	BNE	ETD
538 A	045B	8600	LDAA	#\$500
539 A	045D	B7102D	STAA	\$102D
540 A	0460	BD0264	JSR	LIMDIS
541 A	0463	BD026D	JSR	VD
542 A	0466	CE40E7	LDX	#\$40E7
543 A	0469	861F	LDAA	#\$1F
544 A	046B	97D0	STAA	\$D0
545 A	046D	BD0273	JSR	DISLET
546 A	0470	C60F	LDAB	#\$0F
547 A	0472	BD029B	JSR	TIEMPO
548 A	0475	7E008E	JMP	MENU
549 A				
550 A				
551 A				
552 A				
553 A				

 * RUTINA DE REGISTRO DEL RECORRIDO

554 A	0478	CE2000	RECORRIDO	LDX	#\$2000
555 A	047B	6F00	LIMMEM	CLR	\$00.X
556 A	047D	08		INX	
557 A	047E	8C20FF		CPX	#\$20FF
558 A	0481	26F8		BNE	LIMMEM
559 A	0483	CE5111		LDX	#\$B611
560 A	0486	8602		LDAA	#\$02
561 A	0488	B721D0		STAA	\$21D0
562 A	048B	BD02FA		JSR	GRADAT
563 A	048E	7F1027		CLR	\$1027
564 A	0491	CCB618		LDD	#\$B618
565 A	0494	FD200A		STD	\$200A
566 A	0497	CCB6D8		LDD	#\$B6D8
567 A	049A	FD200C		STD	\$200C
568 A	049D	CE1025		LDX	#\$1025
569 A	04A0	8650	REINICP	LDAA	#\$50
570 A	04A2	B71026		STAA	\$1026
571 A	04A5	6F00	CUENTAP	CLR	\$00.X
572 A	04A7	B61027		LDAA	\$1027
573 A	04AA	B72008		STAA	\$2008
574 A	04AD	BD0569		JSR	RELOJ
575 A	04B0	7C2001		INC	\$2001
576 A	04B3	F620D0		LDAB	\$20D0
577 A	04B6	C101		CMPB	#\$01
578 A	04B8	264C		BNE	CMPCP
579 A	04BA	C601		LDAB	#\$01
580 A	04BC	BD029B		JSR	TIEMPO
581 A	04BF	BD0569	CFIN	JSR	RELOJ
582 A	04C2	F620D0		LDAB	\$20D0
583 A	04C5	C101		CMPB	#\$01
584 A	04C7	2737		BEQ	FIN
585 A	04C9	7C2000		INC	\$2000
586 A	04CC	B62000		LDAA	\$2000
587 A	04CF	8106		CMPA	#\$06

588	A	04D1	26EC		BNE	CFIN
589	A	04D3	B62003		LDAA	\$2003
590	A	04D6	8102		CMPA	#500
591	A	04D8	2715		BEQ	SALTO2
592	A	04DA	810A		CMPA	#50A
593	A	04DC	240E		BHS	SALTO1
594	A	04DE	7F2000		CLR	\$2000
595	A	04E1	B62003		LDAA	\$2003
596	A	04E4	8B06		ADDA	#506
597	A	04E6	B72003		STAA	\$2003
598	A	04E9	7E0534		JMP	PROPAR1
599	A	04EC	7F2000	SALTO1	CLR	\$2000
600	A	04EF	7E053A	SALTO2	JMP	PARCCNF2
601	A	04F2	7F2000		CLR	\$2000
602	A	04F5	7E04A5		JMP	CUENTAP
603	A	04F8	7C2007	ACP	INC	\$2007
604	A	04FB	6F00		CLR	\$00.X
605	A	04FD	7E04A5		JMP	CUENTAP
606	A	0500	BD0500	FIN	JSR	FIN
607	A	0503	7E008E		JMP	MENU
608	A	0506	B61027	CCMPCP	LDAA	\$1027
609	A	0509	F62008		LDAB	\$2008
610	A	050C	11		CBA	
611	A	050D	270D		BEQ	VERPAR
612	A	050F	B62006		LDAA	\$2006
613	A	0512	8101		CMPA	#501
614	A	0514	262A		BNE	VPERICDO
615	A	0516	BD0654		JSR	FINPARADA
616	A	0519	7E04A0		JMP	REINICP
617	A	051C	B62006	VERPAR	LDAA	\$2006
618	A	051F	8100		CMPA	#500
619	A	0521	261A		BNE	PARCONF
620	A	0523	7C2003		INC	\$2003
621	A	0526	B62003		LDAA	\$2003
622	A	0529	8101		CMPA	#501
623	A	052B	2707		BEQ	PROPAR1
624	A	052D	8102		CMPA	#50A
625	A	052F	2406		BHS	PARCONF1
626	A	0531	7E04A5		JMP	CUENTAP
627	A	0534	7E04A5	PROPAR1	JMP	CUENTAP
628	A	0537	BD0626	PARCONF1	JSR	INIPARADA
629	A	053A	BD04EF	PARCONF2	JSR	SALTO2
630	A	053D	7E04A5	PARCONF	JMP	CUENTAP
631	A	0540	BD04EC	VPERIODO	JSR	SALTO1
632	A	0543	B62001		LDAA	\$2001
633	A	0546	810F		CMPA	#50F
634	A	0548	2403		BHS	REGISTRO
635	A	054A	7E04A5		JMP	CUENTAP
636	A	054D	BD05E5	REGISTRO	JSR	REGVARI
637	A	0550	7E04A0		JMP	REINICP
638	A	0553	3C	SEGUNDO	FSHX	
639	A	0554	CE1AFO	LOOP2	LDX	#91AFO
640	A	0557	BD04EF	LOOP3	JSR	SALTO2
641	A	055A	B620D0		LDAA	\$20DC

642 A	055D	8101	CMPA	#S01
643 A	055F	2706	BEQ	SAL
644 A	0561	09	DEX	
645 A	0562	26F0	BNE	LOOP2
646 A	0564	5A	DECB	
647 A	0565	26F0	BNE	LOOP3
648 A	0567	38	SAL	PULX
649 A	0568	39		RTS

.....
 * RUTINA DE CONTROL DE TIEMPO

655 A	0569	7F20D0	RELOJ	CLR	\$20D0
656 A	056C	C605		LDAB	#S05
657 A	056E	B00553		JSR	SEGUNDO
658 A	0571	B620D0		LDAA	\$20D0
659 A	0574	810C0C		CMPA	#S01
660 A	0576	2733		BEQ	RTSREL
661 A	0578	B620CC		LDAA	\$20CC
662 A	057B	4C		INCA	
663 A	057C	813C		CMPA	#S3C
664 A	057E	2706		BEQ	MIN
665 A	0580	B720CC		STAA	\$20CC
666 A	0583	7E05AB		JMP	RTSREL
667 A	0586	7F20CC	MIN	CLR	\$20CC
668 A	0589	B620CD		LDAA	\$20CD
669 A	058C	4C		INCA	
670 A	058D	813C		CMPA	#S3C
671 A	058F	2706		BEQ	HOR
672 A	0591	B720CD		STAA	\$20CD
673 A	0594	7E05AB		JMP	RTSREL
674 A	0597	7F20CD	HOR	CLR	\$20CD
675 A	059A	B620CE		LDAA	\$20CE
676 A	059D	4C		INCA	
677 A	059E	8119		CMPA	#S19
678 A	05A0	2706		BEQ	LIMHOR
679 A	05A2	B720CE		STAA	\$20CE
680 A	05A5	7E05AB		JMP	RTSREL
681 A	05A8	7F20CE	LIMHOR	CLR	\$20CE
682 A	05AB	39	RTSREL	RTS	
683 A	05AC				

.....
 * RUTINA DE FIN DEL RECORRIDO

688 A	05AC	8600	RFIN	LDAA	#S00
689 A	05AE	B71026		STAA	\$1026
690 A	05B1	B620CE		LDAA	\$20CE
691 A	05B4	B72100		STAA	\$2100
692 A	05B7	B620CD		LDAA	\$20CD
693 A	05BA	B72101		STAA	\$2101
694 A	05BD	CEB613		LDX	#SB613
695 A	05C0	8602		LDAA	#S02

696	A	05C2	B721D0	STAA	\$21D0
697	A	05C5	BD02FA	JSR	GRADAT
698	A	05C8	CEB615	LDX	#3B615
699	A	05CB	3602	LDAA	#502
700	A	05CD	B721D0	STAA	\$21D0
701	A	05D0	BD02FA	JSR	GRADAT
702	A	05D3	B62009	LDAA	\$2009
703	A	05D6	B72100	STAA	\$2100
704	A	05D9	CEB617	LDX	#3B617
705	A	05DC	8601	LDAA	#501
706	A	05DE	B721D0	STAA	\$21D0
707	A	05E1	BD02FA	JSR	GRADAT
708	A	05E4	39	RTS	
709	A				
710	A				
711	A				
712	A				
713	A				

.....
 * RUTINA DE REGISTRO DE VARIABLES EN MEMORIA

714	A	05E5	3C	REGVARI	FSHX	
715	A	05E6	C605		LDAB	#505
716	A	05E8	BD029B		JSR	TIEMPO
717	A	05EB	6600		LDAA	#500
718	A	05ED	B71026		STAA	\$1026
719	A	05F0	FC2007		LDD	\$2007
720	A	05F3	FD2100		STD	\$2100
721	A	05F6	B61032		LDAA	\$1032
722	A	05F9	B72102		STAA	\$2102
723	A	05FC	8100		CMPA	#500
724	A	05FE	2705		BEQ	EST
725	A	0600	8600		LDAA	#500
726	A	0602	7E0607		JMP	SALTORV
727	A	0605	8601	EST	LDAA	#501
728	A	0607	B72103	SALTORV	STAA	\$2103
729	A	060A	FE200C		LDX	\$200C
730	A	060D	8604		LDAA	#504
731	A	060F	B721D0		STAA	\$21D0
732	A	0612	BD02FA		JSR	GRADAT
733	A	0615	FC200C		LDD	\$200C
734	A	0618	C30004		ADDD	#504
735	A	061B	FD20JC		STD	\$200C
736	A	061E	7F1027		CLR	\$1027
737	A	0621	7F2001		CLR	\$2001
738	A	0624	38		PULX	
739	A	0625	39		RTS	
740	A					
741	A					
742	A					
743	A					
744	A					

.....
 * RUTINA DE REGISTRO DE PARADAS EN EL RECORRIDO

745	A	0626	3C	INIPARADA	FSHX	
746	A	0627	B620CE		LDAA	\$20CE
747	A	062A	B72100		STAA	\$2100
748	A	062D	B620CD		LDAA	\$20CD
749	A	0630	B72101		STAA	\$2101

750 A 0633 FE200A	LDX	\$200A
751 A 0636 8602	LDAA	#\$02
752 A 0638 B721D0	STAA	\$21D0
753 A 063B BD02FA	JSR	GRADAT
754 A 063E FC200A	LDD	\$200A
755 A 0641 C30004	ADDD	#\$04
756 A 0644 8F	XGDX	
757 A 0645 8602	LDAA	#\$02
758 A 0647 B721D0	STAA	\$21D0
759 A 064A BD02FA	JSR	GRADAT
760 A 064D 8601	LDAA	#\$01
761 A 064F B72006	STAA	\$2006
762 A 0652 38	PULX	
763 A 0653 39	RTS	
764 A		
765 A 0654 3C	FINPARADA PSHX	
766 A 0655 8620CE	LDAA	\$20CE
767 A 0658 B721D0	STAA	\$21D0
768 A 065B 8620CD	LDAA	\$20CD
769 A 065E B72101	STAA	\$2101
770 A 0661 FC200A	LDD	\$200A
771 A 0664 C30002	ADDD	#\$02
772 A 0667 8F	XGDX	
773 A 0668 8602	LDAA	#\$02
774 A 066A B721D0	STAA	\$21D0
775 A 066D BD02FA	JSR	GRADAT
776 A 0670 FC200A	LDD	\$200A
777 A 0673 C30006	ADDD	#\$06
778 A 0676 8F	XGDX	
779 A 0677 8602	LDAA	#\$02
780 A 0679 B721D0	STAA	\$21D0
781 A 067C BD02FA	JSR	GRADAT
782 A 067F FC200A	LDD	\$200A
783 A 0682 C30009	ADDD	#\$08
784 A 0685 FD200A	STD	\$200A
785 A 0688 7F2001	CLR	\$2001
786 A 068B 7F2006	CLR	\$2006
787 A 068E 8600	LDAA	#\$00
788 A 0690 B71026	STAA	\$1026
789 A 0693 7F1027	CLR	\$1027
790 A 0696 7F2003	CLR	\$2003
791 A 0699 38	PULX	
792 A 069A 39	RTS	
793 A		
794 A	END	

SYMBOL TABLE: Total Entries= 99

ACP	04F8	PARCONF1	0537
BORMEM	02E0	PARCONF2	053A
CERO	0389	FDT	03BB
CFIN	04BF	PROPAR1	0534
CINCO	03A2	RECPEC	00CB
COMPCP	0506	RECHOR	0131

COMUNIC	0413	RECOPER1	019A
CUATRO	039D	RECOPER2	01E4
CUENTAP	04A5	RECORR	0212
DISLET	0273	RECORRID	0478
DOK	03F6	RECPASS	0050
DOS	0393	RECREC	022E
DPAR	02D7	REGISTRO	054D
EST	0605	REGRE1	0347
ETD	0442	REGRE2	03F7
FECHA	00AF	REGRE3	0361
FIN	0500	REGRESA	0292
FINPARAD	0654	REGRESA1	02B1
GD	02FE	REGVARI	05E5
GRADAT	02FA	REINICP	04A0
GRAFEC	010A	RELOJ	0569
GRAHOR	0173	REN	0348
GRAOPER1	018D	REN1	0362
GRAOPER2	0207	REN2	0371
GRAREC	0251	REN3	0380
HCR	0597	RETY	0451
HORA	0115	RFIN	05AC
IMP	02A7	RTSREL	05AB
IMP1	02B2	SAL	0567
INI	0006	SALTO1	04EC
INIPA	0039	SALTO2	04EF
INIPARAD	0626	SALTORV	0607
LIMDIS	0264	SCF	03C9
LIMHOR	05A8	SEGUNDO	0553
LIMHEM	047B	SEIS	03A7
LOOP	02F0	SENTEC	0325
LOOP10	029F	SF	00E7
LOOP11	029C	SH	0158
LOOP2	0554	SIETE	03AC
LOOP3	0557	SMENU	009B
LOOPA	0313	SQ30	03BE
MALPASS	0088	SSUM	03DB
MENU	008E	TIEMPO	029B
MIN	0586	TRANS	00AC
MP	02BD	TRES	0398
NUEVE	03B6	UNO	038E
OCHO	03B1	VD	026D
OPER1	017E	VERPAR	051C
OPER2	01C8	VPERIODO	0540
PARCONF	053D		

Total errors: 0

PROGRAMACION DEL SOFTWARE DEL SISTEMA

***** Pantalla de Presentación UNAM *****

```
Private Sub Form_Load()  
frmPresentaUNAM.Caption = TITULO_APLICACION  
End Sub
```

***** Pantalla de Presentación Facultad de Ingeniería *****

```
Private Sub Form_Load()  
tmrRetardo.Interval = MILISEGUNDOS / 5  
tmrRetardo.Enabled = True  
End Sub
```

***** Pantalla de Presentación Sistema *****

```
Private Sub Form_Load()  
gbModoDepuracion = InStr(Command$, "/DEBUG") > 0  
End Sub
```

***** Pantalla de Login *****

```
Private Sub sscAceptar_Click()  
On Error GoTo Error_sscAceptar  
Dim iParamsSalvados As Boolean  
Dim sString As String  
If (txtUsuario.Text <> Empty) Then  
If (txtContraseña.Text <> Empty) Then  
If (ExisteUsrPwd(txtUsuario.Text, txtContraseña.Text) = True) Then  
Unload Me  
mdiComputDeViaje.Show  
Else  
txtUsuario.Text = Empty  
txtContraseña.Text = Empty  
txtUsuario.SetFocus  
End If  
End Sub
```


***** Pantalla Principal *****

```

Private Sub MDIForm_Load()
    Dim vParamsObtenidos As Variant
    Dim iNumParams As Integer
    Dim vParametro As Variant

    Dim iNumPanels As Integer

    For iNumPanels = 1 To 2
        stbCompViaje.Panels.Add ' Add 2 Panel objects.
    Next iNumPanels

    With stbCompViaje.Panels
        .Item(1).Style = sbrNum ' Number lock
        .Item(2).Style = sbrCaps ' Caps lock
        .Item(3).Style = sbrScrt ' Scroll lock
    End With

```

***** Comunicación con el MCU *****

```

Public Function AbrePuerto(comMSComm As Control) As Boolean

```

```

    On Error GoTo ErrorAbrePuerto

```

```

    Dim bError As Boolean

```

```

    Dim msg As String

```

```

    Dim gsError As String

```

```

    Dim sSettings As String

```

```

' Declaracion de Variables

```

```

    Dim sMensaje As String

```

```

    Dim iEstilo As Integer

```

```

    Dim sTitulo As String

```

```

    Dim sArchivoDeAyuda As String

```

```

    Dim iContexto As Integer

```

```

    Dim iRespuesta As Integer

```

```

'Call InitTables

```

```

iHandleComm = comMSComm.CommID

```

```

' Establece y regresa el número de puerto de comunicaciones.

```

```

' Esta propiedad debe ser establecida antes de abrir el puerto.

```

```

comMSComm.CommPort = giPuertoCOM

```

```

comMSCComm.Settings = giVelocidadEnBauds & ", " & _
    gsParidad & ", " & _
    giBitsDeDatos & ", " & _
    giBitsDeParada

' Establece y regresa el número de caracteres que la propiedad Input lee
' a partir del buffer de recepción.
' Lee por completo el contenido del buffer de recepción.

comMSCComm.InputLen = 0

' Establece y recupera el estado del puerto de comunicaciones (abierto o
' cerrado).
' estado = comMSCComm.PortOpen

'If (estado = False) Then
comMSCComm.InBufferSize = 1024

If (comMSCComm.PortOpen = False) Then
    ' En el momento en que se intenta abrir el puerto de comunicaciones,
    ' se detecta si este ya está abierto o no.

    comMSCComm.PortOpen = True

    AbrePuerto = True
End If

Exit Function

```

***** Configuración de Parámetros de Comunicación *****

```

Private Sub Form_Load()

'chkGuardarConfig.Enabled = False

'giPuertoCOM = 4
'If (chkGuardarConfig.Value = True) Then
'Call ObtieneParamsComunic

ssoCOM(giPuertoCOM - 1).Value = True

Select Case giVelocidadEnBauds
Case 300
    optVelocidadBauds(0).Value = True
Case 600
    optVelocidadBauds(1).Value = True
Case 1200
    optVelocidadBauds(2).Value = True
Case 2400
    optVelocidadBauds(3).Value = True
Case 4800
    optVelocidadBauds(4).Value = True

```

```

Case 9600
  optVelocidadBauds(5).Value = True
End Select
End Sub

```

***** Consultas*****

```
Private Sub Form_Activate()
```

```

  Call DatosFormatearGridConsultas(TAB_CNS_VEHICULOS, 6)
  Call FormatearGrid(grdVehiculos, 6, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)

  Call DatosFormatearGridConsultas(TAB_CNS_OPERADORES, 11)
  Call FormatearGrid(grdOperadores, 11, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)

  Call DatosFormatearGridConsultas(CONSULTA_RUTAS, 2)
  Call FormatearGrid(grdRutas, 2, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)

  Call DatosFormatearGridConsultas(TAB_CNS_PARADAS, 8)
  Call FormatearGrid(grdParadas, 8, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)

  Call DatosFormatearGridConsultas(TAB_CNS_PERIODOS, 5)
  Call FormatearGrid(grdPeriodos, 5, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)

  Call DatosFormatearGridConsultas(TAB_CNS_RECORRIDOS, 6)
  Call FormatearGrid(grdRecorridos, 6, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)

```

```
End Sub
```

```
Public Sub ConOpera(Grid As Control, sTablaTemporal As String, ssoNombre As Control)
```

```

  Const SPS_OPERADORES_NOMBRE As String = "spSOperadorNombre"
  Const SPS_OPERADOR As String = "spSOperador"

```

```

  Dim iRenglon As Integer
  Dim iNumRegTmp As Integer
  Dim rstRecordset As Recordset

```

```
  Dim sClave As String
```

```

  Dim miClaveOperador As Integer
  Dim miClaveRecorrido As Integer
  Dim msPlaca As String
  Dim miClaveRuta As Integer
  Dim msNombre As String
  Dim msApellidoPaterno As String
  Dim msApellidoMaterno As String
  Dim miOperadorTipo As Integer
  Dim msDireccion As String
  Dim msTelefono As String
  Dim msRFC As String

```

```
Dim miNoLicencia As String
```

```
Dim miSexo As String
```

```
Dim qryQuery As QueryDef
```

```
gsQuery = "DELETE FROM " & sTablaTemporal
```

```
gdbDatabase.Execute gsQuery
```

```
Call LimpiarGrid(grdOperadores)
```

```
Call DatosFormatearCnsltXOPerador(CONSULTA_OPERADORES, 11, ssoNombre)
```

```
Call FormatearGrid(grdOperadores, 11, gaiAnchoCols(), _  
gasEncabezadosCols(), gaiAlineacionCols)
```

```
Set grsRecordset = gdbDatabase.OpenRecordset(sTablaTemporal, dbOpenTable)
```

```
Public Sub ConParadas(grdParadas As Control, sTablaTemporal As String)
```

```
Const SPS_COMPUTADORA_TODAS As String = "spS_ComputViaje_Todas"
```

```
Const SPS_COMPUTADORA_FILTRO As String = "spS_ComputViaje_Filtro"
```

```
Dim iRenglon As Integer
```

```
Dim iNumRegTmp As Integer
```

```
Dim rstRecordset As Recordset
```

```
Dim sClave As String
```

```
Dim miComputadora As Integer
```

```
Dim miSensadoParada As Integer
```

```
Dim msHoraArribo As String
```

```
Dim msHoraPartida As String
```

```
Dim miCombustArribo As Integer
```

```
Dim miCombustArribo As Double
```

```
Dim miCEArribo As Integer
```

```
Dim miCombustPartida As Integer
```

```
Dim miCombustPartida As Double
```

```
Dim miCEPartida As Integer
```

```
gsQuery = "DELETE FROM " & sTablaTemporal
```

```
gdbDatabase.Execute gsQuery
```

```
Call LimpiarGrid(grdParadas)
```

```
Call DatosFormatearGridConsultas(TAB_CNS_PARADAS, 8)
```

```
Call FormatearGrid(grdParadas, 8, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)
```

```
Set grsRecordset = gdbDatabase.OpenRecordset(sTablaTemporal, dbOpenTable)
```

```
Public Sub ConPeriodos(Grd As Control, sTablaTemporal As String)
```

```
Const SPS_SENSADO_PERIODO_TODAS As String = "spS_Sensado_Periodo_Todas"
```

```
Const SPS_SENSADO_PERIODO_FILTRO As String = "spS_Sensado_Periodo_Filtro"
```

```
Dim iRenglon As Integer
Dim iNumRegTmp As Integer
Dim rstRecordset As Recordset
```

```
Dim sClave As String
```

```
Dim miComputadora As Integer
Dim nuSensadoPeriodo As Integer
Dim miDistancia As Integer
Dim nuTemperatura As Integer
Dim miCETemperatura As Integer
```

```
gsQuery = "DELETE FROM " & sTablaTemporal
gdbDatabase.Execute gsQuery
```

```
Call LimpiarGrid(Grid)
Call DatosFormatearGridConsultas(TAB_CNS_PERIODOS, 5)
Call FormatearGrid(Grid, 5, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)
```

```
Set gsrRecordset = gdbDatabase.OpenRecordset(sTablaTemporal, dbOpenTable)
```

```
Public Sub ConRecorridos(Grid As Control, sTablaTemporal As String)
```

```
Dim iRenglon As Integer
Dim iNumRegTmp As Integer
Dim rstRecordset As Recordset
Dim sClave As String
Dim miComputadora As Integer
Dim nuSensadoPeriodo As Integer
Dim miDistancia As Integer
Dim mfDistancia As Double
Dim mfVelocidad As Double
Dim miTemperatura As Integer
Dim mfTemperatura As Double
Dim miCETemperatura As Integer
```

```
gsQuery = "DELETE FROM " & sTablaTemporal
gdbDatabase.Execute gsQuery
```

```
Call LimpiarGrid(Grid)
Call DatosFormatearGridConsultas(TAB_CNS_RECORRIDOS, 6)
Call FormatearGrid(Grid, 6, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)
```

```
Set gsrRecordset = gdbDatabase.OpenRecordset(sTablaTemporal, dbOpenTable)
```

```
Public Sub ConRutas(Grid As Control, sTablaTemporal As String, ssoClaveRutas As Control)
```

```
Const SPS_RUTAS_CLAVE As String = "spSRutasClave"
Const SPS_RUTAS_NOMBRE As String = "spSRutasNombre"
```

```
Dim iRenglon As Integer
```

```
Dim iNumRegTmp As Integer
Dim rstRecordset As Recordset
```

```
Dim sClave As String
```

```
Dim miClaveRuta As Integer
Dim msRuta As String
```

```
Dim qryQuery As QueryDef
```

```
gsQuery = "DELETE FROM " & sTablaTemporal
gdbDatabase.Execute gsQuery
```

```
Call LimpiarGrid(grdRutas)
Call DatosFormatearCnsitXOPERador(CONSULTA_RUTAS, 2, ssoClaveRutas)
Call FormatearGrid(grdRutas, 2, gaiAnchoCols(), _
    gaiEncabezadosCols(), gaiAlineacionCols())
```

```
Set grsRecordset = gdbDatabase.OpenRecordset(sTablaTemporal, dbOpenTable)
```

```
Const SPS_VEHICULOS_FILTRO As String = "spSVehiculosFiltro"
Const SPS_VEHICULOS_TODOS As String = "spSVehiculosTodos"
```

```
Dim iRenglon As Integer
Dim iNumRegTmp As Integer
Dim rstRecordset As Recordset
```

```
Dim sClave As String
```

```
Dim msPlacas As String
Dim miCombustTipo As Integer
Dim miMarca As Integer
Dim msMarcaNombre As String
Dim miModelo As Integer
Dim miAño As Integer
Dim miClaveVehiculo As Integer
Dim miClaveModelo As Integer
Dim msCombustibleNombre As String
Dim msModeloNombre As String
```

```
' FFC-Or: If (cboCliente.Text = NULL_STRING) Then
' FFC-Or: gsMensaje = "Seleccione un cliente para poder efectuar la consulta"
' FFC-Or: DesMsg CUIDADO
' FFC-Or: Exit Sub
' FFC-Or: End If
```

```
' Elimina los datos de la tabla temporal
```

```
'gsDBConexion.Execute "DELETE FROM " & sTablaTemporal
```

```
gsQuery = "DELETE FROM " & sTablaTemporal
gdbDatabase.Execute gsQuery
```

```

Call LimpiarGrid(Grid)
Call DatosFormatearGridConsultas(TAB_CNS_VEHICULOS, 6)
Call FormatearGrid(Grid, 6, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)

Set grsRecordset = gdbDatabase.OpenRecordset(sTablaTemporal, dbOpenTable)

```

***** Catálogos *****

Option Explicit

```

Dim mbActualizar As Boolean
Dim mbRealizarProceso As Boolean

```

```
Private Sub Form_Activate()
```

```

Screen.MousePointer = vbDefault
Call DatosFormatearGrid(3, 2)
Call FormatearGrid(grdCatalogos, 2, gaiAnchoCols(), gasEncabezadosCols(), _
gaiAlineacionCols())

```

```

mbActualizar = False
Call sstCatalogos_Click(0)
sscModificar.Enabled = "False"
sscEliminar.Enabled = False
If (txtDescripcion.Enabled = True) Then
    txtClave.Text = giSigCveDsp
    txtDescripcion.SetFocus
End If

```

```
End Sub
```

```
Private Sub grdCatalogos_Click()
```

```
' Declaración de variables a nivel procedimiento
```

```
Dim iIndice As Integer
```

```

grdCatalogos.SelectStartCol = 0
grdCatalogos.SelectEndCol = 1

txtClave.Enabled = False
txtDescripcion.Enabled = False

grdCatalogos.Col = 0
If (grdCatalogos.Text <> Empty) Then
    mbActualizar = False
    mbRealizarProceso = True
    txtClave.Text = grdCatalogos.Text
    grdCatalogos.Col = 1
    txtDescripcion.Text = grdCatalogos.Text
    sscAceptar.Enabled = False
    sscModificar.Enabled = True

```

```
sscEliminar.Enabled = True
Else
    ' FFC-Rmk: Convertir a procedimiento
    'txtClave.Enabled = True
    Call ObtSigCveDsp(txtClave)
    txtDescripcion.Enabled = True
    txtDescripcion.SetFocus
    sscAceptar.Enabled = True
    sscModificar.Enabled = False
    sscEliminar.Enabled = False
End If

End Sub

Private Sub sstCatalogos_Click(PreviousTab As Integer)

Call LimpiarGrid(grdCatalogos)

Select Case sstCatalogos.Tab
Case CATALOGO_RUTAS
    gsTabla = "Ruta"
    gsCampo = "Ruta_Id"
    frmCatalogos.Caption = "Mantenimiento a Catálogos - " & "(Rutas)"
Case CATALOGO_MARCAS
    gsTabla = "Marca"
    gsCampo = "Marca_Id"
    frmCatalogos.Caption = "Mantenimiento a Catálogos - " & "(Marcas)"
Case CATALOGO_COMPUTADORAS
    gsTabla = "Computadora_de_Viaje"
    gsCampo = "Computadora_de_Viaje_Id"
    frmCatalogos.Caption = "Mantenimiento a Catálogos - " & "(Computadoras de Viaje)"
Case CATALOGO_TIPOS_OPERADORES
    gsTabla = "Operador_Tipo"
    gsCampo = "Operador_Tipo_Id"
    frmCatalogos.Caption = "Mantenimiento a Catálogos - " & "(Tipos de Operadores)"
Case CATALOGO_COMBUSTIBLES
    gsTabla = "Combustible"
    gsCampo = "Combustible_Id"
    frmCatalogos.Caption = "Mantenimiento a Catálogos - " & "(Combustibles)"
End Select

    gsQuery = "SELECT " & "
gsQuery = gsQuery & "FROM " & gsTabla & " "
gsQuery = gsQuery & "ORDER BY " & gsCampo

Call LlenarGrid(grdCatalogos)
Call ObtSigCveDsp(txtClave)
txtDescripcion.Enabled = True

End Sub

Private Sub Form_Activate()
```


' Establece el tipo de apuntador del mouse a Reloj de Arena.

```
Screen.MousePointer = vbHourglass
```

```
Call DatosFormatearGridCatalogos(CATALOGO_MODELOS, 4)
```

```
Call FormatearGrid(grdModelos, 4, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)
```

```
Call DatosFormatearGridCatalogos(CATALOGO_VEHICULOS, 6)
```

```
Call FormatearGrid(grdVehiculos, 6, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)
```

```
Call DatosFormatearGridCatalogos(CATALOGO_COMPUTS_VEHICLS, 3)
```

```
Call FormatearGrid(grdComputsVehicls, 3, gaiAnchoCols(), gasEncabezadosCols(), _  
gaiAlineacionCols)
```

```
Call DatosFormatearGridCatalogos(CATALOGO_RECORRIDOS, 13)
```

```
Call FormatearGrid(grdRecorridos, 13, gaiAnchoCols(), gasEncabezadosCols(), gaiAlineacionCols)
```

```
mbActualizar = False
```

```
Call sstCatalogos_Click(TAB_CAT_MODELOS)
```

```
sscModificar.Enabled = False
```

```
sscEliminar.Enabled = False
```

```
txtClave.SetFocus
```

' Establece el tipo de apuntador del mouse a Default:

```
Screen.MousePointer = vbDefault
```

End Sub

***** Módulos *****

' Módulo Estándar (Standard Module):

' Es un módulo que contiene únicamente declaraciones y definiciones de procedimientos, tipos y datos. Declaraciones y definiciones a Nivel

' Módulo (Module Level) en un Módulo Estándar son Públicas (Public) por default.

```
Public Const TAB_CNS_VEHICULOS As Integer = 0
```

```
Public Const TAB_CNS_OPERADORES As Integer = 1
```

```
Public Const TAB_CNS_RUTAS As Integer = 2
```

```
Public Const TAB_CNS_PARADAS As Integer = 3
```

```
Public Const TAB_CNS_PERIODOS As Integer = 4
```

```
Public Const TAB_CNS_RECORRIDOS As Integer = 5
```

```
Public Const CATALOGOS_GENERALES As Integer = 1
```

```
Public Const CATALOGO_MODELOS As Integer = 2
```

```
Public Const CATALOGO_VEHICULOS As Integer = 3
```

```
Public Const CATALOGO_COMPUTS_VEHICLS As Integer = 4
```

```
Public Const CATALOGO_RECORRIDOS As Integer = 6
```

```
Public Const CATALOGO_RUTAS As Integer = 0
Public Const CATALOGO_MARCAS As Integer = 1
Public Const CATALOGO_COMPUTADORAS As Integer = 2
Public Const CATALOGO_TIPOS_OPERADORES As Integer = 3
Public Const CATALOGO_COMBUSTIBLES As Integer = 4
```

```
' Identificadores de Tabs para el módulo Catálogos Particulares:
```

```
Public Const TAB_CAT_MODELOS As Integer = 0
Public Const TAB_CAT_VEHICULOS As Integer = 1
Public Const TAB_CAT_COMPUTS_VEHICLS As Integer = 2
Public Const TAB_CAT_RECORRIDOS As Integer = 5
```

```
Public Const GRID_ALINEAR_IZQUIERDA As Integer = 0 ' Alinear a la izquierda el dato dentro de la columna.
```

```
Public Const GRID_ALINEAR_DERECHA As Integer = 1 ' Alinear a la derecha el dato dentro de la columna.
```

```
Public Const GRID_ALINEAR_CENTRO As Integer = 2 ' Centra el dato dentro de la columna.
```

```
' Nombres de QueryDefs
```

```
Public Const SPS_MARCA_VEHICULO As String = "spSMarcaVehiculo"
Public Const SPS_MODELO_VEHICULO As String = "spSModeloVehiculo"
Public Const SPS_MODELO_VEHICULO_AÑO As String = "spSModeloVehiculoAño"
Public Const SPS_COMPUTADORA_VIAJE As String = "spS_Computadora_Viaje"
Public Const SPS_VEHICULO_PLACA As String = "spS_Vehiculo_Placa"
Public Const SPS_RUTA As String = "spS_Ruta"
Public Const SPS_SIG_NUM_RECORRIDO As String = "spS_Sig_Num_Recorrido"
Public Const SPS_RECORRIDO_TODOS As String = "spS_Recorrido_Todos"
Public Const SPS_RECORRIDO_FILTRO As String = "spS_Recorrido_Filtro"
```

```
' Constantes
```

```
Public Const gsIndicePorDefault = "PrimaryKey"
```

```
' Variables Públicas
```

```
Public gsApiDirLoc As String
Public gsApiBDD As String
```

```
Type typTabla
gsNombre As String
gsCampos As String
gsIndices As String
End Type
```

```
Public gsBDDTablas(0 To 4) As typTabla
```

```
Sub db_Init(Filerame As String)
```

```
Dim DefaultWorkspace As Workspace
Dim CurrentDatabase As Database
Dim dbxComputViaje As Database
```

```
Dim sBDDVersion As String
Dim sVersion As String
Dim sRelease As String
Dim iUbicacionPunto As Integer
```

```
Dim sTextoVersion As String
```

```
Set DefaultWorkspace = DBEngine.Workspaces(0)
```

```
' Create new, encrypted database.
' Crea una nueva base de datos:
```

```
Set dbxComputViaje = DefaultWorkspace.CreateDatabase(Filename, _
    dbLangGeneral, _
    dbVersion20)
```

```
Function FileExists(Filename As String) As Integer
```

```
On Error GoTo FileExistsError
```

```
FileExists = (Dir$(Filename, vbNormal) <> NULL_STRING)
Exit Function
End Function
```

```
Function TableExists(Filename As String, TableName As String)
```

```
Dim i As Integer
Dim found As Integer
Dim Db As Database
Dim dbxComputViaje As Database
Dim DefinicionTabla As TableDef
```

```
found = False
Set dbxComputViaje = OpenDatabase(Filename)
End Function
```

```
Sub tb_Int(szFilename As String, AnyTable As typTabla)
```

```
Dim dbxComputViaje As Database
Dim defComputViaje As New TableDef
Dim fldComputViaje As Field
Dim idxComputViaje As Index
```

```
Dim iStart As Integer, iMid As Integer, iEnd As Integer
```

```
Dim szTemp As String, szAux As String
```

```
Dim sSetting As String
```

```
' fldComputViaje:
```

```
' Un objeto Field representa una columna de datos con un tipo de datos
```

```
' común y un conjunto de propiedades común
```

```
' Una colección Fields contiene todos los objetos Field almacenados de un
```

```
' Index, QueryDef, Recordset, Relation, u objeto TableDef.
```

```
If (AnyTable.gsCampos = NULL_STRING) Then
```

```
Exit Sub
```

```
End If
```

```
Set dbxComputViaje = OpenDatabase(szFilename)
```

```
defComputViaje.Name = AnyTable.gsNombre
```

```
szTemp = AnyTable.gsCampos
```

```
If (iPosComaAux <> 0) Then
```

```
    sSetting = Mid$(szAux, 1, iPosComaAux - 1)
```

```
    Else
```

```
        sSetting = szAux
```

```
    End If
```

```
If (iPosComaAux <> 0) Then
```

```
    If (fldComputViaje.Type > 9) Then
```

```
        fldComputViaje.Size = Val(Mid$(szAux, iPosComaAux + 1))
```

```
    End If
```

```
End If
```

```
Public Sub ObtDscErrCom(iNumeroDeError As Integer)
```

```
' Obtiene las descripciones en español de los errores atrapables para el
```

```
' Control de comunicaciones.
```

```
Select Case iNumeroDeError
```

```
Case 8000 ' Operation not valid while the port is opened
```

```
    gsDscError = "Operación no válida mientras el puerto está siendo"
```

```
    gsDscError = gsDscError & "abierto"
```

```
Case 8001 ' Timeout value must be greater than zero
```

```
    gsDscError = "El valor del timeout debe ser mayor que cero"
```

```
Case 8002 ' Invalid Port Number
```

```
    gsDscError = "Número de Puerto Inválido"
```

```
Case 8003 ' Property available only at run time
```

```
    gsDscError = "Propiedad disponible sólo en tiempo de ejecución"
```

```
Case 8004 ' Property is read only at runtime
```

```
    gsDscError = "La propiedad es de sólo lectura en tiempo de"
```

```
    gsDscError = gsDscError & "ejecución"
```

```
Case 8005 ' Port already open
```

```
    gsDscError = "Puerto ya abierto"
```

```
Case 8006 ' The device identifier is invalid or unsupported
```

```
    gsDscError = "El identificador del dispositivo es inválido o no"
```

```
    gsDscError = gsDscError & "es soportado"
```

```
Case 8007 ' The Device 's baud rate is unsupported
gsDscError = "La velocidad en bauds del dispositivo no es"
gsDscError = gsDscError & " soportado"
Case 8008 ' The specified byte size is invalid
gsDscError = "El tamaño de byte especificado es inválido"
Case 8009 ' The default parameters are in error
gsDscError = "Los parámetros por default son incorrectos"
Case 8010 ' The hardware is not available (locked by another
' device)
gsDscError = "El hardware no está disponible (bloqueo por otro"
gsDscError = gsDscError & " dispositivo)"
Case 8011 ' The function cannot allocate the queues
gsDscError = "La función no puede asignar las colas"
Case 8012 ' The device is not open
gsDscError = "El dispositivo no está abierto"
Case 8013 ' The device is already open
gsDscError = "El dispositivo ya está abierto"
Case 8014 ' Could not enable comm notification
gsDscError = "No se puede habilitar notificación comm"
Case 8015 ' Could not set comm state
gsDscError = "No se puede establecer estado comm"
Case 8016 ' Could not set comm event ntask
gsDscError = "No se puede establecer máscara comm event"
Case 8018 ' Operation valid only when the port is open
gsDscError = "Operación válida únicamente cuando el puerto está"
gsDscError = gsDscError & " abierto"
Case 8019 ' Device busy
gsDscError = "Dispositivo ocupado"
Case 8020 ' Error reading comm device
gsDscError = "Error leyendo dispositivo comm"
Case Else
gsDscError = Empty
End Select
```

```
End Sub
```

```
Sub Txt2Tb(sNombreTabla As String, _
sNombreArchivo As String, _
iRegistroInicial As Integer, _
iRegistroFinal As Integer)
```

```
Dim iSigArcDsp As Integer
Dim bArchivoAbierto As Integer
Dim iRegistroNumero As Integer

Dim iLocalizacion As Integer

Dim sRegistro As String

Dim iInicioRegistro As Integer
Dim iFinRegistro As Integer
```

```
Dim iInicioData As Integer
Dim iFinData As Integer

Dim iIndice As Integer
Dim iNumDatReg As Integer

'gdbDatabase
'AnyTable As typTabla
Dim dbxComputViaje As Database
' Dim defComputViaje As TableDef
Dim rstComputViaje As Recordset

Dim j As Integer

' array de ayuda
Const MAXTEMPCOLS = 12
Dim sztemp(0 To MAXTEMPCOLS) As String

Dim asDatosRegistro() As String

bArchivoAbierto = AbrirArchivo(iSigArcDsp, sNombreArchivo, "Input")

If (bArchivoAbierto = True) Then

'Call db_Open(szFilename, AnyTable)

Set rstComputViaje = gdbDatabase.OpenRecordset(sNombreTabla, _
                                                dbOpenTable)

End If

rstComputViaje.Index = iIndicePorDefault

'Set idxIndice = rstComputViaje.Index

iRegistroNumero = 1

Do Until (EOF(iSigArcDsp) = True)
iInicioData = 1

Line Input #iSigArcDsp, sRegistro

If (iRegistroFinal >= iRegistroInicial) Then
If (iRegistroNumero >= iRegistroInicial And _
iRegistroNumero <= iRegistroFinal) Then
sRegistro = sRegistro & Chr$(vbKeyTab)

iNumDatReg = 0

Do While (iInicioData < Len(sRegistro))
iFinData = InStr(iInicioData, sRegistro, Chr$(vbKeyTab))
iInicioData = iFinData + 1
iNumDatReg = iNumDatReg + 1
Loop
```

```
ReDim asDatosRegistro(1 To iNumDatReg)

For iIndice = 1 To iNumDatReg
  asDatosRegistro(iIndice) = Empty
Next iIndice

iIndice = 0
iInicioDato = 1

Do While iInicioDato < Len(sRegistro)
  ' Localiza el siguiente caracter tabulador dentro de la cadena

  iFinDato = InStr(iInicioDato, sRegistro, Chr$(vbKeyTab))
  iIndice = iIndice + 1
  asDatosRegistro(iIndice) = Mid$(sRegistro, iInicioDato, iFinDato - iInicioDato)
  iInicioDato = iFinDato + 1
Loop

' Localiza en un objeto Recordset tipo tabla, un registro que
' satisfice el criterio especificado por el indice actual:

rstComputViaje.Seek "=", asDatosRegistro(1), asDatosRegistro(2)

If (rstComputViaje.NoMatch = True) Then
  rstComputViaje.AddNew
Else
  rstComputViaje.Edit
  Beep
  MsgBox "Datos ya descargados", 48
  Exit Do
End If

For j = 0 To iIndice - 1
  rstComputViaje.Fields(j) = asDatosRegistro(j + 1)
Next j

rstComputViaje.Update
End If
iRegistroNumero = iRegistroNumero + 1
Else
  gsMensaje = "No existen datos que insertar en la tabla "
  gsMensaje = gsMensaje & sNombreTabla

  MsgBox "No existen datos que insertar en la tabla " & sNombreTabla, 48
  Call MsgBox(Chr(gsMensaje), vbExclamation + vbOKOnly, "Error", _
    App.HelpFile, 1)
  'txtEstatus.Text = "Termino recepcion de datos"
  'ndiComputDeViaje.stbCompViaje.Style = sbrSimple
  'ndiComputDeViaje.stbCompViaje.SimpleText = "Termino recepcion de datos"
End If
Loop
```

```
rstComputViaje.Close
```

```
Close iSigArcDsp
```

```
End Sub
```

Option Explicit

```
* Variables Publicas
```

```
* -----
```

- * Son utilizadas a nivel de módulo para declarar variables públicas y asignar espacio de almacenamiento.
- * Las variables declaradas utilizando la sentencia Public están disponibles para todos los procedimientos en todos los módulos en todas las aplicaciones.

```
Public Const NULL_STRING = ""
```

```
Public Const NULL_INTEGER = 0
```

```
Public Const KEY_ENTER = 13
```

```
Public Const NOM_BDD = "COMPVIAJ.MDB"
```

```
Public Const OFN_OVERWRITEPROMPT = &H2&
```

```
Public Const OFN_EXTENSIONDIFFERENT = &H400&
```

```
Public Const OFN_PATHMUSTEXIST = &H800&
```

```
Public Const OFN_HIDEREADONLY = &H4&
```

- * Constante que establece el número de segundos que las pantallas de presentación serán mostradas. El valor está expresado en milisegundos.

- * Cinco mil MILISEGUNDOS equivalen a 5 segundos

```
Public Const MILISEGUNDOS = 5000
```

```
Public Const ENCENDIDO = 1
```

```
Public Const APAGADO = 0
```

```
Public Const CERO As Integer = 0
```

```
Public Const UNO As Integer = 1
```

```
Public Const DOS As Integer = 2
```

```
Public Const TRES As Integer = 3
```

```
Public Const CUATRO As Integer = 4
```

```
Public Const CINCO As Integer = 5
```

```
Public Const SEIS As Integer = 6
```

```
Public Const SIETE As Integer = 7
```

```
Public Const OCHO As Integer = 8
```

```
Public Const NUEVE As Integer = 9
```

```
* ... Mask Edit Box
```

```
Public Const INCREMENTAR_FECHA_EN_UNO As Integer = 1
```

Public Const DECREMENTAR_FECHA_EN_UNO As Integer = -1

Public Const MASCARA_DE_ENTRADA_FECHA_NULA As String = " _/ _/ _"

' Data Access constants

' Option argument values (CreateDynaset, etc)

Global Const DB_DENYWRITE = &H1
Global Const DB_DENYREAD = &H2
Global Const DB_READONLY = &H4
Global Const DB_APPENDONLY = &H8
Global Const DB_INCONSISTENT = &H10
Global Const DB_CONSISTENT = &H20
Global Const DB_SQLPASSTHROUGH = &H40

' SetDataAccessOption

Global Const DB_OPTIONINIPATH = 1

' Field Attributes

Global Const DB_FIXEDFIELD = &H1
Global Const DB_VARIABLEFIELD = &H2
Global Const DB_AUTOINCRFIELD = &H10
Global Const DB_UPDATABLEFIELD = &H20

' Field Data Types

Global Const DB_BOOLEAN = 1
Global Const DB_BYTE = 2
Global Const DB_INTEGER = 3
Global Const DB_LONG = 4
Global Const DB_CURRENCY = 5
Global Const DB_SINGLE = 6
Global Const DB_DOUBLE = 7
Global Const DB_DATE = 8
Global Const DB_TEXT = 10
Global Const DB_LONGBINARY = 11
Global Const DB_MEMO = 12

' TableDef Attributes

Global Const DB_ATTACHEXCLUSIVE = &H10000
Global Const DB_ATTACHSAVEMPWD = &H20000
Global Const DB_SYSTEMOBJECT = &H80000002
Global Const DB_ATTACHEDTABLE = &H40000000
Global Const DB_ATTACHEDODBC = &H20000000

' ListTables TableType

Global Const DB_TABLE = 1
Global Const DB_QUERYDEF = 5
Global Const DB_SORTGENERAL = 256 ' Sort by EFGPI rules (English, French, German, Portuguese, Italian)
Global Const DB_SORTSPANISH = 258 ' Sort by Sparush rules
Global Const DB_SORTDUTCH = 259 ' Sort by Dutch rules

```
Global Const DB_SORTSWEDFIN = 260 ' Sort by Swedish, Finnish rules
Global Const DB_SORTNORWDAN = 261 ' Sort by Norwegian, Danish rules
Global Const DB_SORTICELANDIC = 262 ' Sort by Icelandic rules
Global Const DB_SORTPDXINTL = 4096 ' Sort by Paradox international rules
Global Const DB_SORTPDXSWE = 4097 ' Sort by Paradox Swedish, Finnish rules
Global Const DB_SORTUNDEFINED = -1 ' Sort rules are undefined or unknown
```

```
Public Function SumarSoloDigitos(sRegistro As String) As Integer
```

```
    Dim iInicioDato As Integer
    Dim iSumaDigitos
```

```
    iInicioDato = 1
```

```
    Do While iInicioDato < Len(sRegistro)
```

```
        If ((Asc(Mid$(sRegistro, iInicioDato, 1)) >= 48) And _
            (Asc(Mid$(sRegistro, iInicioDato, 1)) <= 57)) Then
            iSumaDigitos = iSumaDigitos + CInt(Mid$(sRegistro, iInicioDato, 1))
        End If
```

```
        iInicioDato = iInicioDato + 1
    Loop
```

```
    SumarSoloDigitos = iSumaDigitos
```

```
End Function
```

APENDICE B

HOJAS DE ESPECIFICACION DE COMPONENTES ELECTRONICOS

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.
 Input Voltage (V_{IO}) = 5V, 12V and 15V 35V
 Internal Power Dissipation (Note 1) Internally Limited
 Operating Temperature Range (T_A) 0°C to +70°C

Maximum Junction Temperature (K Package) 150°C
 (T Package) 150°C
 Storage Temperature Range -65°C to +150°C
 Lead Temperature (Soldering, 10 sec)
 TO-3 Package K 300°C
 TO-220 Package T 230°C

Electrical Characteristics LM78XXC (Note 2) 0°C ≤ T_J ≤ 125°C unless otherwise noted

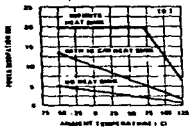
Output Voltage			5V			12V			15V			Units
Input Voltage (unless otherwise noted)			10V			18V			23V			
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V_{O1}	Output Voltage	T _J = 25°C, 5 mA ≤ I _O ≤ 1 A	4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		I _O = 15V, 5 mA ≤ I _O ≤ 1 A	4.75	5.25	11.4	12.6	14.25	15.75	V			
		V _{MIN} ≤ V _{IN} ≤ V _{MAX}	(7.5 ≤ V _{IN} ≤ 20)	(14.5 ≤ V _{IN} ≤ 27)	(17.5 ≤ V _{IN} ≤ 30)	V						
ΔV_{O1}	Line Regulation	I _O = 500 mA	3 50			4 120			4 150			mv
		T _J = 25°C	(7 ≤ V _{IN} ≤ 25)			14.5 ≤ V _{IN} ≤ 30)			(17.5 ≤ V _{IN} ≤ 30)			
		0°C ≤ T _J ≤ +125°C	50			120			150			mv
		ΔV_{IN}	(8 ≤ V _{IN} ≤ 20)			(15 ≤ V _{IN} ≤ 27)			(18.5 ≤ V _{IN} ≤ 30)			
		I _O = 1 A	50			170			150			mv
		T _J = 25°C	(7.5 ≤ V _{IN} ≤ 20)			(14.8 ≤ V _{IN} ≤ 27)			(17.7 ≤ V _{IN} ≤ 30)			
ΔV_{O2}	Load Regulation	T _J = 25°C	5 60			6 120			6 150			mv
		5 mA ≤ I _O ≤ 1.5 A	10 50			12 220			12 150			
		250 mA ≤ I _O ≤ 750 mA	25			60			75			
		0°C ≤ T _J ≤ +125°C	(8 ≤ V _{IN} ≤ 12)			(16 ≤ V _{IN} ≤ 22)			(20 ≤ V _{IN} ≤ 25)			
		5 mA ≤ I _O ≤ 1 A, 0°C ≤ T _J ≤ +125°C	50			120			150			
		0°C ≤ T _J ≤ +125°C	6.5			8			8.5			
I_{O1}	Quiescent Current	I _O = 1 A	8			8			8			mA
		T _J = 25°C	6.5			8.5			8.5			
		0°C ≤ T _J ≤ +125°C	0.5			1.0			0.5			
ΔI_{O1}	Quiescent Current Change	5 mA ≤ I _O ≤ 1 A	1.0			1.0			1.0			mA
		T _J = 25°C, I _O = 1 A	(7.5 ≤ V _{IN} ≤ 20)			(14.8 ≤ V _{IN} ≤ 27)			(17.9 ≤ V _{IN} ≤ 30)			
		V _{MIN} ≤ V _{IN} ≤ V _{MAX}	1.0			1.0			1.0			
		I _O = 500 mA, 0°C ≤ T _J ≤ +125°C	(7 ≤ V _{IN} ≤ 25)			(14.5 ≤ V _{IN} ≤ 30)			(17.5 ≤ V _{IN} ≤ 30)			
V_{IN}	Output Noise Voltage	T _A = 25°C, 10 Hz ≤ f ≤ 100 kHz	40			75			54			mV
		ΔV_{IN}	40			75			54			
ΔV_{OUT}	Ripple Rejection	f = 120 Hz	62 80			55 72			54 70			dB
		I _O = 1 A, T _J = 25°C or	62			55			54			
		I _O = 500 mA, 0°C ≤ T _J ≤ +125°C	(8 ≤ V _{IN} = 16)			(15 ≤ V _{IN} ≤ 25)			(18.5 ≤ V _{IN} ≤ 28.5)			
		V _{MIN} ≤ V _{IN} ≤ V _{MAX}	2.0			2.0			2.0			
R_{O1}	Dropout Voltage	T _J = 25°C, I _{O,OUT} = 1 A	8			18			15			V
		f = 1 kHz	2.1			1.5			1.2			
		T _J = 25°C	2.4			2.4			2.4			
		T _J = 25°C	0.6			1.5			1.8			
		Average TC of V _{O,OUT}	0.6			1.5			1.8			
V_{IN}	Input Voltage Required to Maintain Line Regulation	T _J = 25°C, I _O = 1 A	7.5			14.6			17.7			V

Note 1: Thermal resistance of the TO-3 package (R_{θJC}) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 4°C/W junction to case and 50°C/W case to ambient.

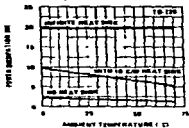
Note 2: All characteristics are measured with capacitor across the input of 0.22 μF, and a capacitor across the output of 0.1 μF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques (t_{ON} = 10 ms, duty cycle = 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

Typical Performance Characteristics

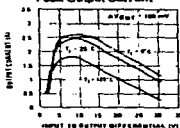
Maximum Average Power Dissipation



Maximum Average Power Dissipation



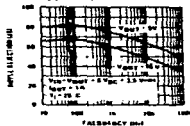
Peak Output Current



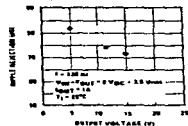
Output Voltage (Normalized to 1V at 1) - 25°C



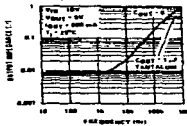
Ripple Rejection



Ripple Rejection



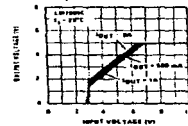
Output Impedance



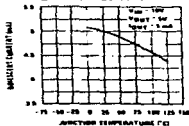
Dropout Voltage



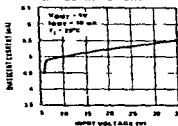
Dropout Characteristics



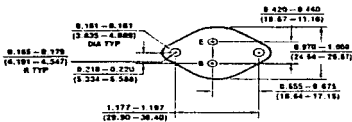
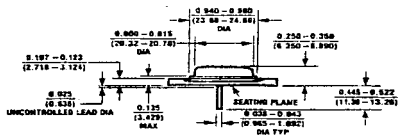
Quiescent Current



Quiescent Current



Physical Dimensions inches (millimeters)



KC02A (REV. 6/81)

Aluminum Metal Can Package (KC)
 Order Number LM7805CK, LM7812CK or LM7815CK
 NS Package Number KC02A

MM54HC00/MM74HC00

Quad 2-Input NAND Gate

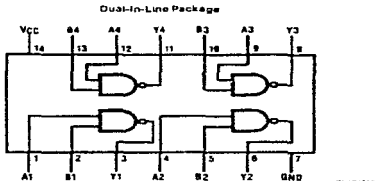
General Description

These NAND gates utilize advanced silicon-gate CMOS technology to achieve operating speeds similar to LS-TTL gates with the low power consumption of standard CMOS integrated circuits. All gates have buffered outputs. All devices have high noise immunity and the ability to drive 10 LS-TTL loads. The 54HC/74HC logic family is functionally as well as pin-out compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

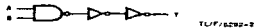
Features

- Typical propagation delay: 8 ns
- Wide power supply range: 2–5V
- Low quiescent current: 20 μ A maximum (74HC Series)
- Low input current: 1 μ A maximum
- Fanout of 10 LS-TTL loads

Connection and Logic Diagrams



Top View
Order Number MM54HC00 or MM74HC00



MM54HC00/MM74HC00 Quad 2-Input NAND Gate

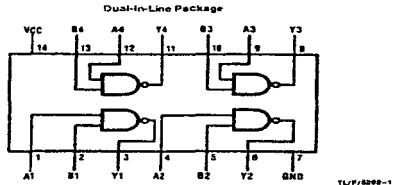
General Description

These NAND gates utilize advanced salicpn-gate CMOS technology to achieve operating speeds similar to LS-TTL gates with the low power consumption of standard CMOS integrated circuits. All gates have buffered outputs. All devices have high noise immunity and the ability to drive 10 LS-TTL loads. The 54HC/74HC logic family is functionally as well as pin-out compatible with the standard 54LS/74LS logic family. All outputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

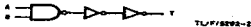
- Typical propagation delay: 8 ns
- Wide power supply range: 2–5V
- Low quiescent current: 20 μ A maximum (74HC Series)
- Low input current: 1 μ A maximum
- Fanout of 10 LS-TTL loads

Connection and Logic Diagrams



Top View

Order Number MM54HC00 or MM74HC00



Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributor for availability and specifications.

Supply Voltage (V_{CC})	-0.5 to +7.0V
DC Input Voltage (V_{IN})	-1.5 to V_{CC} + 1.5V
DC Output Voltage (V_{OUT})	-0.5 to V_{CC} + 0.5V
Clamp Diode Current (I_{PK} , IOK)	± 20 mA
DC Output Current, per pin (I_{OUT})	± 25 mA
DC V_{CC} or GND Current, per pin (I_{CC})	± 50 mA
Storage Temperature Range (T_{STG})	-85°C to +150°C
Power Dissipation (P_D) (Note 3)	800 mW
S.D. Package only	500 mW
Lead Temperature (T_L) (Soldering 10 seconds)	260°C

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	V_{CC}	74HC			Units	
				$T_A = -25^\circ\text{C}$	$T_A = -40$ to 85°C	$T_A = -55$ to 125°C		
				Guaranteed Limits				
				Typ				
V_{IH}	Minimum High Level Input Voltage		2.0V	1.5	1.5	1.5	V	
			4.5V	3.15	3.15	3.15	V	
			8.0V	4.2	4.2	4.2	V	
V_{IL}	Maximum Low Level Input Voltage**		2.0V	0.5	0.5	0.5	V	
			4.5V	1.35	1.35	1.35	V	
			8.0V	1.8	1.8	1.8	V	
V_{OH}	Minimum High Level Output Voltage	$V_{IN} = V_{IH}$ of V_{IL} $I_{OUT} \leq 20 \mu\text{A}$	2.0V	2.0	1.9	1.9	V	
			4.5V	4.5	4.4	4.4	V	
			8.0V	8.0	5.9	5.0	V	
		$V_{IN} = V_{IH}$ of V_{IL} $I_{OUT} \leq 4.0$ mA $I_{OUT} \leq 5.2$ mA	4.5V	4.2	3.98	3.84	3.7	V
			8.0V	5.7	5.48	5.34	5.2	V
V_{OL}	Maximum Low Level Output Voltage	$V_{IN} = V_{IH}$ of V_{IL} $I_{OUT} \leq 20 \mu\text{A}$	2.0V	0	0.1	0.1	V	
			4.5V	0	0.1	0.1	V	
			8.0V	0	0.1	0.1	V	
		$V_{IN} = V_{IH}$ of V_{IL} $I_{OUT} \leq 4.0$ mA $I_{OUT} \leq 5.2$ mA	4.5V	0.2	0.26	0.33	0.4	V
			8.0V	0.2	0.26	0.33	0.4	V
I_{IN}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	8.0V	± 1.0		± 1.0	μA	
I_{CC}	Maximum Quiescent Supply Current	$V_{IN} = V_{CC}$ or GND $I_{OUT} = 0 \mu\text{A}$	8.0V	2.0	2.0	4.0	μA	

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package — 13 mW/°C from 85°C to 85°C, ceramic "J" package — 13 mW/°C from 185°C to 125°C.

Note 4: For a power supply of 5V, 1.0% the worst case output voltages (V_{OH} and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{OH} and V_{OL} occur at $V_{CC} = 5.5$ V and 4.5V respectively (The V_{OH} value at 5.5V is 2.65V). The worst case leakage current (I_{CC} and I_{IN}) occur for CMOS at the higher voltage and so the 8.0V values should be used.

** V_{IL} limits are currently tested at 20% of V_{CC} . The above V_{IL} specification (30% of V_{CC}) will be implemented no later than D1, CY 89.

AC Electrical Characteristics $V_{CC} = 5V$, $T_A = 25^\circ C$, $C_L = 15 pF$, $t_r = t_f = 6 ns$

Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PHL} , t_{PLH}	Maximum Propagation Delay		8	15	ns

AC Electrical Characteristics $V_{CC} = 2.0V$ to $6.0V$, $C_L = 50 pF$, $t_r = t_f = 8 ns$ (unless otherwise specified)

Symbol	Parameter	Conditions	V_{CC}	$T_A = 25^\circ C$		74HC		54HC		Units
				Typ		$T_A = -40$ to $85^\circ C$		$T_A = -55$ to $125^\circ C$		
						Guaranteed Limits				
t_{PHL} , t_{PLH}	Maximum Propagation Delay		2.0V	45	90	110	134	ns		
			4.5V	9	18	23	27	ns		
			6.0V	8	15	19	23	ns		
t_{PLH} , t_{PHL}	Maximum Output Rise and Fall Time		2.0V	30	75	95	110	ns		
			4.5V	8	15	19	22	ns		
			6.0V	7	13	16	19	ns		
C_{PD}	Power Dissipation Capacitance (Note 5)	(per gate)	20						pF	
C_{IN}	Maximum Input Capacitance			5	10	10	10		pF	

Note 5: C_{PD} determines the no load dynamic power consumption $P_{DYN} = C_{PD} V_{CC}^2 f$, V_{CC} and the no load dynamic current consumption $I_{DYN} = C_{PD} V_{CC} f$, V_{CC} .

MM54HC04/MM74HC04 Hex Inverter

General Description

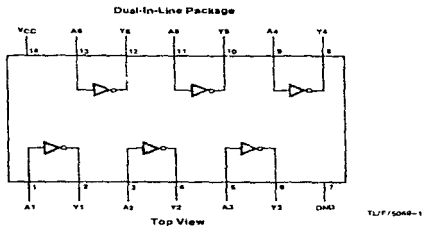
These inverters utilize advanced silicon-gate CMOS technology to achieve operating speeds similar to LS-TTL gates with the low power consumption of standard CMOS integrated circuits.

The MM54HC04/MM74HC04 is a triple buffered inverter. It has high noise immunity and the ability to drive 10 LS-TTL loads. The 54HC/74HC logic family is functionally as well as pin-out compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

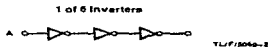
Features

- Typical propagation delay: 8 ns
- Fan out of 10 LS-TTL loads
- Quiescent power consumption, 10 μ W maximum at room temperature
- Low input current: 1 μ A maximum

Connection and Logic Diagrams



Order Number MM54HC04 or MM74HC04



Absolute Maximum Ratings (Notes 1 & 2)

if Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/distributors for availability and specifications.

Supply Voltage (V_{CC})	-0.5 to +7.0V
DC Input Voltage (V_{in})	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage (V_{out})	-0.5 to $V_{CC} + 0.5V$
Clamp Diode Current (I_{CL} , I_{CK})	± 20 mA
DC Output Current, per pin (I_{O1})	± 25 mA
DC V_{CC} or GND Current, per pin (I_{CC})	± 25 mA
Storage Temperature Range (T_{STG})	-65°C to +150°C

Power Dissipation (P_D)

(Note 3)	600 mW
S.O. Package only	500 mW

Lead Temperature (T_L)

(Soldering 10 seconds) 260°C

Operating Conditions

Supply Voltage (V_{CC})	Min	Max	Units
DC Input or Output Voltage (V_{in} , V_{out})	0	V_{CC}	V
Operating Temp. Range (T_A)			
MM74HC	-40	+85	°C
MM54HC	-55	+125	°C
Input Rise or Fall Times (t_r , t_f)	$V_{CC} = 2.0V$	1000	ns
	$V_{CC} = 4.5V$	500	ns
	$V_{CC} = 6.0V$	400	ns

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	V_{CC}	$T_A = 25^\circ C$			Units	
				Typ	74HC $T_A = -40$ to $85^\circ C$	54HC $T_A = -55$ to $125^\circ C$		
V_{IH}	Minimum High Level Input Voltage		2.0V	1.5	1.5	1.5	V	
			4.5V	3.15	3.15	3.15	V	
			6.0V	4.2	4.2	4.2	V	
V_{IL}	Maximum Low Level Input Voltage**		2.0V	0.5	0.5	0.5	V	
			4.5V	1.35	1.35	1.35	V	
			6.0V	1.8	1.8	1.8	V	
V_{OH}	Minimum High Level Output Voltage	$V_{IH} = V_{IL}$ $ I_{OUT} > 20 \mu A$	2.0V	2.0	1.9	1.9	V	
			4.5V	4.5	4.4	4.4	V	
		6.0V	6.0	5.9	5.9	V		
		$V_{IH} = V_{IL}$ $ I_{OUT} > 4.0$ mA	4.5V	4.2	3.98	3.84	3.7	V
			6.0V	5.7	5.48	5.34	5.2	V
		V_{OH}	Maximum Low Level Output Voltage	$V_{IH} = V_{IH}$ $ I_{OUT} > 20 \mu A$	2.0V	0	0.1	0.1
4.5V	0				0.1	0.1	V	
6.0V	0	0.1	0.1	0.1	V			
V_{OL}	Maximum Low Level Output Voltage	$V_{IH} = V_{IH}$ $ I_{OUT} > 4.0$ mA	4.5V	0.2	0.26	0.33	0.4	V
			6.0V	0.2	0.26	0.33	0.4	V
		6.0V	0.2	0.26	0.33	0.4	V	
I_{IN}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	6.0V	± 0.1	± 1.0	± 1.0	μA	
I_{CC}	Maximum Quiescent Supply Current	$V_{IH} = V_{CC}$ or GND $I_{OUT} = 0 \mu A$	6.0V	2.0	2.0	4.0	μA	

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating in plastic "P" package: -12 mW/°C from 85°C to 85°C; no derate; "J" package: -12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of 5V $\pm 10\%$, the worst case output voltages (V_{OH} and V_{OL}) occur for I_C of 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{OH} and V_{OL} occur at $V_{CC} = 5.5V$ and 4.5V respectively. (The V_{OH} value at 5.5V is 3.85V.) The worst case leakage currents (I_{IN} , I_{CC} and I_{O1}) occur for CMOS at the higher voltage and at the 6.0V value should be used.

** V_{IL} limits are currently tested at 30% of V_{CC} . The above V_{IL} specifications (30% of V_{CC}) will be implemented no later than CI, CY 89.

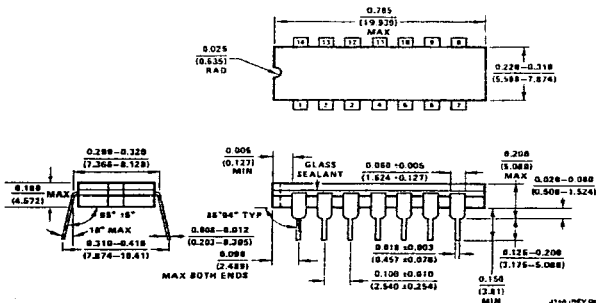
AC Electrical Characteristics $V_{CC} = 5V$, $T_A = 25^\circ C$, $C_L = 15 pF$, $t_r = t_f = 6 ns$

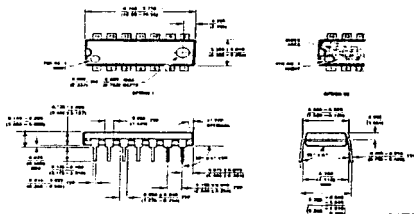
Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PHL} , t_{PLH}	Maximum Propagation Delay		8	15	ns

AC Electrical Characteristics $V_{CC} = 2.0V$ to $6.0V$, $C_L = 50 pF$, $t_r = t_f = 6 ns$ (unless otherwise specified)

Symbol	Parameter	Conditions	V_{CC}	Guaranteed Limits			Units
				$T_A = 25^\circ C$	$T_A = -40$ to $85^\circ C$	$T_A = -55$ to $125^\circ C$	
t_{PHL} , t_{PLH}	Maximum Propagation Delay		2.0V	55	95	120	ns
			4.5V	11	19	24	ns
			6.0V	9	16	20	ns
t_{RLH} , t_{FHL}	Maximum Output Rise and Fall Time		2.0V	30	75	95	ns
			4.5V	6	15	19	ns
			6.0V	7	13	16	ns
C_{PD}	Power Dissipation Capacitance (Note 5)	(per gate)	20				pF
C_{IN}	Maximum Input Capacitance		5	10	10	10	pF

Note 5: C_{PD} determines the no load dynamic power consumption. $P_D = C_{PD} V_{CC}^2 f$, C_{IN} , V_{CC} and the no load dynamic current consumption. $I_D = C_{IN} V_{CC} f$.

Physical Dimensions inches (millimeters)


Physical Dimensions inches (millimeters) (Continued)

Molex Dual-In Line Package (N)
Order Number MM74HC04N
See NS Package M14A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

2 National Semiconductor Corporation
1111 West Sherman Road
Arlington, TX 76010
Tel: (800) 272-9693
Fax: (800) 737-7618

National Semiconductor Europe
Fax: (+49 0-180-530 66 66)
E-mail: info@nsc-europe.com
Dutch: Tel: (+49 0-180-530 66 66)
English: Tel: (+49 0-180-532 76 22)
Finnish: Tel: (+49 0-180-532 65 66)
Hebrew: Tel: (+49 0-180-532 18 60)

National Semiconductor Hong Kong Ltd.
12th Floor, Street 8 Block
Queen Center, 3 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2722-1800
Fax: (852) 2722-0000

National Semiconductor Japan Ltd.
Tel: 81-045-296-2306
Fax: 81-045-296-2308

MM54HC32/MM74HC32 Quad 2-Input OR Gate

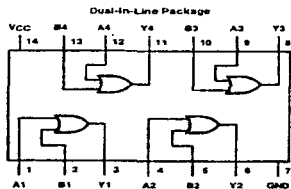
General Description

These OR gates utilize advanced silicon-gate CMOS technology to achieve operating speeds similar to LS-TTL gates with the low power consumption of standard CMOS integrated circuits. All gates have buffered outputs, providing high noise immunity and the ability to drive 10 LS-TTL loads. The 54HC/74HC logic family is functionally as well as pin-out compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

- Typical propagation delay: 10 ns
- Wide power supply range: 2-6V
- Low quiescent current: 20 μ A maximum (74HC Series)
- Low input current: 1 μ A maximum
- Fanout of 10 LS-TTL loads

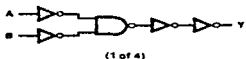
Connection and Logic Diagrams



TL778132-1

Top View

Order Number MM54HC32 or MM74HC32


 $Y = A + B$

TL778132-2

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	-0.5 to +7.0V
DC Input Voltage (V_{IN})	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage (V_{OUT})	-0.5 to $V_{CC} + 0.5V$
Clamp Diode Current (I_{PK} , I_{CK})	± 20 mA
DC Output Current, per pin (I_{OUT})	± 25 mA
DC V_{CC} or GND Current, per pin (I_{CC})	± 50 mA
Storage Temperature Range (T_{STG})	-65°C to +150°C
Power Dissipation (P_D)	
(Note 3)	600 mW
S.O. Package only	500 mW
Lead Temperature (T_L)	
(Soldering 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{CC})	2	6	V
DC Input or Output Voltage (V_{IN} , V_{OUT})	0	V_{CC}	V
Operating Temp. Range (T_A)			
MM74HC	-40	+85	°C
MM54HC	-55	+125	°C
Input Rise or Fall Times (t_r , t_f)		1000	ns
$V_{CC} = 2.0V$		500	ns
$V_{CC} = 4.5V$		400	ns
$V_{CC} = 6.0V$			

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	V_{CC}	74HC			Units	
				$T_A = -25^\circ C$	$T_A = -40$ to $85^\circ C$	$T_A = -55$ to $125^\circ C$		
V_{IH}	Minimum High Level Input Voltage		2.0V	1.5	1.5	1.5	V	
			4.5V	3.15	3.15	3.15	V	
			6.0V	4.2	4.2	4.2	V	
V_{IL}	Maximum Low Level Input Voltage**		2.0V	0.5	0.5	0.5	V	
			4.5V	1.35	1.35	1.35	V	
			6.0V	1.8	1.8	1.8	V	
V_{OH}	Minimum High Level Output Voltage	$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} \leq 20 \mu A$	2.0V	2.0	1.9	1.9	1.9	V
			4.5V	4.5	4.4	4.4	4.4	V
			6.0V	6.0	5.9	5.9	5.9	V
			4.5V	4.7	3.98	3.94	3.7	V
			6.0V	5.2	5.48	5.34	5.2	V
V_{OL}	Maximum Low Level Output Voltage	$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} \leq 4.0$ mA $I_{OUT} \leq 5.2$ mA	2.0V	0	0.1	0.1	0.1	V
			4.5V	0	0.1	0.1	0.1	V
			6.0V	0	0.1	0.1	0.1	V
			4.5V	0.2	0.26	0.33	0.4	V
			6.0V	0.2	0.26	0.33	0.4	V
I_{IH}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	6.0V	1.0	+1.0	+1.0	μA	
I_{CC}	Maximum Quiescent Supply Current	$V_{IN} = V_{CC}$ or GND $I_{OUT} = 0 \mu A$	6.0V	2.0	2.0	4.0	μA	

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

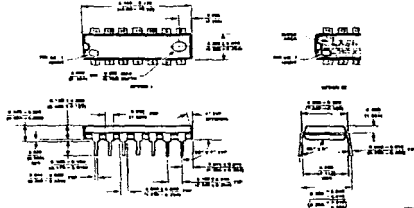
Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package — 12 mW/°C from 85°C to 85°C; ceramic "J" package — 12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of 5V $\pm 10\%$ the worst case output voltages (V_{OH} and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{OH} and V_{OL} occur at $V_{CC} = 6.0V$ and 4.5V respectively. (The V_{OH} value at 6.0V is 3.95V.) The worst case leakage current (I_{CC} and I_{OH}) occur for CMOS at the higher voltage and so the 6.0V values should be used.

** V_{IL} limits are currently tested at 20% of V_{CC} . The above V_{IL} specification (30% of V_{CC}) will be implemented no later than D1, C1, B1.

Physical Dimensions inches (millimeters) (Continued)



Dual-In-Line Package (N)
Order Number MM74HC32N
MS Package Number N14A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION, AS USED HEREIN:

- Life support devices or systems are devices or systems which: (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
- A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor
Corporation
111 West Garden Street
Folsom, CA 95630
Tel: (916) 752-0000
Fax: (916) 752-7010

Synapse Semiconductor
Europe
Fax: (+49 0-180-530 98 98)
E-mail: oscar@synapse.com
Dutch: Tel: (+31 0-180-530 98 98)
English: Tel: (+49 0-180-530 98 98)
Finnish: Tel: (+358 0-180-530 98 98)
German: Tel: (+49 0-180-530 16 80)

National Semiconductor
Hong Kong Ltd.
12th Floor, Shing Lee Building
Corner Connaught & Canton Road
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1800
Fax: (852) 2737-2600

National Semiconductor
Japan Ltd.
Tel: #1-043-296-2000
Fax: #1-043-296-2400

MM54HC138/MM74HC138 3-to-8 Line Decoder

General Description

This decoder utilizes advanced silicon-gate CMOS technology, and is well suited to memory address decoding or data routing applications. The circuit features high noise immunity and low power consumption usually associated with CMOS circuitry, yet has speeds comparable to low power Schottky TTL logic.

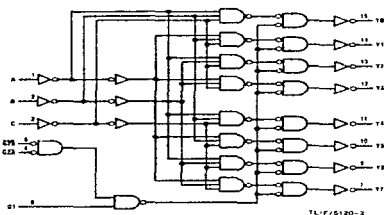
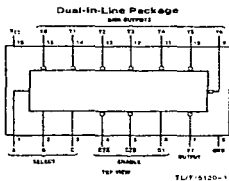
The MM54HC138/MM74HC138 has 3 binary select inputs (A, B, and C). If the device is enabled these inputs determine which one of the eight normally high outputs will go low. Two active low and one active high enables (G1, G2A and G2B) are provided to ease the cascading of decoders.

The decoder's outputs can drive 10 low power Schottky TTL equivalent loads, and are functionally and pin equivalent to the 74LS138/74ALS138. All inputs are protected from damage due to static discharge by diodes to V_{CC} and ground.

Features

- Typical propagation delay: 20 ns
- Wide power supply range: 2V-6V
- Low quiescent current: 80 μ A maximum (74HC Series)
- Low input current: 1 μ A maximum
- Fanout of 10 LS-TTL loads

Connection and Logic Diagrams



Order Number MM54HC138
or MM74HC138

Truth Table

Inputs			Outputs							
Enable	Select									
G1	G2 ^a	C B A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X X X	H	H	H	H	H	H	H	H
L	X	X X X	H	H	H	H	H	H	H	H
H	L	L L L	H	H	H	H	H	H	H	H
H	L	L L H	H	H	L	H	H	H	H	H
H	L	L H L	H	H	H	L	H	H	H	H
H	L	L H H	H	H	H	H	L	H	H	H
H	H	L L H	H	H	H	H	H	L	H	H
H	H	L H L	H	H	H	H	H	H	L	H
H	H	L H H	H	H	H	H	H	H	H	L
H	H	H L L	H	H	H	H	H	H	H	L
H	H	H L H	H	H	H	H	H	H	H	L
H	H	H H L	H	H	H	H	H	H	H	L
H	H	H H H	H	H	H	H	H	H	H	L

G3 = G2A = G2B

H = high level, L = low level, X = don't care

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	-0.5 to +7.0V
DC Input Voltage (V_{in})	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage (V_{out})	-0.5 to $V_{CC} + 0.5V$
Clamp Diode Current (I _{CL})	+20 mA
DC Output Current, per pin (I _{OUT})	+25 mA
DC V_{CC} or GND Current, per pin (I _{CC})	+50 mA
Storage Temperature Range (T_{STG})	-85°C to +150°C
Power Dissipation (P _D)	
(Nominal)	600 mW
S.O. Package only	500 mW
Lead Temp. (T _L) (Soldering 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{CC})	2	6	V
DC Input or Output Voltage (See V_{OUT})	0	V_{CC}	V
Operating Temp. Range (T_A)			
MM74HC	-40	+85	°C
MM54HC	-55	+125	°C
Input Rise or Fall Times (N ₁)			ns
$V_{CC} = 2.0V$		1000	ns
$V_{CC} = 4.5V$		500	ns
$V_{CC} = 6.0V$		400	ns

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	V_{CC}	$T_A = 25^\circ C$			Units	
				Typ	74HC $T_A = -40$ to $85^\circ C$	54HC $T_A = -55$ to $125^\circ C$		
V_{OH}	Minimum High Level Input Voltage		2.0V	1.5	1.5	1.5	V	
			4.5V	3.15	3.15	3.15	V	
			6.0V	4.2	4.2	4.2	V	
V_{OL}	Maximum Low Level Input Voltage**		2.0V	0.5	0.5	0.5	V	
			4.5V	1.35	1.35	1.35	V	
			6.0V	1.8	1.8	1.8	V	
V_{OH}	Minimum High Level Output Voltage	$V_{OH} = V_{OH}$ or V_{IL} I _{OUT} ≤ 20 μA	2.0V	2.0	1.9	1.9	V	
			4.5V	4.5	4.4	4.4	V	
			6.0V	6.0	5.9	5.9	V	
		$V_{OH} = V_{OH}$ or V_{IL} I _{OUT} ≤ 4.0 mA I _{CC1} ≤ 5.2 mA	4.5V	4.2	3.98	3.84	3.7	V
			6.0V	5.7	5.48	5.34	5.2	V
V_{OL}	Maximum Low Level Output Voltage	$V_{OH} = V_{OH}$ or V_{IL} I _{OUT} ≤ 20 μA	2.0V	0	0.1	0.1	V	
			4.5V	0	0.1	0.1	V	
			6.0V	0	0.1	0.1	V	
		$V_{OH} = V_{OH}$ or V_{IL} I _{OUT} ≤ 4.0 mA I _{CC1} ≤ 5.2 mA	4.5V	0.2	0.26	0.33	0.4	V
			6.0V	0.2	0.26	0.33	0.4	V
I_{IN}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	6.0V	±0.1	±1.0	±1.0	μA	
I_{CC}	Maximum Quiescent Supply Current	$V_{IN} = V_{CC}$ or GND I _{OUT} = 0 μA	6.0V	8.0	8.0	160	μA	

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation Temperature (Soldering - plated) - 74HC: 12 mW/°C from 85°C to 85°C; 64HC: 12 mW/°C from 100°C to 125°C. Note 4: For a power supply of 5V ± 10%, the worst case output voltages (V_{OH} and V_{OL}) occur for I_{CC} at 4.5V. Thus the 4.5V values should be used when dealing with this supply. Worst case V_{OH} and V_{OL} occur at $V_{CC} = 3.5V$ and $4.5V$, respectively. (The V_{OH} value at 3.5V is 3.85V.) The worst case leakage current (see I_{CC} and I_{IN}) occurs for CMOS at the higher voltage and for the 8V output should be used.

** V_{OL} level is currently tested at 20% of V_{CC} . The above V_{OL} specification (20% of V_{CC}) will be implemented no later than CI, CV 99.

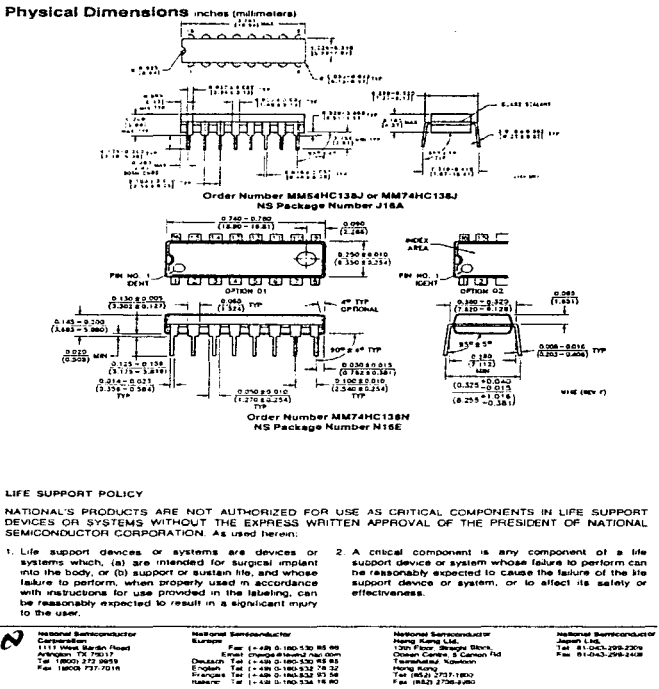
AC Electrical Characteristics $V_{CC} = 5V$, $T_A = 25^\circ C$, $C_L = 15 pF$, $t_r = t_f = 6 ns$

Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PLH}	Maximum Propagation Delay, Binary Select to any Output		18	25	ns
t_{PHL}	Maximum Propagation Delay, Binary Select to any Output		20	35	ns
t_{PHL} , t_{PLH}	Maximum Propagation Delay, G1 to any Output		18	25	ns
t_{PHL}	Maximum Propagation Delay G2A or G2B to Output		23	30	ns
t_{PLH}	Maximum Propagation Delay G2A or G2B to Output		18	25	ns

AC Electrical Characteristics $C_L = 50 pF$, $t_r = t_f = 6 ns$ (unless otherwise specified)

Symbol	Parameter	Conditions	V_{CC}	$T_A = 25^\circ C$		74HC		54HC		Units
				Typ	Guaranteed Limits	$T_A = -40$ to $85^\circ C$	$T_A = -55$ to $125^\circ C$			
t_{PLH}	Maximum Propagation Delay Binary Select to any Output Low to High		2.0V	75	150	189		224	ns	
				4.5V	15	30	38	45	ns	
				6.0V	13	26	32	38	ns	
t_{PHL}	Maximum Propagation Delay Binary Select to any Output High to Low		2.0V	100	200	252		298	ns	
				4.5V	20	40	50	60	ns	
				6.0V	17	34	43	51	ns	
t_{PHL} , t_{PLH}	Maximum Propagation Delay G1 to any Output		2.0V	75	150	189		224	ns	
				4.5V	15	30	38	45	ns	
				6.0V	13	26	32	38	ns	
t_{PHL}	Maximum Propagation Delay G2A or G2B to Output		2.0V	82	175	221		261	ns	
				4.5V	28	35	44	52	ns	
				6.0V	22	30	37	44	ns	
t_{PLH}	Maximum Propagation Delay G2A or G2B to Output		2.0V	75	150	189		224	ns	
				4.5V	15	30	38	45	ns	
				6.0V	13	26	32	38	ns	
t_{TLH} , t_{THL}	Output Rise and Fall Time		2.0V	30	75	95		110	ns	
				4.5V	8	15	19	22	ns	
				6.0V	7	13	16	19	ns	
C_{IN}	Maximum Input Capacitance		3	10	10		10	pF		
C_{PD}	Power Dissipation Capacitance	(Note 5)	75						pF	

Note 5: C_{PD} determines the no load dynamic power consumption $P_D = C_{PD} V_{CC}^2 f = I_{CC} V_{CC}$, and the no load dynamic current consumption $I_{CC} = C_{PD} V_{CC}^2 / V_{CC}$.



MM54HC373/MM74HC373 TRI-STATE® Octal D-Type Latch

General Description

These high speed octal D-type latches utilize advanced six-congate CMOS technology. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS-TTL loads. Due to the large output drive capability and the TRI-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

When the LATCH ENABLE input is high, the Q outputs will follow the D inputs. When the LATCH ENABLE goes low, data at the D inputs will be retained at the outputs until LATCH ENABLE returns high again. When a high logic level is applied to the OUTPUT CONTROL input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

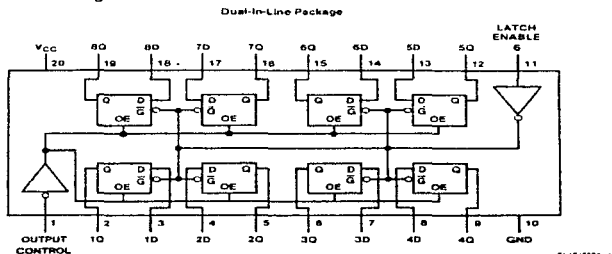
The 54HC/74HC logic family is speed function, and pin-out compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

The 54HC/74HC logic family is speed function, and pin-out compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

- Typical propagation delay: 18 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1 μ A maximum
- Low quiescent current: 80 μ A maximum (74 Series)
- Output drive capability: 15 LS-TTL loads

Connection Diagram



Truth Table

Output Control	Latch Enable	Data	373 Output
L	H	H	H
L	H	L	L
L	X	X	Z^1
H	X	X	Z^2

- H = high level, L = low level
- Z_1 = level of output before steady-state input conditions were established
- Z_2 = high impedance

TM 81117 is a registered trademark of National Semiconductor Corporation.

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributor for availability and specifications.

Supply Voltage (V_{CC})	-0.5 to +7.0V
DC Input Voltage (V_{in})	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage (V_{out})	-0.5 to $V_{in} + 1.5V$
Clamp Diode Current (I_{CL})	+25 mA
DC Output Current, per pin (I_{OL})	-35 mA
DC V_{CC} or GND Current, per pin (I_{CC})	+70 mA
Storage Temperature Range (T_{stg})	-55°C to +150°C

Power Dissipation (P_D)

(Note 3)	n/mW
5-C Package only	n/mW

Lead Temp (T_L) (Soldering) 10 seconds

Operating Conditions

Supply Voltage (V_{CC})	Min	2	Max	5	Units	V
DC Input or Output Voltage (V_{in} , V_{out})	0		V_{CC}			V
Operating Temp. Range (T_A)						
MM74HC	-40		+85			°C
MM54HC	-55		+125			°C
Input Rise or Fall Times (t_r , t_f)						
$V_{CC} = 2.0V$			1000			ns
$V_{CC} = 4.5V$			500			ns
$V_{CC} = 6.0V$			400			ns

DC Electrical Characteristics

Symbol	Parameter	Conditions	V_{CC}	$T_A = -25^\circ\text{C}$			$T_A = -40$ to 85°C		$T_A = -55$ to 125°C		Units
				Typ	Guaranteed Limits		Guaranteed Limits				
V_{IH}	Minimum High Level Input Voltage		2.0V	1.5	1.5	1.5	1.5			V	
			4.5V	3.15	3.15	3.15	3.15		V		
			6.0V	4.2	4.2	4.2	4.2		V		
V_{IL}	Maximum Low Level Input Voltage**		2.0V	0.5	0.5	0.5	0.5			V	
			4.5V	1.35	1.35	1.35	1.35		V		
			6.0V	1.8	1.8	1.8	1.8		V		
V_{OH}	Minimum High Level Output Voltage	$V_{IH} = V_{IH}$ or V_{IL} $I_{OUT} = 20 \mu\text{A}$	2.0V	2.0	1.9	1.9	1.9			V	
			4.5V	4.5	4.4	4.4	4.4		V		
			6.0V	6.0	5.9	5.9	5.9		V		
V_{OL}	Maximum Low Level Output Voltage	$V_{IH} = V_{IH}$ or V_{IL} $I_{OUT} = 20 \mu\text{A}$	2.0V	0	0.1	0.1	0.1			V	
			4.5V	0	0.1	0.1	0.1		V		
			6.0V	0	0.1	0.1	0.1		V		
V_{IL}	Maximum Input Current	$V_{IH} = V_{IH}$ or V_{IL} $I_{OUT} = 0 \text{ mA}$	4.5V	0.2	0.25	0.33	0.4			V	
			6.0V	0.2	0.25	0.33	0.4		V		
			6.0V	0.1	0.1	0.1	0.1		V		
I_{IN}	Maximum Input Current	$V_{IH} = V_{CC}$ or GND	6.0V	0.1	0.1	0.1	0.1		μA		
I_{OZ}	Maximum TRI-STATE Output Leakage Current	$V_{IH} = V_{IH}$ or V_{IL} , $I_{CC} = V_{IH}$ $V_{OUT} = V_{CC}$ or GND	6.0V	+0.5	1.5	1.5	1.5		μA		
I_{CC}	Maximum Quiescent Supply Current	$V_{IH} = V_{CC}$ or GND $I_{OUT} = 0 \text{ mA}$	6.0V	8.0	8.0	8.0	8.0		μA		

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating = plastic T1 package = 12 mW/°C from 85°C to 150°C; ceramic J package = 12 mW/°C from 150°C to 125°C.

Note 4: For a power supply of 0V, 100% maximum case output voltages (V_{OH} and V_{OL}) occur for HC at 4.5V. These are 4.5V values should be used when designing with this supply. Worst case V_{OH} and V_{OL} occur at $V_{CC} = 5.5V$ and 4.5V respectively. The worst case leakage current (I_{IL} , I_{OL} and I_{OZ}) occur for CMOS at the higher voltage and at the 6.0V values should be used.

** V_{IL} limits are currently tested at 20% of V_{CC} . The above V_{IL} specifications (20% of V_{CC}) will be implemented no later than Q1 '93.

AC Electrical Characteristics $V_{CC} = 5V$, $T_A = 25^\circ C$, $t_r = t_f = 6 ns$

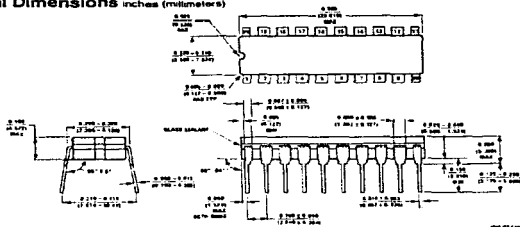
Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PHL} , t_{PLH}	Maximum Propagation Delay, Data to Q	$C_L = 45 pF$	18	25	ns
t_{PHL} , t_{PLH}	Maximum Propagation Delay, LE to Q	$C_L = 45 pF$	21	30	ns
t_{PZH} , t_{PZL}	Maximum Output Enable Time	$R_L = 1 k\Omega$ $C_L = 45 pF$	20	29	ns
t_{PHZ} , t_{PLZ}	Maximum Output Disable Time	$R_L = 1 k\Omega$ $C_L = 5 pF$	18	25	ns
t_S	Minimum Set Up Time			3	ns
t_H	Minimum Hold Time			10	ns
t_W	Minimum Pulse Width		3	16	ns

AC Electrical Characteristics $V_{CC} = 2.0 - 6.0V$, $C_L = 50 pF$, $t_r = t_f = 6 ns$ (unless otherwise specified)

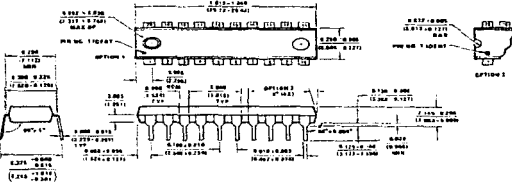
Symbol	Parameter	Conditions	V_{CC}	$T_A = 25^\circ C$		74HC		54HC		Units
				Typ		Guaranteed Limits		Guaranteed Limits		
t_{PHL} , t_{PLH}	Maximum Propagation Delay, Data to Q	$C_L = 50 pF$ $C_L = 150 pF$	2.0V	50	150	18R	25S	225	300	ns
			4.5V	22	30	37	45	45	ns	
			4.5V	30	40	50	60	60	ns	
			6.0V	19	26	31	39	39	ns	
t_{PHL} , t_{PLH}	Maximum Propagation Delay, LE to Q	$C_L = 50 pF$ $C_L = 150 pF$	2.0V	63	175	220	290	263	328	ns
			2.0V	110	225	280	328	328	ns	
			4.5V	25	35	44	52	52	ns	
			4.5V	35	45	55	64	64	ns	
t_{PZH} , t_{PZL}	Maximum Output Enable Time	$R_L = 1 k\Omega$ $C_L = 50 pF$ $C_L = 150 pF$	2.0V	21	30	37	45	45	53	ns
			2.0V	29	39	49	59	59	ns	
			4.5V	21	30	37	45	45	ns	
			4.5V	29	39	49	59	59	ns	
t_{PHZ} , t_{PLZ}	Maximum Output Disable Time	$R_L = 1 k\Omega$ $C_L = 50 pF$	2.0V	50	150	18R	25S	225	300	ns
			2.0V	80	200	250	300	300	ns	
			4.5V	21	30	37	45	45	ns	
			4.5V	30	40	50	60	60	ns	
t_{g}	Minimum Set Up Time	$C_L = 50 pF$ $C_L = 150 pF$	2.0V	19	26	31	39	39	47	ns
			2.0V	26	35	44	53	53	ns	
			4.5V	19	26	31	39	39	ns	
			4.5V	26	35	44	53	53	ns	
t_H	Minimum Hold Time	$C_L = 50 pF$ $C_L = 150 pF$	2.0V	5	5	5	5	5	5	ns
			2.0V	9	13	15	15	15	ns	
			4.5V	5	5	5	5	5	ns	
			4.5V	9	11	11	13	13	ns	
t_W	Minimum Pulse Width	$C_L = 50 pF$ $C_L = 150 pF$	2.0V	30	80	100	120	120	120	ns
			2.0V	30	80	100	120	120	ns	
			4.5V	10	18	20	24	24	ns	
			4.5V	9	14	18	20	20	ns	
t_{PHL} , t_{PLH}	Maximum Output Rise and Fall Time	$C_L = 50 pF$	2.0V	25	60	75	90	90	90	ns
			2.0V	25	60	75	90	90	ns	
			4.5V	7	12	15	18	18	ns	
			4.5V	6	10	13	15	15	ns	
C_{PD}	Power Dissipation Capacitance (Note 5)	(per I _{sch}) $C_C = V_{CC}$ $C_C = GND$	30	50					pF	
C_{IN}	Maximum Input Capacitance		5	10	10	10	10	10	pF	
C_{OUT}	Maximum Output Capacitance		15	20	20	20	20	20	pF	

 Note 5: C_C is the value that the total dynamic power consumption, $P_D = C_{IN} V_{CC}^2 f + C_{OUT} V_{CC}^2 f$, and the no-load minimum current consumption, $I_C = C_{PD} V_{CC} f$, is 100%

Physical Dimensions inches (millimeters)



Ceramic Dual-In-Line Package (J)
Order Number MM54HC373J or MM74HC373J
NS Package Number J20A



Molded Dual-In-Line Package (N)
Order Number MM74HC373N
NS Package Number N20A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor
Corporation
1111 West Herd Road
Folsom, CA 95630
Tel: (916) 252-0000
Fax: (916) 757-7018

National Semiconductor
Europe
Tel: (+49) 0-180-530 84 86
Email: europe@ns.com
Duesseldorf Tel: (+49) 0-180-530 86 85
Singapore Tel: (+65) 0-180-532 72 32
Frankfurt Tel: (+49) 0-180-532 60 90
Munich Tel: (+49) 0-180-534 16 80

National Semiconductor
Hong Kong Ltd.
1200, Cross Street, #09-00
Ocean Centre, 5 Canton Road
Tel: (852) 2533-1000
Fax: (852) 2796-9990

National Semiconductor
Japan Ltd.
Tel: 01-043-296-7330
Fax: 01-043-296-7428

NOTICE: Does not constitute any contract or warranty, and is subject to change without notice. All dimensions are nominal. National Semiconductor Corporation reserves the right to change these dimensions without notice.

MM54HC244/MM74HC244 Octal TRI-STATE® Buffer

General Description

These TRI-STATE buffers utilize advanced silicon-gate CMOS technology and are general purpose high speed non-inverting buffers. They possess high drive current outputs which enable high speed operation even when driving large bus capacitances. These circuits achieve speeds comparable to low power Schottky devices, while retaining the advantage of CMOS circuitry, i.e., high noise immunity, and low power consumption. All three devices have a layout of 15 LS-TTL equivalent inputs.

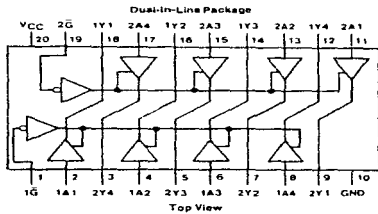
The MM54HC244/MM74HC244 is a non-inverting buffer and has two active low enables (1G and 2G). Each enable independently controls 4 buffers. This device does not have Schmitt trigger inputs.

All inputs are protected from damage due to static discharge by diodes to V_{CC} and ground.

Features

- Typical propagation delay: 14 ns
- TRI-STATE outputs for connection to system buses
- Wide power supply range: 2-6V
- Low quiescent supply current: 80 μ A (74 Series)
- Output current: 6 mA

Connection Diagram



TUF/6287-1

Order Number MM54HC244 or MM74HC244

Truth Table

*HC244

1G	1A	1Y	2G	2A	2Y
L	L	L	L	L	L
L	H	H	L	H	H
H	L	Z	H	L	Z
H	H	Z	H	H	Z

H = high level, L = low level, Z = high impedance

TRI-STATE® is a registered trademark of National Semiconductor Corp.

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	-0.5 to +7.0V
DC Input Voltage (V_{IN})	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage (V_{OUT})	-0.5 to $V_{CC} + 0.5V$
Clamp Diode Current (I_{FK}) (ok)	± 20 mA
DC Output Current, per pin (I_{OUT})	± 35 mA
DC V_{CC} or GND Current, per pin (I_{CC})	± 75 mA
Storage Temperature Range (T_{STG})	-65°C to +150°C
Power Dissipation (P_D)	
(Note 3)	600 mW
S.O. Package only	500 mW
Lead Temperature (T_L)	
(Soldering 10 seconds)	260°C

Operating Conditions

Supply Voltage (V_{CC})	Min 2	Max 6	Units V
DC Input or Output Voltage (V_{IN} , V_{OUT})	0	V_{CC}	V
Operating Temp. Range (T _{OP})	-40	+85	°C
M474HC		125	°C
M484HC		100	°C
Input Rise or Fall Times (t _r , t _f)	$V_{CC} = 2.0V$		ns
	$V_{CC} = 4.5V$		500
	$V_{CC} = 6.0V$		400

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	V_{CC}	$T_A = -25^\circ C$		$T_A = -40 \text{ to } 85^\circ C$		$T_A = -55 \text{ to } 125^\circ C$		Units
				Typ	Guaranteed Limits	Typ	Guaranteed Limits	Typ	Guaranteed Limits	
V_{IH}	Minimum High Level Input Voltage		2.0V	1.5	1.5	1.5	1.5	V		
			4.5V	3.15	3.15	3.15	V			
V_{IL}	Maximum Low Level Input Voltage**		2.0V	0.5	0.5	0.5	0.5	V		
			4.5V	1.35	1.35	1.35	V			
V_{OH}	Minimum High Level Output Voltage	$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} \leq 20 \mu A$	2.0V	2.0	1.9	1.9	1.9	V		
			4.5V	4.5	4.4	4.4	V			
V_{OL}	Maximum Low Level Output Voltage	$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} \leq 6.0$ mA $I_{OUT} \leq 7.8$ mA	4.5V	4.2	3.98	3.94	3.7	V		
			6.0V	5.7	5.48	5.34	5.2	V		
V_{IL}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	6.0V	± 0.1	0.1	0.1	0.1	μA		
			4.5V	0.1	0.1	0.1	V			
I_{OZ}	Maximum TRI-STATE Output Leakage Current	$V_{IN} = V_{OH}$ or V_{IL} $V_{OUT} = V_{CC}$ or GND $I_{S} = V_{IH}$	6.0V	± 0.5	0.5	0.5	0.5	μA		
			4.5V	0.2	0.26	0.33	0.4	V		
I_{CC}	Maximum Quiescent Supply Current	$V_{IN} = V_{CC}$ or GND $I_{OZ} = 0 \mu A$	6.0V	8.0	8.0	8.0	16.0	μA		

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package — 12 mW/°C from 85°C to 89°C; ceramic "J" package — 12 mW/°C from 100°C to 135°C.

Note 4: For a power supply of 5V $\pm 10\%$, the worst case output voltages (V_{OH} and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with the supply. Worst case V_{OH} and V_{OL} occur at $V_{CC} = 5.1V$ and 4.9V respectively. (The V_{OH} value at 5.1V is 3.85V.) The worst case leakage current (I_{OZ} and I_{CC}) occur for CM485 at the higher voltage and to the 6.0V values should be used.

** V_{IL} limits are currently tested at 20% of V_{CC} . The above V_{IL} specification (30% of V_{CC}) will be implemented no later than Q1 '89.

AC Electrical Characteristics MM54HC244/MM74HC244

 $V_{CC} = 5V$, $T_A = 25^\circ C$, $t_r = t_f = 6 ns$

Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PHL} , t_{PLH}	Maximum Propagation Delay	$C_L = 45 pF$	14	20	ns
t_{PZH} , t_{PZL}	Maximum Enable Delay to Active Output	$R_L = 1 k\Omega$ $C_L = 45 pF$	17	28	ns
t_{PHZ} , t_{PLZ}	Maximum Disable Delay from Active Output	$R_L = 1 k\Omega$ $C_L = 5 pF$	15	25	ns

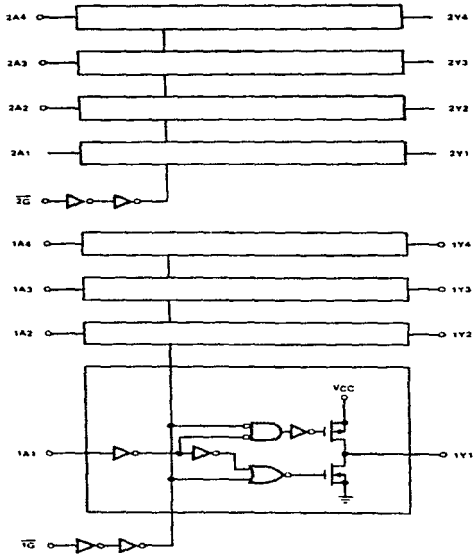
AC Electrical Characteristics $V_{CC} = 2.0V$ to $6.0V$, $C_L = 50 pF$, $t_r = t_f = 6 ns$ (unless otherwise specified)

Symbol	Parameter	Conditions	V_{CC}	$T_A = 25^\circ C$		74HC $T_A = -40$ to $85^\circ C$		54HC $T_A = -55$ to $125^\circ C$		Units
				Typ	Max	Max	Max	Max	Max	
t_{PHL} , t_{PLH}	Maximum Propagation Delay	$C_L = 50 pF$	2.0V	58	115	145	171	ns	ns	
		$C_L = 150 pF$	2.0V	83	165	208	246	ns	ns	
		$C_L = 50 pF$	4.5V	14	23	23	34	ns	ns	
		$C_L = 150 pF$	4.5V	17	33	42	49	ns	ns	
		$C_L = 50 pF$	6.0V	10	20	25	29	ns	ns	
		$C_L = 150 pF$	6.0V	14	28	35	42	ns	ns	
t_{PZH} , t_{PZL}	Maximum Output Enable Time	$R_L = 1 k\Omega$								
		$C_L = 50 pF$	2.0V	75	150	188	224	ns	ns	
		$C_L = 150 pF$	2.0V	100	200	252	298	ns	ns	
		$C_L = 50 pF$	4.5V	15	30	38	45	ns	ns	
		$C_L = 150 pF$	4.5V	30	40	50	60	ns	ns	
		$C_L = 50 pF$	6.0V	7	26	32	38	ns	ns	
		$C_L = 150 pF$	6.0V	17	34	43	51	ns	ns	
t_{PHZ} , t_{PLZ}	Maximum Output Disable Time	$R_L = 1 k\Omega$	2.0V	75	150	188	224	ns	ns	
		$C_L = 50 pF$	4.5V	15	30	38	45	ns	ns	
		$C_L = 50 pF$	6.0V	13	26	32	38	ns	ns	
t_{TLH} , t_{TFL}	Maximum Output Rise and Fall Time		2.0V		60	75	90	ns	ns	
			4.5V		12	15	18	ns	ns	
			6.0V		10	13	15	ns	ns	
C_{PD}	Power Dissipation Capacitance (Note 5)	(per buffer) $G = V_{OH}$ $G = V_{OL}$		12 50				pF	pF	
C_{IN}	Maximum Input Capacitance			5	10	10	10	pF	pF	
C_{OUT}	Maximum Output Capacitance			10	20	20	20	pF	pF	

Note 5: C_{PD} determines the no-load dynamic power consumption: $P_D = C_{PD} V_{CC} f$ ($V_{CC} = V_{OH}$ and the no-load dynamic current consumption: $I_D = C_{PD} V_{CC} f / V_{OH}$)

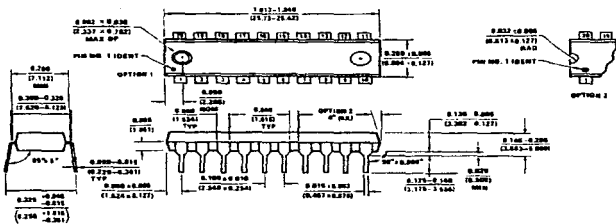
Logic Diagram

*HC244



TL/F/8387-2

Physical Dimensions inches (millimeters) (Continued)



Dual-In-Line Package
Order Number MM74HC244N
MS Package K30A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

- Life support devices or systems are devices or systems which: (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
- A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

2

National Semiconductor
Corporation
1111 West Garden Road
San Jose, CA 95128
Tel: (408) 737-2600
Fax: (408) 737-7100

National Semiconductor
Europe
Paris: (+48) 0-180-520 88 88
Germany: (+48) 0-180-520 88 88
Dachau: (+48) 0-180-520 88 88
Singapore: (+48) 0-180-520 88 88
France: Tel: (+48) 0-180-520 88 88
Madrid: Tel: (+48) 0-180-520 88 88

National Semiconductor
Hong Kong Ltd.
2301, Floor, Wing Lok
Commercial Centre, 3 Canton Road
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2727-1800
Fax: (852) 2726-3800

National Semiconductor
Japan Ltd.
Tel: (045) 296-2700
Fax: (045) 296-2400

MM54HC273/MM74HC273 Octal D Flip-Flops with Clear

General Description

These edge triggered flip-flops utilize advanced silicon-gate CMOS technology to implement D-type flip-flops. They possess high noise immunity, low power, and speeds comparable to low power Schottky TTL circuits. The device contains 8 master-slave flip-flops with a common clock and common clear. Data on the D input having the specified setup and hold times is transferred to the Q output on the low to high transition of the CLOCK input. The CLEAR input when low, sets all outputs to a low state.

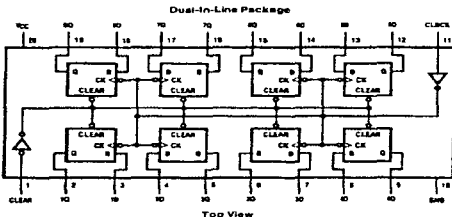
Each output can drive 10 low power Schottky TTL equivalent loads. The MM54HC273/MM74HC273 is functionally

as well as pin compatible to the 54LS273/74LS273. All inputs are protected from damage due to static discharge by diodes to V_{CC} and ground.

Features

- Typical propagation delay: 18 ns
- Wide operating voltage range
- Low input current: 1 μ A maximum
- Low quiescent current: 80 μ A (74 Series)
- Output drive: 10 LS-TTL loads

Connection Diagram



Order Number MM54HC273 or MM74HC273

Truth Table

(Each Flip-Flop)

Inputs		Outputs	
Clear	Clock	D	Q
L	X	X	L
H	T	H	H
H	T	L	L
H	L	X	Q_0

H = high level (steady state)
 L = low level (steady state)
 X = don't care
 T = transition from low to high level
 Q_0 = the level of Q before the indicated steady state input conditions were established

Absolute Maximum Ratings (Notes 1 and 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	-0.5 to +7.0V
DC Input Voltage (V_{IN})	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage (V_{OUT})	-0.5 to $V_{CC} + 0.5V$
Clamp Diode Current (I_{IK}, I_{OK})	± 20 mA
DC Output Current, per pin (I_{OUT})	± 25 mA
DC V_{CC} or GND Current, per pin (I_{CC})	± 50 mA
Storage Temperature Range (T_{STG})	-65°C to +150°C
Power Dissipation (P_D)	600 mW
(Note 3)	500 mW
SQ Package only	260°C
Lead Temp. (TL) (Soldering 10 seconds)	260°C

Operating Conditions

Supply Voltage (V_{CC})	Min	Max	Units
DC Input or Output Voltage	0	6	V
(V_{IN}, V_{OUT})	2	V_{CC}	V
Operating Temp. Range (T_A)			
MM74HC	-40	+85	°C
MM54HC	-55	+125	°C
Input Rise or Fall Times			
(t_r, t_f)	$V_{CC} = 2.0V$	1000	ns
	$V_{CC} = 4.5V$	500	ns
	$V_{CC} = 6.0V$	400	ns

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	$T_A = 25^\circ C$		74HC		54HC		Units
			V_{CC}	Typ	$T_A = -40$ to $85^\circ C$	Guaranteed Limits	$T_A = -55$ to $125^\circ C$		
V_{IH}	Minimum High Level Input Voltage		2.0V	1.5	1.5	1.5	1.5	V	
			4.5V	3.15	3.15	3.15	3.15	V	
			6.0V	4.2	4.2	4.2	4.2	V	
V_{IL}	Maximum Low Level Input Voltage**		2.0V	0.5	0.5	0.5	0.5	V	
			4.5V	1.35	1.35	1.35	1.35	V	
			6.0V	1.8	1.8	1.8	1.8	V	
V_{OH}	Maximum High Level Output Voltage	$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} \leq 20 \mu A$	2.0V	2.0	1.9	1.9	1.9	V	
			4.5V	4.5	4.4	4.4	4.4	V	
			6.0V	6.0	5.9	5.9	5.9	V	
		$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} = 4.0$ mA $I_{OUT} = 5.2$ mA	4.5V	4.2	3.98	3.84	3.7	V	
			6.0V	5.7	5.48	5.24	5.2	V	
V_{OL}	Maximum Low Level Output Voltage	$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} \leq 20 \mu A$	2.0V	0	0.1	0.1	0.1	V	
			4.5V	0	0.1	0.1	0.1	V	
			6.0V	0	0.1	0.1	0.1	V	
		$V_{IN} = V_{IH}$ or V_{IL} $I_{OUT} = 4.0$ mA $I_{OUT} = 5.2$ mA	4.5V	0.2	0.28	0.33	0.4	V	
			6.0V	0.2	0.28	0.33	0.4	V	
I_{IN}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	6.0V	± 0.1	± 1.0	± 1.0	μA		
I_{CC}	Maximum Quiescent Supply Current	$V_{IN} = V_{CC}$ or GND $I_{OUT} = 0 \mu A$	6.0V	8	80	160	μA		

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation Rating during — plastic "N" package — 12 mW/°C from 55°C to 85°C, ceramic "J" package — 12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of 5V $\pm 10\%$ the worst case output voltages (V_{OL} and V_{OH}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this family. Worst case V_{OH} and V_{OL} occur at $V_{CC} = 5.5V$ and 4.5V respectively. (The V_{OH} value at 8.5V or 2.65V). The worst case leakage current (I_{IL} , I_{CC} and I_{OZ}) occur for CMOS at the higher voltage and so the 6.0V values should be used.

** V_{IL} errors are currently tested at 20% of V_{CC} . The above V_{IL} specification (30% of V_{CC}) will be implemented no later than Q1, CY 88.

AC Electrical Characteristics $V_{CC} = 5V$, $T_A = 25^\circ C$, $C_L = 15 pF$, $t_r = t_f = 6 ns$

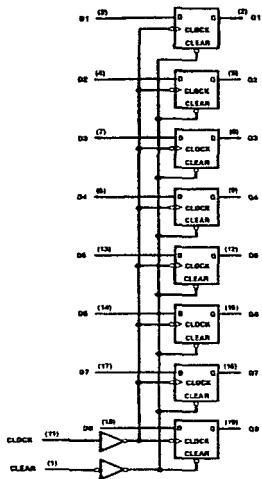
Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
f_{MAX}	Maximum Operating Frequency		50	30	MHz
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Clock to Output		18	27	ns
t_{PHL}	Maximum Propagation Delay, Clear to Output		18	27	ns
t_{REM}	Minimum Removal Time, Clear to Clock		10	20	ns
t_s	Minimum Setup Time, Data to Clock		10	20	ns
t_h	Minimum Hold Time, Clock to Data		-2	0	ns
t_w	Minimum Pulse Width, Clock or Clear		10	16	ns

AC Electrical Characteristics $C_L = 50 pF$, $t_r = t_f = 6 ns$ (unless otherwise specified)

Symbol	Parameter	Conditions	V_{CC}	Guaranteed Limits			Units	
				$T_A = 25^\circ C$	$T_A = -40$ to $85^\circ C$	$T_A = -55$ to $125^\circ C$		
f_{MAX}	Maximum Operating Frequency		2.0V	16	5	4	3	MHz
			4.5V	74	27	21	18	MHz
			6.0V	78	31	24	20	MHz
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Clock to Output		2.0V	38	135	170	205	ns
			4.5V	14	27	34	41	ns
			6.0V	12	23	29	35	ns
t_{PHL}	Maximum Propagation Delay, Clear to Output		2.0V	42	135	170	205	ns
			4.5V	19	27	34	41	ns
			6.0V	18	23	29	35	ns
t_{REM}	Minimum Removal Time, Clear to Clock		2.0V	0	25	32	37	ns
			4.5V	0	5	6	7	ns
			6.0V	0	4	5	6	ns
t_s	Minimum Setup Time, Data to Clock		2.0V	28	100	125	150	ns
			4.5V	7	20	25	30	ns
			6.0V	5	17	21	25	ns
t_h	Minimum Hold Time, Clock to Data		2.0V	-15	0	0	0	ns
			4.5V	-8	0	0	0	ns
			6.0V	-4	0	0	0	ns
t_w	Minimum Pulse Width, Clock or Clear		2.0V	34	80	100	120	ns
			4.5V	11	16	20	24	ns
			6.0V	10	14	18	20	ns
t_r, t_f	Maximum Input Rise and Fall Time, Clock		2.0V	1000	1000	1000	ns	
			4.5V	500	500	500	ns	
			6.0V	400	400	400	ns	
t_{THL}, t_{TLH}	Maximum Output Rise and Fall Time		2.0V	28	75	95	110	ns
			4.5V	11	15	18	22	ns
			6.0V	9	13	16	19	ns
C_{PD}	Power Dissipation Capacitance (Note 5)	(per flip-flop)	45				pF	
C_{IN}	Maximum Input Capacitance		7	10	10	10	pF	

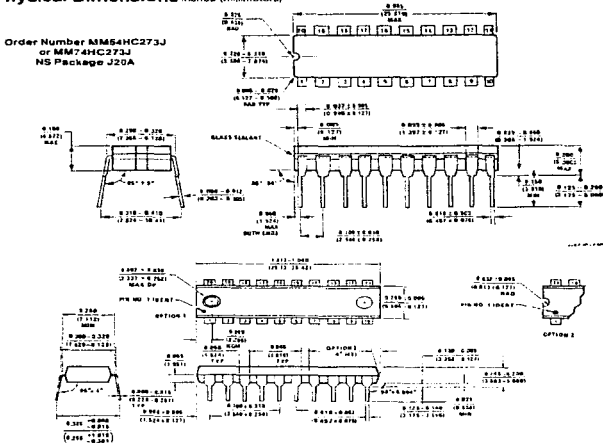
Note 5: C_{PD} determines the total dynamic power consumption: $P_{DYN} = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$, and the total dynamic current consumption: $I_D = C_{PD} V_{CC} f + I_{CC}$.

Logic Diagram



TL/F/8281-2

Physical Dimensions inches (millimeters)



LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which: (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor
Corporate Office
1111 North California Avenue
Folsom, CA 95630
Tel: (916) 752-2000
Fax: (916) 752-1718

National Semiconductor
Customer Support
Tel: (408) 685-3000 (ext. 200)
Fax: (408) 685-3000 (ext. 200)
E-mail: csupport@nsc.com
Tel: (408) 685-3270 (ext. 200)
Tel: (408) 685-3270 (ext. 200)
Tel: (408) 685-3270 (ext. 200)
Tel: (408) 685-3270 (ext. 200)

National Semiconductor
Hong Kong Ltd.
1001 Plover Square
Queen's Centre, 5 Canton Road
Tsimshatsi, Kowloon
HONG KONG
Tel: (852) 2722-1800
Fax: (852) 2726-2500

National Semiconductor
Japan Ltd.
Tel: (81) 045-296-2000
Fax: (81) 045-296-2000

27C64 65,536-Bit (8,192 x 8) UV Erasable CMOS PROM Military Qualified

General Description

The 27C64 is a high speed 64K UV erasable and electrically reprogrammable CMOS EPROM, ideally suited for applications where fast turnaround, pattern experimentation and low power consumption are important requirements.

The 27C64 is designed to operate with a single +5V power supply with $\pm 10\%$ tolerance. The CMOS design allows the part to operate over Military Temperature Ranges.

The 27C64 is packaged in a 28-pin dual-inline package with transparent lid and a 32-pin windowed LCC. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written electrically into the device by following the programming procedure.

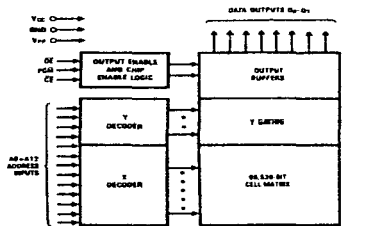
This EPROM is fabricated with National's proprietary, time proven CMOS double-poly silicon gate technology which combines high performance and high density with low power consumption and excellent reliability.

The 27C64 specified on this data sheet is fully compliant with MIL-STD-883, Revision C.

Features

- Clocked sense amps for fast access time down to 200 ns
- Low CMOS power consumption
 - Active Power: 55 mW max
 - Standby Power: 0.55 mW max
- Performance compatible to NSC800™ CMOS micro-processor
- Single 5V power supply
- Pin compatible with NMOS 64K EPROMs
- Fast and reliable programming
- Static operation—no clocks required
- TTL CMOS compatible inputs/outputs
- TRI-STATE® output
- Optimum EPROM for total CMOS systems
- Manufacturer's identification code for automatic programming control
- Windowed DIP and LCC Package Options
- Specifications guaranteed over full military temperature range (-55°C to $+125^{\circ}\text{C}$)
- This device is processed in compliance with SMD85102, and the DIP version is dual marked

Block Diagram



TL/D/10331-1

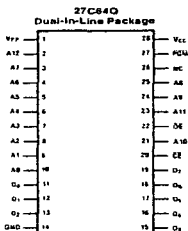
Pin Names

A0 - A12	Addresses
CE	Chip Enable
OE	Output Enable
O ₀ - O ₇	Outputs
PGM	Program
NC	No Connect

TRI-STATE® is a registered trademark of National Semiconductor Corporation.
NSC8000™ is a trademark of National Semiconductor Corporation.

Connection Diagram

27C512	27C256	27C128	27C32	27C16
A15	Vpp	Vpp	2732	2716
A12	A12	A12		
A7	A7	A7	A7	A7
A6	A6	A6	A6	A6
A5	A5	A5	A5	A5
A4	A4	A4	A4	A4
A3	A3	A3	A3	A3
A2	A2	A2	A2	A2
A1	A1	A1	A1	A1
A0	A0	A0	A0	A0
O0	O0	O0	O0	O0
O1	O1	O1	O1	O1
O2	O2	O2	O2	O2
GND	GND	GND	GND	GND

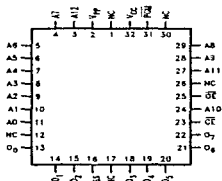


27C16	27C32	27C128	27C256	27C512
2716	2732	27128	27256	27512
		Vcc	Vcc	Vcc
		PGM	A14	A14
Vcc	Vcc	A13	A13	A13
A8	A8	A8	A8	A8
A9	A9	A9	A9	A9
Vpp	A11	A11	A11	A11
CE	CE/Vpp	CE	CE	CE/Vpp
A10	A10	A10	A10	A10
CE/PGM	CE	CE	CE/PGM	CE
O7	O7	O7	O7	O7
O6	O6	O6	O6	O6
O5	O5	O5	O5	O5
O4	O4	O4	O4	O4
O3	O3	O3	O3	O3

TU/D/10331-2

Note: Socket compatible EPROM pin configurations are shown in the boxes adjacent to the 27C64 pin.

NS Package Number J28AQ



Top View

NS Package Number EA32CO

Military Temp Range (-55°C to +125°C)
Vcc = 5V ± 10%

Parameter/Order Number	Access Time (ns)
27C64Q350/883	350
27C64Q250/883	250
27C64Q200/883	200
27C64E350/883	350
27C64E250/883	250
27C64E200/883	200

Absolute Maximum Ratings (Note 1)

Temperature under Bias	-55°C to +125°C
Storage Temperature	-65°C to +150°C
All Input Voltages except A9 with Respect to Ground (Note 10)	+6.5V to -0.5V
All Output Voltages with Respect to Ground (Note 10)	$V_{CC} + 1.0V$ to $GND - 0.5V$
V_{DD} Supply Voltage and A9 with Respect to Ground during Programming	+14.0V to -0.6V

V_{CC} Supply Voltage with Respect to Ground	+7.0V to -0.6V
Power Dissipation	1.0W
Lead Temperature (Soldering, 10 sec.)	300°C
ESD Rating (Mil Spec B53C, Method 3015.2)	2000V

Operating Conditions (Note 7)

Temperature Range (T_{case})	-55°C to +125°C
V_{CC} Power Supply	+5V \pm 1%

READ OPERATION**DC Electrical Characteristics**

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{IH}	Input Load Current	$V_{IH} = V_{CC}$ or GND			10	μA
I_{LO}	Output Leakage Current	$V_{OLT} = V_{CC}$ or GND, CE = V_{IH}			10	μA
I_{CC1} (Note 8)	V_{CC} Current (Active) TTL Inputs	CE = V_{IH} , f = 5 MHz Inputs = V_{IH} or $V_{LH}/O = 0$ mA		5	20	mA
I_{CC2} (Note 8)	V_{CC} Current (Active) CMOS Inputs	CE = GND, f = 5 MHz Inputs = V_{CC} or GND, I/O = 0 mA		3	10	mA
I_{CCS1}	V_{CC} Current (Standby) TTL Inputs	CE = V_{IH}		0.1	1	mA
I_{CCS2}	V_{CC} Current (Standby) CMOS Inputs	CE = V_{CC}		0.5	100	μA
I_{DP}	V_{PP} Load Current	$V_{PP} = V_{CC}$			100	μA
V_{IL}	Input Low Voltage		-0.1		0.8	V
V_{IH}	Input High Voltage		2.0		$V_{CC} + 1$	V
V_{OL1}	Output Low Voltage	$I_{OL} = 2.1$ mA			0.45	V
V_{OH1}	Output High Voltage	$I_{OH} = -400$ μA		2.4		V
V_{OL2}	Output Low Voltage	$I_{OL} = 0$ μA			0.1	V
V_{OH2}	Output High Voltage	$I_{OH} = 0$ μA		4.4		V

AC Electrical Characteristics

Symbol	Parameter	Conditions	27C64						Units
			200		250		350		
			Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay	CE = DE = V_{IL} PGM = V_{IH}		200		250		350	ns
t_{CE}	CE to Output Delay	CE = V_{IH} , PGM = V_{IH}		200		250		350	ns
t_{OE}	OE to Output Delay	CE = V_{IH} , PGM = V_{IH}		60		70		120	ns
t_{OF}	OE High to Output Float	CE = V_{IH} , PGM = V_{IH}	0	55	0	55	0	105	ns
t_{OH}	Output Hold from Addresses, CE or OE, Whichever Occurred First	CE = DE = V_{IL} PGM = V_{IH}	0		0		0		ns

Capacitance $T_A = +25^\circ\text{C}$, $f = 1\text{ MHz}$ (Note 2)

Symbol	Parameter	Conditions	Typ.	Max.	Units
C_{IN}	Input Capacitance	$V_{IN} = 0\text{V}$	6	10	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0\text{V}$	8	12	pF

AC Test Conditions

Output Load

1 TTL Gate and
 $C_L = 100\text{ pF}$ (Note 8)

Timing Measurement Reference Level

Inputs
 Outputs

0.8V and 2V
 0.5V and 2V

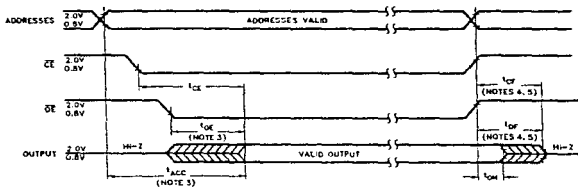
Input Rise and Fall Times

$\leq 5\text{ ns}$

Input Pulse Levels

0.45V to 2.4V

AC Waveforms (Notes 6 & 9)



Note 1: Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: This parameter is only specified and is not 100% tested.

Note 3: OE may be delayed up to $t_{OCC} = t_{OE}$ after the falling edge of CE without impacting t_{OCC} .

Note 4: The t_{CV} and t_{CO} combine level is determined as follows:
 High to TRI-STATE, the measured $V_{OH}(DC) = 0.10\text{V}$
 Low to TRI-STATE, the measured $V_{OL}(DC) = 0.10\text{V}$

Note 5: TRI-STATE may be entered using CE or OE.

Note 6: The power switching characteristics of EPLDs require careful device decoupling. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND.

Note 7: The inputs must be referenced to $V_{CC} = 1.0\text{V}$ to avoid latch-up and device damage.

Note 8: 1 TTL Gate $I_{OL} = 1.0\text{ mA}$, $I_{OH} = -400\text{ }\mu\text{A}$.

Note 9: 100 pF includes fixture capacitance.

Note 10: V_{CC} may be operated to V_{CC} during programming.

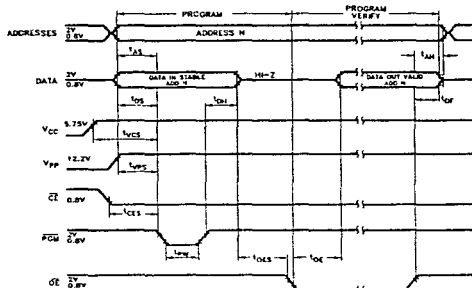
Note 11: Inputs and outputs can under shoot to -2.0V for 20 ns max.

TL/D10331-3

Programming Characteristics (Notes 1, 2, 3 & 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t _{AS}	Address Setup Time		2			μs
t _{OES}	OE Setup Time		2			μs
t _{OCS}	CE Setup Time		2			μs
t _{OS}	Data Setup Time		2			μs
t _{OPS}	V _{PP} Setup Time		2			μs
t _{OCS}	V _{CC} Setup Time		2			μs
t _{AH}	Address Hold Time		0			μs
t _{DH}	Data Hold Time		2			μs
t _{OE}	Output Enable to Output Float Delay	CE = V _{IL}	0		130	ns
t _{OW}	Program Pulse Width		0.45	0.5	0.55	ms
t _{OE}	Data Valid from OE	CE = V _{IL}			150	ns
I _{PP}	V _{PP} Supply Current during Programming Pulse	CE = V _{IL} PCIN = V _{IL}			30	mA
I _{CC}	V _{CC} Supply Current				10	mA
T _A	Temperature Ambient		20	25	30	°C
V _{CC}	Power Supply Voltage		5.75	6.0	6.25	V
V _{PP}	Programming Supply Voltage		12.2	13.0	13.3	V
t _{RI}	Input Rise, Fall Time		5			ns
V _{IL}	Input Low Voltage			0.0	0.45	V
V _{IH}	Input High Voltage		2.4	4.0		V
V _{IN}	Input Timing Reference Voltage		0.8	1.5	2.0	V
V _{OUT}	Output Timing Reference Voltage		0.8	1.5	2.0	V

Programming Waveforms (Note 3)



TL/D/10321-4

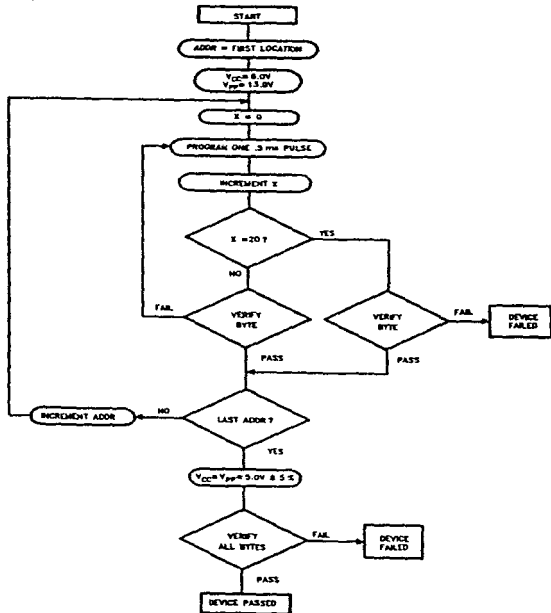
Note 1: Intel's standard product warranty applies to devices programmed to specifications described herein.

Note 2: V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The EP1M10K10 must not be inserted into or removed from a board with voltage applied to V_{PP} or V_{CC}.

Note 3: The maximum absolute address voltage which may be applied to the V_{PP} pin during programming is 14V. Care must be taken when entering the V_{PP} supply to prevent any overshoot from exceeding this 14V maximum specification. At least a 0.1 μ F capacitor is required across V_{PP} to GND to suppress spurious voltage transients which may damage the device.

Note 4: Programming and program verify are tested with the interactive Program Algorithm, at typical power supply voltages and settings. The min and max test parameters are design parameters, not tested or guaranteed.

Interactive Programming Algorithm Flow Chart



TL/D/40031-6

Functional Description

DEVICE OPERATION

The six modes of operation of the 27C64 are listed in Table I. It should be noted that all inputs for the six modes are at TTL levels. The power supplies required are V_{CC} and V_{PP} . The V_{PP} power supply must be at 13.0V during the three programming modes, and must be at 5V in the other three modes. The V_{CC} power supply must be at 5V during the three programming modes, and at 5V in the other three modes.

Read Mode

The 27C64 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable (CE) is the power control and should be used for device selection. Output Enable (OE) is the output control and should be used to gate data to the output pins, independent of device selection. The programming pin (PGM) should be at V_{PP} except during programming. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from CE to output (CG). Data is available at the outputs t_{CG} after the falling edge of CE, assuming that CE has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

The sense amps are clocked for fast access time. V_{CC} should therefore be maintained at operating voltage during read and verify. If V_{CC} temporarily drops below the spec. voltage (but not to ground) an address transition must be performed after the drop to insure proper output data.

Standby Mode

The 27C64 has a standby mode which reduces the active power dissipation by 99%, from 55 mW to 0.55 mW. The 27C64 is placed in the standby mode by applying a CMOS high signal to the CE input. When in standby mode, the outputs are in a high impedance state, independent of the CE input.

Output OR-Tying

Because 27C64s are usually used in larger memory arrays, National has provided a 2-line control function that accommodates the use of multiple memory connections. The 2-line control function allows for:

- the lowest possible memory power dissipation, and
- complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that CE (pin 20) be decoded and used as the primary device selecting function, while OE (pin 22) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This insures that all deselected memory devices are in their low power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

CAUTION: Exceeding 14V on pin 1 (V_{PP}) will damage the 27C64.

Initially, all bits of the 27C64 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "1s" will be programmed, both "1s" and "0s" can be presented in the data word. A "0" cannot be changed to a "1" once the bit has been programmed.

The 27C64 is in the programming mode when the V_{PP} power supply is at 13.0V and CE is at V_{PP} . It is required that at least a 0.1 μ F capacitor be placed across V_{PP} . V_{CC} to ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

For programming, CE should be kept TTL low at all times while V_{PP} is kept at 13.0V.

When the address and data are stable, an active low, TTL program pulse is applied to the PGM input. A program pulse must be applied at each address location to be programmed. The 27C64 is designed to be programmed with interactive programming, where each address is programmed with a series of 5 ms pulses until it verifies (up to a maximum of 20 pulses or 10 ms). The 27C64 must not be programmed with a DC signal applied to the PGM input. Programming multiple 27C64s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 27C64s may be connected together when they are programmed with the same data. A low level TTL pulse applied to the PGM input programs the paralleled 27C64s.

TABLE I. Mode Selection

Mode	Pins	CE (20)	OE (22)	PGM (27)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read		V_{IL}	V_{IL}	V_{IH}	5V	5V	Output
Standby		V_{IH}	Don't Care	Don't Care	5V	5V	$H-Z$
Output Disable		Don't Care	V_{IH}	V_{IH}	5V	5V	$H-Z$
Program		V_{IL}	V_{IH}	V_{IH}	13V	5V	D_H
Program Verify		V_{IL}	V_{IL}	V_{IH}	13V	6V	Output
Program Inhibit		V_{IH}	Don't Care	Don't Care	13V	6V	$H-Z$

Functional Description (Continued)

Program Inhibit

Programming multiple 27C64s in parallel with different data is also easily accomplished. Except for CE all like inputs (including OE and PGM) of the parallel 27C64 may be common. A TTL low level program pulse applied to an 27C64's PGM input with CE at V_{IH} and V_{PP} at 13.0V will program that 27C64. A TTL high level CE input inhibits the other 27C64s from being programmed.

Program Verify

A verify should be performed on the programmed bits to determine whether they were correctly programmed. The verify may be performed with V_{PP} at 13.0V. V_{PP} must be at VCC, except during programming and program verify.

MANUFACTURER'S IDENTIFICATION CODE

The 27C64 has a manufacturer's identification code to aid in programming. The code, shown in Table II, is two bytes wide and is stored in a ROM configuration on the chip. It identifies the manufacturer and the device type. The code for the 27C64 is "8F02", where "8F" designates that it is made by National Semiconductor, and "02" designates a 64k part. The code is accessed by applying $12V \pm 0.5V$ to address pin A9. Addresses A1-A8, A10-A12, CE, and OE are held at V_{IL} . Address A0 is held at V_{IL} for the manufacturer's code, and at V_{IH} for the device code. The code is read out on the 8 data pins. Proper code access is only guaranteed at $25^\circ C \pm 5^\circ C$.

The primary purpose of the manufacturer's identification code is automatic programming control. When the device is inserted in an EPROM programmer socket, the programmer reads the code and then automatically calls up the specific programming algorithm for the part. This automatic programming control is only possible with programmers which have the capability of reading the code.

ERASURE CHARACTERISTICS

The erasure characteristics of the 27C64 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å-4000Å range. After programming, opaque lacquer should be placed over the 27C64's window to prevent unintentional erasure. Covering the window will also prevent temporary functional failure due to the generation of photo currents.

The recommended erasure procedure for the 27C64 is exposure to short wave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15W-sec/cm².

The 27C64 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure. Table III shows the minimum 27C64 erasure time for various light intensities.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at one inch. The erasure time increases as the square of the distance. (If distance is doubled the erasure time increases by a factor of 4.) Lamps lose intensity as they age. When a lamp is changed, the distance has changed or the lamp has aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and even system designs have been erroneously suspected when incomplete erasure was the problem.

SYSTEM CONSIDERATION

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC} , has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by voltage transients on input pins. The magnitude of these transient current peaks is dependent on the output capacitance loading of the device. The associated V_{CC} transient voltage peaks can be suppressed by properly selected decoupling capacitors. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance. In addition, at least a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for each eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage drop caused by the inductive effects of the PC board traces.

TABLE II. Manufacturer's Identification Code

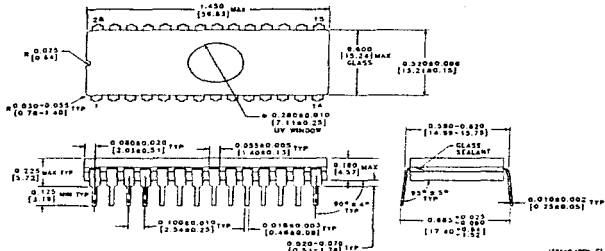
Bits	A0 (10)	O7 (10)	O6 (18)	O5 (17)	O4 (16)	O3 (15)	O2 (13)	O1 (12)	O0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	1	1	1	8F
Device Code	V_{IH}	1	1	0	0	0	0	1	0	02

TABLE III. Minimum 27C64 Erasure Time

Light Intensity (Micro-Watts/cm ²)	Erasure Time (Minutes)
15,000	20
10,000	25
5,000	50

Physical Dimensions inches (millimeters) (Continued)

LN # 114710



28 Lead EPROM Dual-In-Line Package (JQ) Small Window
 Order Number 27C64Q350/883, 27C64Q250/883 or 27C64Q200/883
 NS Package Number J28AQ

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

- Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
- A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor
 Corporation
 1111 West Sixth Street
 Austin, TX 78717
 Tel: 1800-272-9888
 Fax: 1800-371-7018

National Semiconductor
 Europe
 Fax: 1-48 0-180-530 88 88
 Email: europe@ns.com
 Dusseldorf Tel: 1-48 0-180-530 88 88
 England Tel: 1-48 0-180-530 78 78
 France Tel: 1-48 0-180-530 88 88
 Germany Tel: 1-48 0-180-530 18 80

National Semiconductor
 Hong Kong Ltd.
 12th Floor, Summer House
 Chinese Centre, 6 Canton Rd.
 Tsimshatsui, Kowloon
 Hong Kong
 Tel: 862-2 770-1800
 Fax: 862-2 770-2000

National Semiconductor
 Japan Ltd.
 Tel: 81-043-296-3308
 Fax: 81-043-296-3436

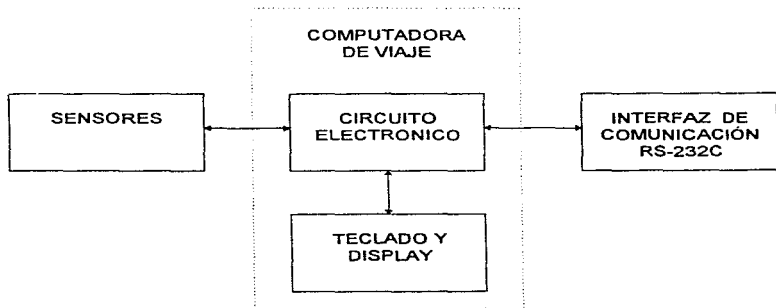
APENDICE C

MANUAL DE USUARIO

MANUAL DE OPERACION DEL SISTEMA**I. HARDWARE**

A continuación se presenta el manual de operación de la Computadora de Viaje, no se pretende llegar a detalle en el funcionamiento del sistema, por lo que únicamente se proporciona la información necesaria para la puesta en operación del dispositivo electrónico, la transferencia de información hacia la PC y el manejo de software del sistema.

Las partes que conforman en hardware del sistema se presentan en el siguiente diagrama de bloques.



Los elementos que integran el sistema son :

- Sensores. *
- Circuito Electrónico. *
- Interfaz de comunicación RS-232C.
- Batería de 9 Volts D.C.
- Cable de comunicación serial de tres hilos.
- Módulo de display y teclado (desmontables).
- Software del sistema.

* Nota: Estos elementos se encuentran acoplados al vehículo.

Los sensores y los circuitos electrónicos que conforman la computadora de viaje se encuentran acoplados directamente al vehículo, por lo que el usuario no tiene acceso a ellos. El display y el teclado son desmontables, lo que permite al operador del sistema conectar este módulo únicamente cuando se requiera o bien dejarlo integrado al vehículo. La interfaz RS-232C, es un puerto de comunicaciones muy similar al que se encuentra en la mayoría de las computadoras personales (PC's) y sólo se requiere un cable de tres hilos (proporcionado con el sistema) para la transferencia de la información.

A continuación se detalla el procedimiento que permite poner en funcionamiento la computadora de viaje.

- a) Conecte las terminales de la batería del sistema, y si es el caso conecte el módulo del display y el teclado.
- b) Una vez alimentado el sistema, usted observará un mensaje de bienvenida en el display.

FACULTAD DE INGENIERIA. UNAM
COMPUTADORA DE VIAJE

- c) A continuación la computadora de viaje le solicitará el password de arranque del sistema. Este password esta integrado por cuatro digitos del 0 al 9.

PASSWORD : _

Si el password es incorrecto, se desplegará un mensaje y le solicitará nuevamente el password.

PASSWORD
INCORRECTO

- d) Una vez que se ha proporcionado el password correcto, se desplegará un menú solicitando la tarea a realizar.

- 1) RECORRIDO
2) TRANSFERENCIA

Aquí deberá de teclear "1" (RECORRIDO), si desea iniciar la operación de registro de datos durante un viaje, o bien "2" (TRANFERENCIA) indicando al dispositivo que establezca la comunicación con la computadora personal (PC) para la transferencia de la información.

Nota: Si se tecldea la opción "1", puede continuar con los siguientes pasos, en el caso contrario pase al inciso (k).

e) En la opción "1" (RECORRIDO), el sistema le solicitará la fecha. La cual deberá proporcionarse en el formato numérico día, mes y año (DDMMAA).

FECHA : _

Una vez proporcionada la fecha, aparecerá un mensaje preguntando si el dato que acaba de ingresar esta correcto, si este es el caso deberá teclear "1" en caso contrario deberá teclear "2" para reingresar el dato.

DATO OK ?
1) Si 2) No

De aquí en adelante, esta pantalla de comprobación, aparecerá cada vez que se solicite un dato nuevo.

- f) A continuación se le solicitará la hora de inicio del recorrido. Esta se debe proporcionar en el formato numérico de horas y minutos (HHMM).

HORA : _

- g) Ahora el sistema le solicitará la clave del recorrido, que será un dato numérico de cuatro dígitos.

RECORRIDO : _

- h) El último dato solicitado es la clave del(os) operador(es), que es un dato numérico de cuatro dígitos. En el caso de que solo exista un solo operador, entonces deberá teclear ceros (0000) en el segundo campo.

OPERADOR 1 : _

OPERADOR 2 : _

- i) Una vez proporcionados todos los datos anteriores, aparecerá un mensaje en el display indicando que se esta realizando el registro del recorrido. Aquí se tiene la opción de retirar el módulo del display y teclado para evitar detener el sistema como consecuencia de una pulsación de tecla accidental.

EJECUTANDO
RECORRIDO ...

En este momento el sistema esta listo y se encontrará registrando todos los eventos del recorrido del vehículo.

- j) Una vez que el vehículo llegue a su destino, se deberá presionar la tecla "SR" del módulo del teclado. En este momento aparecerá un mensaje preguntando si esta seguro de detener el registro de datos.

SUSPENDER RECORRIDO ?
1) Si 2) No

Si no desea suspender la tarea de recorrido (por tratarse de una pulsación accidental) deberá teclear "2", en caso contrario teclee "1". Si se realizó la suspensión el recorrido, el display volverá a mostrar el menú principal.

Nota: En este paso, el sistema esperará 1 minuto para elegir una opción, si al término de este período de tiempo no hay respuesta por parte del usuario, se continuará con el registro de datos del recorrido.

- k) Como ya se mencionó, la opción "2" del menú principal establece la comunicación del dispositivo con la PC. Antes de elegir esta opción, deberá asegurarse de que el cable de conexión esta adecuadamente conectado al dispositivo y a la computadora personal (PC). Una vez verificado este procedimiento puede continuar con la transferencia presionado la opción "2", en seguida aparecerá un mensaje en el display indicando que se esta llevando acabo la transferencia de la información.

TRANSMITIENDO
DATOS ...

Una vez concluida la transferencia, el sistema regresará al menú principal.

- l) Si se desea suspender el funcionamiento de la computadora, únicamente es necesario retirar la alimentación del sistema una vez que aparece el menú principal.

II. SOFTWARE

Requerimientos de Hardware y Software

- **Requerimientos de Hardware**

Componente	Mínimo necesario para el "Sistema de Computadora de Viaje"
Computadora	Procesador 386DX (o superior)
Memoria	4 MB de RAM: se recomienda 8MB
Monitor	VGA (o superior)
Disco Duro	10 MB
Impresora	
Cable serial	

- **Requerimientos de Software**

Sistema Operativo 5.0 o posterior (se recomienda 6.2)

Windows 95

Manejador de Bases de datos Access v. 2.0

INSTALACION

Debe introducir el disco de instalación numerado como 1 y desde el escritorio de Windows 95, en el menú de Start (Inicio), elegir la opción Run (Ejecutar) y escribir A: Instalar.

FUNCIONAMIENTO DEL SISTEMA

Para ejecutar el Sistema de la Computadora de Viaje, basta con dar click en el icono asociado a la aplicación.

La primer pantalla que aparece en su monitor es la de presentación del sistema, la cual después de unos segundos desaparecerá, dando paso a la ventana de seguridad del sistema en la cual deberá de introducir su identificación de usuario y la contraseña. Ver figura 1 y 2.

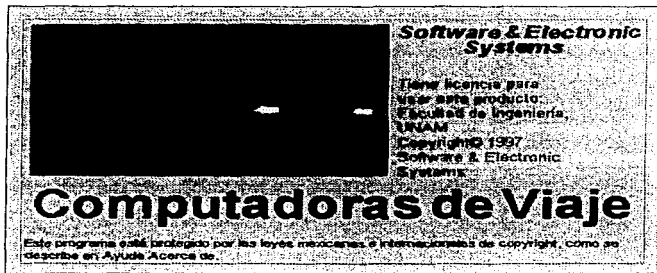


Figura 1 Presentación del Sistema.

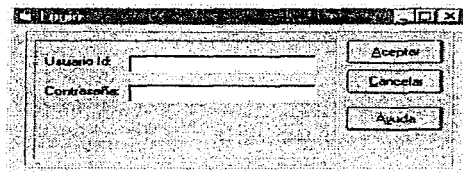


Figura 2. Pantalla de Login.

De acuerdo a la categoría del usuario (Usuario Id), en la pantalla del menú principal, se habilitarán algunas o todas las opciones de ésta, por lo que se tienen las siguientes clasificaciones de usuarios:

- **MGR (Administrador del Sistema)**

Tiene acceso a todos los privilegios del sistema. Puede ejecutar Bajas, Cambios, Modificaciones de Passwords, Reportes y Desactivación total del sistema.

- **UsuarioA**

Puede realizar casi las mismas tareas que el MGR, excepto que no puede desactivar el sistema ni tampoco consultar la Bitácora del Administrador.

- **UsuarioB**

Sólo puede realizar consultas, generar reportes y estadísticas.

De acuerdo al privilegio de acceso que se tenga, se presentará la ventana principal del sistema habilitada en algunas o todas sus partes.

Una vez tecleados los datos correctos en la pantalla de Password, se despliega el menú principal del sistema. Ver figura 3.

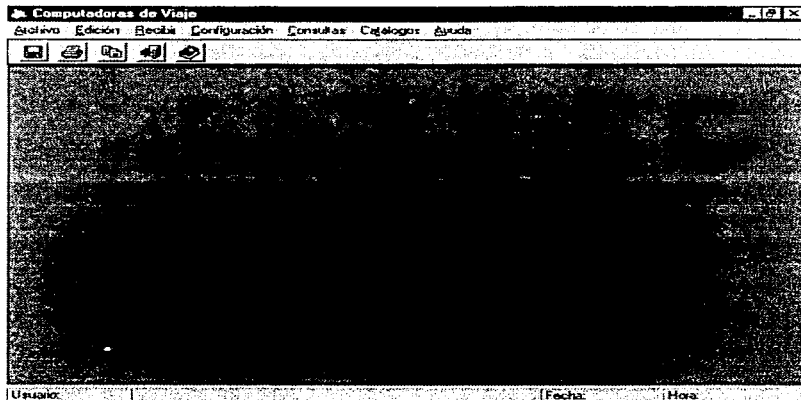


Figura 3. Menú principal del sistema.

En esta pantalla se engloban todos los submódulos en los que está construido el sistema.

Empezamos por "Archivo", al seleccionarla se despliegan las opciones de: Abrir, Cerrar, y Guardar. Si seleccionamos Abrir se despliega una ventana en donde podemos seleccionar de los drives disponibles en el sistema, el archivo que necesitamos para cargarlo al sistema. Con la opción de Cerrar el archivo que se tenga

trabajando en ese momento se puede cerrar para proceder con otros. En la opción de Guardar es para salvar el archivo que estemos trabajando.

En "Edición" tenemos Buscar, Borrar y Reemplazar. Todas ellas efectúan las funciones sobre alguna frase o palabra que tengamos en nuestra aplicación.

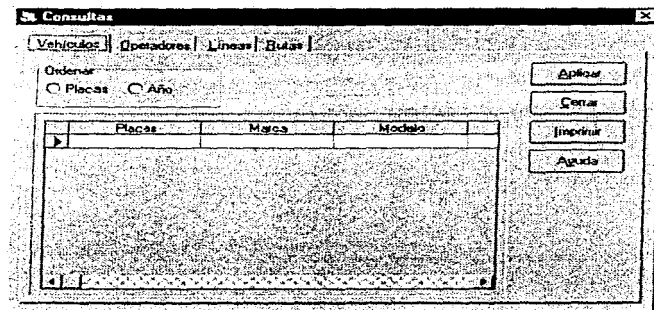
En la opción de "Recibir", es donde se establece la conexión del microcontrolador con la PC, para el inicio de la transferencia de información.

En la opción de "Configuración", se utiliza cuando no se tiene como puerto de comunicación el Com2, pues se encuentra ocupado por algún otro dispositivo por lo que se define en esta pantalla que puerto se va a usar y que parámetros va a manejar.

En la opción de "Consultas" se tienen varios tipos y son :

- Consulta de Vehículos
- Consulta de Operadores
- Consulta de Líneas
- Consulta de Rutas

Pantalla de Consultas



Como se puede observar, el diseño de pantalla es distinto a los acostumbrados en otras aplicaciones en donde los diferentes tipos de consultas se efectúan en pantallas distintas. Ultimamente en aplicaciones comerciales se agrupan las consultas de esta forma, pues permiten al usuario un rápido acceso a la información que desea y no se pierde en un cantidad exagerada de ventanas.

Pantalla de Consulta de Vehículos

En esta pantalla se muestra el tipo de consulta de vehículos, donde podemos pedir la información ya sea por el número de placas o el año del vehículo.

Otro tipo de consulta es :

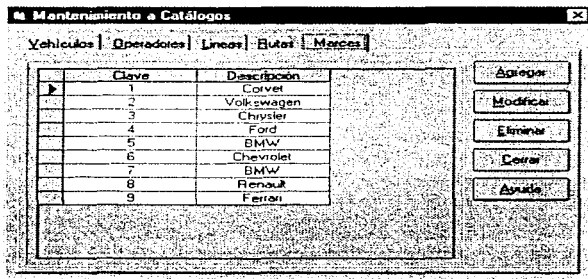
Pantalla de Consulta de Operadores

The screenshot shows a window titled "Consultas" with a menu bar containing "Vehículos", "Operadores", "Lineas", and "Buses". The "Operadores" tab is selected. Below the menu, there are two radio buttons: "Nombre" (selected) and "Operador". A table with the following columns is displayed: "Clave Operador", "Nombre", "Apellido Paterno", and "Apellido Materno". The table is currently empty. On the right side of the window, there are four buttons: "Aplicar", "Cerrar", "Imprimir", and "Ayuda".

En esta pantalla se efectúan consultas de los operadores de la línea de autobuses.

Los otros tipos de consultas : Líneas y Rutas siguen la misma filosofía de diseño que las dos pantallas anteriores.

Menú de Catálogos

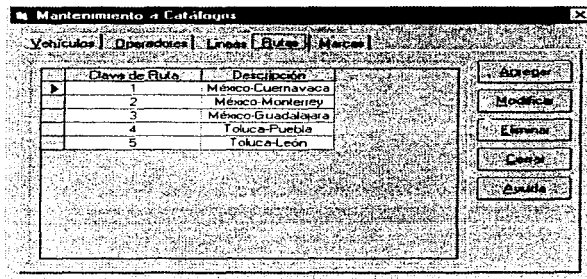


En esta opción del menú principal, se tiene el "Mantenimiento a Catálogos", donde se agrupan los diferentes tipos de mantenimientos del sistema. En la pantalla que se presenta, se tiene el Mantenimiento al catálogo de Vehículos, en donde se tienen las opciones de : Agregar, Modificar, Eliminar, Cerrar y Ayuda.

En la opción de "Agregar", se añaden nuevos registros a la tabla de vehículos. Aquí se cargan los nombres de las diferentes clases de automóviles que existen. En "Modificar", se cambian los datos del vehículo que se desee. Con la opción de

"Eliminar", se borran los registros de los tipos de vehículos que ya no necesiten en el sistema. Con "Cerrar", se cierra la ventana. Y la opción de "Ayuda", le proporciona información en línea sobre alguna duda que tenga el usuario de algún punto en particular.

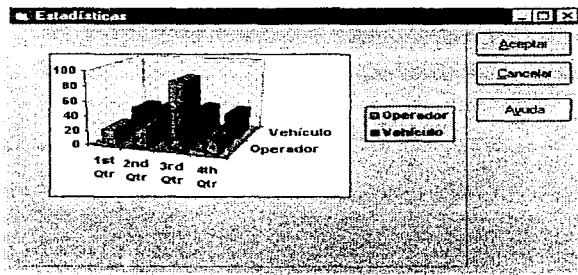
Pantalla de Mantenimiento del Catálogo de Rutas



En esta pantalla de mantenimiento se observa la misma filosofía de diseño que se utilizó en las consultas que realiza el sistema. En general todos los distintos mantenimientos que proporciona el sistema son parecidos. A través de estos diseños damos al usuario uniformidad y que el mantenimiento de sus catálogos no le resulte tedioso al estar pasando de pantalla en pantalla, pues con el uso del Tab, que simula

un pequeño archivero en el que sólo hay que seleccionar la pestaña deseada y en ese momento se entra a la opción deseada.

Pantalla de Estadísticas



Dentro de esta opción se visualizan las diferentes gráficas que se pueden crear con los datos almacenados en la base de datos del sistema, permitiendo tener un panorama general de los mismos. Por ejemplo, es posible obtener una gráfica de las estadísticas de los operadores y los vehículos, de la frecuencia de recorridos, del combustible consumido, etc.

AYUDA

Dentro del menú de ayuda tenemos: Acerca de y Ayuda del programa. El diseño y uso de esta interface explicativa está basada en la misma que proporciona el ambiente gráfico Windows, es decir, se aprovechan las bondades del uso de hipertexto, búsqueda por palabras y búsqueda por tópicos, entre otras.

Otra opción de ayuda es Acerca de, que tiene una nota informativa sobre el sistema

