

33
2 es.



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO Y CONSTRUCCION DE UN SISTEMA
DE PEAJE PARA VEHICULOS DE
TRANSPORTE PUBLICO

T E S I S

QUE PARA OBTENER EL TITULO DE :
INGENIERO MECANICO ELECTRICISTA
EN EL AREA ELECTRICA-ELECTRONICA

P R E S E N T A N :
GUILLERMO CARBALLO MIRANDA
RAYMUNDO OGARRIO OLVERA



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

MEXICO, D.F.

1998

TESIS CON
FALLA DE ORIGEN

110808



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México, a la Facultad de Ingeniería y a todos los profesores por habernos dado una educación profesional.

Al Instituto de Ingeniería y sobre todo al M. en I. Lauro Santiago Cruz por el tiempo dedicado durante la realización de este trabajo.

Y a todos quienes colaboraron con el desarrollo de este trabajo.

El presente trabajo está dedicado:

A mi madre Virginia Miranda,
por su amor y dedicación durante
toda mi vida, gracias.

A mi padre José M. Carballo,
quien me ha apoyado durante toda
mi vida, como hijo y como estudiante.

A mis hermanas Verónica y Ana Luisa,
quienes me han brindado su cariño y apoyo.

A Araceli L. por cada una de sus palabras
de aliento para llegar a este logro,
gracias por todo.

A todos y cada uno de los amigos que he
conocido a lo largo de mi vida y que me han
apoyado.

A Raymundo, un agradecimiento especial,
por permitirme trabajar con él.

A todos muchas gracias.

Guillermo Carballo M.

Gracias a Dios, por permitirme cumplir una meta más en mi vida.

A mis padres, por que ellos siempre han sido el cimiento de mis metas, con todo mi amor para ustedes.

A mis hermanos, por todo su cariño y apoyo.

A toda mi familia, por que siempre puedo contar con ella.

A Oriana por su gran amor y por estar siempre a mi lado.

A todos mis amigos, por estar siempre conmigo.

A Guillermo un agradecimiento especial por ser mi amigo y por que juntos hemos logrado terminar este trabajo.

Gracias.

Raymundo.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I. ANTECEDENTES	4
I.1 Evolución de los sistemas de peaje	4
I.2 Los sistemas de peaje actuales	7
I.3 Normativas para los sistemas de peaje en México	9
I.4 Planteamiento del problema y propuesta de solución	13
CAPÍTULO II. CONCEPTOS BÁSICOS	16
II.1. Microcontroladores	16
II.2. Memorias	23
II.3. Transductores	27
II.4. Visualizadores	44
II.5. Estándares de comunicación	47
CAPÍTULO III. ANÁLISIS DEL SISTEMA DE PEAJE	53
III.1. Descripción y análisis del sistema	53
III.2. Análisis de cada uno de los módulos que integran al sistema	61

CAPÍTULO IV. DISEÑO DEL HARDWARE DEL SISTEMA DE PEAJE	68
IV.1. Especificaciones de diseño del sistema	68
IV.1.1. Arquitectura COP8SAC	69
IV.1.2. Señal de entrada (rpm's)	95
IV.1.3. Teclado	98
IV.1.4. Despliegado de la información	99
IV.2. Diseño de accesorios	100
CAPÍTULO V. DESARROLLO DEL SOFTWARE DEL SISTEMA DE PEAJE	115
V.1. Software del sistema	115
V.1.1 Diagrama de flujo general	118
V.1.2 Diagramas de flujo de subprogramas	121
V.2. Desarrollo del software para la computadora	133
V.2.1 Selección del software para el desarrollo	133
V.2.2 Desarrollo del software	137
CAPÍTULO VI. RESULTADOS Y CONCLUSIONES	143
BIBLIOGRAFÍA	150
APÉNDICES	
Apéndice A: Fotos del prototipo	A-1
Apéndice B: Hojas de especificaciones	B-1
Apéndice C: Listado parcial de programas	C-1

INTRODUCCIÓN

Además de las funciones de investigación que tiene el Instituto de Ingeniería tiene una muy importante que es la vinculación con el sector empresarial para generar nuevas propuestas en el desarrollo de prototipos que cumplan normas preestablecidas a nivel nacional e incluso internacional, o en su defecto generarlas, para la comercialización de nuevos productos.

Ante esta situación la empresa PADI solicitó el desarrollo de un sistema de peaje para vehículos de transporte público. El Instituto de Ingeniería, a través de la Coordinación de Instrumentación generó una serie de propuestas que podrían satisfacer las necesidades de dicha empresa privada. La empresa PADI al ver las propuestas decidió contratar los servicios del Instituto para desarrollar dicho prototipo.

Al obtener una respuesta satisfactoria de parte de la empresa se procedió a trabajar en la propuesta de desarrollo. Durante el desarrollo del proyecto se requirió cumplir con las normas establecidas para este tipo de sistemas, dado que va a ser un producto comercial.

El crear un sistema que cumpla con ciertas reglamentaciones y nuevas adiciones hacen que nuestro sistema sea más competitivo y eficiente, por lo que nuestros objetivos estarán enfocados a esto último sin descuidar que nuestro sistema supere una cantidad fija de dinero

para los costos de producción de una unidad.

El presente trabajo está estructurado de la siguiente forma: en el capítulo I hacemos una reseña de los que han sido los sistemas de transporte a través del tiempo, los respectivos sistemas de cobro hasta nuestros días y la normativa vigente para dichos sistemas en nuestro país; al final del capítulo hacemos el planteamiento del problema y las propuestas posibles para su solución.

En el capítulo II hacemos un breve recordatorio teórico en el cual nos basamos para realizar el presente proyecto tal es el caso de microcontroladores, transductores, etc.

En el capítulo III se hace una descripción detallada del sistema, que es lo que debe contener, que normas debe cumplir, la operación normal del sistema y cada uno de los módulos tanto en hardware como en software que lo integran y su posible realización.

En el capítulo IV se hace una descripción de las especificaciones del sistema así como la selección de los dispositivos que cumplan con la norma oficial y el diseño a bloques de toda la arquitectura del módulo central. También hacemos la descripción de uno de los accesorios que va a soportar el sistema de peaje.

En el capítulo V se describe la operación a nivel de software del sistema, esto basándonos en diagramas de flujo para generar los algoritmos usados en el desarrollo de cada una de las funciones que debe realizar el sistema de peaje. También realizamos una breve reseña de la operación del sistema de actualización del sistema de peaje, que es un software para la computadora personal mediante el cual podemos cambiar los parámetros de operación del sistema haciendo una comunicación serial de la PC al sistema de peaje. De dicho sistema presentamos algunas ventanas de trabajo y su descripción para la operación correcta del mismo.

En el último capítulo presentamos los resultados obtenidos así como algunas pruebas de operación que realiza SECOFI para aprobar un sistema que va a salir al mercado. También

se presentan pruebas realizadas a los transductores de velocidad que van a ser utilizados en la instalación de cada uno de los sistemas de peaje. Por último presentamos las conclusiones obtenidas del desarrollo de este trabajo y las expectativas del sistema a futuro para posibles mejoras y/o cambios.

En la sección de apéndices mostramos los listados parciales de la programación del sistema de peaje y el del software para la computadora personal, así como algunas especificaciones de los circuitos usados y algunas fotos del prototipo final.

CAPÍTULO I

ANTECEDENTES

El presente capítulo tiene por objetivo el dar un panorama general de lo que han sido los sistemas de peaje o de cobro en el transporte público de la Ciudad de México; posteriormente se analiza la evolución que se ha ido dando en los mismos, que tipo de tecnologías se han incluido en ellos a través del tiempo. Un punto importante es saber que normas, a nivel nacional e internacional, rigen el diseño de este tipo de sistemas. Por último, analizaremos las características del equipo a desarrollar, para poder definir la solución más viable al mismo.

I.1. Evolución de los sistemas de peaje

El transporte ha sido con el paso del tiempo, un proceso tecnológico, económico y social cuya función es trasladar personas y cosas de un lugar a otro a través del espacio (físico). Decimos que es un proceso debido a que consta de una serie de actividades que se repiten constantemente (por ejemplo: cargar, desplazar, descargar, cargar, etc.); es tecnológico por que hacemos uso de los conocimientos técnicos que se van desarrollando, para mejorar y agilizar este medio; en cuanto a lo económico y social, se recalca debido a que es un medio indispensable para el desarrollo de las actividades productivas y de interrelación social.

Con la evolución que han tenido los medios de transporte a través del tiempo se han generado cambios en la forma de cobro o en los sistemas de cobro para dichos servicios.

El primer medio que conoció el hombre para poder transportar cosas o transportar a otras personas fue el mismo cuerpo del hombre, por este servicio no se daba una bonificación. Posteriormente el uso de animales para carga y transporte fue y es muy frecuente, por el uso de este medio de transporte se daba una remuneración económica o en prenda, es decir, se daba un producto a cambio de un servicio.

Con la creación de la rueda, el hombre creó las primeras formas de transporte de dos y cuatro ruedas. En los transportes conocidos como diligencias y coches de alquiler, que eran arrastrados por caballos, se hacían los pagos con monedas de oro u otra piedra o material precioso.

El uso de barcos para el transporte de carga o de personas en caminos fluviales como ríos o mares, fue durante la época de las colonias, de vital importancia debido a que se tenían que transportar las mercancías a través de los mares y de una forma rápida.

Con el advenimiento de las máquinas de vapor se idearon sistemas de transporte mucho más rápidos y seguros. La creación del ferrocarril como medio de transporte terrestre y el barco de vapor como medio acuático dieron un gran salto en cuanto a la implantación de un primer sistema de peaje o de cobro. Estos sistemas eran muy sencillos en cuanto al cobro, el uso de estos medio de transporte se hacía mediante el pago de un boletaje en los lugares asignados dentro de la estación o puerto. Haciendo el pago el usuario recibía un boleto que lo acreditaba como pasajero de un medio de transporte definido y con un destino fijo. Con estos modernos medios de transporte la actividad mercantil se elevó en grandes proporciones debido a la rapidez de los mismos.

Posteriormente se creó la máquina de combustión interna y esto dio paso a la creación de vehículos automotores, tal como los conocemos en la actualidad. Dichos automóviles se crearon, como ha sucedido en la creación de los diferentes medios de transporte, para uso

particular, es decir, si una persona deseaba transportarse y tenía las posibilidades de adquirir un vehículo, lo compraba. Con el tiempo se crearon los vehículos de alquiler, conocidos como "taxis", autobuses, trenes, etc.

Ya en estos sistemas se hacen uso de sistemas de cobro sofisticados para su tiempo como lo que son la compra de boletos mediante maquinas automáticas. Cabe señalar que en cada uno de los medios de transporte los principales parámetros para cobrar un servicio de transporte era, y sigue siendo, la distancia del recorrido, el número de personas o el peso de la carga, el tipo de carga, etc.

En los vehículos de alquiler o mejor conocido como "taxis" los sistemas de cobro eran mecánicos y lo que hacían era medir la distancia recorrida por el vehículo y/o el tiempo que tardaban en realizar un viaje de un punto a otro.

Es importante recalcar que los medios de transportes se han ido mejorando a través del tiempo, debido a la gran importancia que tienen para con los medios de producción económicos es decir, el hacer uso de este servicio implica una remuneración económica así como una reducción en costos al transportar grandes volúmenes de mercancías o personas en trayectorias largas; para fines militares, el transporte ha sido de vital importancia dada la rapidez del transporte de tropas y víveres a los puntos en donde se tenga un conflicto, el transporte aunado con las comunicaciones son vitales para el desarrollo de cualquier país, ya sea militar, comercial o socialmente.

A lo largo de la historia el transporte ha sido de vital importancia, por señalar unos ejemplos, durante la Primera y Segunda Guerra Mundial gran parte de las batallas se libraron en un nuevo medio de transporte creado por el hombre, el avión, mediante vehículos motorizados y navíos de guerra.

Como podemos ver actualmente los más grandes desarrollos se han generado en la industria militar y posteriormente se liberan esas tecnologías para uso industrial y comercial. Actualmente un medio de transporte creado en sus inicios para fines militares son los

transbordadores espaciales, vehículos a los cuales no se tiene un acceso comercial todavía.

1.2. Los sistemas de peaje actuales

Actualmente son muy variados los sistemas de cobro del servicio de transporte público. Podemos observar uno, que es uno con los que tenemos más contacto. El sistema de transporte colectivo. Metro, en este medio de transporte no se toma en cuenta la distancia y/o el tiempo, es decir sólo se toma en cuenta el viaje. Una persona como usuario paga en una taquilla el monto asignado para un viaje; para tener acceso al sistema de transporte el usuario debe introducir el boleto en una máquina automática que reconoce si es válido el boleto para dicho viaje y permite la entrada activando los torniquetes que están hechos para ese fin. En caso que no sea un boleto válido no se activan los torniquetes y el usuario no tiene acceso al transporte

En el caso del tren ligero de la Ciudad de México, la expedición de boletos se hace de forma automática, es decir. una máquina recibe el número de boletos que uno desee y el dinero correspondiente a este número de boletos. En caso de ser correcto el monto de efectivo la máquina dará los boletos que se pidieron. Para tener acceso el transporte los boletos se introducen una máquina de característica similares a las del metro.

Otro sistema de cobro, y al cual haremos referencia durante el resto del presente trabajo es el conocido como *sistema de peaje*. Este sistema está implantado en los vehículos de alquiler para el transporte de pasajeros. Dichos sistemas de cobro basan su funcionamiento en la distancia recorrida por dicho vehículo y/o en el tiempo que se tarda en realizar dicho recorrido o partes del recorrido. Estos sistemas han ido evolucionando desde su implantación, en un inicio su funcionamiento era totalmente mecánico (ver figura 1.2.1), actualmente son totalmente electrónicos, lo que aumenta la eficiencia en su funcionamiento; no obstante los anteriores son susceptibles de modificaciones en las tarifas correspondientes por personal no autorizado, práctica que se realiza en talleres clandestinos, por lo que los requerimientos en la creación de un sistema de peaje se deben tomar las medidas necesarias para poder

imposibilitar dicha práctica. Cabe señalar que estas tarifas las define la Secretaría de Hacienda y Crédito Público (SHCP) a través de la Secretaría de Comunicaciones y Transportes (SCT).

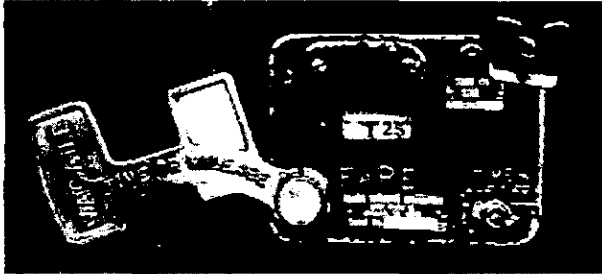


Figura 1.2.1 Sistema de peaje mecánico de la marca Japonesa Omaha.

Los sistemas de peajes cuentan con una gran variedad de aditamentos; en el Distrito Federal, así como en el interior de la república, todavía no se hace una reglamentación de estos accesorios, como lo pueden ser impresoras de recibos, lectores especiales, comunicación por radiofrecuencia, por satélite, etc.

Los sistemas de peajes que están en servicio tanto en la ciudad de México como en otras en todo el país se apegan exclusivamente a las normas establecidas por el gobierno. En contraste con sistemas de peajes de países desarrollados, éstos cuentan con avanzados sistemas de comunicaciones, lectores de tarjetas, impresoras de recibos, conexiones para sensores de presencia, múltiples tarifas, etc. En la figura 1.2.2a y 1.2.2b podemos observar sistemas de peajes con el tipo de características anteriormente mencionadas, la desventaja que tienen estos sistemas es su costo, el cual en un país como el nuestro, el tener un sistema tan sofisticado y del cual no se tiene todavía una reglamentación de los accesorios, hace incoesteable su importación.



Figura I.2.2a. Sistema de peaje Pulsar.

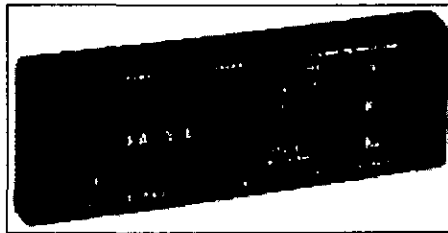


Figura I.2.2b. Sistema de peaje Cetrodyne

Los sistemas de peajes mexicanos se rigen por un conjunto de normas establecida por la Secretaría de Comercio y Fomento Industrial (SECOFI), aquí en la Ciudad de México, y un comité de evaluación de los sistemas de medición e instrumentación. Para saber de una forma precisa el funcionamiento y las características con las que debe contar el sistema de peaje veremos en el siguiente apartado las normas que rigen a éstos.

I.3. Normativas para los sistemas de peaje en México

Como lo señalamos en el apartado anterior describiremos algunos de los incisos que indican las normas oficiales mexicanas para el desarrollo de sistemas de medición y en particular las de un sistema de peaje.

La norma por la que se rigen este tipo de sistemas es la siguiente: **NOM-007-SCFI-1993** "Instrumentos de Medición, Taxímetros Electrónicos"¹.

¹Fuente: Diario Oficial, 13 de Octubre de 1993. pp 53-59. Dicha norma cancela a la NOM-CH-48-1987.

Uno de los medios de transporte más utilizados en las grandes ciudades son los vehículos de alquiler, mejor conocidos como taxis. Para realizar el cobro por los servicios prestados por este tipo de transporte, el vehículo cuenta con un aparato de medición, como se señaló en el apartado anterior, llamado sistema de peaje, el cual tiene como fin el contabilizar tiempo y/o distancias recorridas para poder generar un importe a pagar por el servicio prestado

La necesidad de crear una norma nace de garantizar el buen funcionamiento de dicho sistema de cobro, esto con el fin de poder garantizar al usuario y al prestador del servicio que el sistema funciona de una manera exacta y confiable sin importar la marca, modelo y/o año del sistema de cobro

En si la norma tiene la finalidad de "garantizar la calidad de un instrumento denominado sistema de peaje electrónico que reúna las características, especificaciones, tolerancias, métodos de prueba y garantías, que un instrumento de esta naturaleza deba cumplir"²

Dentro de esta norma se encuentran términos que no son de un uso frecuente por lo que nos dispondremos a explicar los más sobresalientes y están enfocados al diseño electrónico de un sistema de cobro, el resto de los puntos se pueden consultar en la norma previamente mencionada.

El sistema de peaje cuenta con una serie de parámetros para poder realizar la función encomendada:

Importe inicial: Es el valor monetario inicial con el que cuenta el sistema de peaje cuando se inicia el préstamo de dicho servicio.

Relación de incrementos: esta relación es el cociente entre el valor de un salto inicial y los saltos subsecuentes, cabe señalar que los saltos son tiempos en los que se aumenta el costo

²Diario Oficial: 13 de Octubre de 1993, pp. 53

del servicio, estos pueden ser generados por tiempo o distancia.

Para poder hacer una diferenciación entre un cobro por tiempo y un cobro por distancia se toma como referencia la *velocidad de cambio de arrastre* la cual nos indica un límite para cuando se tiene el mismo valor de cobro, es decir, cuando se tiene una velocidad inferior a la de arrastre se toma una función de tiempo y cuando se supera dicha velocidad se toma una función de distancia.

La *función de tiempo* se encarga de ir acumulando el importe de los tiempos en los cuales el vehículo no está en circulación ó que la velocidad sea menor a la velocidad de cambio de arrastre.

En el caso de la *función de distancia* lo que se hace es ir acumulando el importe de la distancia que se va recorriendo

Ahora bien, el sistema de peaje debe contar con ciertas constantes para poder ser usadas para calcular las distancias recorridas por el vehículo, dichas constantes son las siguientes y se toman tal cual están señaladas en la norma:

“*Constante del sistema de peaje ‘K’*: característica intrínseca del instrumento, que indica la clase y el número de señales que debe recibir para indicar correctamente la distancia recorrida de un kilómetro. Se expresa en pulsos por kilómetro.

Constante del vehículo ‘W’: Característica intrínseca del vehículo que indica la clase y el número de señales que envía al sistema de peaje para una distancia recorrida de un kilómetro. Se expresa en pulsos por kilómetro. Esta constante ‘W’ es función del tipo, desgaste y presión de los neumáticos y de la carga del automóvil. Esta constante debe ser igualada a la constante ‘K’.”³

³Diario Oficial: 13 de Octubre de 1993

Avocándonos en la parte de la electrónica, el sistema debe contener lo siguiente:

- Pulsadores o teclas para el funcionamiento del sistema.
- Un transductor de velocidad, para poder hacer el cálculo de distancias.
- Una memoria de respaldo de información la cual deberá contener lo siguiente:
 - Una memoria de programa fijo con una duración mínima de 5 años sin alteraciones en el programa .
 - Una memoria para el almacenamiento de datos acumulados de por lo menos 1 año de servicio del vehículo.
- Una fuente de energía que será alimentada por la batería del vehículo de tal forma que no repercuta en el buen funcionamiento del mismo.

El sistema deberá contar con protecciones contra interferencias electromagnéticas y ruido ocasionado por el automóvil⁴; contra descargas eléctricas; deberá soportar vibraciones⁵ de 10 a 55 Hz, temperaturas de 60° C, una humedad relativa del 90%.

Un aspecto importante que señalan la norma es la tolerancia que debe tener el sistema al realizar los cálculos, ya sea por distancia y/o por tiempo. Para el caso del cálculo de tiempo se tiene una tolerancia máxima en el error de $\pm 1\%$ para un tiempo de prueba mayor a 180 segundos, en el caso de ser menor el tiempo de prueba el error máximo permitido debe ser de $\pm 1.8\%$. En distancia el error permitido es de $\pm 1\%$, para una distancia mayor a 1 kilómetro, para distancias menores el error máximo permitido es de 10 metros, por ejemplo, si se realiza una prueba a 2.5 kilómetros, el error permitido es de 25 metros, por el contrario, si la distancia de prueba es de 600 metros el error permitido es de 10 metros..

El sistema de peaje también cuenta con un ciertos estados de operación los cuales deben ser respetados por el orden que indica la norma, esto es: "*Libre-En servicio-A pagar*", este proceso no debe ser interrumpido por algún tipo de anomalía en el sistema. Solamente

⁴Se debe cumplir con la norma NMX-I-93, "Ruido radioeléctrico producido por los sistemas en ignición de motores de combustión interna"

⁵Se debe cumplir la norma "NMX-I-7/9, "Equipos y componentes electrónicos-métodos de prueba ambientales y de durabilidad-Prueba Fc vibración (senoidal)"

cuando se termina un ciclo puede ser modificado.

Por último, en lo que respecta a las normas para sistemas de cobro, se hará una breve descripción de los datos que deben ser almacenados en el sistema:

- Distancia total recorrida por el automóvil.
- Distancia total recorrida en servicio.
- Número total de viajes en servicio.
- Número total de cambios o saltos del indicador de precio.
- Importe acumulado.

A todos estos parámetros se les denomina totalizadores y tienen la función de llevar un registro del servicio que se presta. Más adelante se explicará en forma más detallada cada uno de estos totalizadores y como se calculan.

I.4. Planteamiento del problema y propuesta de solución

Para los primeros meses de 1998, el parque vehicular de la Zona Metropolitana de la Ciudad de México (ZMCM) era de alrededor de 4.2 millones⁶ de vehículos de diversas características. De este total de vehículos el 2.8%⁷ correspondía exclusivamente a taxis, es decir, en la ZMCM se contaban con 117,600 vehículos que prestaban sus servicios como taxis. Este total de vehículos se ha visto incrementado considerablemente. Si tomamos los viajes que se realizan por medio de transporte tendríamos la distribución que se presenta en la tabla I.4.1 y en la figura I.4.1.

⁶FUENTE: Dirección General de Prevención y Control de Contaminantes, DDF.1998

⁷Idem

Medio de transporte	Porcentaje de Viajes[%]
Autobuses	32
Microbuses	30.5
Autos Privados	21
Metro, Trolebuses y Tren ligero	12.5
Taxis	3.5
Bicicletas	0.5

Tabla I.4.1. Porcentajes de viajes por medio del transporte público.⁸

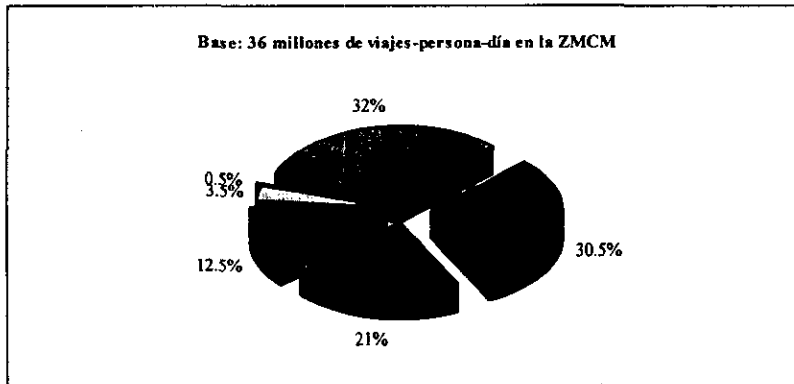


Figura I.4.1. Distribución de viajes por medio de transporte público.

Como podemos observar, del 100 % de los viajes que se realizan, el 3.5 % lo hacen exclusivamente los taxis. Aun cuando es una cantidad pequeña de viajes los que realiza en comparación con el transporte colectivo, vemos que si es muy usado.

Con la idea de participar en el desarrollo de sistemas de cobro para estos vehículos, la empresa PADI solicitó al personal de la coordinación de Instrumentación del Instituto de Ingeniería el “Diseño y construcción de un sistema de peaje para vehículos de transporte público”.

⁸ FUENTE: INEGI, Estudio de Origen-Destino 1998

Para realizar este proyecto se requiere diseñar un dispositivo electrónico que pueda llevar el control y registro de un servicio prestado. Este sistema deberá ser lo más eficiente posible y deberá cumplir con las normas ya establecidas a nivel nacional e, en su momento, internacional. Deberá ser de bajo costo, de gran seguridad, de fácil manejo, con un diseño ergonómico, sin descuidar los puntos que señala la norma, y que sea de una programación breve y fácil.

Dado el planteamiento del problema la propuesta de solución que proponemos es la de crear un sistema totalmente electrónico basado en un microcontrolador. El objetivo de usar el microcontrolador como principal circuito de control en un sistema de peaje es la de realizar las operaciones necesarias de una forma eficaz, rápida y sobretodo con muy buena precisión. Otra razón para hacer el uso de un microcontrolador es la posibilidad de poder colocar no solamente sistema de seguridad, sino también tener la facilidad de colocar otros accesorios como lo son impresoras, lectores, etc.

Como ya hemos observado a través de los apartados anteriores nuestro objetivo es crear un sistema de medición, para esto necesitamos tener conocimiento de los principios básicos de operación de un microcontrolador, así como de cada uno de los componentes externos que se van a colocar para armar un sistema ya en su conjunto; también es necesario saber como debemos realizar las mediciones y los métodos que se usan y que deben ser lo más exactos posibles. Dado lo anterior proseguiremos con el siguiente capítulo del presente trabajo que nos dará las bases para poder realizar el diseño que pretendemos hacer.

CAPÍTULO II

CONCEPTOS BÁSICOS

Este capítulo está enfocado a describir las bases teóricas en que se fundamenta la operación de cada uno de los elementos que integran el sistema de peaje.

II.1. Microcontroladores

Un microcontrolador es un circuito integrado (IC) de una gran escala de integración (LSI) extremadamente complejo, que tiene la capacidad de ser programable. Este dispositivo consta básicamente de los siguientes bloques: Unidad Aritmética Lógica, Registros, Memoria, Puertos, Temporización y Control. Con la interacción de estos bloques, además de la existencia de un programa, el microcontrolador es capaz de realizar operaciones lógicas, aritméticas, controles de memoria, así como el control de dispositivos periféricos de entrada y salida. Para realizar esta interacción el microcontrolador necesita un bloque de instrucciones, el cual toma la función del programa.

Un microcontrolador es un dispositivo que fue creado específicamente para controlar máquinas. Su parte principal es un microprocesador, al cual se le agregan dispositivos, tales como, contadores programables, puertos seriales de comunicación (UART's), puertos paralelos de entrada/salida (I/O), convertidores analógico/digital (A/D), etc.

De los microcontroladores que se encuentran en el mercado, existen varias familias que manejan palabras de ocho bits, como son las de National, Motorola, Hitachi, Zilog, Intel, etc.

A continuación tendremos una descripción de las familias más importantes del mercado que cumplen con las características antes mencionadas.

Familia COP de National

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

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

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

Algunas otras características que ofrece esta familia son:

- a) Costo bajo, en casi todos sus microcontroladores de 8 bits.
- b) Frecuencia de operación de 10 MHz (ciclo de instrucción de 1 microsegundo).

-
- c) 1 kbytes de memoria ROM y 64 bytes de memoria RAM en los MCU's COP820C. 2 kbytes de memoria ROM y 128 bytes de memoria RAM en los MCU's COP840C.

 - d) Reloj de 16 bits de lectura-escritura que opera en tres modos: reloj con registro de autocarga de 16 bits, contador de eventos externo de 16 bits, *reloj con registro de captura de 16 bits*.

 - e) Fuente de interrupción cuyos modos son: interrupción activada por un *reset* de limpieza, modo externo de interrupción con flanco de interrupción programable, reloj de interrupciones o captura de interrupciones y modo de interrupción de *software*.

 - f) Registro *stack-pointer* de 8 bits.

 - g) *Set* de instrucciones para aritmética BCD.

 - h) *Interface* serial de entrada y salida.

 - i) Opciones de uso del *software* para los modos de entrada-salida (tres estados, *Push-Pull*, etc.).

 - j) Entradas tipo *Schmitt-Trigger* en el puerto G.

 - k) La memoria puede ser expandida externamente hasta 32 kbytes.

 - l) Rangos de temperatura de -40°C a +85°C y -55°C a +125°C.

Familia MC68HC11 de MOTOROLA

Esta familia está formada por un grupo sofisticado de microcontroladores de 8 bits, con tecnología HCMOS. Su velocidad de bus nominal es de 2 MHz, cada elemento de esta familia tiene las siguientes características.

- a) Un sistema de memoria de 8 kbytes de memoria ROM, 512 bytes de memoria EEPROM y 256 bytes de memoria RAM.
- b) Un puerto de 8 líneas multiplexadas a un convertidor Analógico/Digital con una resolución de 8 bits.
- c) Sistema de monitoreo tipo *watchdog* de la circuitería interna del circuito integrado, para prevenir fallas.
- d) Una *interface* para comunicación serial asíncrona SCI y una *interface* para periféricos seriales síncronos SPI.
- e) El sistema principal de reloj tiene tres líneas de captura de entrada, 5 líneas de salida de comparación y una función real para interrupción.
- f) Subsistema acumulador de pulsos de 8 bits, capaz de contar y/o convertir eventos externos con ó sin periodicidad, entre otras aplicaciones.

Además, a través de 2 modos de software, *WAIT* y *STOP*, es posible controlar la potencia requerida por el CI, logrando como consecuencia conservar energía adicional. Estos dos modos hacen que la familia MC68HC11 sea de gran interés para aplicaciones automotrices ó donde se requiere un buen aprovechamiento de una batería.

Es muy importante aclarar que el conjunto de instrucciones es completamente compatible con toda la familia. Así mismo algunas instrucciones son tan versátiles que permiten que varias operaciones sean ejecutadas por una sola instrucción.

A continuación se muestra en la tabla II.1.1, las características de los principales miembros de la familia de microcontroladores de MOTOROLA 68HC11.

	68HC11A8	68HC11E9	68HC11E2	68HC11F1	68HC11A8	68HC11D3	68HC11D3	68HC11E9
ROM	8 k	12 K	0	0	8 K EEPROM	4 K EEPROM	4 K	12 K EEPROM
RAM	256	512	256	1 K	256	192	192	512
EEPROM	512	512	2 K	512	512	0	0	512
RELOJ	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT	16 BIT
SPI	SI	SI	SI	SI	SI	SI	SI	SI
SCI	SI	SI	SI	SI	SI	SI	SI	SI
A/D	SI	SI	SI	SI	SI	NO	NO	SI
INT.T. REAL	SI	SI	SI	SI	SI	SI	SI	SI
WATCH DOG	SI	SI	SI	SI	SI	SI	SI	SI
ENCAP- SULADO	PLCC 52 DIP 48	PLCC 52	PLCC 52	PLCC 68	PLCC 52	PLCC 44 DIP 40	PLCC 44 DIP 40	PLCC 52
I/O	38	38	38	30	38	32	32	38
VEL.DEL BUSMHz	0-2.1	0-2.1	0-2.1	0-4.2	0-2.1	0-2.1	0-2.1	0-2.1

Tabla II.1.1. Características de la familia MC68HC11.

Familia 80C51

La familia SIGNETICS de PHILIPS tiene una amplia variedad de microcontroladores para las aplicaciones en tiempo real, tales como instrumentación médica, control industrial y aplicaciones de control automotor.

Estos microcontroladores son de 8 bits y están disponibles en versiones que cuentan internamente con ROM, EEPROM, o solamente el CPU. Además pueden direccionar externamente una memoria de 64 kbytes para datos y 64 kbytes de memoria de programa.

Algunas características del 80C51 son:

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

En la tabla II.1.2. se listan algunos miembros de esta familia con algunas de sus características.

NOMBRE	VERSION SIN ROM	VERSION EEPROM	A/D	PUERTOS	PWM	RELOJ	PUERTO SERIAL	ROM bytes	RAM bytes	TIPO DE CIRCUITO
80C51	80C31	87C51	-	4	-	2 st.	UART	4 K	128	HMOS
80CL51	-	-	-	4	-	2 st.	UART	4 K	128	COMS
80C52	80C32	87C52	-	4	-	2 st.	UART	8 K	256	COMS
83C053	-	87C054	-	3 1/2	9	2 st.	-	8 K	192	COMS
83CL410	80C410	-	-	4	-	2 st.	I ² C	4 K	128	COMS
83C451	80C451	87C451	-	7	-	2 st.	UART	4 K	128	COMS
83C528	80C528	87C528	-	4	-	2 Reloj. 2 st.	UART I ² C	32 K	512	COMS
83C550	80C550	87C550	8c/8b	4	-	2 st. WATCHDOG	UART	4 K	128	COMS
83C552	80C552	87C552	8c/10b	6	2	2 st. 2 Reloj. WATCHDOG	UART I ² C	8 K	256	COMS
83C562	80C562	-	8c/8b	6	2	2 st. 2 Reloj. WATCHDOG	UART	8 K	256	COMS
83C575	80C575	87C575	4 CO MP	4	5	2 st. 2 Reloj. WATCHDOG PCA	UART	8 K	256	COMS
87C592	-	87C592	8c/10b	6	2	2 st. 2 Reloj. WATCHDOG	UART CAN	16 K	512	COMS
83C652	80C652	87C652	-	4	-	2 st.	UART I ² C	8 K	256	COMS
83C654	-	87C654	-	4	-	2 st.	UART I ² C	16 K	256	COMS
83C751	-	87C751	-	2 3/8	-	1 st.	I ² C	2 K	64	COMS
83C752	-	87C752	5c/8b	2 5/8	-	1 st.	I ² C	2 K	64	COMS
83C851	80C851	-	-	4	-	2 st.	UART	4 K	128	COMS
83C852	-	-	-	4	-	2 st.	-	6 K	256	COMS

Tabla II.1.2. Características de la familia 80C51.

II.2. Memorias

Las memorias son dispositivos o sistemas que almacenan información digital en grandes cantidades. La capacidad en memoria electrónica en sistemas digitales varía desde poco menos de 100 bits, para una simple calculadora de bolsillo de 4 operaciones, a entre 10^4 y 10^6 bits para computadoras personales y hasta 10^9 bits o más para las grandes computadoras comerciales. Los diseñadores de circuitos normalmente hablan de capacidades de memoria en términos de bits; por otro lado, los diseñadores de sistemas suelen hablar de capacidades de memoria en términos de bytes (8 bits), donde cada byte representa un carácter alfanumérico.

Excepto para los sistemas más pequeños, la capacidad de almacenaje se cuantifica en unidades de kilobytes o kbytes. Ya que el direccionamiento de las memorias está basado en códigos binarios, las capacidades más comunes son aquellas que son potencias de 2. Según el convenio tenemos:

$$\begin{aligned} 1 \text{ Kbyte} &= 1024 \text{ bytes} \\ 64 \text{ Kbytes} &= 65536 \text{ bytes} \end{aligned}$$

La figura II.2.1. muestra un registro de almacenamiento con capacidad para 1 byte o palabra de varios bits. Un registro como éste puede diseñarse utilizando un flip-flop SR o D para cada bit. Sin embargo, no es nada económico construir grandes memorias de esta forma. Cada flip-flop o célula de almacenamiento de 1 bit de un registro requiere bastantes transistores y tiene dos o más conexiones para el acceso de datos.

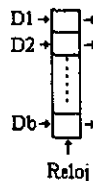


Figura II.2.1. Registro de almacenamiento.

Según su acceso, las memorias se clasifican como:

- a) *De acceso directo o aleatorio*: En estas memorias se asocia una dirección a cada palabra, y al suministrar a la memoria la dirección, determina que se suministre o modifique la información de la palabra asociada a dicha dirección en un tiempo que no depende del valor de la dirección.
- b) *De acceso secuencial*: En estas memorias el tiempo de acceso a una palabra determinada depende de su posición con respecto a una posición de referencia. El dato es accesado mediante una secuencia temporal.
- c) *Asociativas*: En estas memorias el acceso a una palabra determinada se consigna mediante la información contenida en una parte de la propia palabra.

Memorias RAM

La organización preferida en muchas memorias grandes puede verse de una manera esquemática en la figura II.2.2. Esta organización es conocida con el nombre de *memoria de acceso aleatorio* (RAM, Random Acces Memory). Este nombre proviene del hecho de que las direcciones de memoria pueden accederse de una forma aleatoria, a una cadencia fija independiente de su posición física tanto para la lectura como para la escritura. La matriz de almacenamiento de una RAM se construye mediante una agrupación de células, de manera que puedan compartir conexiones en filas horizontales y columnas verticales. Las líneas horizontales, que son gobernadas sólo desde fuera de la matriz de células, a menudo son llamadas *líneas de palabra* (word lines), mientras que las líneas verticales, a lo largo de las cuales los datos entran y salen de las células son llamadas *líneas de bits* (bit lines).

Las memorias de acceso aleatorio de *lectura-escritura* (RWM) pueden almacenar la información en flip-flops o simplemente como cargas en capacitores. Los tiempos necesarios

para lectura y escritura son aproximadamente iguales, oscilando en el rango de 10 a 500 ns. Puesto que las memorias de lectura y escritura almacenan los datos en circuitos activos, son volátiles: es decir, la información almacenada se pierde si la alimentación se interrumpe.

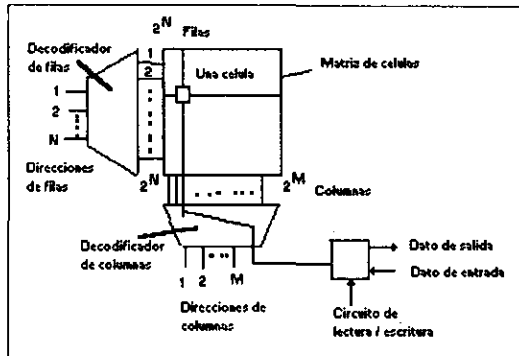


Figura II.2.2. Esquema de las memorias de acceso aleatorio.

Memorias ROM

Las memorias de *sólo lectura* (ROM, Read Only Memory), almacenan información de acuerdo con la presencia o ausencia de diodos o transistores que unen filas o columnas. Su característica principal es que mantienen los datos almacenados aún cuando no se tenga presente la tensión de alimentación a las mismas, es por ello que a estas memorias se les conoce como no volátiles. Las memorias ROM emplean la organización vista en la figura II.2.2. y tienen velocidades de lectura comparables a las memorias de lectura-escritura, este tipo de memorias se pueden clasificar en: PROM, EPROM y EEPROM. Las ROM más sencillas son escritas en el mismo proceso de fabricación mediante la configuración física del circuito integrado (CI); en consecuencia, los cambios en los datos almacenados son imposibles. A este tipo de memorias se les denomina programables por máscara o, más sencillamente, ROM de máscara.

Las memorias *programables de sólo lectura* (PROM) tienen una conexión o camino entre cada fila y cada columna una vez fabricadas, correspondiendo a un "1" en cada posición

de dato. Las celdas de almacenamiento pueden ser selectivamente cambiadas al estado lógico "0" después de su fabricación, aplicando pulsos eléctricos apropiados para abrir de forma selectiva (desconectar) los caminos entre filas y columnas. Una vez programada, o fundida, los ceros no pueden volver a cambiarse a "1".

Las memorias de sólo lectura *programables y borrables* (EPROM, Erasable Programmable Read Only Memory) presentan inicialmente todos los bits en el estado lógico "1". Se programan eléctricamente con pulsos de voltaje considerablemente más grandes que el voltaje de polarización de la memoria. Estos pulsos mueven una compuerta flotante que pone en estado de saturación al transistor que contiene la información lógica (estado lógico "0"), esta compuerta permanece en ese lugar hasta que exista un flujo de electrones, para que regrese a su posición original y esto se logra exponiendo el CI a la luz ultravioleta, ocasionando que las memorias puedan ser borradas. Los encapsulados para estos componentes tienen ventanas transparentes para permitir la irradiación ultravioleta.

Las memorias de sólo lectura *programables y borrables eléctricamente* (EEPROM), estas últimas son las más avanzadas y costosas de las memorias PROM. Al contrario que las EPROM, que deben ser totalmente borrables y reescritas aunque sólo se desee cambiar un solo bit, las EEPROM pueden borrarse selectivamente. La operación de escritura y borrado para las PROM, requieren unos tiempos que oscilan entre 10 microsegundos y algunos milisegundos.

El *tiempo de acceso* de lectura es el retraso que existe desde la aparición de la dirección hasta que el dato almacenado en dicha dirección está disponible en la salida. El *tiempo de ciclo* es el inverso de la frecuencia en la cual puede cambiarse la información de la dirección mientras se lee o se escribe en posiciones aleatorias. Normalmente dentro de las características de una memoria se especifica el mínimo valor del tiempo del ciclo, por debajo del cual pueden producirse errores. Los tiempos mínimos del ciclo para lectura y escritura no son necesariamente iguales, pero por simplicidad algunos diseñadores lo utilizan así tanto para lectura como para escritura, el tiempo de acceso oscila entre el 50% y 90% del ciclo de lectura.

En la figura II.2.3. se observan los parámetros temporales de las memorias. Las memorias de solo lectura se utilizan para el almacenamiento constante de información de control e instrucciones que son básicamente el programa para los sistemas digitales. Pueden ser consideradas como componentes que proporcionan una salida binaria fija para cada dirección binaria dada.

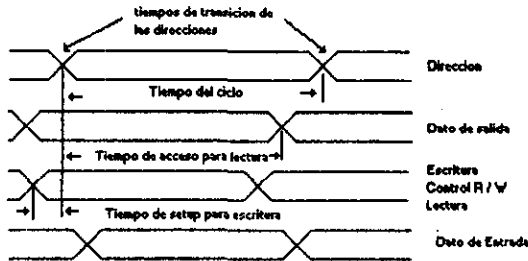


Figura II.2.3. Definición de los parámetros temporales de las memorias.

II.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 por que no todas las cantidades que se necesitan medir se pueden mostrar con tanta facilidad como otras. Generalmente se puede efectuar una mejor medición de una cantidad si ésta se puede convertir a otra forma que se pueda mostrar con facilidad y exactitud. Por ejemplo, el termómetro de mercurio convierte los cambios de temperatura a cambios en la longitud de una columna de mercurio. Como este último cambio es muy fácil de medir, el termómetro de mercurio es un dispositivo conveniente para detectar cambios de temperatura. Por otro lado, el cambio real de temperatura no es tan fácil de mostrar en forma directa.

Los transductores eléctricos constituyen la vasta mayoría de los transductores que se emplean hoy en día. La instrumentación eléctrica está tan adelantada que generalmente hay varios métodos diferentes para convertir la mayor parte de las cantidades físicas en señales

eléctricas. Además, si las señales débiles se convierten a una forma eléctrica, se pueden amplificar con fidelidad hasta que sus amplitudes lleguen a ser bastantes grandes para poder mostrarse con facilidad.

Características de los Transductores

Cuando se intenta hacer una medición de una cantidad no eléctrica, convirtiendo la cantidad a una forma eléctrica, se debe seleccionar un transductor adecuado 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 ha de medir. Esto debe comprender el conocimiento del rango de magnitudes y frecuencias que se espera que exhiba la cantidad. Cuando se ha establecido el problema, se deben examinar los principios del transductor disponible para medir la cantidad deseada. Si uno o más principio de transducción son capaces de producir una señal satisfactoria, se debe decidir si usar un transductor comercial o intentar construir el transductor. Si hay transductores comerciales disponibles a un precio razonable, la elección probablemente sea comprar uno de ellos.

Cuando se examinan las especificaciones de un transductor en especial, se deben considerar los siguientes puntos para determinar su capacidad para una medición en particular:

1. *Rango.* El rango del transductor debe ser lo suficientemente grande tal que abarque todas las magnitudes esperadas de la cantidad a ser medida.
2. *Sensibilidad.* Para obtener datos significativos, el transductor debe producir una señal de salida suficiente por unidad de entrada medida.
3. *Efecto de carga.* Como los transductores siempre consumirán algo de energía del efecto físico que se está probando, debe determinarse si se puede despreciar esta absorción o si se puede aplicar factores de corrección para compensar las lecturas por las pérdidas.

4. *Respuesta a la frecuencia.* El transductor debe ser capaz de responder a la velocidad máxima de cambio en el efecto que se esté observando.
5. *Formato de salida eléctrica.* La forma eléctrica de salida del transductor debe ser compatible con el resto del sistema de medición. Por ejemplo, un voltaje de salida DC no sería compatible con un amplificador que sólo pueda responder a señales de corriente alterna.
6. *Impedancia de salida.* La impedancia de salida del transductor debe tener un valor que lo haga compatible con las siguientes etapas eléctricas del sistema. Si existe incompatibilidad de impedancia, se debe agregar dispositivos modificadores de señal al sistema para superar este problema.
7. *Requerimiento de potencia.* Los transductores pasivos necesitan de excitación externa. Así, si se deben emplear transductores pasivos, es necesario asegurar que haya disponibles fuentes de poder eléctricas adecuadas para operarlos.
8. *Medio físico.* El transductor seleccionado debe poder resistir las condiciones ambientales a las que estará sujeto mientras se efectúe la prueba. Parámetros tales como temperatura, humedad y sustancias químicas corrosivas podrían dañar algunos transductores y a otros no.
9. *Errores.* Los errores inherentes a la operación del mismo transductor, o aquellos errores originados por las condiciones del ambiente en la medición, deben ser lo suficientemente pequeños o controlables para que permitan tomar datos significativos.

Para nuestro sistema de peaje es necesario medir las revoluciones del vehículo, por lo cual mencionaremos a continuación la variable velocidad.

Velocidad

La velocidad lineal comúnmente se detecta mediante dispositivos electromagnéticos, en los que un cambio en el flujo magnético induce una fuerza electromotriz (fem) en un conductor (devanado). La fem inducida es igual al producto del número de vueltas en el devanado y al cambio de flujo magnético en la unidad de tiempo, (ecuación II.3.1.)

$$e = -N \frac{d\Phi}{dt} \quad (\text{II.3.1})$$

Donde:

e = fem inducida

N = número de vueltas del devanado

$$\frac{d\phi}{dt} = \text{cambio de flujo magnético } (\phi) \text{ por unidad de tiempo (t)}$$

El signo negativo de la ecuación indica que la dirección de la fem inducida se opone al cambio de flujo que lo produce.

Determinando la velocidad de rotación del dispositivo transductor y con el conocimiento del radio del mismo, se puede obtener la velocidad lineal.

Ya que la aceleración es la variación de velocidad en el tiempo, la velocidad puede ser determinada integrando la salida de un acelerómetro. En muchas aplicaciones con acelerómetros, se añaden circuitos integradores al circuito acondicionador para poder generar tanto la aceleración como la velocidad.

Para poder medir y tomar en cuenta la velocidad analizaremos algunos transductores de velocidad, de los cuales posteriormente seleccionaremos uno.

Transductores de velocidad electromagnéticos

Este tipo de transductor consiste en un devanado montado en una carcaza de acero pulido y un núcleo coaxial cilíndrico con imán permanente unido a un eje con un tope engarzado. El núcleo se mueve dentro del devanado, con el movimiento del objeto al que se ha unido el extremo del eje, produciendo una salida en el devanado, proporcional a la velocidad. Otro principio diferente es el diseño de devanado fijo como el que se muestra en la figura II.3.1. Un imán permanente se coloca entre dos muelles, en los extremos del imán cilíndrico, unos aros de aleación oro-paladio minimizan la fricción, moviéndose dentro de unos casquillos de acero cromoplateado pulido. El transductor completo se monta sobre el objeto cuya velocidad oscilante quiere medirse. Cuando la frecuencia de este movimiento excede de la frecuencia natural del sistema suspendido (del orden de 15Hz) el imán permanece invariable en una posición determinada. El devanado fijado dentro de la carcaza del transductor, se mueve relativamente al imán, ahora estacionario, con el movimiento del objeto a medir. El consiguiente cambio de flujo resultante provoca una tensión de salida en el devanado proporcional a la velocidad.

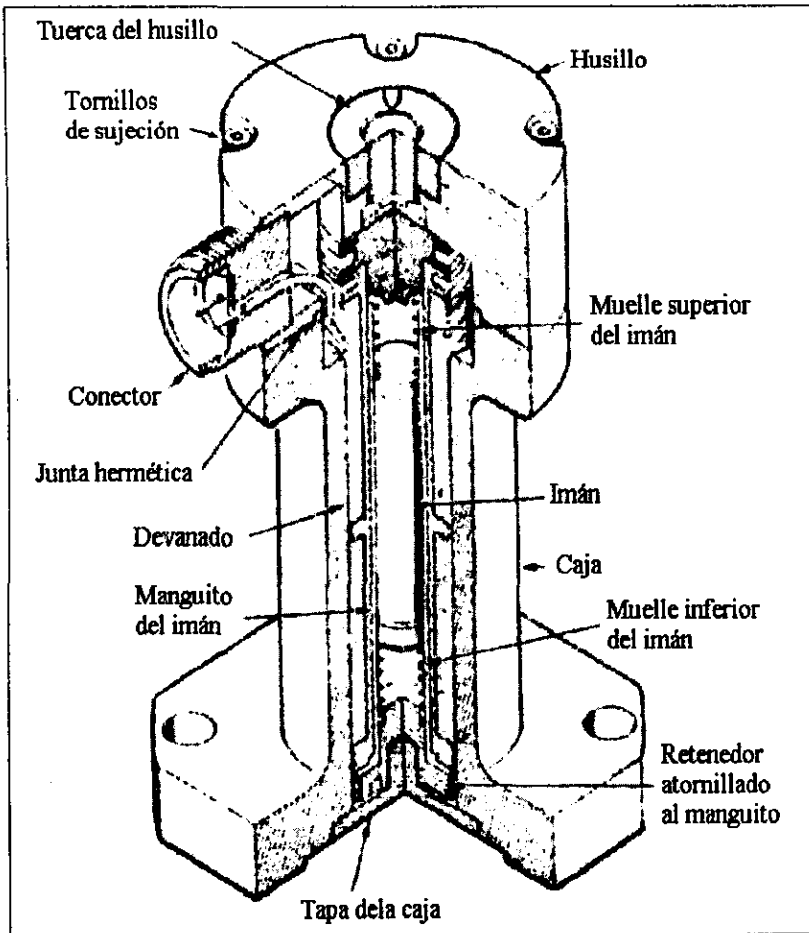


Figura II.3.1. Transductor de velocidad electromagnético.

Velocidad angular (R.P.M.)

Para velocidades angulares, lo más conveniente para una correcta medición es utilizar una combinación de una rueda dentada (engrane), un sensor magnético de proximidad y un contador de pulsos por unidad de tiempo.

Transductores de velocidad angular

- **Tacómetros extensiométricos:** Utilizan una viga en voladizo en la que se le han aplicado unas galgas extensiométricas, un disco excéntrico está unido al eje de rotación. La viga se encuentra en contacto con el disco de manera que se provoca una deflexión en cada revolución del objeto.
- **Tacómetros del tipo interruptor:** Utilizan la apertura y cierre de contactos. Pueden usarse relés magnéticos, actuando por un imán acoplado al eje. En algunos diseños se tienen pares de contactos rotativos entre los que se conecta un condensador de manera que se carga alternativamente con tensiones de polaridad opuesta, los contactos estacionarios se conectan a una fuente de C.C. en serie con un condensador a través del cual se producen pulsos de corriente.
- **Tacómetro electromagnético de rotor dentado:** Este tipo de tacómetro utiliza un rotor ferromagnético con salientes dentadas o configuración de engranaje equiespaciados en conjunción con un devanado de transducción electromagnética. El conjunto devanado transductor puede ser del tipo de “efecto hall” o puede ser inductivo, sin embargo la mayoría son del tipo electromagnético. El conjunto se coloca dentro de una cápsula herméticamente sellada. La parte frontal de la cápsula suele estar atornillada para facilitar la instalación y el ajuste entre el polo y el diente rotor (figura II.3.2).

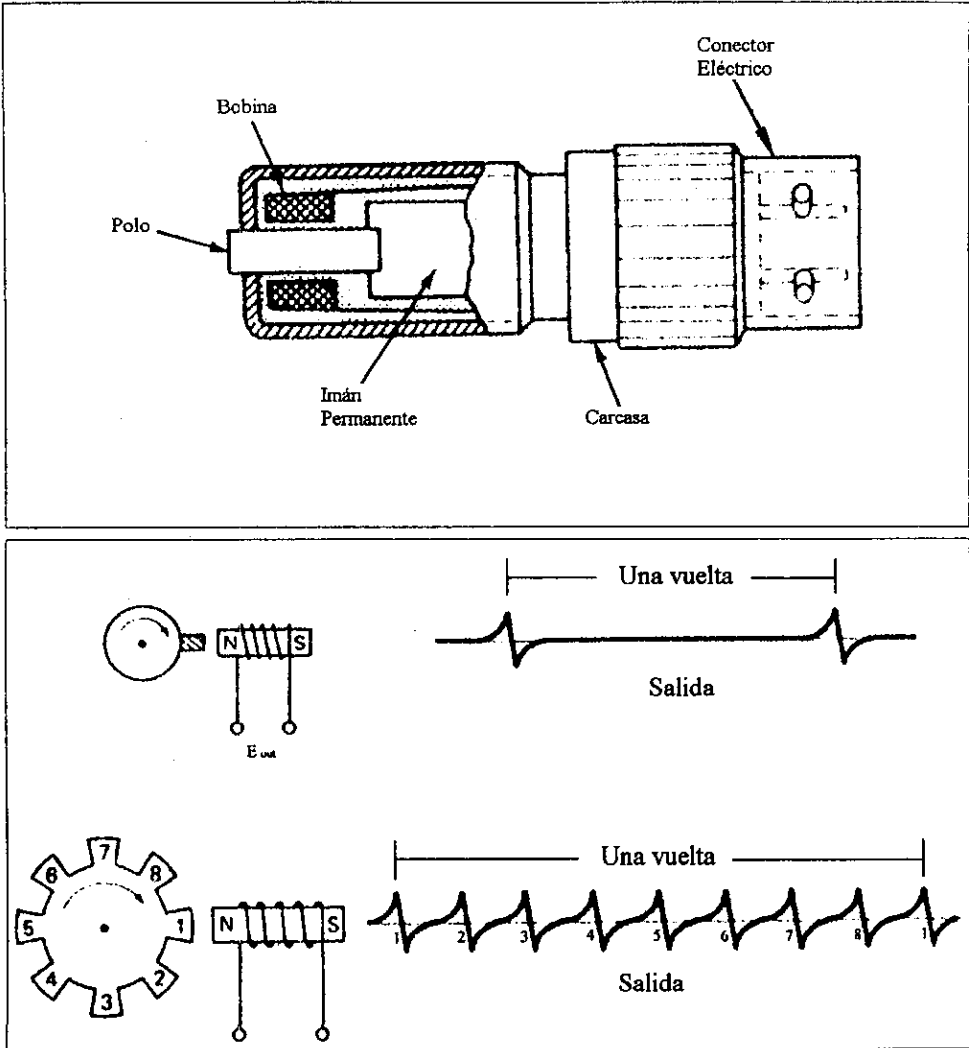


Figura II.3.2. Tacómetro electromagnético de velocidad angular.

Interruptor tipo *Reed*

Este tipo de interruptor cada vez está siendo más utilizado en circuitos electrónicos, ya que es un elemento que dentro de sus principales características están su bajo costo, una respuesta rápida y su tamaño pequeño. En la tabla II.3.1 se muestran algunas de sus características.

Compacto y ligero	El interruptor tipo <i>reed</i> puede montarse en un espacio muy limitado, es ideal para el uso en equipo miniaturizado.
Herméticamente sellado	Los elementos interruptores del interruptor tipo <i>reed</i> se sellan herméticamente en una atmósfera de gas inerte, para que éstos nunca se expongan al ambiente externo.
Vida ilimitada	Los interruptores tipo <i>reed</i> no emplean ninguna parte corrediza, para que no haya degradación relacionada en la calidad de los materiales usados, asegurando una vida mecánica casi ilimitada.
Funcionamiento de alta velocidad	Cada elemento movable tiene una masa muy baja que produce una velocidad alta de funcionamiento. Esto le permite al interruptor tipo <i>reed</i> ser usado como una <i>interface</i> para un transistor o para un circuito integrado.

Tabla II.3.1. Características de los interruptores tipo *Reed*.

Construcción

El interruptor tipo *reed* consiste en un par de terminales flexibles hechas de un material magnético, y sellado en un tubo ó cápsula de vidrio lleno de un gas inerte. Las terminales se traslapan pero tienen una separación muy pequeña. El área de contacto de cada terminal se chapa con un metal noble, como Rodio o Rutenio, para proporcionar características estables y larga vida al interruptor. En la figura II.3.3 se ilustra un interruptor tipo *reed*.

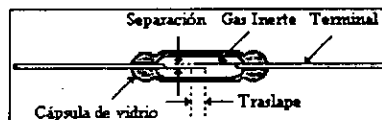


Figura II.3.3. Interruptor tipo *reed*.

Funcionamiento

La aplicación de un campo magnético, generada por un imán permanente o una bobina, en el interruptor tipo *reed* causa que ambas terminales sean magnetizadas. Esto produce un polo Norte en el área de contacto de una terminal, y un polo Sur en la otra terminal, como se ilustra en la figura II.3.4. Si la fuerza magnética de atracción supera la fuerza resistiva causada por la elasticidad de la terminal, las terminales entran en contacto (Pull-In), esto provoca que el circuito se cierre. Una vez que el campo magnético sea retirado, las terminales están de nuevo separadas por el efecto de elasticidad de la terminal (Drop-Out), esto provoca que el circuito se abra. En la figura II.3.4 podemos observar el funcionamiento del interruptor tipo *reed*.

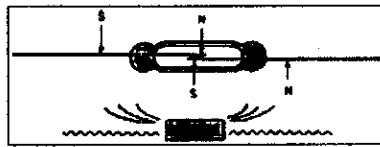


Figura II.3.4. Campo magnético.

Formas de utilizar un Imán con un interruptor tipo *reed*

Las formas más utilizadas a menudo para usar un interruptor tipo *reed* con un imán son mostradas en las siguientes figuras:

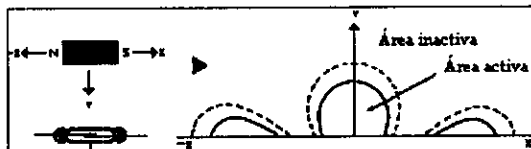


Figura II.3.5. Posición Horizontal.

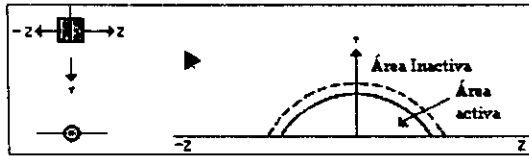


Figura II.3.6. Posición Transversal.

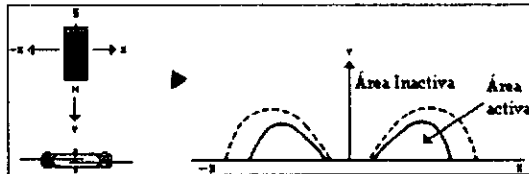


Figura II.3.7. Posición Perpendicular.

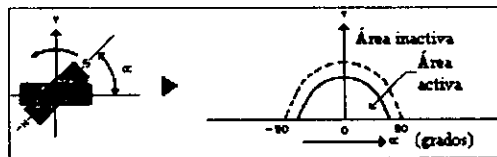


Figura II.3.8. Activación Rotatoria.

Aplicaciones con interruptores tipo reed

Dando forma a las terminales (cortando y doblando).

- Cuando se empieza a cortar ó doblar se debe tener precaución en el extremo para no ejercer cualquier tensión indebida que puede producir daño o deterioración en la cápsula de vidrio. El sujetar apropiadamente las terminales es necesario (ver Fig. II.3.9).

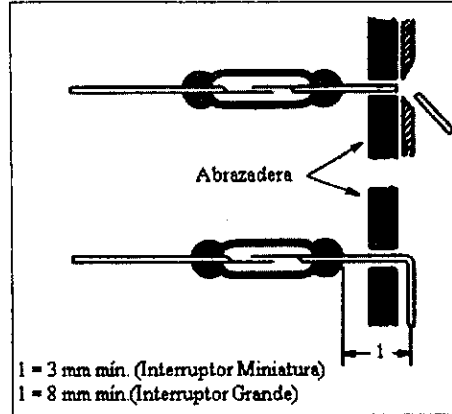


Figura II.3.9. Proceso de doblado y cortado.

- La distancia recomendada, del extremo de la cápsula de vidrio, para empezar a doblar y cortar las terminales (distancia "l" en Fig. II.3.9) es de 3 mm mínimo para el interruptor tipo *reed* miniatura (longitud de vidrio 9 a 20 mm) y 8 mm mínimo para el interruptor tipo *reed* grande (longitud de vidrio de 30 mm ó más).
- Como resultado de cortar las terminales externas, la corriente al momento del contacto y al momento en que se termina el contacto aumentará. Hay que tener en cuenta esto durante el diseño (ver Fig. II.3.10). De igual forma, doblando las terminales externas la corriente aumentarán.

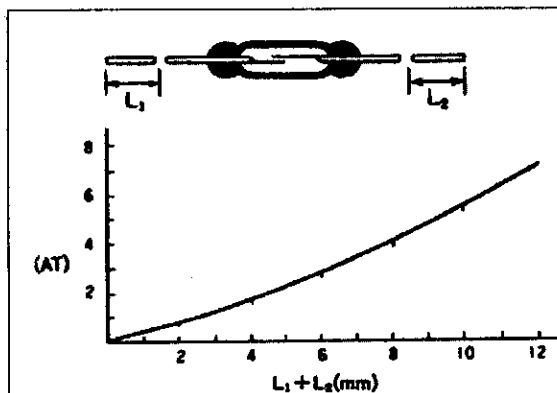


Figura II.3.10. Gráfica de corriente en función de la distancia.

Montaje del interruptor tipo *reed*

- El tiempo excesivo de exposición a altas temperaturas pueden causar daño a la cápsula de vidrio (crujido, goteo, etc.). Es necesario aplicar las técnicas de soldadura recomendables. Las condiciones recomendables de soldadura son: 280° C a 300° C para el caudín, y 250° C a 300° C para la soldadura de ola.
- Cuando se están soldando las terminales del interruptor tipo *reed*, el campo electromagnético generado por la corriente de la soldadura puede operar el interruptor, esto a su vez puede causar daño en el contacto. Se debe tener precauciones especiales durante el proceso de soldadura, estas precauciones son las siguientes: tener en cuenta el voltaje al que se esta soldando, la corriente y el tiempo, ya que si no se tiene precaución el interruptor tipo *reed* podría resultar dañado.

Nunca debe soldar ambos extremos del interruptor tipo *reed* al mismo tiempo.

- Al montar en una tarjeta de circuito impreso (*pcb*) debe prestarse atención a las deformaciones del *pcb* y las características de expansión térmica. La tensión causada por estos factores también puede dañar la cápsula de vidrio.
- Al montar un interruptor tipo *reed* en un *pcb*, se recomienda dar forma a las terminales para proporcionar un espacio adecuado entre el *pcb* y la cápsula, o para colocar el interruptor tipo *reed* en una apertura (*cutout*) ó espacio reservado en el *pcb* (ver Fig. II.3.11).

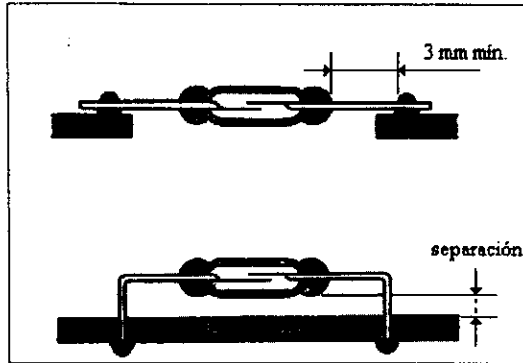


Figura II.3.11. Forma de colocar un interruptor tipo *reed*.

Choque y Vibración

- Cuando un interruptor tipo *reed* se deja caer hacia una superficie dura (suelo) de más de 30 cm de altura, las características eléctricas (Pull-In, Drop-Out, etc.) se alterarán.

Después de que un interruptor tipo *reed* se ha dejado caer y antes de utilizarlo en la aplicación real, se debe asegurar que sus características todavía estén dentro de los límites aceptables.

Esto es también aplicable después de haber aplicado una fuerza de torsión al interruptor tipo *reed*.

- No use el interruptor tipo *reed* por encima de su frecuencia de resonancia.

Protección del Interruptor tipo *Reed*

- Cargas inductivas: Cuando se usa un interruptor tipo *reed* para cargas inductivas como motores, relevadores, solenoides, etc., los contactos estarán sujetos a voltajes inducidos altos durante la apertura de los contactos (circuito de carga). Tales voltajes inducidos altos (transitorios) pueden causar daño al interruptor tipo *reed* o pueden reducir su vida

significativamente. Por lo tanto, los circuitos de protección como: RC (amortiguador), varistores o diodos, son recomendados (ver Fig. II.3.12).

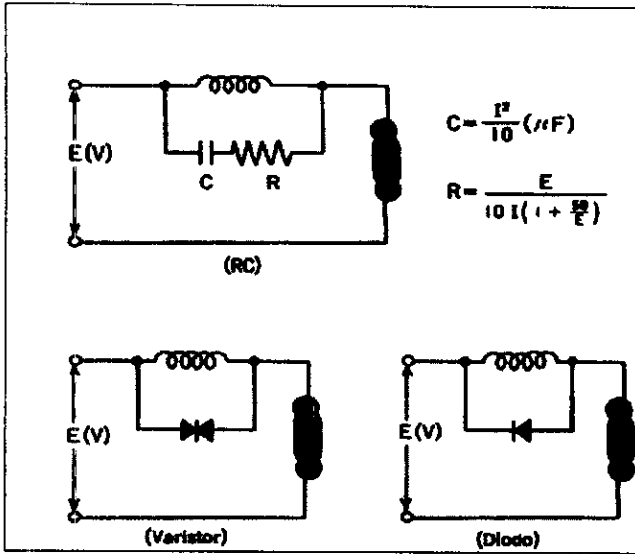


Figura II.3.12. Circuitos de protección para cargas inductivas.

- Cargas capacitivas: Cuando se usa un interruptor tipo *reed* para cargas capacitivas como capacitores, lámparas incandescentes o cables largos (harnesses), los contactos estarán sujetos a un flujo de corriente alta (inrush). Por lo tanto, los circuitos de protección como: supresores de flujo o resistencias limitadoras de corriente, son recomendados (ver Fig. II.3.13).

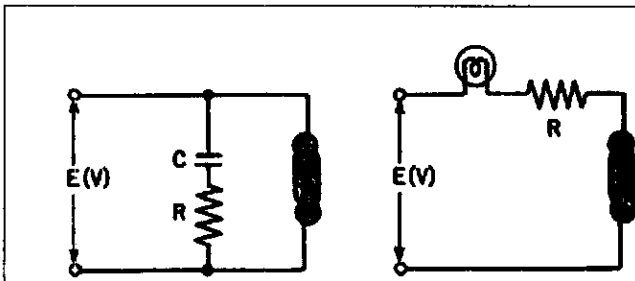


Figura II.3.13. Circuitos de protección para cargas capacitivas.

Interferencia Magnética Externa

- Cuando se encuentra un interruptor tipo *reed* junto con su imán o bobina funcionando, las fuentes cercanas de interferencia magnética fuerte como platos de acero, transformadores, etc., alterarán las características operacionales del interruptor tipo *reed* y probablemente provocaran un mal funcionamiento.

Los casos específicos pueden ser muy diferentes entre uno y otro, y por consiguiente las fuentes de posible interferencia (interacción) deben investigarse en cada aplicación.

- Cuando se usan múltiples interruptores tipo *reed* muy próximos entre sí, la interferencia magnética similar puede causar cambios en las características y provocar un mal funcionamiento. Si tal interferencia es observada, los interruptores tipo *reed* deben espaciarse entre sí más de 15 mm.

En la tabla II.3.2 se muestran algunos interruptores tipo *reed* y sus características.


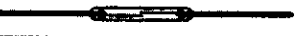
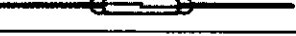
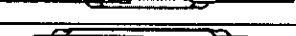
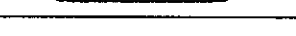


Interruptores para aplicaciones generales.	HYR 1532	
	HYR 2001	
Interruptores de potencia de 10 Watts.	HYR 2034	
	HYR 4001	
	HYR 5051	
Interruptores Ultraminiatura. Con longitud de cápsula de vidrio de 9.5 a 12 mm. Estos interruptores usan un espacio muy pequeño. Las aplicaciones incluyen relevadores de miniatura tipo <i>reed</i> , teclados, etc.	HYR 1002	
	HYR 1003	

Tabla II.3.2. Características de los interruptores tipo *reed* (continúa).

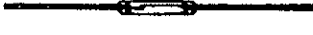





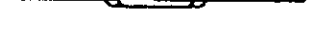





<p>Interruptores de alta potencia. Conmutan potencia que está entre 20 y 70 Watts. Puede usarse para conmutar directamente voltajes de línea, por ejemplo 110 VAC o 220 VAC.</p>	HYR 1531 (para 110 VAC)	
	HYR 2003-1 (para 220 VAC)	
	HYR 2003-2 (alta potencia)	
	HYR 2016 (para más de 220 VAC)	
<p>Interruptores para altos voltajes de ruptura. Dichos voltajes están entre 1,000 a 2,000 Volts. Los interruptores tipo reed son capaces de soportar voltajes de 15 KV. También hay disponibles en tamaños más grandes.</p>	HYR 1559	
	HYR 2016	
	HYR 5006	
	HYR 5007	
	HYR 5008	
<p>Interruptores con alta resistencia de aislamiento. Con un porcentaje de resistencia de aislamiento de más de 10^{13} Ohm.</p>	HYR 2010	
<p>Interruptores de RF y de baja EMF térmica. Con rangos de EMF térmica tan bajo como $5 \mu\text{V} / ^\circ\text{C}$, (otros interruptores tienen aprox. $40 \mu\text{V} / ^\circ\text{C}$). Estos interruptores se aplican a menudo en equipo de prueba. Estos también pueden ser utilizados en aplicaciones de RF debido a sus características de pérdida de inserción.</p>	HYR 2011	
<p>Interruptores de alta corriente de inmersión. Pueden ser usados como interruptores para lámparas incandescentes o cargas capacitivas sin resistencias externas que limiten la corriente.</p>	HYR 1506	

Tabla II.3.2. Características de los interruptores tipo *reed* (continúa).

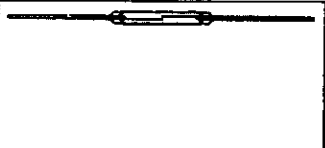
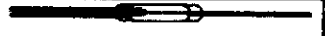
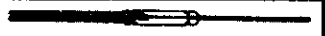

Interruptores de cierre diferencial Una característica es que presenta un histéresis magnético bajo entre el cierre y la apertura del contacto.	HYR 2031-1	
Interruptores de forma tipo "C" Estos interruptores pueden ser de un polo y dos contactos.	HYR 1554	
	HYR 1555	
	HYR 4003	

Tabla II.3.2. Características de los interruptores tipo reed.

Para poder visualizar la información en nuestro sistema, requerimos de un *display* ó visualizador, por lo cual en el tema siguiente mencionaremos algunos visualizadores.

II.4. Visualizadores

A continuación se presenta una descripción general de algunos visualizadores analizados.

Light-Emitting Diode Displays (LED's)

Es un dispositivo de estado sólido que convierte la electricidad en luz. Se trata de elementos semiconductores que utilizan el fenómeno de recombinación radiactiva, también conocida como fenómeno de inyección electroluminiscense, el cual ocurre cuando cargas minoritarias son inyectadas dentro de una juntura P. Dependiendo del material semiconductor usado, la frecuencia de onda de la luz emitida puede ser alterada. De tal forma que existen LED's que efectúan emisión de luz visible y LED's con emisión de infrarrojos, los primeros se definen cuando su frecuencia de onda se encuentra en la región visible y son usados en la elaboración de indicadores y *displays*. Los LED's de la región infrarroja son usados para comunicación óptica y otras formas de información, transmisión o proceso de señales.

Las ventajas que poseen este tipo de dispositivos son básicamente tres:

1. Los LED's pueden proporcionar pequeñas fuentes de luz de alta capacidad luminosa.
2. Los LED's requieren un bajo voltaje de manejo (desde 2V), lo que permite una perfecta integración en diversos sistemas.
3. Son capaces de responder a transiciones de alta velocidad.

El fenómeno de luminiscencia es producido por la recombinación radioactiva entre dos niveles referidos a los estados de energía de los electrones. El rango de la recombinación está determinado por el producto de la densidad de electrones en el nivel alto de energía. La densidad de los estados vacíos en el nivel bajo de energía y la probabilidad de transición entre estos dos niveles, desde un punto de vista de equilibrio térmico es expresado por la distribución de Boltzmann, la relación entre las densidades de electrones. Dicha densidad de electrones se puede observar de manera representativa en la figura II.4.1 donde se tienen los niveles de energía E_1 y E_2 , así como la densidad de electrones n_1 y n_2 .

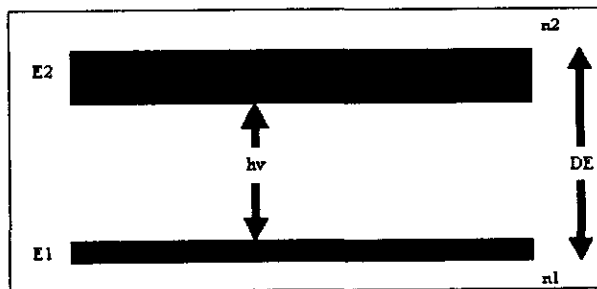


Figura II.4.1. Funcionamiento del LED.

La relación entre las diferentes densidades de electrones está definida por la ecuación II.4.1.

$$\frac{n_2}{n_1} = e^{\frac{-DE}{kT}} \quad (\text{II.4.1})$$

Donde k es la constante de Boltzmann, $k = 1.38 \times 10^{-23}$ J/K, T es la temperatura absoluta (°K), y DE es la diferencia de energía. Si la densidad de electrones en el nivel E_2

aumenta por alguna razón (con respecto al nivel E_1), visto desde el punto de vista de equilibrio térmico, el sistema tratará de emitir luz y de esta forma liberar energía y regresar el equilibrio térmico. Esto se reconoce como la recombinación radiactiva.

Liquid Cristal Displays (LCD's)

El cristal líquido es un material por lo general orgánico, que fluirá como un líquido pero cuya estructura molecular tiene algunas propiedades que se asocian normalmente con los sólidos. En la figura II.4.2 se representa el funcionamiento del cristal líquido, en ella se muestran las regiones clara y opaca. En las unidades de dispersión luminosa, el mayor interés se encuentra en el cristal líquido nemático¹, sus moléculas tienen una apariencia alargada. En los extremos del LCD se tendrá una capa conductora de óxido de indio, la cual es transparente y en condiciones normales la luz atravesará sin problemas esta capa. Así la estructura del líquido aparecerá clara. Si se aplica un voltaje (entre 6-20V) a través de las superficies conductoras de óxido de indio, el arreglo molecular se distorsionará, de tal forma que se crearán regiones con mayores índices de refracción, lo cual genera un fenómeno llamado dispersión dinámica, en donde se tiene como resultado que la luz dispersada tiene la apariencia de un vidrio opaco. Por lo tanto de esto se tiene que un LCD no genera su propia luz sino que depende de una fuente interna o externa.

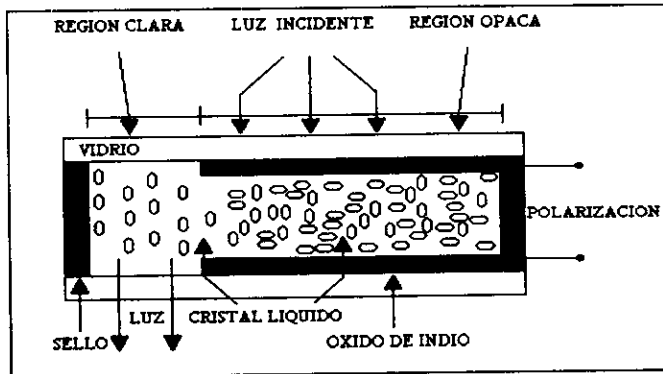


Fig. II.4.2. Funcionamiento del cristal líquido.

¹ Es el estado líquido intermedio entre el estado cristalizado y el estado amorfo (sin estructura cristalina), en el cual las moléculas están orientadas en una dirección dada.

Cuando el LCD va a ser colocado en un sitio donde la fuente de luz externa no es suficiente para ver su despliegue, existe una posibilidad de iluminación llamada *back-light*, ésta consiste en un pequeño foco que ilumina el cristal líquido que en ese momento dispersa luz auxiliado por una fibra óptica.

En la tabla II.4.1 se mencionan las características principales de los *displays* anteriormente descritos (LED's y LCD's).

Propiedades	LCD	LED
Voltaje de operación	2 - 5V	2 - 5V
Consumo de corriente	1-10 mA/cm ²	10 mA/cm ²
Contraste	10-20	50
Tiempo de Respuesta	30-150 ms	40 ms
Luminancia (fL)	Mala	50
Eficiencia Luminosa	-	0.1 - 1.5 lm/W
Temperatura de Operación	-25 a 95°C	-40 a 8-0°C
Horas de vida útil	50,000	100,000
Tipo de tecnología	CMOS	Bipolar

Tabla II.4.1. Características de algunos visualizadores.

Como el sistema que desarrollamos requiere una comunicación con la PC, mencionaremos algunos estándares de comunicación.

II.5. Estándares de comunicación

Las *interfaces* de nivel físico se utilizan para conectar dispositivos de usuarios al circuito de comunicaciones. Para llevar a cabo esta importante función, en la mayoría de las especificaciones relativas a *interfaces* del nivel físico, se describen cuatro atributos de la *interface*. Los atributos eléctricos son los que determinan los niveles de tensión (o corriente) y la temporización de los cambios eléctricos que representan los unos y ceros. Muchos de los

protocolos de nivel físico clasifican estas funciones en: control, sincronismo, datos y tierra. Los atributos mecánicos describen los conectores y los hilos de la *interface*. Por lo general, todas las líneas de datos, de señalización y de control están incluidas en un mismo cable, y se conectan a enchufes situados en ambos extremos del cable.

RS-232C

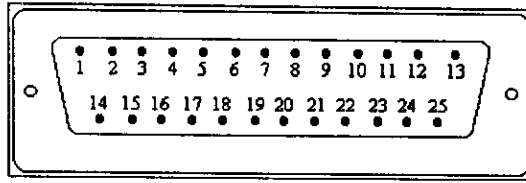
Los Equipos Terminales de Datos (ETD) y los Equipos de Terminación del Circuito de Datos (ETCD) suelen conectarse mediante la *interface* estándar RS-232C². Los ETD son por lo general dispositivos de usuario final. Ejemplo de ellos son los terminales y los ordenadores. Los ETCD proporcionan al ETD una conexión con el circuito de comunicaciones. En las especificaciones de la *interface* RS-232C (V.24/V.28) se describen cuatro funciones del mismo:

- Definición de las señales de control que atraviesan la *interface*.
- Movimiento de los datos de usuario a través de la *interface*.
- Transmisión de las señales de tiempos necesarias para sincronizar el flujo de datos.
- Conformación de las características eléctricas concretas de la *interface*.

RS-232C transmite los datos que los atraviesan mediante cambios en los niveles de tensión. Un 0 binario se representa como un nivel de tensión comprendido entre +3V y +12V, mientras que un 1 binario se expresa como un nivel comprendido entre -3V y -12V. La longitud del cable RS-232C depende de las características eléctricas del mismo, aunque algunos fabricantes prohíben longitudes superiores a unos 16 metros. La norma internacional V.28 establece un *interface* eléctrico similar al RS-232C.

En la fig. II.5.1 se ilustra el circuito RS-232C, que cuenta con 25 líneas. No se utilizan la totalidad de los 25 canales. Para los ETD y ETCD suelen bastar de cuatro a ocho canales.

² La C se refiere a la cuarta versión, aprobada en 1981. El organismo de normalización CCITT ha editado estándares similares, conocidos como V.24/V.28



Línea	Circuito	Fuente	Descripción
1	AA	-	Tierra de protección
2	BA	ETD	Datos transmitidos
3	BB	ETCD	Datos recibidos
4	CA	ETD	Solicitud de transmisión
5	CB	ETCD	Permiso para transmitir
6	CC	ETCD	Equipo de datos preparado
7	AB	-	Tierra de Señal
8	CF	ETCD	Detector de señal de línea recibida
9	-	-	Reservado para pruebas del equipo de datos
10	-	-	Reservado para pruebas del equipo
11	-	-	No asignado
12	SCF	ETCD	Detector de señal de línea secundaria recibida
13	SCB	ETCD	Permiso para transmitir secundario
14	SBA	ETD	Datos secundarios transmitidos
15	DB	ETCD	Sincronismo del elemento de señal en transmisión
16	SBB	ETCD	Datos secundarios recibidos
17	DD	ETCD	Sincronismo del elemento de señal en recepción
18	-	-	No asignado
19	SCA	ETD	Solicitud de transmisión secundaria
20	CD	ETD	Terminal de datos preparado
21	CG	ETCD	Detector de calidad de señal
22	CE	ETD	Timbre indicador
23	CH	ETD	Selector de velocidad de la señal de datos
24	DA	ETD	Sincronismo del elemento de señal en transmisión
25	-	-	No asignado

Fig. II.5.1. Descripción del circuito RS-232C.

RS-422A

Un problema importante con el RS-232C es que únicamente puede transmitir los datos sin pérdida alguna a una distancia máxima de 16.4 m (50 ft) con su valor máximo de 20,000 bauds. Si se usan líneas más largas, el valor de la transmisión se vera drásticamente atenuado.

Una norma más nueva, RS-422A, especifica que cada señal se enviará diferencialmente sobre dos cables adyacentes en un cable de listón ó en un par trenzado de cables como se muestra en fig.II.5.2. Las señales diferenciales son producidas por los conductores diferenciales de línea tal como el MC3487 y convertidos a niveles TTL por receptores diferenciales de línea tal como el MC3486. Los rangos de datos para esta norma son 10 Mbd para una distancia de 16.4 m (50 ft) ó 100,000 Bd para una distancia de 1220 m (4000 ft). Una ventaja de las señales diferenciales es que en modo común cualquier ruido eléctrico inducido en las dos de líneas será rechazado por la línea diferencial del receptor. Para RS-422A un nivel lógico alto o la marca es indicada por la línea de señal B siendo más positivo que la línea de señal A. Un nivel lógico bajo o el espacio es indicado por la línea de señal A siendo más positivo que la línea de señal B. El voltaje diferencial entre las dos de líneas deber ser mayor que 0.4 V, pero no mayor que 12 V. En modo común el voltaje sobre las líneas de señal deben estar en un rango de - 7 V a +7V.

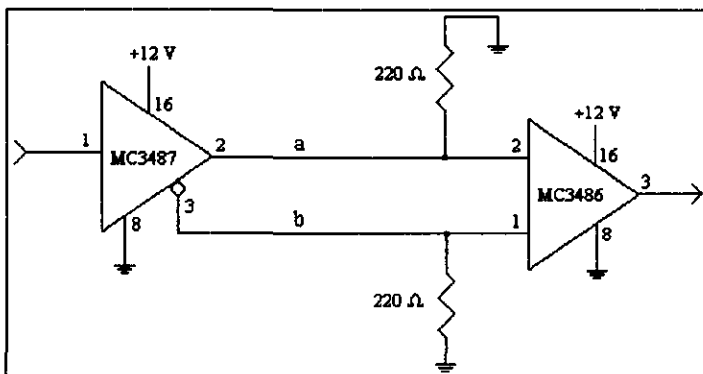


Figura II.5.2. Señales diferenciales para el RS-422A

El inconveniente de esta norma es que en aplicaciones de *bus*, se deben conectar varios emisores sobre el mismo par de conductores. En estos casos, sólo uno debe emitir información, permaneciendo todos los demás en estado de alta impedancia, ya que, de otra forma, pueden aparecer altas intensidades de voltaje entre ellos. Esta característica no está exigida por la norma, que sólo impone que el emisor adopte el estado de alta impedancia cuando se le retira la alimentación.

RS-485

Para superar los inconvenientes de RS-422A se definió un nuevo estándar, el RS-485, que permite 32 emisores y 32 receptores sobre la misma línea, satisfaciendo, en todo lo demás, el estándar RS-422A.

En el RS-485, el emisor tiene protección para el caso eventual de un intento de transmisión simultánea. Además de salida en tres estados y limitación de corriente en ambos estados activos, alto y bajo, tienen incorporada una protección térmica que los desconecta sin daños en caso de una elevación anómala de la temperatura de la unión. En la figura II.5.3 se representa un esquema de una red de sistemas configurada basándose en el estándar RS-485.

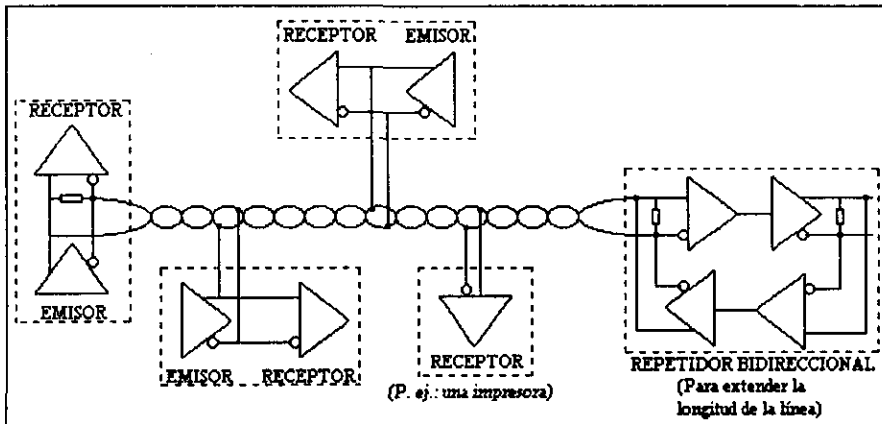


Figura II.5.3. Esquema de una red configurada basándose en el estándar RS-485.

Una vez que hemos descrito brevemente algunos conceptos básicos sobre el sistema de peaje que se desarrollo, pasaremos al análisis del sistema de peaje, en el cual se analizara cada uno de los módulos que integran al sistema de peaje.

CAPÍTULO III

ANÁLISIS DEL SISTEMA DE PEAJE

En este capítulo vamos a hacer una explicación detallada del sistema de peaje a desarrollar, así mismo definiremos todas las funciones que van a ser llevadas a cabo por el mismo. El análisis del sistema se realiza con base en varios submódulos, los cuales se van a estudiar de una manera breve más no por eso menos detallada.

III.1 Descripción y análisis del sistema

En la figura III.1.1 podemos observar un diagrama de bloques general del sistema.

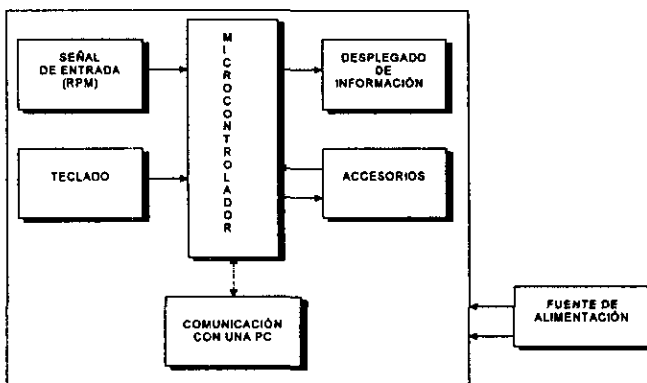


Figura III.1.1. Esquema general del sistema de peaje.

El sistema está integrado alrededor de un microcontrolador; como elementos de entrada tenemos a la señal de la velocidad del vehículo y al teclado; como elementos de salida, el desplegado de la información solicitada: costo del viaje, estadísticas, etc., los accesorios que pueden ser varios, entre ellos: impresora, lectores de tarjetas prepago, de crédito, otros desplegados, etc. Un elemento que se considera de entrada/salida es el de comunicación con una computadora personal, dicho elemento se encarga de actualizar o revisar los datos del sistema de peaje.

Para alimentar el sistema disponemos de la batería de 12 volts del vehículo. De dicha batería obtendremos un voltaje de corriente directa (CD), el cual tendremos que regular para suministrar la tensión requerida por los circuitos que conforman cada submódulo para su correcta operación.

El sistema de peaje deberá contar con un conjunto de funciones mínimas, que establece la norma que ya mencionamos en el primer capítulo.

La principal función de un sistema de peaje de este tipo es la de cuantificar la distancia recorrida y el tiempo en el cual se presta el servicio cuando se conduce a una velocidad menor a la especificada por la norma. Cuando realizamos la cuantificación de estos parámetros éstos se relacionan con un pago monetario el cual genera el sistema de peaje y que será el importe a pagar.

Actualmente el cobro de dicho servicio se hace mediante un banderazo, es decir, un cobro inicial y posteriormente cobros equitativos cada 250 metros o su equivalente en tiempo. Para poder obtener la equivalencia de tiempo, se hace uso de la velocidad de cambio de arrastre. Esta velocidad está definida como 20 km/hr para la Ciudad de México y que se obtiene mediante análisis estadístico y en base al parque vehicular que haya en la ciudad, cabe señalar que un estudio preliminar sobre esta velocidad en 1997 arrojó una velocidad media o de cambio de arrastre en la Ciudad de México de 18 km/hr.¹

¹Fuente: *Propuesta de un ciclo e manejo representativo de la ZMCM para homologar vehículos ligeros*. José Luis Ramírez Mellado. Tesis de Maestría en Ingeniería Mecánica. 1997

Tomando como dato para la velocidad el de 20 km/hr y una distancia de 250 metros podemos obtener el tiempo que tarda en recorrer el vehículo dicha distancia y, como señala la norma, el tiempo en que se deben realizar los incrementos cuando se aplica la función tiempo.

$$v = \frac{d}{t} \quad \text{despejando } t = \frac{d}{v} \quad (\text{III.1.1})$$

Realizando las operaciones con los datos de distancia igual a 250 metros y la de velocidad de 20 km/hr, obtenemos el tiempo que es de 45 segundos. Entonces ahora ya conocemos mediante que parámetros podemos realizar los cobros adecuados y cada cuando cuantificarlos: cada 250 metros cuando la velocidad del vehículo sobrepase la velocidad de cambio de arrastre y cada 45 segundos cuando la velocidad del móvil sea menor a 20 km/hr.

Como pudimos observar la figura III.1.1 el microcontrolador va a organizar y controlar cada uno de los eventos que se van generando y guardar el programa que se va a ejecutar para realizar dichas funciones. También tendrá que controlar las acciones que tomen cualesquiera de los accesorios conectados al sistema de peaje.

En la tabla III.1.1 hacemos una reseña de las funciones que se van a controlar y que deberá contener nuestro programa y que lo señala la norma oficial.

PARÁMETROS	FUNCIÓN
Tarifa de día	Es el Importe inicial del servicio durante el día (Banderazo diurno).
Incremento de día	Es el importe que se acumula al generarse un incremento durante el día.
Tarifa de noche	Importe inicial del servicio durante la noche (Banderazo nocturno).
Incremento de noche	Es el importe que se acumula al generarse un incremento durante la noche.

Tabla. III.1.1 Totalizadores mínimos de un taxímetro (continúa).

Distancia total recorrida por el automóvil	Es el total de kilómetros recorridos por el vehículo.
Distancia total recorrida en servicio	Es el total de kilómetros acumulados durante todos los viajes que haya realizado el vehículo.
Número total de viajes en servicio	Nos indica el número de viajes que ha realizado un vehículo en una jornada de trabajo.
Número total de cambios o saltos del indicador del precio	Este parámetro acumula el total de incrementos que se generaron sin importar si fueron por tiempo o distancia recorrida.
Importe acumulado	El importe acumulado de todos los viajes realizados por el vehículo.

Tabla. III.1.1 Totalizadores mínimos de un taxímetro.

Aún siendo los requisitos mínimos que pide la Dirección General de Normas (DGN), el fabricante de sistemas de peaje puede incluir otro tipo de funciones como lo son:

- Indicador de tiempo.
 - Fecha
 - Hora
- Información acerca del vehículo
 - Número de placas
- Información propia del sistema de peaje
 - Modelo
 - Folio
 - Serie
- Indicadores de cobros extras

Otra función que deberá ejecutar nuestro sistema es el de indicar de forma externa al usuario si está ocupado o no el vehículo. Esto se realiza haciendo uso del llamado “copete” el cual tiene la leyenda “TAXI” y la bandera la cual tiene la leyenda “LIBRE”. Estos dos accesorios externos también son solicitados por la DGN.

Operación normal de un sistema de peaje

La operación de un sistema de peaje se realiza como se ilustra en la figura III.1.2.

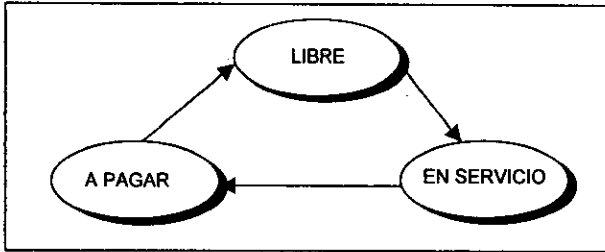


Figura III.1.2 Operación de un sistema de peaje.

La norma nos señala que la operación del sistema de peaje deberá realizarse con un solo botón de operación, es decir, un pulsador será el único que podrá colocarnos en modo de servicio y posteriormente ese mismo botón en el modo de pagar. Este ciclo no se verá interrumpido por ninguna acción externa o interna al sistema de peaje, a menos de que se presione el botón de fin de ciclo de operación.

Para que nosotros podamos realizar la contabilización de distancia y la velocidad a la que se mueve el vehículo, debemos hacer uso de las constante K y W que nos especifica la norma. Como se mencionó en el capítulo I, la constante K nos dirá cuantos pulsos nos debe entregar el coche para cuantificar la distancia de un kilómetro, y la constante W son los pulsos que se van acumulando y que se debe de igualar a la constante K del sistema de peaje. Para poder realizar el cálculo de la constante K para nuestro sistema de peaje debemos usar los siguientes parámetros:

- ◆ Velocidad de cambio de arrastre (VCA).
- ◆ Distancia de prueba (DP).
- ◆ Diámetro de la llanta del vehículo.

Para conocer el valor del número de pulsos que se generan en un kilómetro, disponemos de los siguientes datos:

$$VCA = 20 \frac{km}{hr} = 5.555 \frac{m}{s}$$

$$DP = 250 m$$

Para obtener el perímetro de la llanta del vehículo se hace uso del tamaño del rin. Para realizar dicho cálculo es necesario tener las condiciones de presión y tipo de llanta que recomienda el fabricante (ver tabla III.1.2), tomando o no en cuenta el achatamiento que le ocurre al neumático debido al peso propio del vehículo.

Vehículo	Llanta	Ancho de piso [mm]	Relación de aspecto. %ancho de piso	Diámetro del rin ["]	Presión de aire [psi]	Presión de aire kPa
V. W Sedán	Pasajero	155	80	15	28	193.05
Golf	Pasajero	175	70	14	30	206.84
Jetta	Pasajero	175	70	13	30	206.84
Nissan	Pasajero	155	80	13	30	206.84
Spirit	Pasajero	205	70	14	30	206.84
Shadow	Pasajero	185	70	14	30	206.84
Chevi	Pasajero	185	65	14	28	193.05

Tabla III.1.2 Condiciones de presión y tipo de rin.²

En la tabla anterior podemos ver que tenemos de referencia el tipo de llanta, ancho de piso y relación de aspecto. La primera se refiere al tipo de uso que tiene el vehículo, la segundo es el ancho de la llanta que hace contacto con el piso y, por último la relación de aspecto que es la distancia que hay desde la circunferencia del rin hasta la circunferencia de la llanta, este perímetro se da con respecto al ancho de piso.

La nomenclatura que se utiliza para definir estos parámetros en una llanta, es la siguiente:

P155/80/15

²Fuentes: The tyre and Rim Association, Inc. 1996.
Recomendaciones del fabricante del cada vehículo.

La letra **P** corresponde al tipo de llanta; **155**, corresponde al ancho de piso; **80** a la relación de aspecto y **15** al diámetro del rin.

En la tabla anterior no se ha tomado en cuenta el achatamiento. Este achatamiento se toma de $\frac{1}{2}$ pulgada, es decir que se reduce el radio del neumático en $\frac{1}{2}$ pulgada debido a que sufre el achatamiento solo en la zona de contacto. Esto lo podemos visualizar en la figura III.1.3.

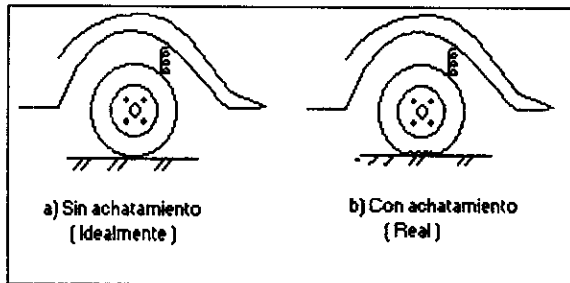


Fig. III.1.3 Efecto del peso del vehículo en los neumáticos.

Tomando en cuenta lo anterior se procede a calcular los perímetros de los neumáticos haciendo uso de la siguiente ecuación:

$$P = 2\pi r \quad (\text{III.1.2})$$

Para hacer uso de la ecuación anterior es necesario conocer el radio de las llantas del vehículo para conocer exactamente la distancia que se recorre en una revolución de la llanta. La medición del diámetro de la llanta se realizó en base a la tabla III.1.2.

Con los diámetros obtenidos y haciendo uso de la ecuación (III.1.2) se obtuvieron los perímetros de las llantas para diferentes vehículos, así como los respectivos valores de pulsos por kilómetro para una velocidad de cambio de arrastre de 20 km/hr. Esto se muestra en la tabla III.1.3 y III.1.4.

Vehículos	Tamaño de rin ["]	Perímetro [m]	Pulsos por km.
VW Sedán	15	1.91	1047
VW Golf	14	1.83	1092
VW Jetta	14	1.75	1142
Chrysler Spirit	14	1.99	1005
Chrysler Shadow	14	1.83	1092
Nissan Tsuru	13	1.75	1142

Tabla III.1.3 Perímetro y total de pulsos sin tomar en cuenta el achatamiento.

Vehículos	Tamaño de rin ["]	Perímetro [m]	Pulsos por km.
VW Sedán	15	1.83	1092
VW Golf	14	1.75	1142
VW Jetta	14	1.67	1197
Chrysler Spirit	14	1.91	1047
Chrysler Shadow	14	1.75	1142
Nissan Tsuru	13	1.67	1197

Tabla III.1.4 Perímetro y total de pulsos tomando en cuenta el achatamiento.

Como podemos observar en las tablas anteriores, tenemos una diferencia del 8% aproximadamente entre los cálculos tomando y no tomando en cuenta el achatamiento. Por esta razón debemos tener cuidado en seleccionar que dato vamos a usar como la constante K de nuestro sistema de peaje.

Habiendo realizado una investigación con los fabricantes de sistemas de peaje en nuestro país, específicamente en la Ciudad de México, nos dimos cuenta que estos fabricantes no toman en cuenta el achatamiento que tienen los neumáticos en todos los vehículos.

Habiendo revisado algunas características de dichos sistemas, procedemos a realizar un análisis de cada uno de los submódulos por los que está compuesto dicho sistema de peaje.

Módulo del Teclado

El teclado lleva a cabo las funciones que puede generar el sistema de peaje, como lo son visualizar totalizadores, colocarlo en modo de servicio, visualizar fecha, reloj, cambio de tarifa, etc. Un esquema que podemos tomar de base es como el que muestra en la figura III.2.2.

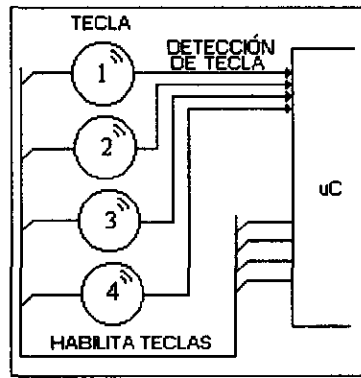


Figura III.2.2. Configuración del teclado.

Cada una de las teclas o pulsadores deberán cumplir con las normas establecidas. Como podemos observar el esquema básico es un tanto grande en cuanto al total de líneas que se van a usar para poder realizar el barrido del teclado y detectar que tecla se presionó. Esto lo estudiaremos más en detalle en el diseño de la arquitectura y si se tienen que realizar cambios o modificaciones las haremos enfatizando el por que de los cambios.

La norma nos señala que con una sola tecla debemos colocar el sistema de peaje en funcionamiento y con la misma terminarlo; podemos escoger el orden que deseemos para asignar las funciones de cada tecla. En nuestro caso asignaremos dos teclas para los diferentes tipos de cobro que hay, el de día y el de noche. Las otras teclas serán para otras funciones que va a contener el sistema de peaje, visualización de totalizadores, impresiones, fecha, hora, etc.

Módulo de desplgado de información

En la figura III.2.3 se muestra un esquema propuesto para la visualización de los parámetros que tiene el sistema de peaje.

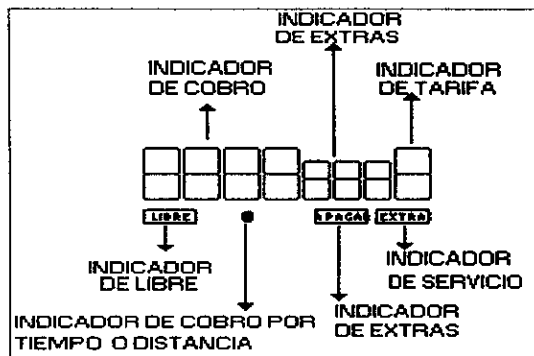


Figura III.2.3 Desplgado de información.

Como podemos observar en la figura anterior es un sistema de desplgado básico y que satisface las necesidades del sistema. Cabe señalar que es solamente una propuesta que posteriormente se ajustará a costos y apariencia estética. Como primera instancia propondríamos un desplgado de cristal líquido, pero esto no sería lo suficientemente apropiado debido a que es demasiado caro para las especificaciones que nos indica la DGN y sobre todo perderíamos visibilidad de los indicadores a pequeñas distancias y viendo el desplgado en cierto ángulo de vista. La DGN nos dice que el desplgado de la información deberá ser visible a por lo menos tres metros de distancia tanto en el día como en la noche, y que los dígitos que se usen sean de por lo menos 12.7 mm de altura.

Debido a estas normativas se opta por usar desplgados de 7 segmentos para cada uno de los dígitos que se van a usar y *leds* del tipo "light bar" y bicolores.

Módulo de accesorios

El tener un sistema que pueda interactuar con otros dispositivos nos da una gran ventaja, por que de esta forma podemos no solamente tener acciones en las funciones del sistema de peaje sino que también podemos agregar servicios tanto para operadores como a usuarios.

La conexión de estos dispositivos se hará mediante un *bus*, el cual se va a encargar de transportar los datos al dispositivo seleccionado. Esto lo podemos visualizar en la figura III.2.4.

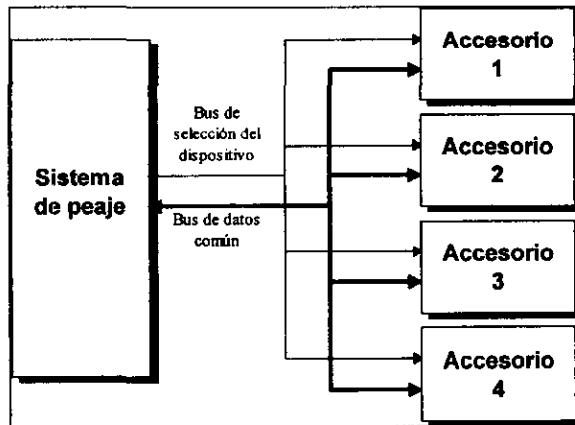


Figura III.2.4. Conexión de accesorios al sistema de peaje.

Dentro de los accesorios que se pueden tener nombramos impresoras, lectores de tarjetas, módulos de localización vía satélite, etc.

De los accesorios nombrados anteriormente solamente hablaremos de una impresora de recibos de pago. La conexión de esta impresora al sistema de peaje se debe realizar como se muestra en la figura III.2.5. La interconexión entre estos dos dispositivos debe ser lo más adecuada posible y sobre todo transparente, es decir, solamente llegaremos a conectar el dispositivo y el sistema de peaje detectará de que dispositivo se trata.

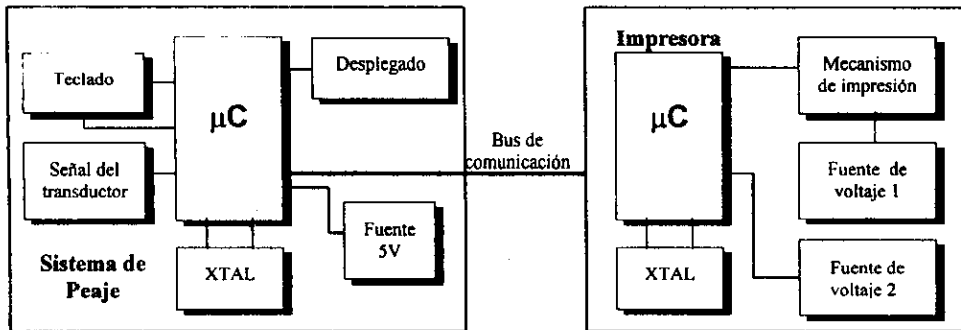


Figura III.2.5. Conexión de la impresora.

La impresora tiene ciertas características que se deben tomar en cuenta en cuanto a que no solamente se debe controlar desde el propio sistema de peaje, sino que también requiere de cierta electrónica de control para poder hacerla operar de una manera eficiente sin dañar la impresora o el sistema de peaje.

Dado que no solamente va estar conectada la impresora, debemos prevenir posibles conflictos durante la interconexión de los otros accesorios. Un esquema general de la conexión se hace mediante un bus común y líneas de habilitación. De esta forma podemos escoger solamente un dispositivo a la vez sin que se creen conflictos de comunicación.

Debemos tener en cuenta que nuestro sistema tendrá que tener la capacidad de cada uno de los accesorios solicitados por la empresa que pidió dicho proyecto para que se realice un protocolo sencillo y/o en su defecto utilizar algunos protocolos de comunicación que ya se tienen incluidos en algunos microcontroladores, y realizar los controles de cada accesorio.

Módulo de comunicación serial con una computadora personal

Actualmente los sistemas de peaje que se encuentran en el mercado no cuentan con una comunicación serial con una computadora, por lo que la actualización de tarifas, banderazo e incrementos, no se realizan de una manera rápida y eficiente. Para poder realizar este cambio

de tarifas debe ser abierto el sistema de peaje y romper los sellos de inviolabilidad que pide la SECOFI que se coloquen.

El tener la posibilidad de cambiar dichos parámetros y otros que son propios del sistema de peaje en un tiempo muy corto y sin tener la necesidad de realizar alguna operación directa sobre el sistema de peaje, propusimos el implementar una comunicación serial basándonos en un estándar de comunicación como lo es RS-232.

En la figura III.2.6 se muestra una configuración para esta tipo de conexiones. Esta conexión es del tipo *null-modem*, es decir que transmitimos en banda base sin hacer algún tipo de modulación. Como podemos observar el sistema de peaje deberá ser montado en una base que contenga una fuente de alimentación y la conexión para la computadora personal. Después de haber sido conectado podemos pedir conexión con el sistema ya sea para lectura o escritura de los parámetros.

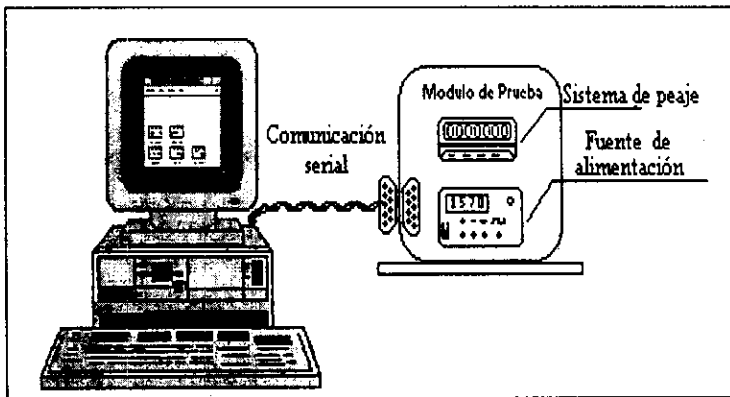


Figura III.2.6. Configuración de comunicación con una PC.

El realizar este tipo de comunicación no debe incrementar el número de circuitos integrados que se van a colocar en el sistema de peaje para evitar un incremento en los costos de producción.

Un punto importante cuando se realiza una transmisión de datos es la seguridad de los

mismos durante la transmisión. Los datos transmitidos deben ser fidedignos para que no tengamos algún error en la interpretación de los mismos. Para realizar una verificación de datos se usan algoritmos de detección de errores; en sistemas más complejos se usan algoritmos para la corrección de los errores detectados.

En nuestro caso utilizaremos un código de detección de error en la transmisión de datos. Dicho código es conocido como *checksum*. Este algoritmo lo que hace es ir sumando cada dato transmitido y enviar al final de la trama la suma, esto último del lado del transmisor. En el lado del receptor cada dato recibido se va sumando y cuando llega el último dato de la trama se recibe la suma de todos los datos, la cual debe coincidir con la que se suma en el receptor.

Módulo del circuito de seguridad

Uno de los esquemas con los que trabajamos es el poder colocar un sistema de seguridad para prevenir posibles robos de los sistemas de peaje, situación que se ha dado en los últimos años y que ha generado centros de revistas, reparación y venta de una forma "ilegal".

Para evitar esto se pensó en un sistema de seguridad en el que se pueda implementar un código único para cada sistema de peaje y una llave de seguridad para poder iniciar la operación del mismo. La primera llave tiene el fin de poder establecer una comunicación con una computadora para poder alterar los datos del sistema de peaje.

Después de haber analizado el funcionamiento que debe tener cada módulo que conforman a nuestro sistema procederemos a realizar el diseño de toda la arquitectura tanto para el sistema de peaje como para los accesorios, aunque en el presente diseño solamente hablaremos de un único accesorio: la impresora.

CAPÍTULO IV

DISEÑO DEL HARDWARE DEL SISTEMA DE PEAJE

En este capítulo comenzamos por mostrar de forma específica como será resuelto el diseño de *hardware* del sistema y se definirán cuales serán los recursos empleados en la solución. Además se mostrarán algunos de los aspectos de la implementación del sistema.

IV.1. Especificaciones de diseño del sistema

En la figura IV.1.1 se presenta en un diagrama a bloques la estructura del sistema a desarrollar.

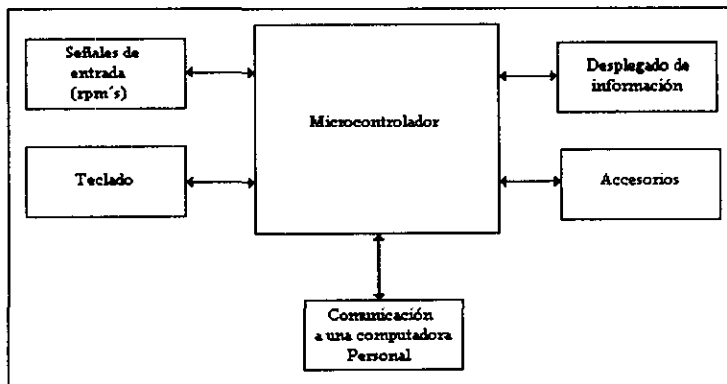


Fig. IV.1.1. Diagrama a bloques del sistema.

Como se puede ver en el diagrama anterior, el elemento principal del sistema es el microcontrolador, se decidió utilizar un microcontrolador COP8SAC de National Semiconductor. A diferencia de otros microcontroladores, de marcas conocidas como Motorola, Intel y Philips, este microcontrolador es más barato, y es lo suficientemente poderoso para la aplicación que se va a desarrollar. Este último se consigue con mayor facilidad y es bastante sencillo de programar. Además de que existen las herramientas para facilitar la programación de los sistemas basados en este microcontrolador (emuladores de tiempo real, emuladores a 20 KHz, simuladores, etc.), así como suficiente bibliografía y *software* para su uso. Este es el elemento central del sistema y es el que lleva a cabo la mayoría de las funciones del sistema que desarrollaremos.

Además el sistema requiere detectar señales de rpm's, atender a un teclado, procesar la información, desplegarla, establecer comunicación con una PC y controlar algún accesorio externo, por ejemplo una impresora. Todos y cada uno de estos bloques del sistema se analizarán más adelante por separado.

IV.1.1 Arquitectura COP8SAC

El microcontrolador COP8SAC, es un miembro de la familia COP de National, cuenta con las características principales de esta familia, además de sus características propias.

En la figura IV.1.2. se muestra un diagrama a bloques del microcontrolador COP8SAC. Sus características principales son:

- Memoria RAM no volátil de 128 bytes para datos.
- Dos relojes-contadores de 16 bits y 12 bits, además de un WATCHDOG.
- Cuatro puertos bidireccionales de 8 bits de I/O y un puerto de salida de 8 bits.
- *Interface* Serial MICROWIRE/PLUS (SPI Compatibles).
- Multi-entradas Wake-up con opción de interrupciones.

- Cuenta hasta con 12 salidas de alta corriente.
- Baja emisión de radiación.
- Opciones programables de entrada de reloj.

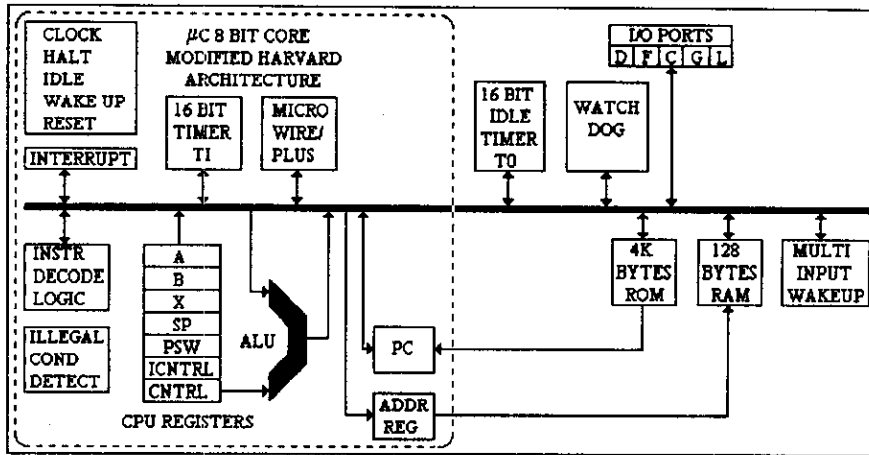


Fig. IV.1.2. Diagrama a bloque del microcontrolador COP8SAC.

La presentación de este dispositivo es en empaque tipo DIP de 40 terminales. Este microcontrolador divide la memoria en espacio de datos y espacio de programa, esta separación lógica de memoria permite el acceso a la memoria de datos por direccionamiento de 8 bits, el cual nos facilita el almacenamiento y la manipulación en un CPU de 8 bits. La memoria de programa EPROM solamente se puede leer y no escribir, esta puede ser de hasta 4kbytes.

Para poner en operación al microcontrolador se requieren 5V, y una señal de reloj la cual puede ser por un oscilador R/C interno, externo ó por un oscilador de cristal de cuarzo.

Cada terminal bidireccional puede ser configurado como línea de salida y como entrada con alta impedancia. Un ejemplo típico del uso de estas líneas bidireccionales es cuando se tiene las líneas de un teclado matricial. Las salidas de alta corriente están disponibles para diferentes dispositivos, como pueden ser *leds*, motores y bocinas. Esta última característica

permite garantizar un claro desarrollo, con un menor número de componentes externas y a un bajo costo.

Como ya se mencionó anteriormente, este microcontrolador cuenta con cuatro puertos bidireccionales de 8 bits de entrada y salida (C, G, L, y F), donde cada bit individual puede ser configurado independientemente como entrada (entradas tipo Schmitt trigger en los puertos L y G), o salidas tres estados mediante un control por *software*. Cada puerto de I/O tiene asociado tres registros en el mapa de memoria de 8 bits, el registro de configuración, el registro de datos de entrada y el registro de dato de salida. La dirección en el mapa de memoria es también reservada para las líneas de entrada de cada puerto de I/O. La tabla IV.1.1 muestra la configuración de los puertos de I/O, el registro de dato y configuración permite que cada bit de cada puerto sea configurado individualmente mediante un control por *software*.

REGISTRO DE CONFIGURACION	REGISTRO DE DATO	PUERTO CONFIGURADO
0	0	Alta impedancia de entrada Salida de tres estados
0	1	Entrada con tenue Pull-up
1	0	Cero a la salida Push-Pull
1	1	Uno a la salida Push-Pull

Tabla IV.1.1. Registros de configuración.

El puerto G es de 8 bits. Las terminales G0, G2-G5 son líneas bidireccionales de I/O. La terminal G6 es siempre una entrada de alta impedancia de propósito general. Todas las terminales tienen entradas tipo *Schmitt Triggers*. La terminal G1 está dedicada para la salida WDOUT del WATCHDOG, esta función especial tiene que ser activada por el registro ECON. Cuando no se habilita el WATCHDOG esta terminal es de propósito general y se utiliza como I/O. La terminal G7 es puesta en uno o en cero dependiendo del tipo de oscilador que se seleccione. Si se selecciona el oscilador con cristal, la terminal G7 sirve de salida para el reloj CKO. Con la opción de oscilador R/C interno o externo, la terminal G7 es una entrada de alta impedancia de propósito general. Hay tres registros asociados al puerto G que son el registro de datos de entrada, el registro de datos de salida y el registro de configuración. Haciendo uso

de estos registros, cada una de las 5 terminales de I/O (G0, G2-G5), las cuales pueden ser configuradas individualmente mediante un control por *software*.

El puerto C es de 8 bits de I/O. En el microcontrolador de 40 terminales el puerto C no está completo, sólo tiene 4 terminales. En este dispositivo, los registros asociados al puerto C que son el registro de datos y el registro de configuración, no pueden ser utilizados.

El puerto F es de 8 bits de I/O, mediante su registro de configuración se pueden declarar las terminales como entradas o como salidas.

El puerto D es de 8 bits con salidas en alto, cuando el RESET se va a un bajo, este puerto es exclusivamente de salida y es de alta corriente.

Registros de la CPU

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

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

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

Memoria de Programa

La memoria de programa consiste en 1024, 2048 o 4096 bytes de EPROM. En la tabla IV.1.2 se muestran algunos tamaños de la memoria de programa de algunos dispositivos de la misma familia de este microcontrolador. Estos bytes pueden tener instrucciones de programa o datos de algunas constantes. La memoria de programa es direccionada por medio del contador de programa (PC).

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

Tabla IV.1.2. Muestra diferentes tamaños de memoria.

Todas las interrupciones en este dispositivo tienen un vector en el mapa de memoria localizado en la dirección 0FF Hex.

Memoria de Datos

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

Circuito oscilador

Existen cuatro opciones disponibles de osciladores: Cristal oscilador con o sin resistencia en el circuito integrado, arreglo R/C de resistencia y capacitor externo, y un oscilador externo. Estas distintas opciones pueden ser habilitadas mediante el registro ECON, como se muestra claramente en la tabla IV.1.3.

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

Tabla IV.1.3. Opciones del Oscilador.

Cristal Oscilador. La opción del cristal oscilador, con o sin resistencia, puede ser habilitada si en el bit # 4 del registro ECON se coloca un 1. La terminal del microcontrolador que se utiliza como entrada del reloj es la CKI, mientras que la terminal G7/CKO es la salida que se utiliza como alimentación del cristal. La resistencia en el circuito integrado puede ser habilitada si en el bit # 3 del registro ECON se coloca un 1, con la opción del cristal oscilador habilitado. El valor de la resistencia R2 está en un rango de entre 0.5 M Ω y 2 M Ω (típicamente de 1 M Ω). La tabla IV.1.4. muestra los distintos valores de los componentes para varios valores conocidos del cristal. La resistencia R2 únicamente se utiliza cuando la resistencia del circuito integrado es deshabilitada por el usuario. La figura IV.1.3 se muestra el circuito de configuración para este tipo de oscilador.

R1 (K Ω)	R2 (M Ω)	C1 (μ F)	C2 (μ F)	CKI FREQ. (MHZ)
0	1	100	30	15
0	1	45	32	10
0	1	32	30 - 36	4
5.6	1	30	100 - 156	0.455

Tabla IV.1.4. Configuración del Cristal Oscilador.

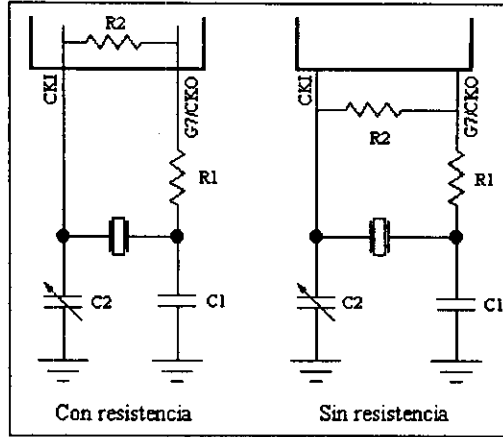


Figura IV.1.3. Oscilador de Cristal.

Oscilador Externo. El modo del oscilador externo puede ser habilitado si el bit 3 y el bit 4 tienen un cero en el registro ECON. Por la terminal CKI se puede introducir una señal de reloj, con una frecuencia y un nivel determinado. Cuando se utiliza la opción de oscilador externo, la terminal G7/CKO está disponible para un propósito general. En la figura IV.1.4. se muestra este oscilador externo.

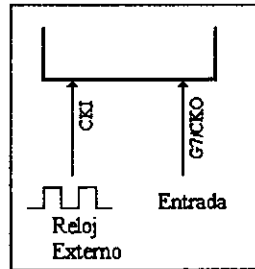


Figura IV.1.4. Oscilador Externo.

Oscilador R/C. El modo del oscilador R/C puede ser habilitado si el bit 3 tiene un 1 y el bit 4 tiene un 0 del registro ECON. En el modo de oscilador R/C, la terminal CKI está desconectada, mientras que la terminal G7/CKO es una entrada de propósito general o para el control del HALT. La máxima frecuencia del oscilador R/C es de 5 MHz \pm 35 % para una alimentación que puede estar entre 4.5 V y 5.5 V, con un rango de temperatura entre - 40 °C hasta + 85 °C. Para frecuencias mayores la terminal CKI debe desconectarse. Para frecuencias menores debe conectarse un capacitor externo entre la terminal CKI y tierra. La tabla IV.1.5 muestra las diferentes frecuencias para este tipo de oscilador, así como el valor del capacitor externo. La figura IV.1.5 muestra la configuración del oscilador R/C.

CAPACITOR EXTERNO (pF)	FREQ. DEL OSC. R/C (MHZ)	CYCLE DE INSTRUCCIÓN (μS)
0	5	2.0
9	4	2.5
52	2	5.0
150	1	10
TBD	32 kHz	312.5

Tabla IV.1.5. Configuración del oscilador R/C.

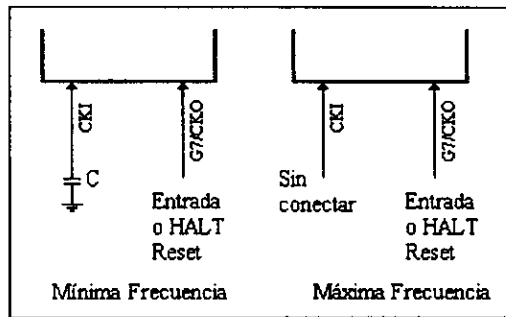


Figura IV.1.5. Oscilador R/C.

Registros de Control

Este microcontrolador tiene registros de control que son muy importantes para las diversas actividades que se quieran desarrollar. Estos registros son: el CNTRL, el PSW y el ICNTRL, los cuales describiremos a continuación.

Con el registro CNTRL, con dirección 00EE, se puede controlar al temporizador T1 y al MICROWIRE/PLUS. En la figura IV.1.6 se muestra la disposición de las diferentes opciones, las cuales son:



Figura IV.1.6. Registro de control CNTRL.

- SL1 y SL0 Selecciona al reloj del MICROWIRE/PLUS, dividido por (00 = 2, 01 = 4, 1x=8).
- IEDG Interrupción externa por flanco de subida o flanco de bajada. 0 = Flanco de subida, 1 = Flanco de bajada.
- MSEL Selecciona G5 y G4 como las señales SK y S0 respectivamente, para el MICROWIRE/PLUS.
- TIC0 Bit de control del Star/Stop en los modos 1 y 2 del temporizador. En el modo 3 del temporizador, T1 es la bandera para una interrupción por sobreflujo.
- TIC1 Bit de control para el temporizador T1.
- TIC2 Bit de control para el temporizador T1.
- TIC3 Bit de control para el temporizador T1.

Con el registro **PSW**, con dirección 00EF, se puede controlar las interrupciones y el tipo de captura de pulsos. En la figura IV.1.7 se muestra el registro de control, el cual contiene las siguientes opciones:



Figura IV.1.7. Registro de control PSW.

GIE	Habilita las interrupciones globales.
EXEN	Habilita las interrupciones externas.
BUSY	Es la bandera de ocupado del MICROWIRE/PLUS.
EXPND	Interrupción externa pendiente.
T1ENA	Es la interrupción del temporizador T1 habilitada por un sobreflujo del temporizador o por la captura de un flanco por la entrada T1A.
T1PND	Es la bandera de interrupción pendiente (autorecarga RA en el modo 1, T1 sobreflujo en el modo 2, T1A flanco de captura en el modo 3).
C	Bandera del Carry.
HC	Bandera del Half Carry.

Con el registro **ICNTRL**, con dirección 00E8, se puede controlar al temporizador 0, además de habilitar y controlar las interrupciones del MICROWIRE/PLUS. En la figura IV.1.78 se muestra el registro de control, el cual contiene las siguientes opciones:



Figura IV.1.8. Registro de control ICNTRL.

T1ENB	La interrupción del temporizador T1 es habilitada por la captura de un flanco por la entrada de T1B.
--------------	--

T1PNDB	Es la bandera de interrupción pendiente por T1B para la captura por flanco.
μ WEN	Habilita la interrupción del MICROWIRE/PLUS.
μ WPND	Interrupción pendiente por el MICROWIRE/PLUS.
TOEN	Habilita la interrupción del temporizador T0.
T0PND	Interrupción pendiente por el temporizador T0.
LPEN	Habilita la interrupción del puerto L.

Bit 7 puede ser usado como bandera para propósito general.

Temporizadores

Este dispositivo contiene un juego muy versátil de temporizadores. El temporizador T1 y los registros asociados de captura y con autorecarga contienen datos aleatorios.

Temporizador T0 (Temporizador IDLE)

El dispositivo soporta aplicaciones que operan en tiempo real y un consumo bajo de energía, con el modo de espera (IDLE mode). El temporizador T0 corre continuamente al mismo tiempo de cada ciclo de reloj de instrucción. El usuario no puede leer o escribir en este temporizador. Este temporizador cuenta hacia atrás.

El temporizador T0 puede realizar las siguientes funciones:

- Permite salir del modo de espera al microcontrolador.
- Puede manejar el WATCHDOG.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

El temporizador T0 puede generar una interrupción cuando llega a cero. Cuando esto ocurre se ve registrado en el bit TOPND, que es la bandera que está pendiente del temporizador, esta interrupción ocurre cada 4.096 ms a la máxima frecuencia ($t_c = 1 \mu s$). El bit T0EN permite habilitar ó deshabilitar la bandera TOPND del temporizador T0. Este bit está deshabilitado mientras ocurre una señal de restablecimiento.

Temporizador T1

Una de las principales funciones del microcontrolador es la de proveer coordinación y contar con la capacidad de controlar tareas en tiempo real. La familia de COP888 ofrece algunos temporizadores y contadores de 16 bits, y dos de ellos cuentan con autorecarga y registros de captura R1A y R1B. El temporizador T1 tiene dos terminales asociadas con T1A y T1B. La terminal T1A puede usarse como entrada/salida (I/O), y el T1B es únicamente de entrada.

Este temporizador tiene tres opciones de operación: Proceso independiente de modulador por ancho de pulso (PWM), Contador de eventos externos y captura de datos. Los bits de control T1C3, T1C2 y T1C1 permiten seleccionar las diferentes opciones de operación.

Opción 1: Proceso independiente de PWM

En esta opción el temporizador entra en un proceso independiente que genera una señal de PWM, una vez que el temporizador es seleccionado en esta opción ya no se requiere de ninguna acción adicional desde la CPU. Esta opción es posible ya que este temporizador tiene dos registros independientes con recarga de 16 bits. Uno de estos registros contiene el tiempo en "ON", mientras que el otro contiene el tiempo en "OFF".

Este temporizador puede generar una salida modulada por ancho de pulso y con un ciclo de servicio controlado por los valores colocados en el registro de recarga. La salida de PWM puede ser usada para controlar motores, triacs, la intensidad de *displays*, y para proporcionar entradas para adquirir datos y para generadores de onda senoidal. En esta opción

el temporizador T1 cuenta hacia atrás al mismo tiempo de cada instrucción, en cada sobreflujo el temporizador es alternativamente recargado con el contenido de los registros R1A y R1B. El primer sobreflujo del temporizador causa la carga del temporizador por medio del registro R1A. Los sobreflujos consecutivos que el temporizador sea recargado alternadamente desde los registros R1A y R1B, comenzando con el registro R1B.

Los bits de control del temporizador T1 son: T1C3, T1C2 y T1C1.

Opción 2: Contador externo de eventos

Esta opción es totalmente similar a la anterior. La principal diferencia está en el temporizador T1, el cual puede capturar una señal por medio de la terminal R1A. Los bits de control de este temporizador son T1C3, T1C2 y T1C1, los cuales permiten que el temporizador pueda ser configurado para capturar por flanco positivo o por flanco negativo desde la terminal R1A. Los sobreflujos de este temporizador son registrados por la bandera T1PNDA. El bit TIENA controla la bandera que está pendiente de que ocurra una interrupción por sobreflujo en el temporizador.

En esta opción la terminal T1B puede ser usado como entrada por flanco positivo sensible a una interrupción externa, si la bandera de control TIENB está habilitada. Esta interrupción es registrada por la bandera T1PNDB.

Opción 3: Captura de datos

Este dispositivo puede hacer mediciones precisas de frecuencias externas o controlar eventos externos, poniendo el temporizador T1 en la opción de captura de datos. En esta opción los registros de recarga sirven independientemente como registros de captura, capturando el contenido del temporizador cuando ocurre un evento externo (transición en la terminal de entrada del temporizador). El registro de captura puede ser leído aun mientras se mantiene la cuenta, esta característica permite al usuario medir el tiempo transcurrido y el tiempo entre cada evento. Para salvar el valor del temporizador cuando ocurre un evento externo, el tiempo del evento externo es grabado. La mayoría de los microcontroladores tienen

un temporizador latente por que ellos no pueden determinar el valor del temporizador cuando ocurre un evento externo. El registro de captura elimina al temporizador latente, de esta manera permite en los programas de aplicación a recuperar el valor del temporizador puesto en el registro de captura.

En esta opción el temporizador T1 corre constantemente al mismo tiempo de cada instrucción. Los dos registros R1A y R1B operan como registros de captura. Cada registro opera con una terminal. El registro R1A opera en conjunto con la terminal T1A y el registro R1B opera en conjunto con la terminal T1B. El valor del temporizador será copiado dentro del registro cuando un disparo como evento ocurra en sus terminales correspondientes. Los bits de control T1C3, T1C2 y T1C1 permiten configurar el tipo de disparo ya sea por flanco positivo o negativo. Esta condición de disparo para cada terminal de entrada puede ser configurada independientemente.

La condición de disparo también puede ser programada para generar interrupciones. El caso de la condición específica de disparo en las terminales T1A y T1B serán específicamente registrados en las banderas T1PND A y T1PND B. El control de la bandera T1ENA permite que la interrupción en T1A sea habilitada o deshabilitada. La configuración de la bandera T1ENA habilita las interrupciones para ser generadas cuando la condición de disparo seleccionada ocurra en la terminal T1A. Similarmente la bandera T1ENB controla las interrupciones desde la terminal T1B.

Los sobreflujos desde el temporizador pueden también ser programados para generar interrupciones. Los sobreflujos son registrados dentro de la bandera pendiente del temporizador T1C0 (el bit de control T1C0 sirve como bandera pendiente a las interrupciones por sobreflujo en la opción de captura de datos). Consecuentemente, el bit de control T1C0 debe ser restablecido cuando se entre en la opción de captura de datos. La interrupción por sobreflujo en el temporizador es habilitada con la bandera de control T1ENA. Cuando ocurra una interrupción en T1A en el modo de captura de datos, el usuario debe revisar ambas

banderas T1PNDA y T1C0 en orden para determinar si la entrada de captura T1A o un sobreflujo provoca esta interrupción.

Banderas de control del Temporizador:

T1C0	Bit de control del Start/Stop en las opciones 1 y 2. Proceso independiente de PWM y contador externo de eventos, donde 1 = Start, 0 = Stop. Bandera para la interrupción por sobreflujo en la opción 3.
T1PNDA	Bandera pendiente a la interrupción del temporizador.
T1PNDB	Bandera pendiente a la interrupción del temporizador.
T1ENA	Habilita la bandera de interrupción del temporizador.
T1ENB	Habilita la bandera de interrupción del temporizador. 1 = Habilita la interrupción del temporizador. 0 = Deshabilita la interrupción del temporizador.
T1C3	Opción de control del temporizador.
T1C2	Opción de control del temporizador.
T1C1	Opción de control del temporizador.

WATCHDOG /CLOK Monitor

Este dispositivo contiene un WATCHDOG y un monitor de reloj, los cuales pueden ser configurados por el usuario. El WATCHDOG está diseñado para detectar ciclos infinitos en los programas del usuario, dando como resultado un mal funcionamiento en la operación del dispositivo. El WATCHDOG contiene dos ventanas separadas de servicio. Mientras la parte alta de la ventana programable habilita el tiempo de servicio del WATCHDOG, la parte baja de la ventana provee protección para los ciclos infinitos que contiene el servicio de instrucción del WATCHDOG.

Los dispositivos de la familia COPSAX7 proveen más características de *software*, las cuales proveen protección para evitar que el *stack pointer* encime datos del programa y para no tener localidades de memoria fuera del área válida de programa.

El monitor de reloj es usado para detectar la ausencia de reloj (oscilador) ó un ciclo muy bajo de reloj en la terminal CKI. El WATCHDOG consiste básicamente en dos bloques lógicos independientes: WD parte alta y WD parte baja. WD parte alta establece el límite superior de la ventana de servicio y WD parte baja define el límite inferior de la ventana de servicio.

El servicio del WATCHDOG consiste en escribir un valor específico en el registro de servicio del WATCHDOG, el cual recibe el nombre de WDSVR, el cual está en el mapa de memoria de la RAM. Este valor está compuesto de tres campos, el primer campo consiste de dos bits para seleccionar la ventana, el segundo campo consiste de 5 bits que forman un dato de control y el último campo contiene un bit, el cual sirve para habilitar el monitor del reloj. La tabla IV.1.6. muestra los campos del registro WDSVR.

Selección de Ventana		Dato de control					Monitor del reloj
X	X	0	1	1	0	0	Y

Tabla IV.1.6. Registro de control WDSVR.

El límite inferior de la ventana de servicio es de 256 ciclos de instrucción. Los bits 7 y 6 del registro WDSVR permiten al usuario elegir el límite superior de la ventana de servicio.

La tabla IV.1.7. muestra cuatro posibles combinaciones del límite inferior y superior para la ventana de servicio del WATCHDOG. Esta flexibilidad para escoger la ventana de servicio del WATCHDOG previene cualquier carga indebida en el *software* del usuario. Los bits 5, 4, 3, 2, y 1 del registro WDSVR representan un dato de control. Este dato de control es igual a 01100.

WDSVR Bit 7	WDSVR Bit 6	Ventana de Servicio Límites superior e inferior.
0	0	256 – 8k t_c ciclos
0	1	256 – 16k t_c ciclos
1	0	256 – 32k t_c ciclos
1	1	256 – 64k t_c ciclos

Tabla IV.1.7. Ventana de servicio del WATCHDOG.

Monitor del reloj

El monitor del reloj dentro de este dispositivo puede ser habilitado o deshabilitado mediante el programa de control. El monitor del reloj es garantía para no rechazar el reloj si el ciclo de reloj por instrucción ($1/t_c$) es mayor o igual a 10 kHz.

Operación del WATCHDOG y del monitor del reloj

El WATCHDOG es habilitado por el bit 2 del registro ECON. Cuando este bit es 0, el WATCHDOG está habilitado y la terminal G1 se convierte en la salida del WATCHDOG. El WATCHDOG y el monitor del reloj son deshabilitados durante la señal de restablecimiento (*reset*). De esta manera un error en el monitor del reloj ocurrirá después de salir de la señal de restablecimiento, si la frecuencia del ciclo de reloj por instrucción no es el mínimo valor especificado, incluyendo el caso donde el oscilador falla al iniciar.

El registro WDSVR puede ser escrito sólo una vez después de la señal de restablecimiento, y el dato de control, que está formado por los bits del 5 al 1 del registro WDSVR, no puede ser modificado por el usuario, dando por resultado una escritura válida. Esta escritura en el registro WDSVR involucra dos casos irrevocables: a) la selección de la ventana de servicio del WATCHDOG, b) habilitación o deshabilitación del monitor del reloj. Por lo tanto, la primera escritura en el registro WDSVR es para habilitar o deshabilitar el monitor del reloj y para la ventana de servicio del WATCHDOG, así como el dato de control debe corresponder. Las escrituras subsecuentes en el registro WDSVR serán comparadas con

el valor inicial de este mismo especificado por el usuario, la tabla I.1.8. muestra la serie de eventos que pueden ocurrir.

Dato de control	Dato de la ventana	Monitor del reloj	Evento
Corresponde	Corresponde	Corresponde	Servicio valido: Reiniciar ventana de servicio.
No importa	No corresponde	No importa	Error: Genera salida del WATCHDOG.
No corresponde	No importa	No importa	Error: Genera salida del WATCHDOG.
No importa	No importa	No corresponde	Error: Genera salida del WATCHDOG.

Tabla IV.1.8. Eventos de la ventana de servicio del WATCHDOG.

Resumen del WATCHDOG y del monitor del reloj

- Ambos circuitos detectores, el WATCHDOG y el monitor del reloj, son inhibidos durante la señal de restablecimiento.
- Después de la señal de restablecimiento, ambos circuitos son activados.
- La opción de habilitación o deshabilitación de la ventana de servicio del WATCHDOG y el monitor del reloj únicamente puede ser cambiada una vez durante el inicio del servicio del WATCHDOG.
- Al inicio del servicio del WATCHDOG el dato de control en el registro WDSVR, no debe ser modificado por el usuario, de lo contrario se producirá un error en el WATCHDOG.
- En los servicios subsecuentes del servicio del WATCHDOG el dato que se escribe en el registro de control WDSVR, debe ser el mismo para evitar un error en el WATCHDOG.
- El valor del dato de control no puede ser leído desde el registro WDSVR. En cualquier intento por leer este dato el usuario únicamente leerá ceros.

- En las opciones de HALT o IDLE el circuito detector del WATCHDOG es inhibido.
- El circuito detector del monitor del reloj es activado en las opciones de HALT o IDLE. Consecuentemente, el dispositivo que inadvertidamente entra en la opción de HALT será detectado como un error en el monitor del reloj.

MICROWIRE/PLUS

MICROWIRE/PLUS es un sistema de comunicación serial síncrono que permite a los microcontroladores de la familia COP888 comunicarse con cualquier otro dispositivo que cuente con el sistema de MICROWIRE/PLUS. Los ejemplos de tales dispositivos incluyen a los convertidores A/D, comparadores, memorias EEPROMs, controladores de *displays*, dispositivos para telecomunicaciones, y otros procesadores (e.g., HPC y COP400 procesadores). La *interface* serial MICROWIRE/PLUS usa una conexión simple y barata de tres alambres entre los dispositivos.

Pueden conectarse varios dispositivos de MICROWIRE/PLUS al mismo sistema de tres alambres. Uno de estos dispositivos, operando en lo que se llama el modo maestro, suministra el reloj síncrono para la *interface* serial y comienza el traslado de los datos. Otro dispositivo, operando en lo que se llama el modo esclavo, responde enviando (o recibiendo) los datos pedidos.

El dispositivo esclavo usa el reloj del maestro para el intercambio serial de los datos de salida (o entrada), mientras que el dispositivo maestro cambia los datos de entrada (o de salida). En la familia COP888, las tres señales de la *interface* se llaman SI (Entrada Serial), SO (Salida Serial), y SK (Reloj Serial). Para el dispositivo maestro, SO y SK son salidas (conectadas a las entradas del dispositivo esclavo), y SI es una entrada (conectada a la salida del dispositivo esclavo).

Los microcontroladores de la familia COP888 pueden operar como maestro o como esclavo, dependiendo de cómo sea configurado por el *software*. La figura IV.1.9 muestra un ejemplo de cómo pueden conectarse varios dispositivos juntos usando el sistema de MICROWIRE/PLUS, con el COP888 en la izquierda que opera como maestro y los otros dispositivos que operan como esclavos. El protocolo para seleccionar y habilitar los dispositivos esclavos es determinado por el diseñador del sistema.

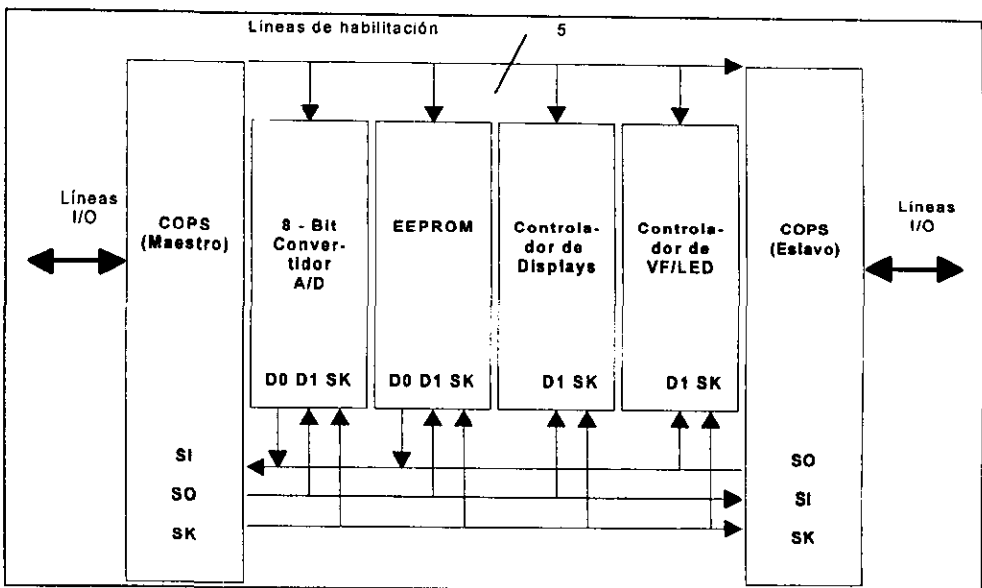


Figura IV.1.9. Aplicación a bloques del MICROWIRE/PLUS.

La figura IV.1.10 es un diagrama a bloques que ilustra el funcionamiento interior del circuito de MICROWIRE/PLUS.

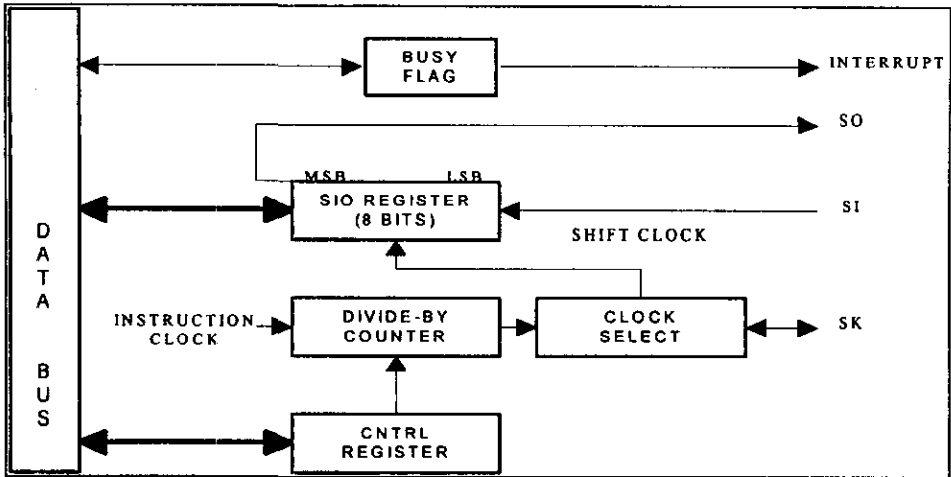


Figura IV.1.10. Diagrama a bloques del MICROWIRE/PLUS.

Un registro de cambio de 8 bits, llamado el registro SIO (de Entrada/Salida Serial), se usa para ambos propósitos, enviando y recibiendo los datos. En cualquier tipo de transferencia de datos, los bits son cambiados a través del registro. Cuando un byte de datos se envía, los bits son transmitidos por medio del registro SO de salida, comenzando con el bit más significativo. Cuando un byte de datos es recibido, los bits se transmiten por medio del registro SI de entrada, comenzando con el bit más significativo.

El registro SIO está contenido en el espacio de memoria de datos del microcontrolador, permitiendo mediante el *software* escribir un byte de datos para ser enviado, o para leer un byte de datos completo que se ha recibido. La bandera de ocupado en el registro PSW indica si el registro SIO está listo para ser leído o escrito. Las interrupciones pueden usarse para sincronizar la lectura o escritura del registro SIO, para realizar la transmisión de cada uno de los 8 bits, o por medio del *software* se puede programar una rutina que cumpla con este propósito.

Un byte de datos se puede escribir en el registro SIO únicamente cuando el reloj SK está en un nivel bajo. Si el byte de datos se escribe en el registro cuando SK está en un nivel alto, datos erróneos pueden ser colocados en el registro.

Ciclos de Reloj

La señal de reloj SK es generada por el dispositivo maestro. El tiempo de la *interface* MICROWIRE/PLUS se sincroniza con esta señal.

Hay dos modos de operación para la *interface*, llamados el modo normal SK y el modo alternado SK. En el modo normal SK, el dato de salida en SO es transmitido en el flanco de bajada del reloj SK, y el dato de entrada en SI se captura en el flanco de subida del reloj SK. En el modo alternado SK, las funciones de los flancos del reloj SK se invierten: el dato de salida en SO es transmitido en el flanco de subida del reloj SK, y el dato de entrada en SI se captura en el flanco de bajada del reloj SK.

Los ciclos de reloj para las señales de la *interface* SK, SO, y SI se muestran en la figura IV.1.11 para el modo normal SK, y en la figura IV.1.12 para el modo alternado SK. Las flechas sólidas indican los flancos del reloj SK que causan que el dato de salida sea transmitido por medio de la terminal SO, y las flechas punteadas indican los flancos del reloj que causan que el dato de entrada en la terminal SI sea capturado.

En el modo normal SK, el bit más significativo se vuelve válido inmediatamente después de que el registro SIO de transmisión del dispositivo está cargado. Así, el dato es capturado por el dispositivo receptor en el flanco de subida inicial del reloj SK. En el modo alternado SK, el bit más significativo es transmitido en el flanco de subida inicial del reloj SK, y es capturado por el dispositivo receptor en el flanco de bajada de ese mismo pulso del reloj. La opción de cual modo se puede usar, depende en la aplicación del protocolo de MICROWIRE/PLUS.

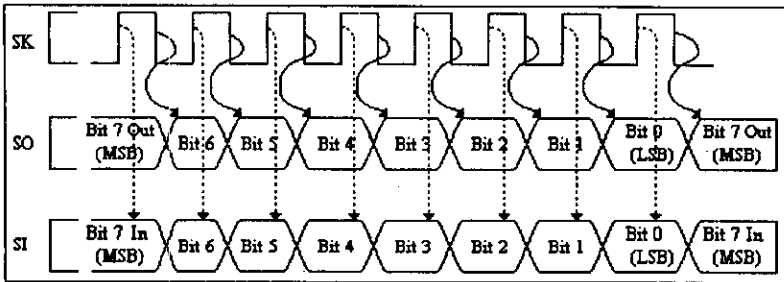


Figura IV.1.11. Modo normal SK del MICROWIRE/PLUS.

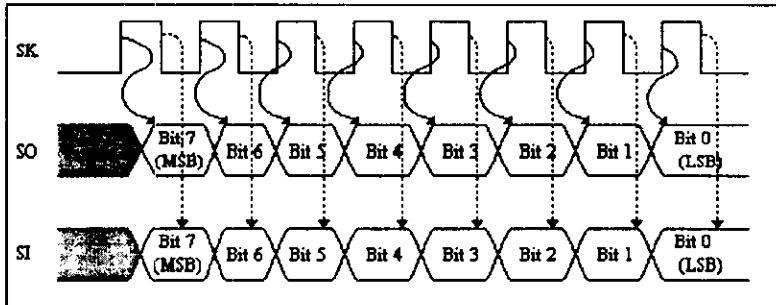


Figura IV.1.12. Modo alternado SK del MICROWIRE/PLUS.

El modo de reloj SK es determinado por el bit SKSEL (bit 6) en el registro de configuración del puerto G. En la señal de restablecimiento, este bit se restablece a 0, produciendo el modo normal SK. Se puede cambiar el funcionamiento de los COP888 al modo alternado SK por medio del bit SKSEL.

Configuración del Puerto G

Las tres señales alternadas del MICROWIRE/PLUS SO, SK, y SI son funciones alternadas de puerto G terminales G4, G5, y G6, respectivamente. Habilitar el uso de estas terminales para la *interface* MICROWIRE/PLUS, el bit MSEL (seleccione MICROWIRE) del registro de CNTRL debe ponerse en 1. (Se usan los bits SL1 y SL0, también en el registro de CNTRL, para controlar la velocidad del reloj SK en el modo maestro, como se describe abajo.)

El puerto G debe configurarse apropiadamente para el funcionamiento de la *interface*. Esto es logrado escribiendo ciertos valores de bit en el registro de configuración del puerto G. La terminal G4 (SO) debe configurarse como una salida para enviar datos. Puede ponerse en el modo de TRI-STATE (alta-impedancia) para recibir los datos, para prevenir que datos desconocidos sean puestos en la terminal de salida SO. La terminal G5 (SK) debe configurarse como una salida en el modo maestro, o como una entrada en el modo esclavo. G6 (SI) sólo sirve como una entrada, así que no necesita ser configurado como tal específicamente. En la tabla IV.1.9 se muestran las diferentes opciones de configuración del puerto G.

Reg. De Config. Del Port G Bits G5-G4	Funcionamiento del MICROWIRE	Función del terminal G4	Función del Terminal G5	Función del terminal G6
0 - 0	Esclavo, dato de entrada.	TRI-STATE (sin usar)	Entrada SK	Entrada SI
0 - 1	Esclavo, dato de entrada y salida.	Salida SO	Entrada SK	Entrada SI
1 - 0	Maestro, dato de entrada.	TRI-STATE (sin usar)	Salida SK	Entrada SI
1 - 1	Maestro, dato de entrada y salida	Salida SO	Salida SK	Entrada SI

Tabla IV.1.9. Bits del registro de configuración del puerto G.

Frecuencia del reloj SK

Cuando los COP888 operan en el modo maestro, generan la señal de reloj SK. Un contador divisor baja la frecuencia de la instrucción de reloj, produciendo un periodo de reloj SK que es 2, 4, o 8 veces el periodo de la instrucción de reloj. El factor divisor es programado escribiendo dos de los bits del registro CNTRL, designados SL1 y SL0 (Seleccione 1 y Seleccione 0), como se indica en la tabla IV.1.10.

SL1 (CNTRL Bit 1)	SL0 (CNTRL Bit 0)	Periodo del reloj SK
0	0	2 veces el periodo de la instrucción de reloj.
0	1	4 veces el periodo de la instrucción de reloj.
1	X	8 veces el periodo de la instrucción de reloj.

Tabla IV.1.10. Bits de selección del reloj en el modo maestro.

El contador divisor interno se restablece cuando la bandera de ocupado del MICROWIRE (descrita anteriormente) está en 1. Debido a esto, el contador siempre empieza su conteo a partir de 0 y se restablece de nueva cuenta en este valor después de haber contabilizado 8 bits, asegurando pulsos uniformes del reloj SK.

Cuando el microcontrolador opera en el modo de esclavo, el reloj SK es generado por el dispositivo externo maestro. En este caso, SK es una entrada, y el SK el circuito generador del reloj de los COP888 está inactivo.

Interrupción y Bandera de ocupado

Un bit de bandera en el registro PSW (estado de la palabra en el procesador) indica el estado del registro de intercambio SIO. Para comenzar una operación de intercambio de 8 bits, el *software* pone este bit en 1. Entonces comienza el intercambio y continúa automáticamente a la velocidad del reloj SK. Con cada cambio, el bit más significativo del registro es transmitido por medio de SO (si está habilitado), y simultáneamente, el bit menos significativo del registro SI es capturado.

Cuando la operación de intercambio de 8 bits termina, la bandera de ocupado se restablece automáticamente a 0 por el *hardware*, y una interrupción de MICROWIRE/PLUS se genera (si está habilitado). Cuando esto ocurre, el *software* debe escribir el próximo byte para

ser enviado y entonces se enciende la bandera de ocupado para comenzar la transferencia del próximo byte.

Si se usa una rutina de poleo, la CPU corre en dicha rutina en la que el estado de la bandera de ocupado se lee. Cuando la bandera se encuentra en 0, el *software* lee o escribe en el registro SIO, se debe habilitar la bandera de ocupado para la próxima transmisión, y después regresar a la rutina de poleo.

Si se usan interrupciones, la CPU puede realizar otras tareas mientras el byte de los datos está transmiéndose por medio del registro SIO. La rutina de servicio de la interrupción del MICROWIRE restablece la bandera pendiente (uWPND) del MICROWIRE, escribe o lee el registro SIO, pone en 1 la bandera de ocupado para la próxima transferencia, y después regresa desde la interrupción. Para habilitar la interrupción del MICROWIRE, se enciende el bit uWEN (interrupción habilitada del MICROWIRE). El bit GIE (habilitación Global de Interrupciones) también debe ponerse en 1. La bandera del uWPND y el bit de activación uWEN son los bits 3 y 2 del registro ICNTRL, respectivamente.

El *software* puede controlar el tiempo de transferencia activando y limpiando la bandera de ocupado directamente. Si el bit de la bandera de ocupado es limpiado directamente por el *software*, inmediatamente se produce un alto. En este caso la interrupción no se genera.

El protocolo de comunicación entre el maestro y el esclavo debe asegurar que el dispositivo esclavo tenga bastante tiempo para responder después de ser habilitado por el maestro.

Algunos dispositivos externos pueden requerir un flujo continuo de bits, sin ninguna pausa entre los bytes. Este modo, llamado el modo continuo del MICROWIRE, se cumple escribiendo mediante *software* una rutina que ejecuta un número específico de ciclos. El reloj debe ser dividido por un factor de 8.

Cuando los COP888 operan en modo de esclavo, la bandera de ocupado sólo debe activarse cuando la señal del reloj SK es un nivel bajo (una entrada). Esto es porque el bit de ocupado es internamente sumado con la señal SK para producir que la señal del reloj cambie. Si la bandera de ocupado es activada mientras SK es un nivel alto, el pulso SK es reconocido inmediatamente, produciendo un pulso de reloj con un ancho desconocido (quizás muy estrecho), causando un corrimiento.

Después de describir las características del microcontrolador, procederemos a la selección de los diferentes dispositivos que integran al sistema de peaje.

IV.1.2. Señal de entrada (rpm´s)

El sistema de peaje adquiere la señal de revoluciones por minuto (rpm´s), esta señal proviene de un interruptor tipo *reed*. El interruptor tipo *reed* que usamos fue un HYR 2001, marca ALEPH, en la tabla IV.1.2.1 se presentan sus características y sus dimensiones.

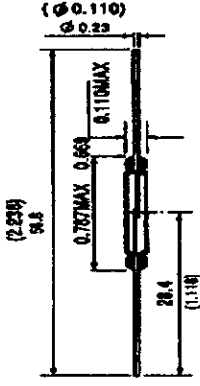
Número de parte	HYR2001
Dimensiones Unidades: mm(inch)	
Forma de Contacto	1A
Posición de Contacto	C
Material de Contacto	Rodio
Máximo Rango de Contacto	10W

Tabla IV.1.2.1. Características del interruptor tipo *reed* (continúa).

Máximo Voltaje de Operación	200VDC
Máxima Corriente de Operación	0.5A
Máxima Resistencia Inicial de Contacto	100 Mohms
Valor de los contactos cerrados (AT)	15-50
Valor mínimo de los contactos abiertos (AT)	6
Mínimo Voltaje de Ruptura	300VDC
Máxima Capacitancia de Contacto	0.4pF
Mínima Resistencia de Aislamiento	10^{10} ohms
Típica Frecuencia de Resonancia	3kHz
Vida Eléctrica (Cargas Resistivas)	10^7 (5VDC, 10mA) 10^6 (100VDC, 100mA)
Bobina de Prueba	TC-0502
Características	Aplicación General

Tabla IV.1.2.1. Características del interruptor tipo *reed*.

La señal que nos entrega este interruptor son pulsos, los cuales son pasados por un filtro paso bajas, para cancelar frecuencias mayores a 200 Hz. Un filtro paso bajas es un circuito que tiene un voltaje de salida constante desde cc hasta una frecuencia de corte f_c . Conforme la frecuencia aumenta arriba de f_c , el voltaje de salida se atenúa (disminuye). En la figura IV.1.2.1 se muestra una gráfica de la magnitud del voltaje de salida del filtro paso bajas comparada con la frecuencia. La línea continua es una gráfica para el filtro ideal paso bajas, en tanto que las líneas punteadas indican las curvas para los filtros prácticos paso bajas. El alcance de frecuencias que se transmiten se conoce como "banda de paso". El alcance de frecuencias que se atenúan se conoce como "banda de paro". La frecuencia de corte, f_c , también se denomina: frecuencia 0.707, frecuencia a 3 dB ó frecuencia de esquina.

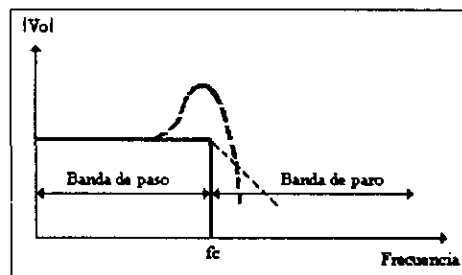


Figura IV.1.2.1. Filtro paso bajas.

Para el cálculo de la frecuencia de corte se utilizó la ecuación IV.1.2.1, la cual se muestra a continuación.

$$\omega_c = \frac{1}{RC} = 2\pi f_c \quad (\text{IV.1.2.1})$$

Donde: ω_c es la frecuencia de corte en radianes por segundo.

f_c es la frecuencia de corte en Hertz.

R está en ohms.

C está en farads.

Sustituyendo los valores de $R = 7.5 \text{ K}\Omega$ y $C = 0.1 \mu\text{F}$, tenemos el siguiente resultado:

$$\omega_c = 1.33 \text{ krad/s.}$$

o bien

$$f_c = \frac{\omega_c}{2\pi} = \frac{1.33\text{k}}{6.28} = 211.67 \text{ Hz.}$$

La frecuencia de corte que calculamos es buena para el sistema de peaje, por el hecho de que no nos interesan frecuencias grandes (p. Ej. 1 kHz), ya que el rango de frecuencias en un vehículo de transporte público va desde 0 Hz a 120 Hz, tomando en cuenta que las velocidades del vehículo no son muy grandes. En la figura IV.1.2.1 se muestra el circuito del filtro paso bajas para el interruptor tipo *reed*.

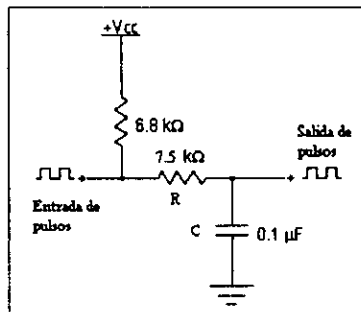


Figura IV.1.2.1 Circuito paso bajas.

La señal que sale del filtro es la que procesa el sistema de peaje para vehículos de transporte público. El siguiente módulo del sistema es el teclado, el cual mencionaremos a continuación.

IV.1.3. Teclado

Un teclado consiste en presionar o tocar para activar interruptores que están colocados en un arreglo matricial. Para descubrir qué tecla se ha presionado, se requiere una combinación de hardware y/o por medio del software hacer una rutina para que realice esta operación.

Hay dos tipos básicos de teclados, los cuales son: codificado y no-codificado. El teclado codificado incluye el hardware necesario para detectar qué tecla fue presionada y para mantener ese dato hasta que se presione nuevamente. El teclado no-codificado no tiene ningún hardware y debe ser analizado por una rutina de software o por hardware especial.

El sistema de peaje tiene un teclado de 4 botones, estos botones son del tipo *push-button*, este tipo de interruptor mecánico se usa bastante en la electrónica, ya que la fuerza que se aplica para realizar la función está dirigida en el mismo sentido que la necesaria para aproximar el contacto móvil sobre el fijo, este interruptor mecánico regresa a su estado inicial por medio de un resorte después de haberse presionado. Los elementos del interruptor están hechos de bronce y las áreas de contacto están chapeadas con oro.

Los interruptores mecánicos son relativamente baratos, pero estos tienen algunas desventajas, una es que estos presentan un rebote al momento de hacer contacto los elementos del interruptor. Cuando un interruptor es presionado puede presentar un rebote durante un tiempo corto antes de que haga un contacto sólido. Este rebote puede tener un tiempo de duración de 10 ms. La solución de este problema es la de realizar una rutina por medio de *software*, que verifique el estado de la tecla que se presionó por un periodo de tiempo de aproximadamente 20 ms. En la figura IV.1.3.1 se muestra una gráfica del rebote en un interruptor mecánico.

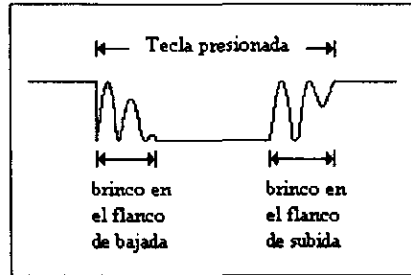


Figura IV.1.3.1. Rebote en un interruptor mecánico.

Otra desventaja es que con el paso del tiempo los contactos pueden oxidarse o ensuciarse, provocando que éstos ya no hagan un buen contacto. Los interruptores mecánicos seleccionados son de buena calidad y tienen una vida de aproximadamente de 1 millón de pulsaciones. En la figura IV 1 3.2 se ilustra un interruptor mecánico.



Figura IV.1.3.2. Interruptor mecánico.

IV.1.4. Despliegado de la información

Para la parte de visualización de la información se utilizan *displays* de 7 segmentos, los cuales están formados por un conjunto de trazos rectos, que contienen un *led* cada uno, con una estructura geométrica similar a un 8. Los segmentos se designan con las letras a, b, c, d, f y g. Esta forma de *displays* permite representar todos los números pero presenta muchas limitaciones a la hora de reproducir caracteres alfabéticos, siendo de fácil utilización desde el punto de vista electrónico. En la figura IV.1.4.1 se muestran *displays* de 7 segmentos.

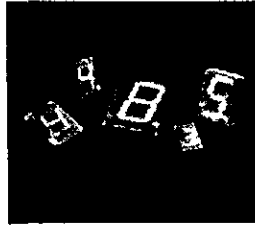


Figura IV.1.4.1. *Displays de 7 segmentos.*

El sistema que desarrollamos utiliza 10 *displays* de 7 segmentos, por lo cual requerimos de un transistor para poder suministrar la corriente necesaria para cada segmento, cada segmento consume 30 mA. Para evitar ocupar mucho espacio al colocar 8 transistores individuales, se utilizaron circuitos integrados, los cuales contienen cada uno 4 transistores, estos transistores nos pueden proporcionar la corriente necesaria.

IV.2. Diseño de accesorios

En este apartado veremos el diseño de un solo accesorio. Este accesorio es una impresora de recibos.

Mecanismo

El mecanismo consiste en una microimpresora térmica. Dado su pequeño tamaño este tipo de impresoras es ideal para aplicaciones en instrumentos de medición y analizadores, aparatos médicos y dispositivos terminales de datos, aunque no descartamos su uso en otras aplicaciones donde se requiera.

El mecanismo es una impresora MTP102-16B de Seiko Instruments Inc. Esta impresora tiene como características generales las siguientes:

- Bajo costo
- Libre de mantenimiento
- Alta calidad de impresión
- Manejada por batería
- Silenciosa debido a su sistema de no-impacto
- Diseño versátil

Las características eléctricas y de impresión se muestran en la tabla IV.2.1 y IV.2.2 respectivamente. Como podemos observar en dichas tablas el voltaje de operación es de 5V por lo que podemos utilizar una lógica de control basado en un microcontrolador y circuitería externa a él, de tal forma que también sea compatible con los voltajes de operación de la impresora.

CARACTERÍSTICAS	ESPECIFICACIONES
Voltaje de operación	5.0 ± 1.0 V de DC
Consumo de corriente (a) durante impresión	3.2 A máximo
temperatura ambiente y 5.0 V) durante la carga de papel	250 mA máximo
Rango de temperatura de operación	0 a 50°C
Rango de temperatura sin operar.	-40 a 60°C
Ruido de operación (a temperatura ambiente y 5.0 V)	70 dB o menor
Dimensiones (ancho x largo x alto)	31x48x13.8 mm

Tabla IV.2.1. Características eléctricas generales.

CARACTERÍSTICAS	ESPECIFICACIONES
Método de impresión	Impresión térmica por punto
Dirección de impresión.	De izquierda a derecha en la dirección que el papel es alimentado
Puntos por línea	7 puntos x 110 puntos
Matriz de caracteres	matriz de 7x5 puntos
Tamaño del carácter	2.4 mm x 1.2 mm
Caracteres por línea	16 caracteres (con un espacio de 2 puntos)
Activación de la cabeza	Mediante un tacogenerador
Detección de posición inicial	Mediante un interruptor mecánico

Tabla IV.2.2. Características de impresión de la impresora.

Las características que se observan en las tablas son las principales, aun cuando se tienen más características.

Características de manejo del motor

El motor con el que opera la impresora es un motor de corriente directa. Este motor es la principal fuente de energía para el movimiento de la cabeza de impresión y la alimentación del papel. Mediante la aplicación de un voltaje de DC al motor, la cabeza automáticamente se mueve hacia adelante o hacia atrás, y el papel es alimentado en el regreso, de ahí que no se requiera cambiar la dirección de movimiento del motor. Las características del motor son las siguientes:

1. Voltaje de operación del motor. 5.0 ± 1.0 VDC
2. Consumo de potencia del motor.
 - Durante la impresión 170 mA máx.
 - Durante la alimentación de papel 250 mA máx.
 - Iniciando el movimiento 700 mA máx.
 - Deteniendo el movimiento 500 mA máx.

Cada una de estas características se definen a temperatura ambiente y con una alimentación de 5.0 VDC. En la figura IV.2.1 se puede apreciar la gráfica de comportamiento de consumo de corriente a través del tiempo con las características antes mencionadas.

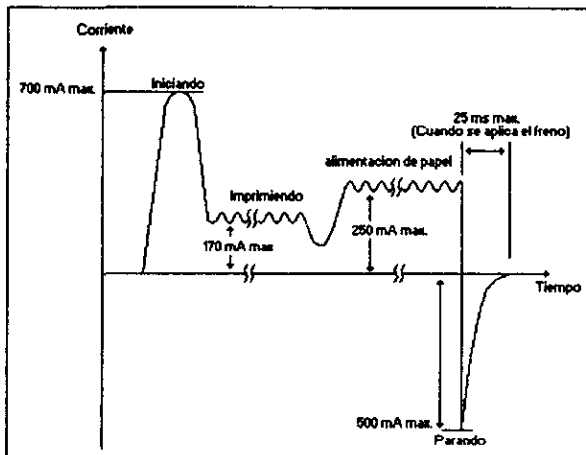


Figura. IV.2.1. Características de manejo del motor.

Al tener las características eléctricas del motor podemos seleccionar algún dispositivo para la polarización de la impresora, dado que se tendrá que generar una fuente independiente de la circuitería alrededor del microcontrolador.

Características de salida del tacogenerador

El tacogenerador es un generador de señales conectado directamente al motor, que genera una señal parecida a una onda senoidal, la cual tiene dos ciclos por cada vuelta del motor. Esta señal es convertida mediante un circuito recortador en una señal rectangular, la cual es usada como una señal de tiempo para la impresión en la cabeza térmica. La figura IV.2.2. nos muestra el circuito que convierte una señal senoidal en una cuadrada.

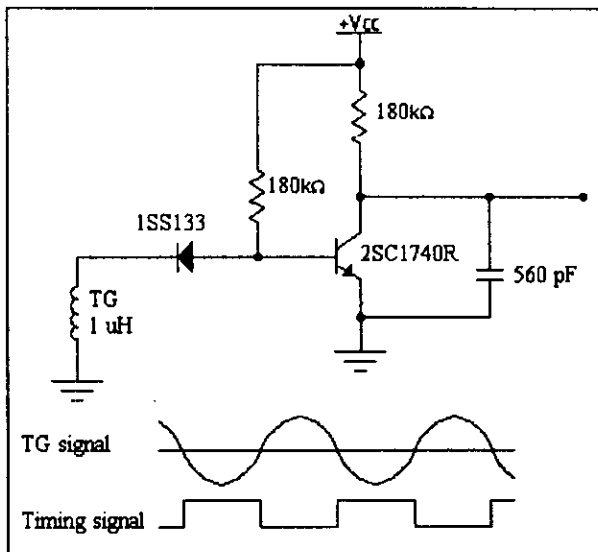


Figura IV.2.2. Circuito convertidor de onda senoidal a cuadrada.

El circuito anterior lo propone el fabricante para una operación correcta del dispositivo. Con cada pulso se imprimirá una columna de 7 puntos para ir formando cada carácter.

Interruptor de posición inicial

El interruptor de posición inicial es un interruptor mecánico el cual conmuta a apagado cuando la cabeza está en la posición inicial. Este interruptor tiene dos funciones principales: la primera es detectar la posición inicial cuando la cabeza pasa por ese punto; pasando por la posición inicial, el interruptor conmuta de “encendido” a “apagado”, y los frenos son aplicados para detener el movimiento de la cabeza de impresión. La segunda función es la de utilizarlo como un punto de referencia para iniciar el conteo de las señales de tiempo de impresión. Cuando la cabeza empieza el movimiento hacia la derecha de la posición inicial, el interruptor conmuta de “apagado” a “encendido”, punto en el cual se inicia el conteo de las señales de impresión.

Sistema de control de la impresora

Ahora estudiaremos el diagrama a bloque que se presenta en la figura IV.2.3 para poder iniciar el diseño de la arquitectura de la impresora.

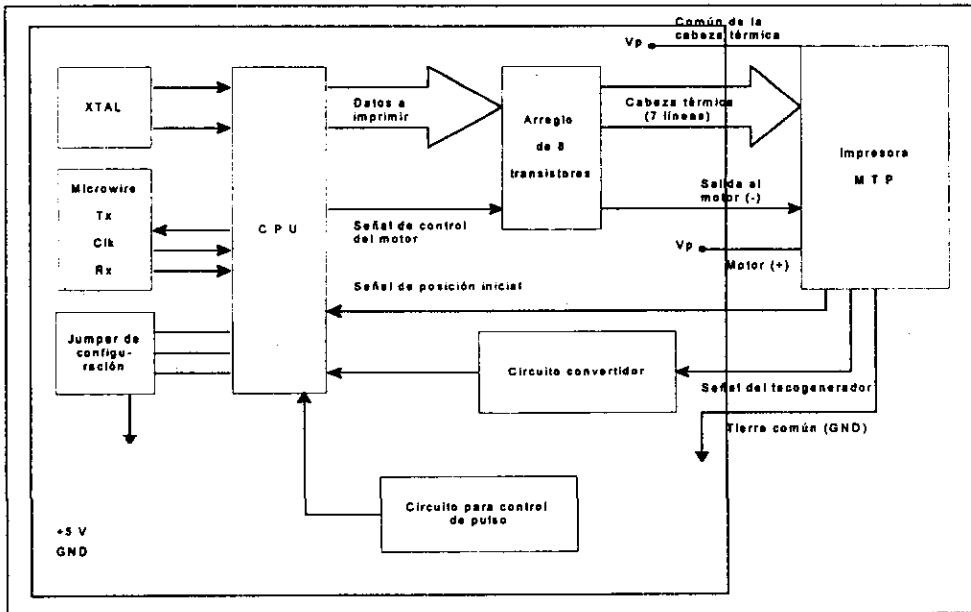


Figura IV.2.3. Diagrama a bloques de la arquitectura básica del control de la impresora.

Para desarrollar el controlador de dicha impresora utilizamos un microcontrolador de National Semiconductor: COP8SAC728. Dicho microcontrolador contiene 4Kb de memoria de programa interno, 128 byte de memoria RAM, dos temporizadores, un watchdog, un puerto de comunicaciones llamado MicroWire, dos puertos digitales bidireccionales de 8 y 4 líneas, un puerto con 4 líneas de salidas digitales de alta corriente; las características de dicho microcontrolador son exactamente las mismas que el microcontrolador descrito en el primer punto del presente capítulo

Considerando que la impresora se debe conectar al sistema de peaje, y que se utilizan microcontroladores de la misma familia en el diseño, decidimos conectarlos haciendo uso del protocolo de comunicaciones MICROWIRE/PLUS que tiene dichos microcontroladores.

Ahora bien el sistema de peaje debe saber cuando se ha conectado la impresora por lo que se tendrá que tener una línea exclusivamente para detectar la impresora y otra para poder habilitarla. También se va a requerir que el sistema de peaje alimente con 12V a la tarjeta de la impresora. Dado que el fabricante nos especifica algunas precauciones que hay que tomar en cuenta para el diseño del control, tenemos que habilitar la alimentación de la circuitería de la impresora desde el sistema de peaje, por lo que las señales de intercambio que se tienen del sistema de peaje a la impresora las podemos observar en la figura IV.2.4.

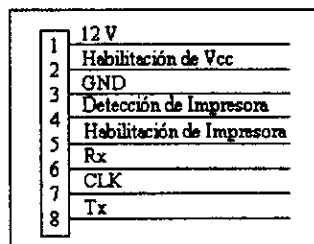


Figura IV.2.4. Señales de entrada a la tarjeta de impresora.

Para la alimentación tanto de la circuitería de control como para la impresora se usaron dos fuentes de voltaje de precisión como el LM2935T. Las características de estos reguladores son las siguientes:

- Dos salidas reguladas de 5 V
- Salida de corriente de 750 mA.
- Entrada y salida diferencial menor de 0.6 V a 0.5 A.
- Baja corriente del regulador en reposo ó standby.
- Protección contra polarización inversa.
- Protección contra sobre carga.
- Protección contra transitorios.
- Protección contra cortocircuito.
- Protección contra sobrecarga térmica interna.
- Disponible en encapsulado tipo TO-220
- Interruptor ON/OFF de control de la salida de alta corriente.

Para nuestro sistema requerimos de dos reguladores de voltaje, uno será activado por el sistema de peaje y el otro será activado cuando se haya encontrado una señal de inicio de impresión, considerando que no podemos tener energizada la cabeza de impresión por periodos de tiempo muy prolongados. La configuración final de los circuitos de alimentación se muestra en la figura IV.2.5

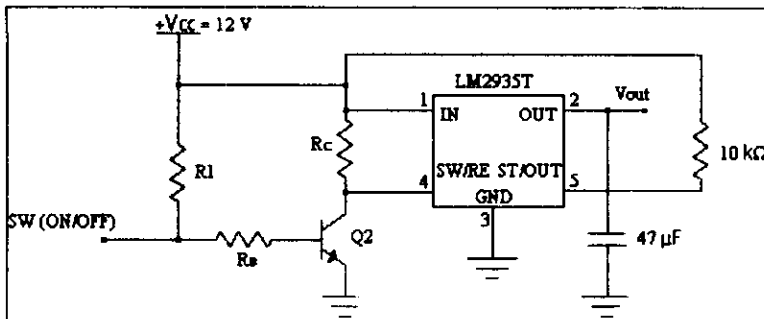


Figura IV.2.5 Fuentes de alimentación de la impresora.

Para calcular la resistencia de colector R_c propusimos una corriente I_c de saturación igual a 10 mA, la constante β del transistor igual a 200, y $V_{ce}=0.2$ V. Se tiene la siguiente ecuación:

$$\begin{aligned} I_{c_{sat}} &= 10 \text{ mA} \\ I_{c_{sat}} &= \frac{V_{cc} - V_{ce}}{R_c} \end{aligned} \quad (\text{IV.2.1})$$

Haciendo uso de la ecuación (IV.2.1) obtenemos el valor de R_c .

$$R_c = 1.1 \text{ k}\Omega$$

En saturación

$$I_R = \frac{I_{c_{sat}}}{\beta} \quad (\text{IV.2.2})$$

Sustituyendo valores en la ecuación IV.2.2, obtenemos que $I_B = 0.05$ mA. Proponiendo una corriente de $I_{BS} = 10I_B$ para asegurar la saturación del transistor dada la relación IV.2.3.

$$I_B > \frac{I_{c_{sat}}}{\beta} \quad (\text{IV.2.3})$$

calculamos la resistencia R_B con la ecuación IV.2.4.

$$R_B = \frac{V_i - V_{BE}}{I_B} \quad (\text{IV.2.4})$$

Sustituyendo los valores de $V_i = 5$ V y $V_{BE} = 1.2$ en la ecuación IV.2.4, tenemos una resistencia $R_B = 7.6 \text{ k}\Omega$. Proponiendo una resistencia de valor comercial de $7.5 \text{ k}\Omega$, se obtiene una corriente $I_{BS} = 506.6 \mu\text{A}$.

Verificando que la relación (IV.2.3) se cumpla tenemos que:

$$506.6 \mu\text{A} > 50.6 \mu\text{A}$$

Dado que la relación (IV.2.3) se cumple, entonces estamos asegurando que el transistor se sature cuando aplicamos una señal de 5 volts y se mantiene en corte cuando aplicamos un voltaje de entrada menor a 1.2 volts.

Con la corriente de base calculada no sobrepasamos el máximo de corriente que nos entrega la terminal del microcontrolador que controla el encendido, pero para asegurar que no se encienda en algún estado desconocido colocamos una resistencia R1 de 33 k Ω para mantenerlo siempre en corte.

El fabricante del regulador con control de encendido nos recomienda colocar un capacitor de por lo menos 10 μF para asegurar una buena estabilidad en el voltaje regulado de salida. En nuestro caso colocamos uno de 47 μF para mantener una correcta regulación durante transitorios.

Debido a que no utilizamos la salida de 10 mA del regulador el fabricante nos recomienda colocar una resistencia de 10 k Ω entre la salida regulada y el voltaje a regular, que en este caso es de 12 volts.

Ambas fuentes (de circuitería y de impresora) tienen la misma configuración, en el caso de la impresora no debemos tener energizada la cabeza de impresión a menos que ya se empiece a imprimir un dato, razón por la que usamos los mismos valores de resistencias para ambas fuentes.

Para saber que señales son las que vamos a controlar debemos conocer el tipo de conector y las señales que se tienen en dichos conectores. Estos conectores se encuentran en el mecanismo de la impresora.

El primer conector que tiene la impresora térmica es el de datos y alimentación de la cabeza. El tipo de conector que se usa es de membrana flexible. Esto lo podemos observar en la figura IV.2.6

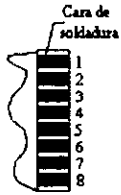
	Numero de terminal	Nombre de la señal	Función
	1	Dot 1	Primer punto de la cabeza térmica
	2	Dot 2	Segundo punto de la cabeza térmica.
	3	Dot 3	Tercer punto de la cabeza térmica.
	4	Dot 4	Cuarto punto de la cabeza térmica.
	5	Dot 5	Quinto punto de la cabeza térmica.
	6	Dot 6	Sexto punto de la cabeza térmica.
	7	Dot 7	Séptimo punto de la cabeza térmica.
	8	H-COM	Común de la cabeza térmica (+5V)

Figura IV.2.6. Asignación de terminales de la cabeza térmica.

Teniendo identificadas las señales de datos procedemos a identificar lo que sería el conector de señales de control del motor y del interruptor de posición inicial. Dicho conector es el que se muestra en la figura IV.2.7.

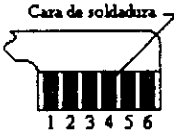
	Número de terminal	Nombre de la señal	Función
	1	HS	Switch de inicio, tierra para el Vcc.
	2	HS	
	3	TG	Tacogenerador, tierra para el Vcc.
	4	TG	
	5	M -	Tierra del motor.
6	M +	Vcc del motor.	

Figura IV.2.7 Asignación de terminales de control del motor e interruptor.

En la figura IV.2.8 se muestra un diagrama de tiempos, los cuales se tomaron en cuenta para realizar la programación para controlar la impresora. En este diagrama se observa el comportamiento de cada una de las señales que son importantes para el controlador de la impresora. Estas señales son generadas por el controlador al momento de estar imprimiendo un dato previamente cargado.

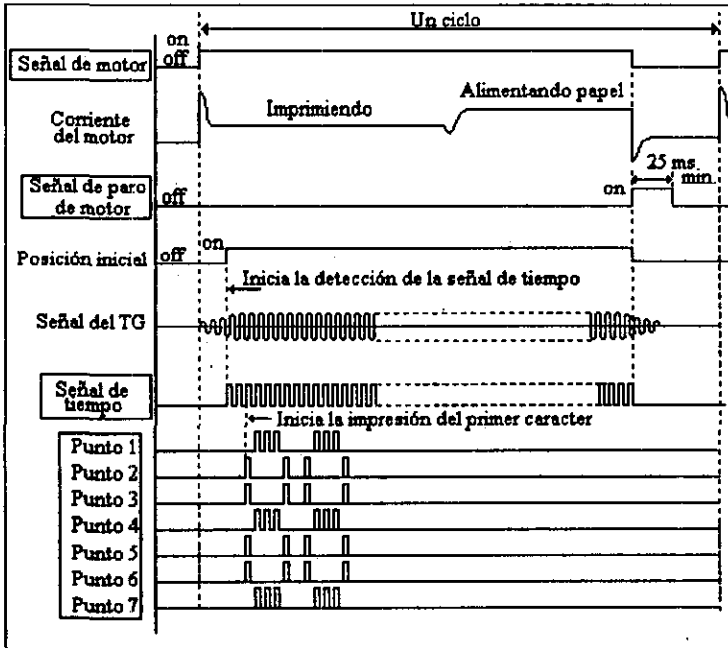


Figura IV.2.8. Diagrama de tiempos.

Como podemos observar en el diagrama de tiempos es necesario transmitir byte por byte hasta terminar el número de bytes que representan un carácter y tener en cuenta que el procesamiento de la información no deberá sobrepasar el tiempo en que se da una transición de bajo a alto en la señal de impresión. Las señales que aparecen dentro de un recuadro son señales que provienen del microcontrolador y las otras son señales que provienen de la impresora.

Ahora bien al inicializar la impresora se debe alimentar el papel en una línea y detener la cabeza de impresión en la posición inicial. La posición de inicio de impresión está determinada par cada línea en fase al flanco de subida del interruptor de posición inicial.

Teniendo la operación de la impresora nos avocamos ahora en la realización del diagrama de flujo para la operación de dicho accesorio, esto lo podemos observar en la figura IV.2.9.

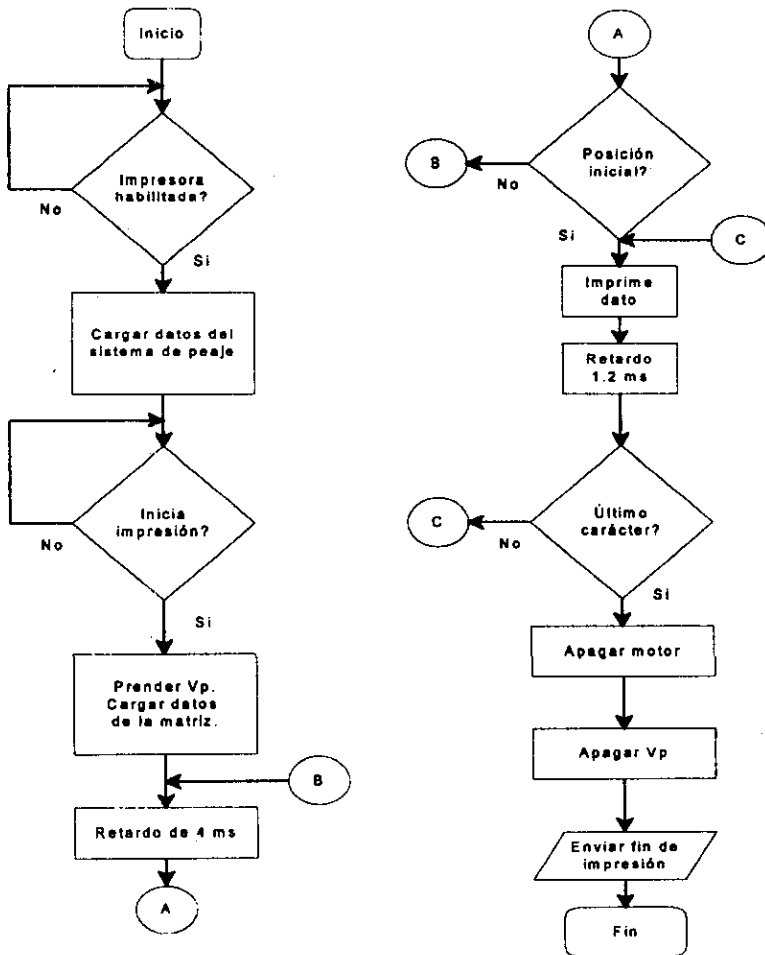


Figura IV.2.9. Diagrama de flujo de operación de la impresora.

Como podemos ver en el diagrama de flujo, lo que realizamos en primera instancia es estar verificando el estado de la bandera de habilitación de impresora. Cuando esta señal pasa de un estado bajo a uno alto se interpreta que el sistema de peaje enviará una serie de parámetros que serán los que se impriman en el recibo. Para esto el sistema carga los datos requeridos y espera un carácter de control para poder iniciar la impresión. Lo primero que hacemos es energizar la cabeza de impresión y el motor para que inicie el movimiento. Cuando se inicia el movimiento estamos verificando la posición inicial y cuando ésta se ha detectado se inicia la cuenta de pulsos de impresión, en cada flanco de subida se coloca el dato a imprimir en el puerto de datos del microcontrolador y éste lo pasa a un arreglo de transistores para generar una diferencia de voltaje entre los extremos de la cabeza de impresión y generar así una columna de la matriz del carácter. Dichos caracteres están formados, como se ha mencionado, de una matriz de 7x5 tal y como se muestra en la figura IV.2.10. En dicha figura podemos observar el arreglo de bytes para cada carácter, con lo que se requieren 5 bytes para poder formar un carácter

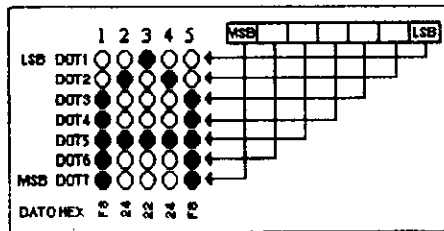


Figura IV.2.10. Formación de un carácter para la impresión de recibos.

Un punto importante a tratar en la impresión de los caracteres es la calidad de impresión. Par asegurar una alta calidad en la impresión, el ancho de pulso de activación de la cabeza debe ser configurado de acuerdo al voltaje y a la resistencia de la cabeza de impresión

La fórmula para calcular la energía aplicada es:

$$E = \frac{V_0 + V}{2V} E_0 \left(1 + \frac{T_0 - T}{100} \right) \quad (\text{IV.2.5})$$

Donde:

E es la energía aplicada [mJ].

V es el voltaje en la terminal FPC.

T es la temperatura [°C].

E₀ es la energía promedio [mJ].

V₀ es el voltaje de polarización [5 V].

T₀ es la temperatura ambiente [24 °C].

La fórmula para calcular el ancho de pulso aplicado es la siguiente:

$$t = \frac{R * E}{V^2} \quad (\text{IV.2.6})$$

Donde:

t es el ancho de pulso para la aplicación de energía a la cabeza [ms]

R es la resistencia de la cabeza de impresión, R=14.0 ohms.

E es la energía aplicada [mJ].

V es el voltaje Vp menos el V_{CE(SAT)}.

Con los datos anteriores podemos saber cuanto es el tiempo máximo que debemos energizar una cabeza para obtener una buena calidad de impresión. Si por descuido aplicamos energía por más del tiempo especificado por t podemos dañar la cabeza de impresión. Si estamos energizando la misma cabeza continuamente es recomendable reducir el tiempo de energizamiento a un 80% de t, para evitar daños y/o acortar la vida útil de la cabeza.

Dado que la temperatura es un factor que afecta las características de operación de gran cantidad de aparatos, para la impresora no hay excepción. por lo que se tienen circuitos de compensación contra variaciones de temperatura y voltaje que son recomendados por el fabricante. Con estos circuitos es posible ajustar el tiempo de energizamiento de las cabezas térmicas.

Para una mayor información sobre el calculo de los distintos valores del circuito de la impresora, se puede consultar el manual *MTP SERIES. Printer Mechanism Technical Reference. Seiko Instruments, 1995.*

Una vez diseñado el hardware del sistema de peaje, procederemos a analizar y diseñar la parte del software del sistema de peaje, el cual realizará el control total del sistema. Así mismo diseñaremos el software de comunicación del sistema de peaje con una PC.

CAPÍTULO V

DESARROLLO DEL SOFTWARE DEL SISTEMA DE PEAJE

En el presente capítulo abordaremos el desarrollo del software para la operación del sistema de peaje ya en su totalidad. Dicho programa se maneja de forma modular por lo que definiremos primeramente las funciones a realizar para posteriormente realizar el diagrama de flujo general del sistema y segmentar éste en programas por módulo.

También comentaremos el desarrollo del software que se realizó para la computadora personal. Este software está diseñado para poder establecer comunicación con el sistema de peaje y poder realizar los cambios de tarifa y/o parámetros requeridos por la SECOFI, según lo marca la norma.

V.1 Software del sistema

Como se vio en el capítulo anterior, en el desarrollo del hardware solamente obtuvimos como resultado un conjunto de circuitos que pueden ser gobernados por el microcontrolador, dicho microcontrolador no puede manejar por sí solo estos dispositivos externos, por lo que se requiere generar un programa que gobierne dichos dispositivos.

Con este programa logramos que el sistema funcione de manera autónoma sin tener que estar cargando un programa cada vez que se requiera operar el sistema.

El sistema de peaje está constituido básicamente por 4 módulos principales, como se puede apreciar en la figura V.1.1: un visualizador, formado con *display's* de 7 segmentos; un teclado, formado por 4 teclas; un sistema de seguridad, formado por dos llaves electrónicas; un circuito de comunicación serial y una entrada de la señal de velocidad.

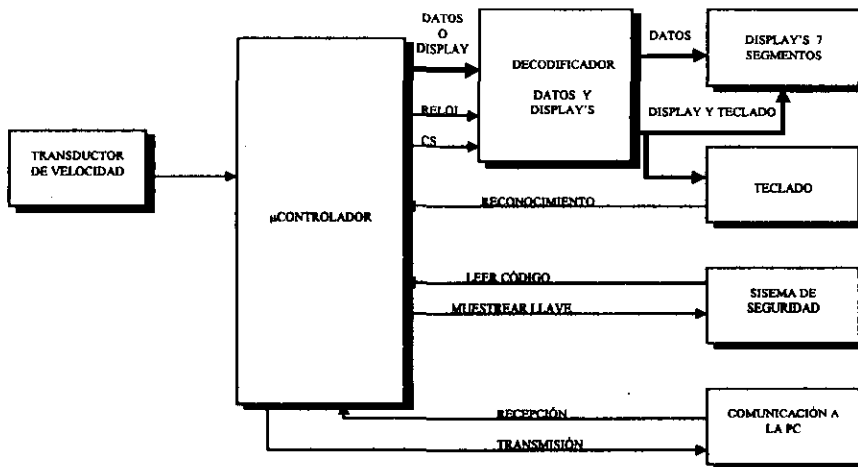


Figura V.1.1 Diagrama modular del sistema.

En la figura podemos apreciar que el desplegado y el teclado comparten un mismo bus de direcciones o habilitación. Esto se realiza para poder minimizar el número de líneas de control para cada una de las teclas y para cada uno de los *display's*.

La forma en que se opera es la siguiente: usamos como decodificador un registro de captura (*latch*), esto es para que podamos mantener el dato a visualizar en el desplegado, por un lado, por el otro habilitar la tecla correspondiente.

En caso de haberse presionado una tecla se envía un reconocimiento al microcontrolador indicando que ha sido presionada una tecla. Cabe señalar que solamente con este reconocimiento sabemos que una tecla se presionó mas no conocemos cual fue, por lo que

tenemos que tomar en cuenta y con tiempos exactos cuando se habilita una tecla, para no interferir con el desplegado de información en los visualizadores.

Dentro del módulo de seguridad se cuenta con dos tipos de llaves una llave maestra y otra de usuario. La llave maestra la tiene el fabricante de los sistemas de peaje y solamente se ocupa cuando se van a realizar las revistas correspondientes, o los cambios de tarifas autorizados por el gobierno. Dicha llave también tiene como función el realizar los enlaces para la comunicación serial.

La llave de usuario la conserva el conductor del vehículo para poder acceder a las funciones del sistema de peaje, una vez energizado este último. Señalamos que el sistema de peaje no va a requerir de la llave de usuario mientras no se le desconecte la alimentación de energía.

Para lograr un control óptimo de todas las acciones debemos tener en mente los tiempos en que se ejecutan cada una de las funciones del sistema y dar prioridad a algunos eventos, tanto internos como externos.

En primera instancia se verifica que tipo de llave se ha introducido al sistema; en caso de que sea la llave de revista, se tendrá un letrero que señale la conexión a una PC para la recepción o transmisión de datos, después de esto el sistema vuelve a reiniciar para poder cargar los nuevos parámetros y esperar que se introduzca la llave de usuario.

Una vez que el sistema está en operación, el programa debe tener como prioridad principal el contabilizar el número de pulsos que se generan al estar en movimiento el vehículo y llevar un reloj de tiempo real para saber cuando se requiere hacer un incremento por tiempo o por distancia.

Cuando el sistema está contabilizando la señal de velocidad se requiere ir actualizando los datos de distancias recorridas y, dependiendo en que modo de operación se encuentre el sistema de peaje, ir acumulando los incrementos en tarifa, el número de viajes, total acumulado en el día, total acumulado en servicio del vehículo, total acumulado sin pasaje, etc.

V.1.1.1. Diagrama de flujo general

Dado que nuestro programa debe contar con una secuencia en la realización de sus funciones, debemos generar el diagrama de flujo correspondiente a una operación normal del sistema y las posibles acciones que se deberán ejecutar cuando se presente una entrada específica (Teclado, accesorios y/o señal de velocidad).

En la figura V.1.1.1 podemos observar un diagrama de flujo para la operación del sistema.

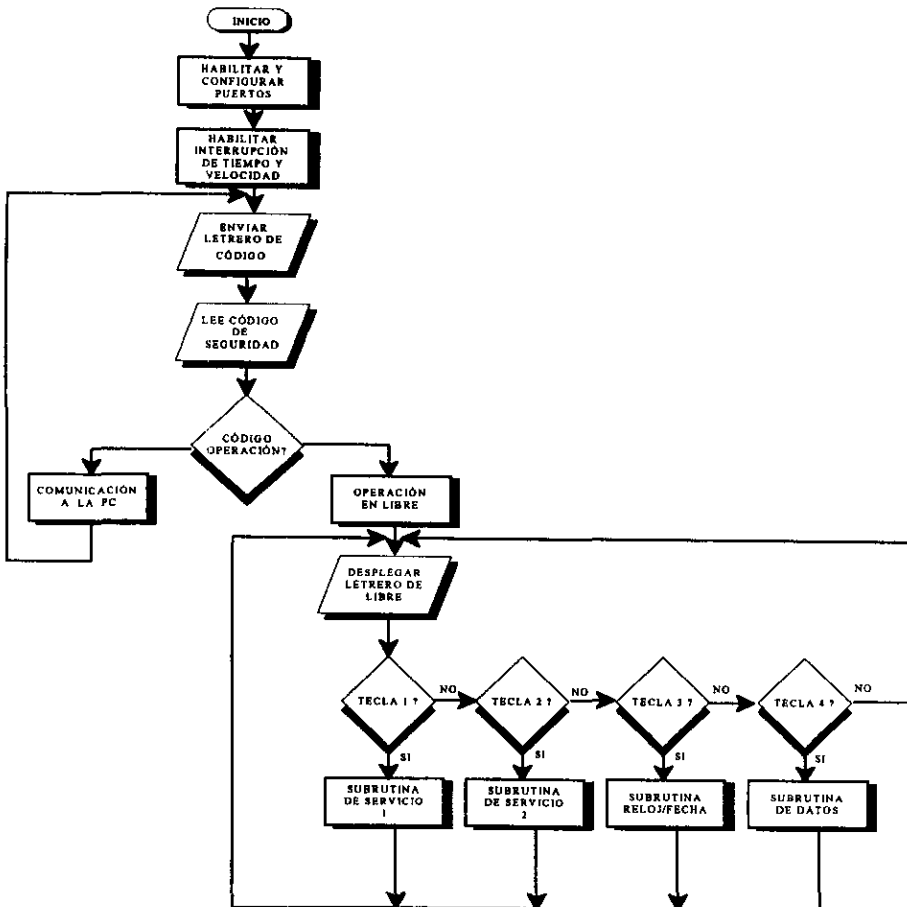


Figura V.1.1.1. Diagrama de flujo de operación inicial.

En el diagrama de flujo anterior primeramente inicializamos todos los puertos digitales

tanto para el uso de los desplegados (líneas de datos y selección de *display*), para el teclado así como para las comunicaciones con los dispositivos externos. Se habilitan las interrupciones de los temporizadores, uno para llevar un reloj de tiempo real y otro para la captura de eventos externos, que en este caso es la señal de velocidad proveniente del transductor magnético

También se inicializan los registros que se usarán temporalmente para guardar los datos que posteriormente se almacenarán en la memoria EEPROM de tipo serial. Al terminar dicha inicialización el sistema se coloca en un estado de lectura de las llaves de acceso. La primera tiene como finalidad el permitir al fabricante el cambio de los parámetros del sistema, si así lo ha requerido el gobierno, es decir, se procede a establecerse una comunicación serial vía RS232 con una estación central en la cual se tiene un software especial par realizar las operaciones antes indicadas. En caso de que no se pida una actualización de los parámetros se procede a realizar la lectura de la segunda llave, la cual tiene el conductor del vehículo y que tiene como fin el lograr la operación normal del sistema , una vez instalado en el vehículo; si el sistema de peaje se retira se tiene que volver a colocar cualesquiera de las dos llaves, la de actualización o la de usuario, dependiendo de la función a realizar.

La siguiente tarea es la de desplegar en los visualizadores la leyenda de LIBRE y durante la cual se van a estar revisando el estado de las teclas. En caso de que se haya presionado una tecla se ejecutará el llamado al subprograma correspondiente para realizar la función ya establecida por dicha tecla.

Aun estando en el modo de operación de LIBRE el sistema tiene que estar cuantificando la velocidad del vehículo y cuantificar, en base a este último parámetro, la distancia recorrida. También durante este modo de operación y durante cualquier otro el sistema de peaje deberá llevar la cuenta del tiempo, es decir, generaremos un reloj de tiempo real mediante software.

Mientras estamos cuantificando la velocidad del vehículo, cuantificamos la distancia recorrida del vehículo, haciendo uso de los parámetros W y K , para cuantificar los siguientes tipos de distancias: 1) la distancia recorrida del vehículo sin pasaje, 2) la distancia recorrida

del vehículo con pasaje, y 3) la distancia recorrida en el último viaje.

Interrupciones de velocidad y tiempo

En la interrupción de tiempo, para generar el reloj de tiempo real, hacemos uso de un temporizador de carrera libre de 12 bits, es decir, dicho temporizador está corriendo a la velocidad de ejecución de un ciclo de máquina (1 microsegundo), y genera una interrupción al microcontrolador cada 4.93 milisegundos. Dado que es un temporizador que está presente en cualquier modo de operación tenemos que llevar la cuenta y saber cuando se ha generado un segundo, un minuto, ó una hora. También es necesario permitirle al usuario el poder modificar la hora del sistema de peaje.

Las rutinas de interrupción las podemos observar en la figura V.1.1.2.

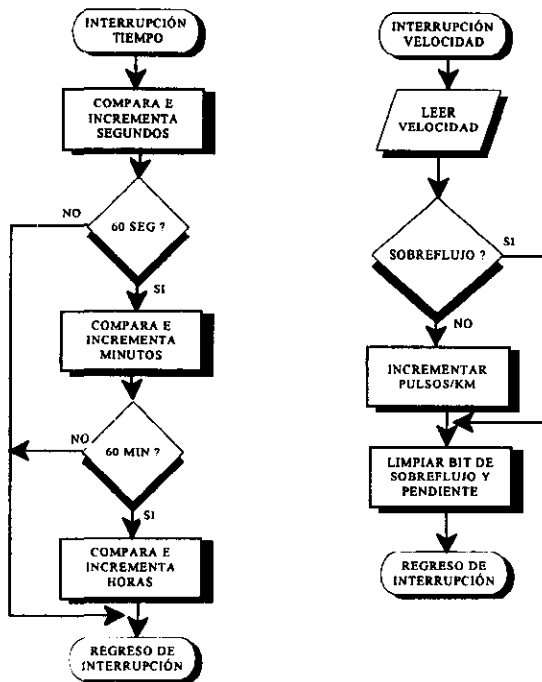


Figura V.1.1.2. Rutinas de interrupción.

En dichas subrutinas de interrupción, la que tiene una prioridad más alta es la de velocidad. Lo primero que hacemos es verificar si se generó un pulso, si es así el microcontrolador generará una petición de interrupción en la dirección indicada del microcontrolador, de aquí el microcontrolador salta a la dirección asignada para esta interrupción. Estando en la rutina de la interrupción lo que hacemos es ir incrementando una variable para llevar la cuenta de los pulsos y el tiempo que duran para posteriormente conocer la velocidad del vehículo.

Ahora después de haber visto las actividades principales que realiza el sistema de peaje, procederemos a describir algunos subprogramas que se realizaron para poder integrarlo.

V.1.2. Diagramas de flujo de subprogramas

Uno de los subprogramas de mayor relevancia es el de desplegado de información, tanto de estado del sistema de peaje como el de visualización de parámetros. Debido a que usamos desplegados de 7 segmentos, tenemos que generar una tabla de caracteres para poder visualizarlos en dichos *display's*. Además de ser una rutina de visualización también generaremos una rutina de barrido para el teclado y detectar cuando se ha presionado alguna tecla.

Para poder realizar esto nos apoyamos en un diagrama de tiempos para sincronizar cuando enviamos la habilitación de un visualizador y al mismo tiempo el de una tecla determinada, y el dato que se quiere visualizar en el desplegado. Aquí señalamos que tenemos cuatro *display's* que nos habilitan a una tecla cada uno, esto es debido a que se necesitaba optimizar al máximo el hardware del sistema, esto nos ayuda que a un mismo tiempo habilitamos un *display* y barremos una tecla predeterminada

En la figura V.1.2.1 podemos apreciar el diagrama de tiempos para la habilitación del

decodificador, así como el reloj y la carga de datos o habilitación de *display*-tecla.

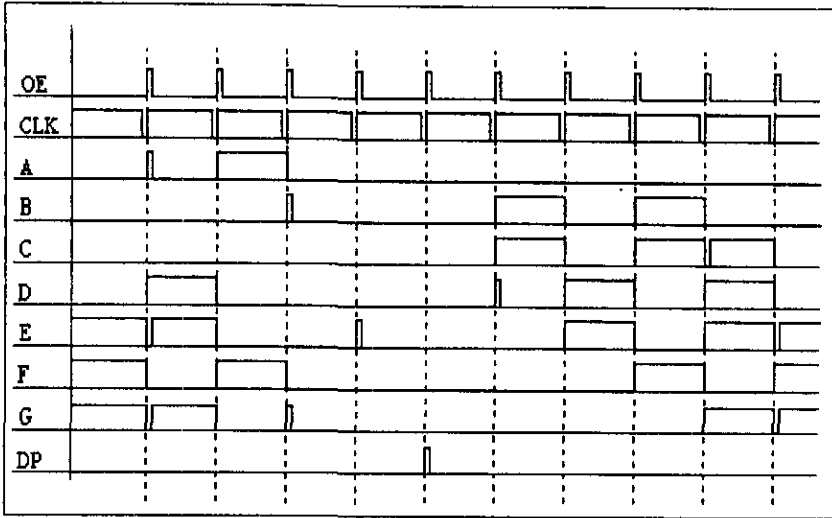


Figura V.1.2.1 Diagrama de tiempos para cargar y visualizar datos.

Del diagrama de tiempos observamos que primero cargamos el dato del *display* a utilizar en el puerto de salida correspondiente y generar el pulso de reloj (CLK) para que el *latch* cargue el dato, posteriormente debemos habilitar la salida del *latch* (*output enable*) para que se habilite el *display* correspondiente. Posteriormente cargamos el dato a visualizar en el puerto, sin generar el reloj para el *latch*. Esto último se tiene que respetar, de lo contrario perderíamos el dato del *display* a encender. De esta forma visualizamos el dato en un *display* a la vez. Para poder visualizar todo un mensaje en los *display's*, tenemos que hacer un barrido de todos los *display's* en poco más de 60 Hz, tomando en cuenta el fenómeno que se observa con un foco incandescente.

Subrutina de desplegado

Basándose en el diagrama de tiempos anterior generamos el diagrama de flujo de la figura V.1.2.2.

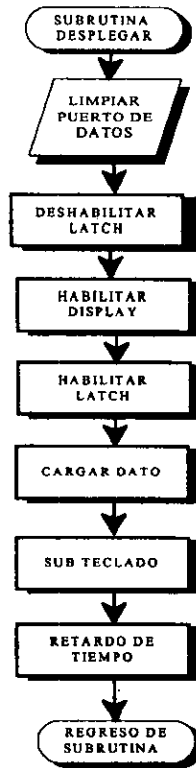


Figura V.1.2.2. Diagrama de flujo para el despliegado.

Este diagrama de flujo solamente se realizó para un solo *display* dado que es repetitivo para los *display's* restantes. En esta subrutina lo primero que realizamos es limpiar el puerto de datos para no visualizar “datos basura” en los *display's*. Posteriormente tenemos que deshabilitar el *latch* y cargar el dato del display a utilizar; para que el *latch* cargue el dato debemos generar un reloj y posteriormente habilitar el *latch*. De esta forma tenemos un display habilitado y solamente falta colocar el dato a desplegar en el puerto de datos, acción que realizamos posteriormente y finalmente hacemos un retardo de tiempo para poder visualizar el dato.

Subrutina de teclado

Mientras estamos visualizando el dato también es posible que estemos habilitando una

tecla, razón por la cual mandamos llamar a una subrutina llamada teclado. En dicha subrutina discernimos si se ha presionado una tecla. Si fue así, prendemos un bit de bandera que nos indica la tecla presionada. Dado que para cada tecla oprimida va a generarse una respuesta del sistema de peaje, debemos también encender un bit de acción. Con esto tenemos dos registros en memoria: el de banderas de teclado y el de acción. Esto lo podemos observar en la figura V.1.2.3

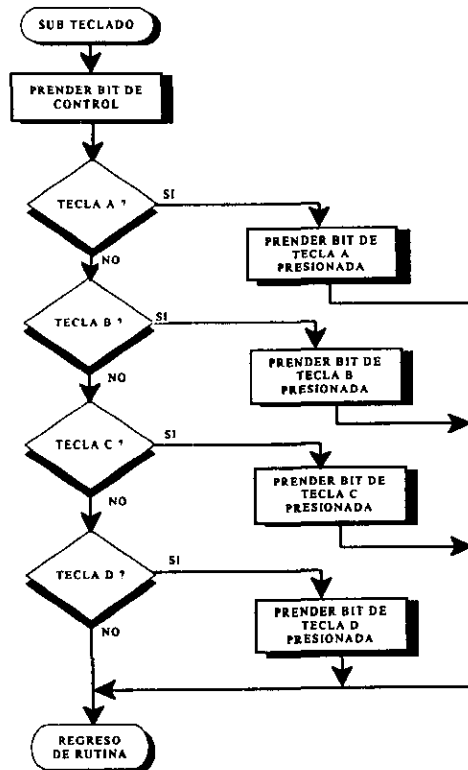


Figura V.1.2.3. Diagrama de flujo para el teclado.

En el diagrama lo primero que hacemos es verificar si se oprimió la tecla, si es así mandamos llamar la subrutina de *buzzer*, esto es para que el usuario sepa que el sistema ha reconocido una acción sobre el mismo; verificamos si se ha soltado la tecla, en cuanto se realiza esto último se procede a prender los bits de bandera correspondientes y se regresa de la rutina donde fue llamada la subrutina de teclado para su posterior decodificación de acción.

Subrutina de *buzzer*

Para la generación de la señal audible del *buzzer* nos basamos en el diagrama de flujo de la figura V.1.2.4.

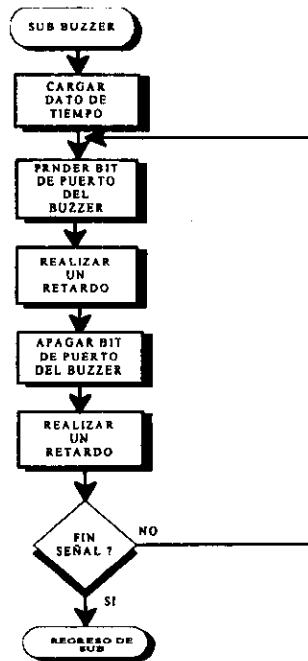


Figura V.1.2.4. Diagrama de flujo para la subrutina de *buzzer*

De este diagrama de flujo observamos que lo primero que hacemos es inicializar el tiempo que dura la señal audible, este tiempo es de aproximadamente 1 segundo, posteriormente tenemos que sacar un nivel de 5 volts en la terminal del puerto al cual está conectado el *buzzer* y esperar un tiempo, después mandamos cero volts en la misma terminal y esperamos un tiempo equivalente al anterior; luego verificamos si ya transcurrió el tiempo de la señal audible, si no es así volvemos a ejecutar las instrucciones anteriores, en caso contrario hacemos un regreso de subrutina.

Es necesario tener un tiempo oscilando de 5 a 0 volts al *buzzer* debido a que el que ocupamos para generar la señal audible no tiene un oscilador integrado. Dicho oscilador nos

ahorraría espacio de programa dado que solamente tendríamos que colocar un nivel de voltaje constante en la terminal a la cual este conectado el *buzzer*. En el mercado encontramos *buzzer's* con oscilador integrado, pero tenían un tamaño demasiado grande, por lo cual optamos el hacer uso de uno sin oscilador.

Subrutina de acceso a memoria

Otros subprogramas que son de vital importancia son los que habilitan, leen, escriben o borran la memoria de datos fijos. Usamos una memoria EEPROM serial que puede almacenar 64 palabras de 16 bits. Para poder realizar las rutinas antes mencionadas debemos saber como opera la memoria y que comandos se le tienen que enviar. La tabla V.1.2.1 nos muestra los comandos válidos para dichas operaciones.

Instrucción	Bit de inicio	Código de operación	Dirección	Datos	Comentarios
WEN	1	00	11XXXX		Habilita modos de programación
ESCRIBIR	1	01	A5-A0	D15-D0	Escribe en la localidad indicada
LEER	1	10	A5-A0		Lee de la localidad indicada
BORRAR	1	11	A5-A0		Borra la localidad indicada
ERALL	1	00	10XXXX		Borra toda la memoria
WRALL	1	00	01XXXX	D15-D0	Escribe en toda la memoria
WDS	1	00	00XXXX		Deshabilita modos de programación

Tabla V.1.2.1. Comandos de operación de la memoria EEPROM serial.

En la tabla anterior observamos que cada vez que deseamos escribir, leer o borrar la memoria, ésta tiene que ser habilitada mediante un comando WEN y al terminar la operación debemos deshabilitar los modo de programación mediante WDS. Esto último es recomendado por el fabricante de la memoria¹. Para poder interactuar con la memoria debemos enviar un bit de inicio, que este caso siempre va a ser uno. También debemos envía el código de la operación a realizar y la dirección de la localidad en memoria en la cual se va a operar. En el caso de la escritura se tienen que tener listos los registros en donde se almacenan los datos a escribir en memoria. A continuación, en la figura V.1.2.5, mostramos los diagramas de flujo para cada uno de los comandos que tiene la memoria.

¹Fuente: National Semiconductor. 1997.COP8 Microcontroller Databook. Application Note.

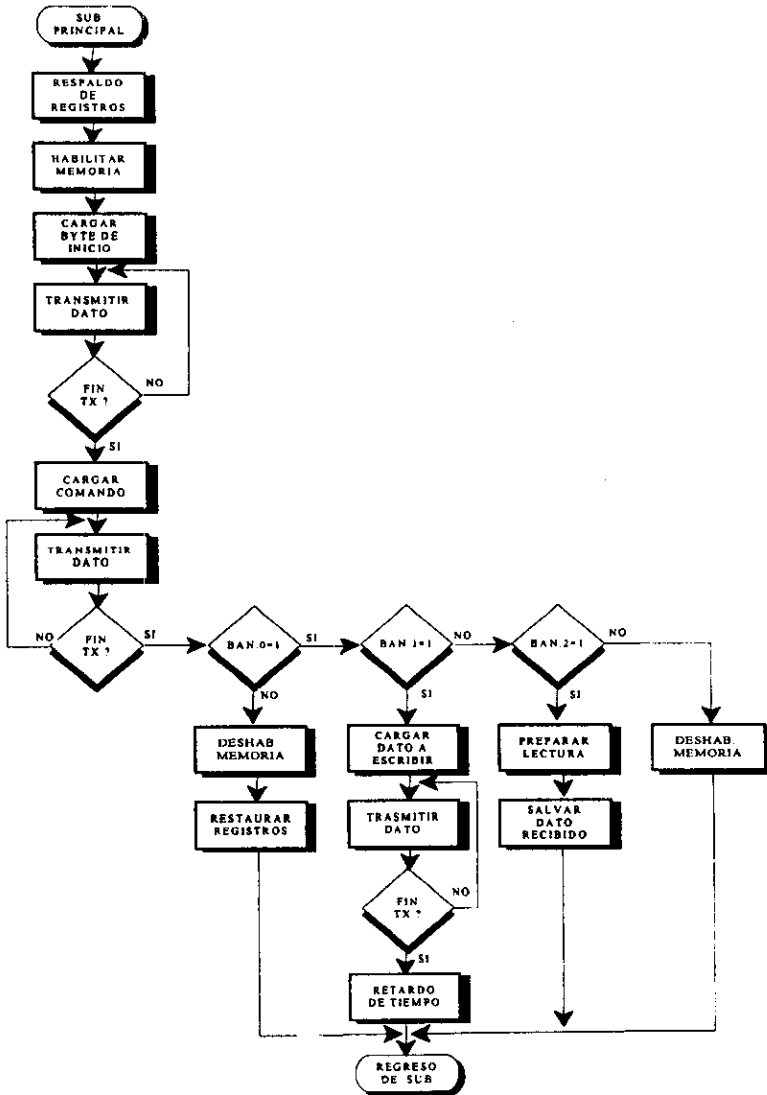


Figura V.1.2.5. Subprograma principal para la memoria serial.

El subprograma del diagrama anterior es usado por los siete comandos que anteriormente se mencionaron. Lo primero que se hace al entrar en la rutina es salvar el contenido de los registros en el *stack* o pila de memoria. Posteriormente cargamos el dato de inicio para comunicarnos con la memoria y trasmitirlo; dado que es una comunicación serial

debemos estar verificando cuando se termina de transmitir el último bit, al terminar cargamos el comando que vayamos a realizar y volvemos a transmitir. Cuando hemos terminado la transmisión verificamos las banderas para saber que es lo que tenemos que realizar en la memoria: leer, escribir o borrarla.

Cada una de las banderas se habilita cuando llamamos a la rutina en particular. Esto lo podemos observar en el diagrama de flujo de la figura V.1.2.6.

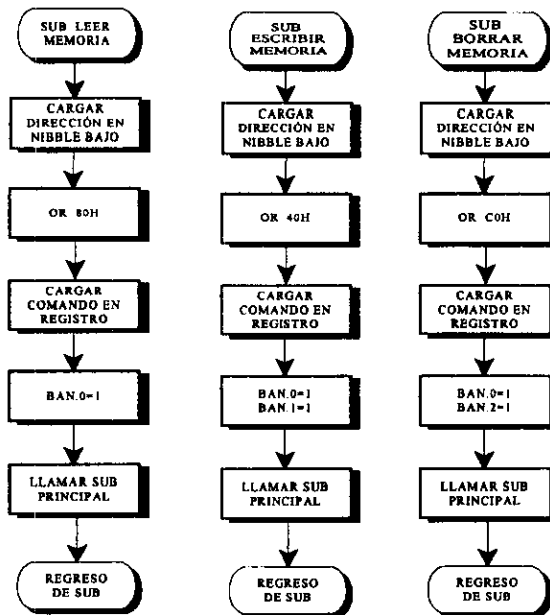


Figura V.1.2.6. Diagramas de flujo para escritura, lectura y borrado de la memoria serial.

Como podemos observar en cada rutina, lo primero que hacemos es cargar la dirección que queremos acceder en los primeros 4 bits o en el *nibble*² bajo; en el *nibble* alto tendremos el comando que deseamos realizar, razón por la que realizamos la operación *OR* con el comando y lo almacenamos en un registro que va a ser usado por la subrutina principal que se describió anteriormente. Por último prendemos las banderas correspondientes para indicar la operación a realizar dentro de la rutina principal.

²*Nibble*: v. *Roer, morder, mordiscar. sust. bocadito*. Se usa para designar un conjunto de 4 bits consecutivos o contiguos en un sistema digital o en software.

Subrutina de transmisión y recepción serial

Por último señalaremos las rutinas de comunicación serial que se desarrollaron para poder realizar un enlace con una computadora personal. El microcontrolador que utilizamos no tiene un puerto serial asíncrono de recepción-transmisión (UART) razón por la que se tuvo que implementar un puerto serial mediante software, haciendo uso de dos terminales de un puerto del μC , uno para recepción y otro para transmisión. Dichos programas nos generan una comunicación *half-duplex*, es decir, podemos enviar y/o recibir. Estas rutinas pueden manejar hasta una velocidad de 9600 bauds, 8 bits de datos, 1 bit de paro, sin paridad.

En la figura V.1.2.7 podemos observar los diagramas de flujo para la recepción y para la transmisión.

La forma en que opera la rutina de recepción serial es la siguiente: primero verificamos la terminal de recepción cada $B/2$ segundos ($415 \mu\text{s}$ para una velocidad de 1200 bauds), para detectar el bit de inicio, donde B es el tiempo de duración de un bit. Posteriormente realizamos la detección del primer bit de dato a los $1.25B$ segundos y los restantes en B segundos. Cada vez que detectamos un bit, vamos salvando ese bit haciendo corrimientos en un registro de recepción de forma repetitiva hasta tener los 8 bits de datos.

La rutina de transmisión serial opera de forma similar, primero cargamos el dato a transmitir en un registro y definimos el número de bits de datos, en nuestro caso es de 8; posteriormente enviamos el bit de inicio y lo mantenemos durante B segundos; luego empezamos a realizar las rotaciones a través del *carry* y verificar en que estado se encuentra, en caso de estar en 1 lógico, colocamos un 1 lógico en la terminal TX en caso contrario colocamos un 0 lógico en la terminal TX. Esto lo repetimos hasta que se complete la transmisión de 8 bits que conforman el byte a transmitir.

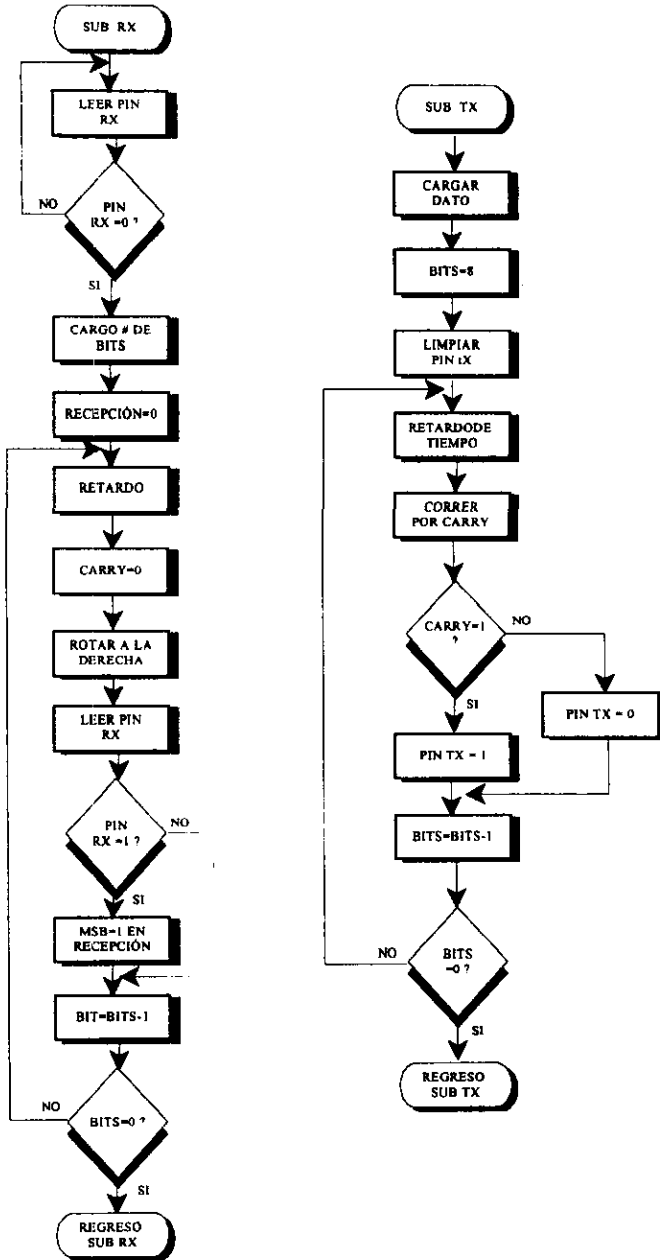


Figura V.1.2.7. Diagramas de flujo para recepción y transmisión serial.

Subrutina de comunicación serial

Dado que estas rutinas requieren de un programa que las mande llamar y que hagan la interacción con la computadora personal, en la figura V.1.2.8 se muestra el diagrama de flujo para realizar la comunicación serial ya de una forma más estructurada con la PC.

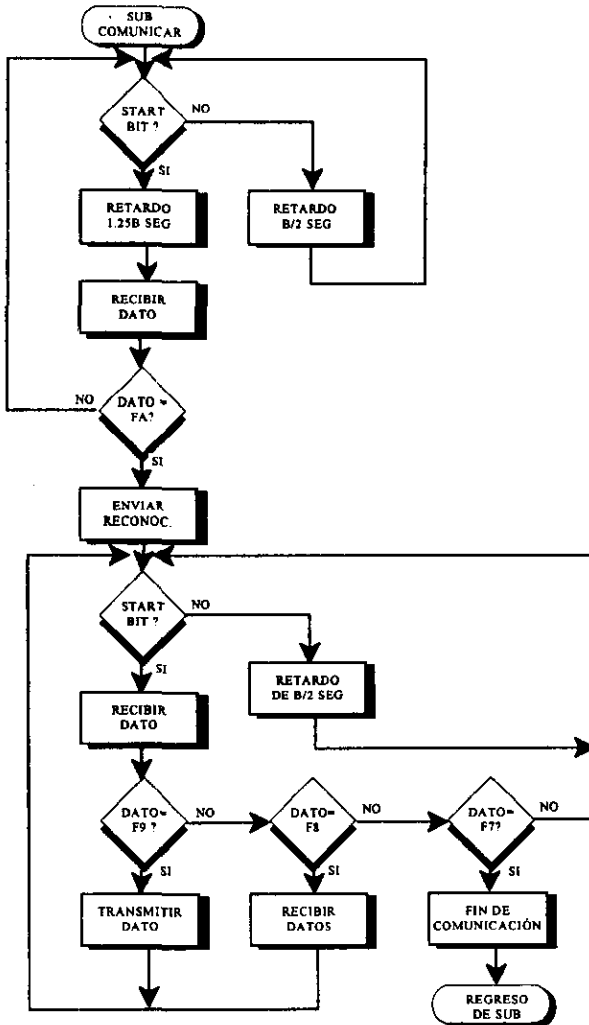


Figura V.1.2.8. Diagrama de flujo para la comunicación serial.

En dicha figura damos ya por hecha las subrutinas para checar la integridad de los datos. Se realiza un *checksum* tanto en la recepción como en la transmisión.

Para iniciar un enlace con la PC se debe introducir la llave del técnico, por lo que el sistema de peaje espera de un bit de inicio en la comunicación serial. En el caso de que se haya detectado dicho bit se genera un retardo de tiempo de aproximadamente $1.25B$ segundos, donde B es el tiempo de duración de un bit, para asegurar que no detectemos en una transición de bajo a alto o viceversa. Posteriormente recibimos el dato y verificamos que sea un dato válido para poder continuar con las proposiciones de condición para realizar la transmisión de datos del sistema de peaje hacia la PC o la recepción de los datos provenientes de la PC. Si el primer dato de comando para hacer el enlace no es válido volvemos al estado de espera de un comando válido.

Para el caso de no detectar un bit de inicio, procedemos a realizar un retardo de tiempo de $B/2$ segundos. El tiempo se escogió de tal forma que no perdamos algún bit y evitar datos falsos. Si queremos hacer una transmisión de datos a una velocidad de 9600 bauds, el tiempo de duración de cada bit sería de 104.16 μ segundos; para no perder algún bit lo que hacemos es estar checando la terminal por donde se realiza la recepción serial en la mitad de ese tiempo, es decir cada 52.08 μ segundos.

El retardo de tiempo de $1.25B$ segundos se realiza después de haber detectado el bit de inicio, esto es, para asegurar la lectura correcta de cada bit de datos recibido. En última instancia se envía un comando de liberación o finalización de la comunicación serial e inmediatamente el sistema de peaje regresa al modo de seguridad, esto significa que el sistema de peaje está listo para colocarlo en su modo de operación normal mediante la llave de usuario.

Debido a que nos llevaría demasiado el explicar todas y cada una de las subrutinas que se hacen uso para el control del sistema de peaje, solamente hemos mencionado las de mayor relevancia a consideración nuestra, por lo que procederemos a describir el software diseñado para la computadora personal.

V.2. Desarrollo del software para la computadora

Para tener acceso y poder actualizar o modificar los parámetros del sistema de peaje, ya sea debido a que se realizan revistas, incrementos de tarifas autorizadas por el gobierno, etc., se desarrolló un programa que pudiera darnos toda la información necesaria sin tener que abrir el sistema de peaje. Esto lo realizamos mediante una comunicación serial con una computadora personal. En dicha comunicación la computadora envía o recibe los datos para, o del sistema de peaje.

Para desarrollar el *software* correspondiente se analizaron diferentes tipos de lenguajes, a continuación presentamos una breve comparación entre los lenguajes existentes en el mercado.

V.2.1 Selección del lenguaje de programación para el desarrollo

Para poder realizar el programa de adquisición de datos para el sistema de peaje, es necesario hacer uso de un lenguaje de alto nivel, que nos permite hacer el acceso a los parámetros del sistema mediante los puertos seriales de la computadora. Para poder realizar esto último tenemos los siguientes lenguajes de programación:

- Quick Basic

BASIC (*Beginner's All-purpose Symbolic Instruction Code*) ha sido durante muchos años uno de los lenguajes de programación en el que la gran mayoría de los programadores aprendió a interactuar con la computadora. BASIC es un lenguaje muy fácil de aprender, no tiene un conjunto de instrucciones muy amplio pero para aplicaciones sencillas es suficiente.

Quick Basic es una versión mejorada de BASIC, fue diseñado para personas que se estuvieran introduciendo en el uso de lenguajes de programación, debido a que nos ofrece un

lenguaje estructurado y poderoso. Una de las ventajas que tiene este lenguaje es el de poder editar, compilar, ejecutar y depurar programas dentro de un solo ambiente de trabajo. Todas las herramientas que el programa requiera para poder desarrollar sus sistema están contenidas en un solo paquete integral.

Algunas de las desventajas que tienen estos lenguajes incluyen la baja velocidad de ejecución. Esto se debe a que la gran mayoría de las versiones originales se escribieron como *intérpretes*, es decir, conforme se va leyendo el programa, la computadora va traduciendo línea a línea, a diferencia de un compilador, el cual lee todo el programa antes de ejecutarlo.

- Turbo Pascal

Turbo Pascal es uno de los lenguajes que más se utilizó en la década de los 80's. Este lenguaje de programación es totalmente estructurado y nos permite un ambiente de edición y compilación muy eficiente y rápido. También nos muestra mejoras para el manejo de puertos y Acceso Directo a Memoria (*DMA*). Además, nos permite hacer uso de un ambiente gráfico si así lo desea desarrollar el programador.

La *interface* que se utiliza en Turbo Pascal es mediante menús con diálogos, botones y otras características gráficas que permiten una interacción más amigable. Una de las desventajas que se pueden presentar al usar este lenguaje es el tiempo de aprendizaje del mismo, ya que es mayor al de aprender BASIC.

- C/C++

Este lenguaje fue creado en 1972 en los Laboratorios Bell. Dicho lenguaje fue desarrollado para facilitar el desarrollo del sistema operativo UNIX. Debido a su flexibilidad y poder, se extendió su uso rápidamente para crear diferentes programas.

Dentro de las ventajas que tiene C para ser usado es su portabilidad, permite una programación estructurada, tiene un manejo de apuntadores poderoso, se puede hacer uso de datos tipificados, de funciones para el manejo de periféricos, tales como , puertos, vídeo, DMA, etc.

La única desventaja que se tiene con este lenguaje es el tiempo de aprendizaje. Debido a su gran capacidad, es considerado un lenguaje de medio nivel, por lo cual su aprendizaje se hace lento y árido en relación a los otros lenguajes de programación ya mencionados.

- Visual Basic

Este lenguaje está disponible para el ambiente Windows, y es la herramienta más rápida para crear aplicaciones generales. El tiempo de aprendizaje es breve, requiere pocos conocimientos de programación y brinda una gran facilidad para una rápida integración y reuso de componentes.

Sus principales ventajas son:

- Controles visuales preconstruidos por terceros (controles Visual Basic eXtention (VBX)).
- Permite ensamblar fácil y rápidamente una *interface* de usuario con componentes prefabricadas.
- Ofrece gran capacidad y velocidad.
- La *interface* de programación es muy amigable.
- El tiempo de aprendizaje es corto.
- Programación orientada a eventos.

Sus principales desventajas son:

- El lenguaje no es muy flexible dado que requiere de otros programas para poder acceder puertos o en su defecto tener las librerías para poder usarlos.
 - El manejo de datos a nivel de bits es muy complicado.
 - Requiere Windows
-
- Visual C++

Este lenguaje, al igual que Visual Basic, se basa en una programación bajo ambiente Windows, lo que permite una mayor amigabilidad con el usuario. Además constituye una herramienta flexible y poderosa para generar soluciones, además de ser rápida para programadores de C/C++.

Sus principales ventajas son:

- Incluye librerías preconstruidas.
- Tiene controles visuales.
- Maneja controles VBX.
- Se pueden programar sistemas multitarea.
- Uso eficiente de variables locales y globales.
- Total libertad en el manejo de memoria y recursos de sistema
- Permite una programación orientada a objetos y orientada a eventos.

Al igual que C/C++ el tiempo de aprendizaje es lento, aún cuando se cuenta con herramientas muy flexibles para su aprendizaje.

Después de haber visto algunas características de los lenguajes de programación decidimos realizar el programa en Visual Basic 4.0 Professional. Esta elección constituye una

solución muy amigable para el usuario debido a que podemos generar pantallas bajo ambiente Windows que son atractivas para el usuario y el tiempo de aprendizaje es muy corto además de que se tiene experiencia previa en el desarrollo de programas para la adquisición de datos sistemas en este lenguaje de programación.

V.2.2. Desarrollo del software

La elaboración del programa consiste en crear ventanas donde se incluyen elementos de control y despliegue, como menús, botones, gráficas, tablas, etc. Una vez construidas las ventanas se establecen las relaciones entre los distintos controles y los demás objetos, cuyos parámetros se verán afectados por las acciones del usuario.

La primera ventana es la de acceso al sistema, tal como se muestra en la figura V.2.2.1 dicha ventana tiene dos cajas de texto en las cuales solicitamos una identificación y una contraseña.

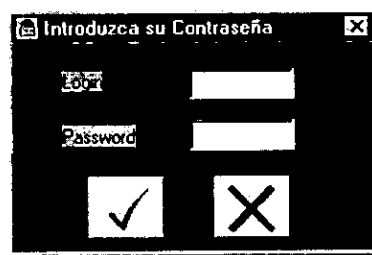


Figura V.2.2.1 Ventana de acceso.

El control acceso al sistema es necesario debido a que no cualquier persona podrá realizar los cambios en los parámetros del sistema de peaje, esto es para evitar cualquier tipo de fraude al no colocar los parámetros correctos. El sistema tiene dos personas con contraseñas maestras, es decir estas dos personas tiene acceso al sistema desde su instalación, la función que tienen estas dos claves es ayudar a la persona encargada de instalar el programa y verificar de su buen funcionamiento, en caso de no tenerlas no se podría acceder al sistema para poder

generar nuevos usuarios. Una vez generado un usuario como administrador se borrarán las dos claves maestras teniendo acceso exclusivamente el administrador y las personas dadas de alta por él.

Una vez introducido el *login* y el *password* el usuario tendrá que hacer presión en el botón con la “palomita”. La ventana siguiente es la que se muestra en la figura V.2.2.2.

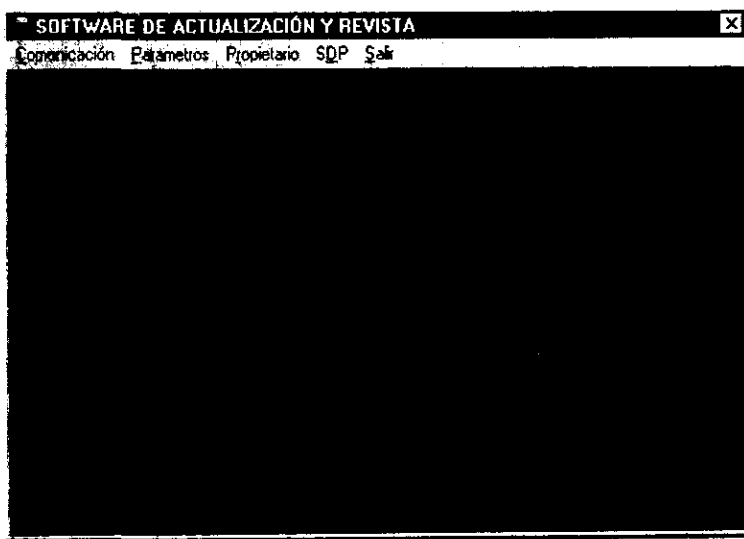


Figura V.2.2.2. Ventana principal del sistema.

Esta es la ventana principal del sistema. A través de la barra de menú de esta ventana tenemos acceso a la selección de Comunicación, con una velocidad de transmisión fija de 1200 *bauds*; a la opción Parámetros, en la cual definimos los parámetros que deben de ir en el sistema de peaje y que serán alterados solamente por personal autorizado, es decir el técnico no tendrá acceso a modificarlos, solamente podrá leerlos; la opción Propietario, nos muestra los datos de la persona que es propietaria del sistema de peaje; la opción SDP, nos muestra la ventana para poder acceder a los datos del sistema de peaje o por el contrario guardar en el sistema de peaje los nuevos datos y por último la opción Salir nos permite finalizar una sesión con el sistema.

Cuando seleccionamos la opción de Comunicación la ventana que nos aparece es la que se muestra en la figura V.2.2.3.



Figura V.2.2.3 Ventana de selección de puerto de comunicación.

En dicha ventana le indicamos al sistema por que puerto se va a realizar un enlace con el sistema de peaje. Una vez que se haya seleccionado el puerto y hacer presión en el botón de enlazar, el sistema nos mostrará un mensaje de enlace correcto, en caso de que se haya realizado con éxito el enlace, si no es así nos aparecerá otra ventana indicando la falla en la transmisión, tal como se muestra en la figura V.2.2.4.

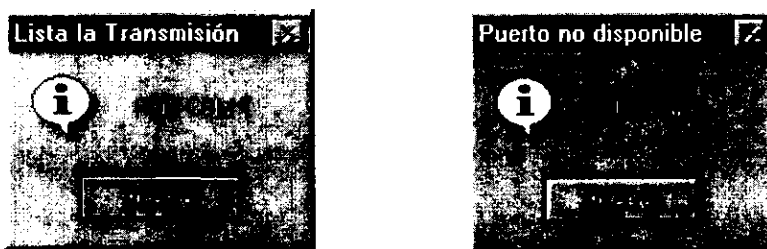


Figura V.2.2.4. Ventanas que indican un enlace correcto o incorrecto.

Una de las ventanas más importantes en el desarrollo de este software fue la de Parámetros.

Dicha ventana se muestra en la figura V.2.2.5, como podemos observar en dicha figura tenemos una caja de *Datos Recibidos* en la cual observamos los datos recibidos del sistema de peaje más el *checksum* de dichos datos. Tenemos también 3 botones de comando: *LEER*, para hacer la lectura de los parámetros; *PROBAR*, nos permite realizar un enlace de prueba; *LIMPIAR*, para limpiar la forma o ventana de los datos no necesarios.

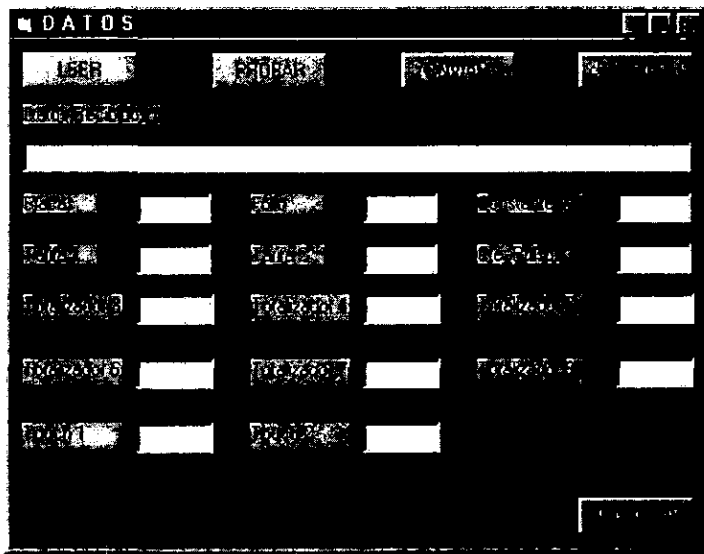


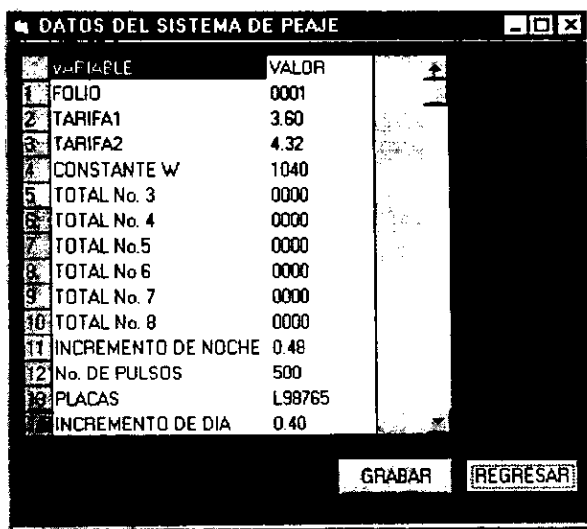
Figura V.2.2.5. Ventana para la lectura de datos del sistema de peaje.

En esta ventana también observamos una serie de totalizadores y parámetros propios del sistema, constante *W* constante *K*, los números de las llaves para usuario y para técnico, etc.

En dicha ventana solamente podemos leer los parámetros que se encuentran en el sistema de peaje en revista.

Cabe señalar que se recibe un mensaje de identificación en la parte inferior izquierda de la ventana cada vez que se coloca el puntero del *ratón* sobre alguna caja de diálogo. Para el caso de la ventana se muestra el identificador para la caja de *Datos Recibidos*.

Ahora bien, en la figura V.2.2.6 observamos la ventana para poder modificar o no los parámetros que se van a guardar en el sistema de peaje.



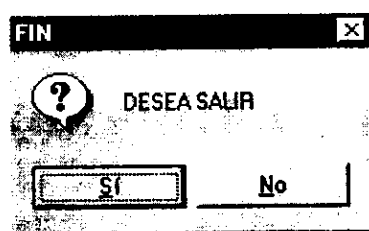
The screenshot shows a window titled "DATOS DEL SISTEMA DE PEAJE" with a table of variables and their values. The table has two columns: "VARIABLE" and "VALOR". The variables listed are: FOLIO (0001), TARIFA1 (3.60), TARIFA2 (4.32), CONSTANTE W (1040), TOTAL No. 3 (0000), TOTAL No. 4 (0000), TOTAL No. 5 (0000), TOTAL No. 6 (0000), TOTAL No. 7 (0000), TOTAL No. 8 (0000), INCREMENTO DE NOCHE (0.48), No. DE PULSOS (500), PLACAS (L98765), and INCREMENTO DE DIA (0.40). At the bottom of the window, there are two buttons: "GRABAR" and "REGRESAR".

VARIABLE	VALOR
1 FOLIO	0001
2 TARIFA1	3.60
3 TARIFA2	4.32
4 CONSTANTE W	1040
5 TOTAL No. 3	0000
6 TOTAL No. 4	0000
7 TOTAL No. 5	0000
8 TOTAL No. 6	0000
9 TOTAL No. 7	0000
10 TOTAL No. 8	0000
11 INCREMENTO DE NOCHE	0.48
12 No. DE PULSOS	500
13 PLACAS	L98765
14 INCREMENTO DE DIA	0.40

Figura V.2.2.6. Ventana para cargar y grabar datos al sistema de peaje.

Los datos se encuentran ordenados en un arreglo "grid" de Visual Basic. Cuando nosotros vamos a grabar los datos se hace un procesamiento de la información para poder guardarla en una forma adecuada y que el sistema de peaje pueda interpretar

Para finalizar el presente capítulo mostramos en la figura V.2.2.7 la ventana para salir del sistema de actualización para sistemas de peaje.



The screenshot shows a dialog box titled "FIN" with a question mark icon. The text inside the dialog box says "DESEA SALIR". Below the text are two buttons: "SI" and "No".

Figura V.2.2.7. Ventana de salida del sistema de actualización.

Los requisitos para el software antes mencionado son los siguientes:

- Una computadora con un procesador 486 o superior.
- Windows 95.
- 16 Mbytes en memoria RAM.
- 10 Mbytes de espacio libre en disco duro.
- 1 puerto serial disponible.

Al haber realizado el diseño y la programación del sistema procedemos a un último capítulo en el cual damos toda la serie de pruebas y procedimientos a los que se sometió el prototipo de dicho sistema y los resultados a los que se llegaron y cuales serían las expectativas a futuro para este prototipo.

CAPÍTULO VI

RESULTADOS Y CONCLUSIONES

En el presente y último capítulo mostraremos los resultados que se obtuvieron durante el desarrollo del sistema de peaje, así como las conclusiones y sugerencias para nuevos prototipos que se lleguen a desarrollar con la empresa PADI.

Resultados

Para alcanzar los objetivos planteados al inicio del presente trabajo se desarrollaron diferentes etapas. En la primera etapa se realizó una investigación de los diferentes sistemas de peaje que hay actualmente en la Ciudad de México. En dicha investigación no se logró recabar suficiente información debido a que ésta solamente la tienen los fabricantes de dichos sistemas y no está disponible. Por esta razón el presente trabajo se desarrolló a partir de la norma oficial para dichos sistemas.

La segunda etapa consistió en el desarrollo del hardware del sistema y la selección de los dispositivos para el mismo. Al tener una propuesta de desarrollo viable se procedió a la construcción del prototipo de pruebas y a su programación.

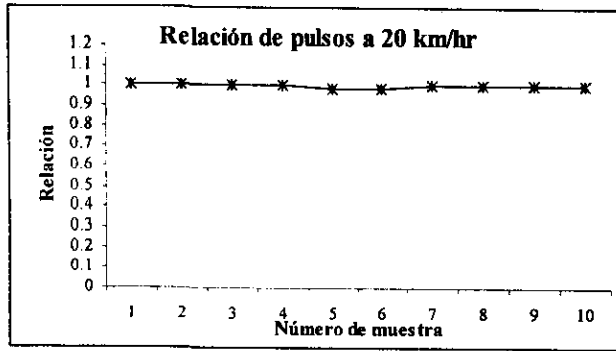
Una vez terminado el prototipo del sistema de peaje se procedió a realizar una fase de pruebas de laboratorio. En dichas pruebas se comprobó el funcionamiento del sistema y que cumpliera la norma ya establecida.

Dentro de estas pruebas la de mayor relevancia es la de cuantificación de la velocidad del vehículo haciendo uso de las constantes K y W. La norma nos señala que no debemos sobrepasar un error del 1%. En la tabla VI.1 podemos apreciar los resultados obtenidos de dichas pruebas.

Velocidad = 20 km/hr		f = 23.11 Hz		
Muestra	Testigo	Prototipo	% Error	Relación
1	99	99	0.00000	1
2	99	99	0.00000	1
3	104	104	0.00000	1
4	98	98	0.00000	1
5	104	103	0.96154	0.99038462
6	102	101	0.98039	0.99019608
7	104	104	0.00000	1
8	105	105	0.00000	1
9	100	100	0.00000	1
10	103	103	0.00000	1
Promedio	101.8	101.6	0.19419	0.99805807

Tabla VI.1. Resultado de las pruebas en el laboratorio.

En la tabla anterior tenemos las mediciones hechas a 20 km/hr. Tenemos un testigo para verificar el número de pulsos enviados al sistema de peaje y la lectura correspondiente del sistema de peaje. Dicho testigo es un generador de onda cuadrada de frecuencia variable con una exactitud de 0.5%, tal como lo señala la norma; este generador también cuenta con un contador de pulsos generados. Como podemos observar el error es casi siempre cero, por lo que supera a lo solicitado en la norma. Si observamos el promedio de error podemos darnos cuenta que es mucho menor a lo solicitado. En velocidades menores a 20 km/hr tenemos un error nulo, esto es debido al uso de interrupciones para la contabilización de pulsos. En la gráfica VI.1 tenemos la relación entre los pulsos enviados y los pulsos recibidos por el sistema de peaje.



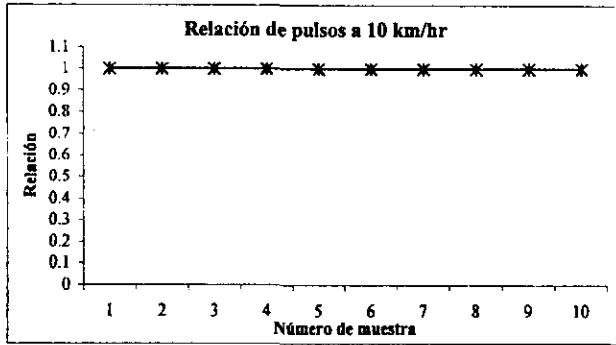
Gráfica VI.1. Relación de pulsos enviados y pulsos recibidos.

Como podemos observar en la gráfica, la relación es casi unitaria, idealmente debería ser unitaria. Con esto comprobamos que nuestro sistema está respondiendo con una gran precisión y que cumple con la norma.

Pruebas como esta se hicieron a diferentes velocidades teniendo casi siempre el mismo porcentaje de error, tal como se puede observar en las tablas VI.2 y VI.3; y en las gráficas VI.2. y VI.3. Para velocidades grandes, por ejemplo 130 km/hr, el sistema tiene un error pequeño.

Velocidad = 10 km/hr		f = 11.55 Hz		
Muestra	Testigo	Prototipo	% Error	Relación
1	102	102	0.0	1
2	103	103	0.0	1
3	104	104	0.0	1
4	102	102	0.0	1
5	102	102	0.0	1
6	102	102	0.0	1
7	102	102	0.0	1
8	101	101	0.0	1
9	104	104	0.0	1
10	102	102	0.0	1
Promedio	102.4	102.4	0.0	1

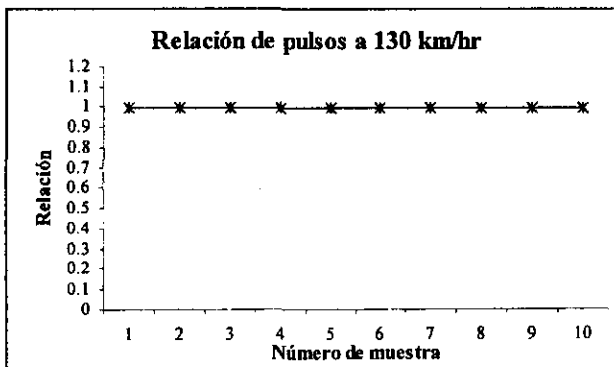
Tabla VI.2. Resultados de pruebas de laboratorio.



Gráfica VI.2. Relación de pulsos enviados y recibidos.

Velocidad =130 km/hr		f = 150.22 Hz		
Muestra	Testigo	Prototipo	% Error	Relación
1	310	308	0.64516	0.99354839
2	316	316	0.00000	1
3	326	326	0.00000	1
4	305	305	0.00000	1
5	293	293	0.00000	1
6	310	310	0.00000	1
7	324	323	0.30864	0.99691358
8	314	314	0.00000	1
9	309	309	0.00000	1
10	318	318	0.00000	1
Promedio	312.5	312.2	0.09538	0.9990462

Tabla VI.3. Resultados de pruebas de laboratorio.



Gráfica VI.3. Relación de pulsos enviados y recibidos.

Las pruebas de velocidad que se muestran anteriormente son las únicas que pudimos desarrollar en el laboratorio; las pruebas restantes como lo son: de humedad, temperatura, etc., serán realizadas en los laboratorios de SECOFI.

Otra prueba que se realizó es la de comunicación serial con la PC y verificar que todos y cada uno de los datos fueran correctos. La comunicación tiene una velocidad de transmisión de 1200 bauds, actualmente no se requieren velocidades mayores por que son muy pocos datos los que se tienen que ir actualizando.

Al sistema de peaje se le conectó una impresora, para dicha impresora se desarrolló el controlador y se probó a nivel de laboratorio en conjunto con el sistema de peaje, dando buenos resultados, es decir, la impresora podía establecer comunicación con el sistema e intercambiar información para la impresión de los montos a pagar por el servicio prestado.

Otro de los objetivos planteados al inicio de la tesis fue el de tener un sistema de seguridad en el sistema de peaje. Dicho sistema de seguridad consistió de dos llaves electrónicas para acceder al sistema de peaje. Con dichas llaves se probó que la detección fuera la correcta y que no se le pudiera introducir algún objeto extraño al contacto y que diera acceso al sistema. Esto funcionó de muy buena forma debido a los códigos únicos que tiene cada llave.

Conclusiones

Para finalizar el presente trabajo y con base en los resultados obtenidos podemos concluir lo siguiente:

Se ha desarrollado un prototipo de sistema de peaje que cumple con las características solicitadas por la empresa que lo va a comercializar. Entre estas características tenemos:

-
- Un sistema que cuantifica la velocidad del vehículo con un error menor al 1%, que es el máximo permitido por la norma.
 - Cuenta con una memoria para guardar datos históricos, como lo son: total acumulado de cobros día tras día, total acumulado de pasajeros, total de kilómetros recorridos por el vehículo con pasaje y sin pasaje, total acumulado de incrementos en la tarifa y verificación de recepción de la señal de velocidad.
 - Cuenta con la capacidad de poder conectarle una impresora para la elaboración de recibos, si es que así lo desea el usuario. También se desarrolló el control para la impresora, dicho control contiene un microcontrolador y recibe toda la información que se quiere imprimir, como lo es: placas del vehículo, fecha, hora, costo del servicio, extras y total a pagar.
 - Tiene capacidad para poder conectar otros dispositivos haciendo uso del *bus* de comunicaciones que contiene el microcontrolador, teniendo como límite el número de dispositivos externos, el número de líneas disponibles para su habilitación.
 - Contiene un reloj de tiempo real con el cual realizamos los incrementos de tiempo y se despliega un reloj.
 - Contiene un sistema de inviolabilidad que consta de dos llaves electrónicas: una para el usuario y otra para el técnico. Una de las llaves es utilizada para poner en operación normal al sistema y la otra es para realizar los cambios de tarifas y/o los parámetros del sistema de peaje por medio de una PC.
 - Por último, tiene la capacidad para comunicarse con un PC para poder hacer los cambios en los parámetros del sistema de peaje.

Un mérito que tiene este proyecto, es el de haberse realizado solamente con los requisitos que indica la norma ya que no se tuvo la suficiente información necesaria de este tipo de sistemas para iniciar su desarrollo.

Por último, el desarrollo del software para la comunicación con el sistema de peaje con una PC funciona de manera eficiente al poder recibir y enviar los datos sin errores. Dentro de este programa se hace el manejo de la información.

Dadas las características técnicas del prototipo desarrollado podemos concluir que el haber trabajado en este proyecto nos facilitó la vinculación que debe haber con la empresa y darnos cuenta que se tiene la capacidad para desarrollar cualquier producto con tecnología propia.

El haber desarrollado este proyecto nos capacitó para enfrentarnos a los problemas reales que tienen los empresarios y como resolverlos, tanto en un aspecto técnico y legal, ya que el desarrollo de un prototipo no solamente requiere una habilidad técnica para su creación sino que también requiere conocer aspectos legales. Para poder presentar un producto a nivel nacional e internacional.

- *Análisis y Diseño de Circuitos Integrados Digitales*
D.A.Hodges, H.G.Jackson
Colección Ciencia Electrónica
Editorial Gustavo Gili,S.A.
Barcelona, 1988.
- *Aprendiendo Visual C ++ 2 en 21 días*
Namir Clement Shanmas
Tercera Edición
Prentice Hall, 1996.
- *Circuitos Digitales y Microprocesadores*
Herbert Taub
Primera Edición
McGraw-Hill, 1983.
- *Circuitos Integrados Lineales y Amplificadores Operacionales*
Robert F. Coughlin y Frederick F. Driscoll
Segunda Edición
Prentice Hall, Marzo de 1987.
- *Comunicaciones y Redes de Procesamiento de Datos*
Néstor González Sainz
McGraw-Hill
Colombia, 1987
- *COP8™ Microcontroller Databook*
National Semiconductor
1997 Edition.
- *Diccionario Enciclopédica de Términos Técnicos*
Javier L. Collazo
McGraw-Hill, 1981.

- *Electronic Fabrication*
Gordon Shimizu
Second Edition
Delmar Publishers Inc.
- *Enciclopedia de Microsoft® Visual Basic 4*
Francisco Javier Ceballos
Alfaomega, 1997
- *Estructura y Desarrollo del sector transporte en México*
Víctor Islas Rivera
El Colegio de México
Primera Edición, México, 1990.
- *Guía Para Mediciones Electrónicas y Prácticas de Laboratorio*
Stanley Wolf, Richard F.M. Smith
Segunda Edición. Prentice Hall.
- *Handbook of Transducers*
Harry N. Norton
Prentice Hall, 1989.
- *Interferencias Electromagnéticas en Sistemas Electrónicos*
Josep Balcells, Rafael Esparza, Frances Daura y Ramón Pallas
Alfaomega, 1992.
- *Lógica Digital y Diseño de Computadores*
M. Morris Mano
Prentice Hall.
- *Manual de Visual Basic 4 Para Windows 95*
Gary Cornell
McGraw-Hill, 1996

- *Métodos Experimentales Para Ingenieros*
Jack P. Holman
Segunda Edición.
McGraw-Hill, 1990

- *MC68HC11 Reference Manual*
Motorola Inc. , 1991

- *Microprocessors and Interfacing Programming and Hardware*
Douglas V. Hall
Prentice Hall, 1987.

- *Microcontrollers 8-bits*
Signetics, 1991.

- *Microprocessor Interfacing Techniques*
Austin Lesea and Rodney Zans
Third Edition
Sybex, 1974.

- *Programación de Bases de Datos con Visual Basic*
Alfons González Pérez
Alfaomega, 1997.

- *Programación en Turbo Pascal Versión 4.0, 5.0 y 5.5*
Luis Joyanes Aguilar
McGraw-Hill, 1990.

- *Semiconductor Technical Data*
Motorola Inc. , 1991

- *Visual Basic for Windows Developer' s Guide*
D.F Scott
Sams Publishing
USA. 1993

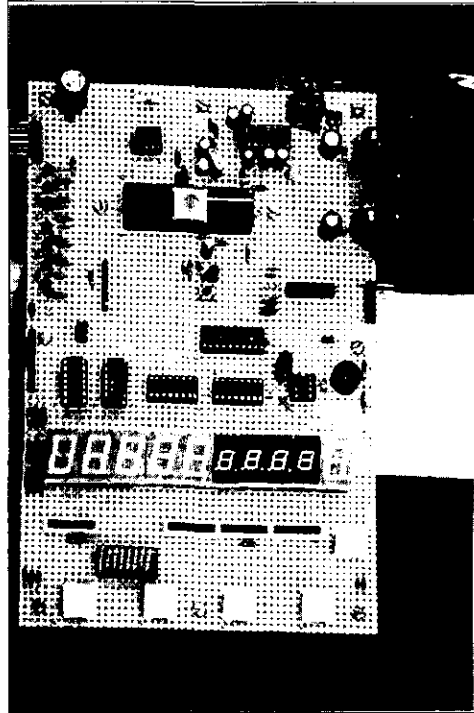


Figura A.1. Primer prototipo de pruebas desarrollado.

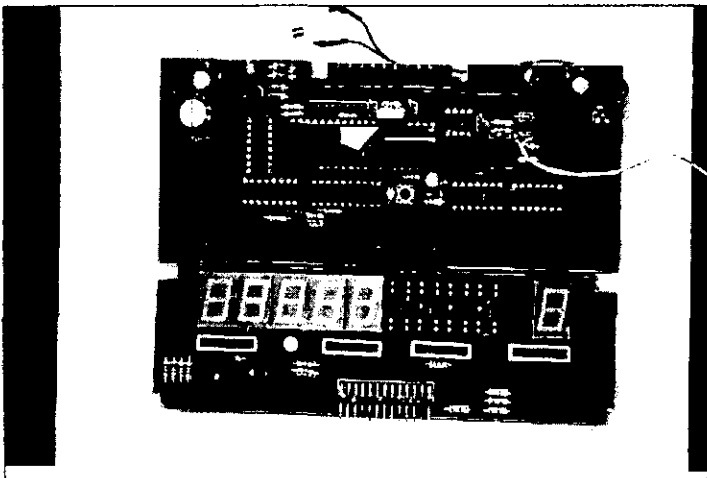


Figura A.2. Prototipo en tarjeta de circuito impreso.

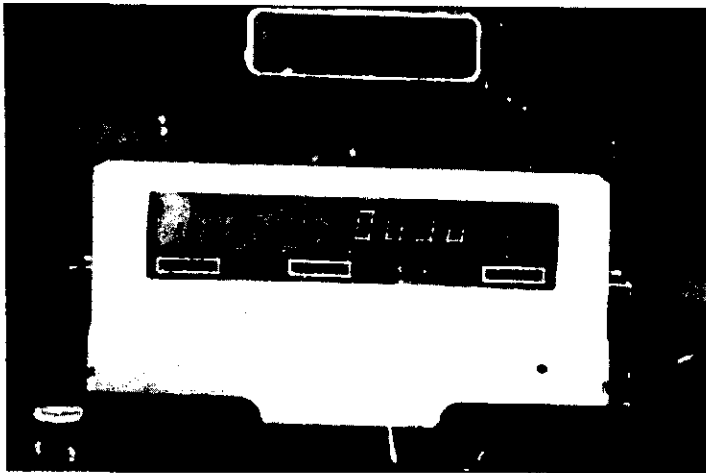


Figura A.3. Prototipo final en espera de la llave de acceso.

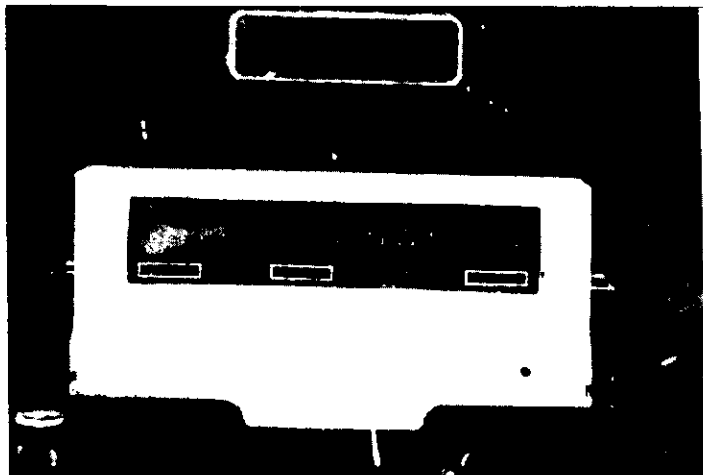


Figura A.4. Prototipo en operación normal.

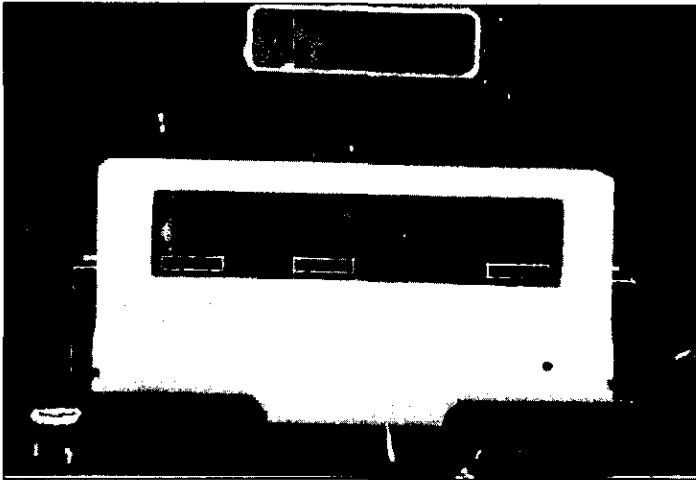


Figura A.5. Prototipo en comunicación con una PC.

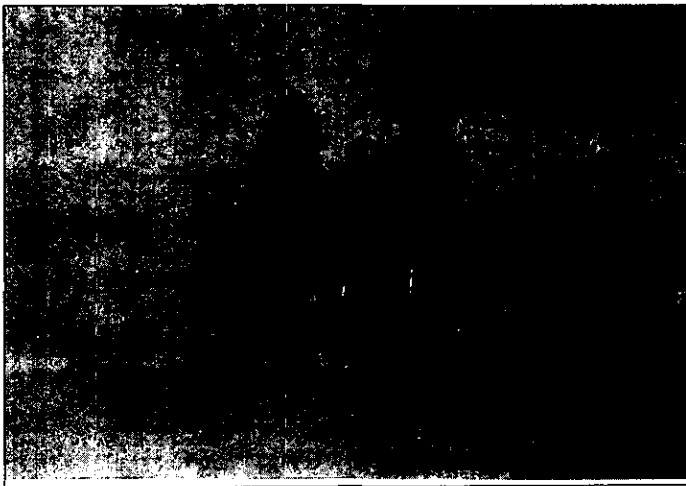
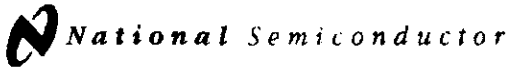


Figura A.6. Llaves electrónicas de acceso.

APÉNDICE B
HOJAS DE ESPECIFICACIONES



February 1995

LM2935 Low Dropout Dual Regulator

General Description

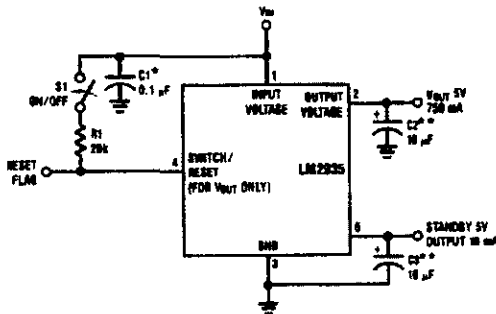
The LM2935 dual 5V regulator provides a 750 mA output as well as a 10 mA standby output. It features a low quiescent current of 3 mA or less when supplying 10 mA loads from the 5V standby regulator output. This unique characteristic and the extremely low input-output differential required for proper regulation (0.55V for output currents of 10 mA) make the LM2935 the ideal regulator for power systems that include standby memory. Applications include microprocessor power supplies demanding as much as 750 mA of output current.

Designed for automotive applications, the LM2935 and all regulated circuitry are protected from reverse battery installations or 2 battery jumps. During line transients, such as a load dump (60V) when the input voltage to the regulator can momentarily exceed the specified maximum operating voltage, the 0.75A regulator will automatically shut down to protect both internal circuits and the load while the standby regulator will continue to power any standby load. The LM2935 cannot be harmed by temporary mirror-image insertion. Familiar regulator features such as short circuit and thermal overload protection are also provided.

Features

- Two 5V regulated outputs
- Output current in excess of 750 mA
- Low quiescent current standby regulator
- Input-output differential less than 0.6V at 0.5A
- Reverse battery protection
- 60V load dump protection
- -50V reverse transient protection
- Short circuit protection
- Internal thermal overload protection
- Available in 5-lead TO-220
- ON/OFF switch controls high current output
- Reset error flag
- P⁺ Product Enhancement tested

Typical Application Circuit



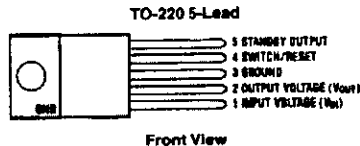
*Required if regulator is located far from power supply filter.

**C_{OUT} must be at least 10 μF to maintain stability. May be increased without bound to maintain regulation during transients. Locate as close as possible to the regulator. This capacitor must be rated over the same operating temperature range as the regulator. The equivalent series resistance (ESR) of this capacitor is critical; see curve.

TL/H/5232-1

FIGURE 1. Test and Application Circuit

Connection Diagram



TL/H/5232-8

Order Number LM2935T
See NS Package Number T05A

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage	
Operating Range	26V
Overvoltage Protection	60V

Internal Power Dissipation (Note 1)	Internally Limited
Operating Temperature Range	-40°C to +125°C
Maximum Junction Temperature	150°C
Storage Temperature Range	-65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	230°C

Electrical Characteristics for V_{OUT}

V_{IN} = 14V, I_O = 500 mA, T_J = 25°C (Note 4), C₂ = 10 μF (unless otherwise specified)

Parameter	Conditions	Typ	Tested Limit (Note 3)	Units Limit
Output Voltage	6V ≤ V _{IN} ≤ 26V, 5 mA ≤ I _O ≤ 500 mA, -40°C ≤ T _J ≤ 125°C (Note 2)	5.00	5.25 4.75	V _{MAX} V _{MIN}
Line Regulation	9V ≤ V _{IN} ≤ 16V, I _O = 5 mA 6V ≤ V _{IN} ≤ 26V, I _O = 5 mA	4 10	25 50	mV _{MAX} mV _{MAX}
Load Regulation	5 mA ≤ I _O ≤ 500 mA	10	50	mV _{MAX}
Output Impedance	500 mA _{DC} and 10 mA _{rms} , 100 Hz - 10 kHz	200		mΩ
Quiescent Current	I _O ≤ 10 mA, No Load on Standby I _O = 500 mA, No Load on Standby I _O = 750 mA, No Load on Standby	3 40 90	100	mA mA _{MAX} mA
Output Noise Voltage	10 Hz - 100 kHz	100		μV _{rms}
Long Term Stability		20		mV/1000 hr
Ripple Rejection	I _O = 120 Hz	66		dB
Dropout Voltage	I _O = 500 mA I _O = 750 mA	0.45 0.82	0.6	V _{MAX}
Current Limit		1.2	0.75	A _{MIN}
Maximum Operational Input Voltage		31	26	V _{MIN}
Maximum Line Transient	V _O ≤ 5.5V	70	60	V
Reverse Polarity Input Voltage, DC		-30	-15	V
Reverse Polarity Input Voltage, Transient	1% Duty Cycle, τ ≤ 100 ms, 10Ω Load	-80	-50	V
Reset Output Voltage				
Low	R1 = 20k, V _{IN} = 4.0V	0.9	1.2	V _{MAX}
High	R1 = 20k, V _{IN} = 14V	5.0	6.0 4.5	V _{MAX} V _{MIN}
Reset Output Current	Reset = 1.2V	5		mA
ON/OFF Resistor	R1 (± 10% Tolerance)		20	kΩ _{MAX}

Note 1: Thermal resistance without a heat sink for junction to case temperature is 3°C/W(TO-220). Thermal resistance for TO-220 case to ambient temperature is 50°C/W.

Note 2: The temperature extremes are guaranteed but not 100% production tested. This parameter is not used to calculate outgoing AQL.

Note 3: Tested Limits are guaranteed and 100% tested in production.

Note 4: To ensure constant junction temperature, low duty cycle pulse testing is used.

Electrical Characteristics for Standby Output

$I_O = 10 \text{ mA}$, $V_{IN} = 14\text{V}$, S1 open, $C_{OUT} = 10 \mu\text{F}$, $T_J = 25^\circ\text{C}$ (Note 4), (unless otherwise specified)

Parameter	Standby Output Conditions	Typ	Tested Limit	Units Limit
Output Voltage	$I_O \leq 10 \text{ mA}$, $6\text{V} \leq V_{IN} \leq 26\text{V}$, $-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	5.00	5.25 4.75	V_{MAX} V_{MIN}
Tracking	V_{OUT} - Standby Output Voltage	50	200	mV_{MAX}
Line Regulation	$6\text{V} \leq V_{IN} \leq 26\text{V}$	4	50	mV_{MAX}
Load Regulation	$1 \text{ mA} \leq I_O \leq 10 \text{ mA}$	10	50	mV_{MAX}
Output Impedance	10 mA_{DC} and 1 mA_{RMS} , 100 Hz - 10 kHz	1		Ω
Quiescent Current	$I_O \leq 10 \text{ mA}$, V_{OUT} OFF (Note 2)	2	3	mA_{MAX}
Output Noise Voltage	10 Hz - 100 kHz	300		μV
Long Term Stability		20		$\text{mV}/1000 \text{ hr}$
Ripple Rejection	$f_O = 120 \text{ Hz}$	66		dB
Dropout Voltage	$I_O \leq 10 \text{ mA}$	0.55	0.7	V_{MAX}
Current Limit		70	25	mA_{MIN}
Maximum Operational Input Voltage	$V_O \leq 6\text{V}$	70	60	V_{MIN}
Reverse Polarity Input Voltage, DC	$V_O = -0.3\text{V}$, 510Ω Load	-30	-15	V_{MIN}
Reverse Polarity Input Voltage, Transient	1% Duty Cycle $T_r = 100 \text{ ms}$ 500Ω Load	-80	-50	V_{MIN}

Typical Circuit Waveforms

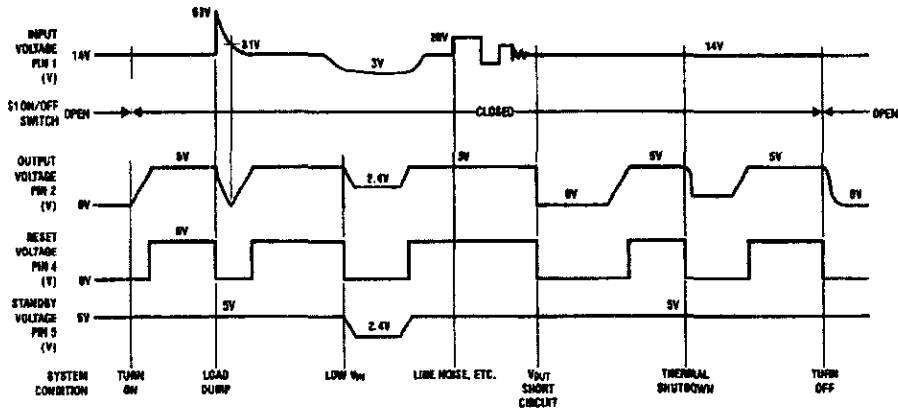
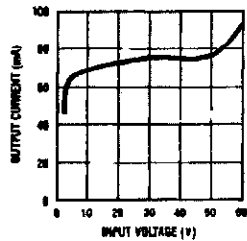
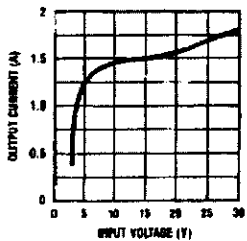
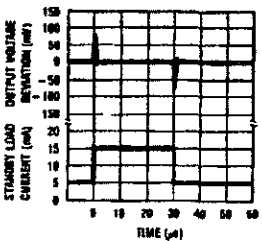
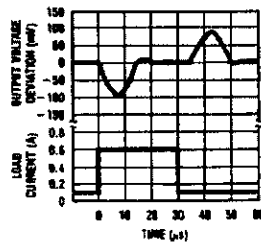
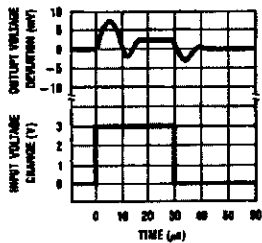
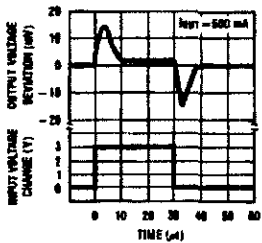
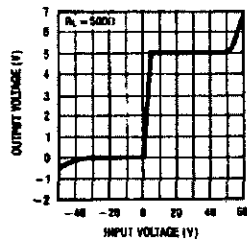
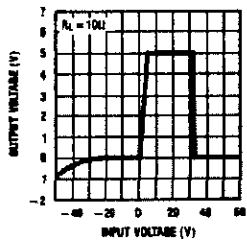
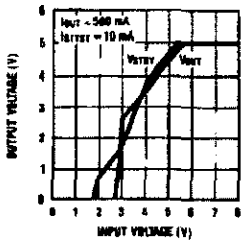
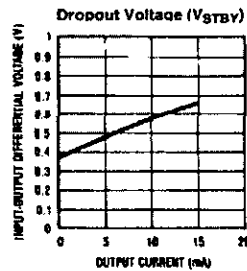
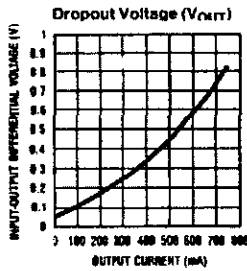
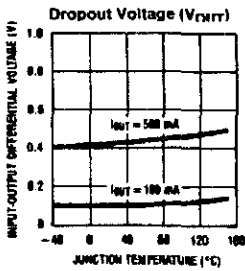


FIGURE 2

TL/H/6232-Z

Typical Performance Characteristics

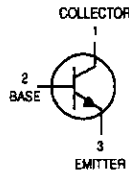


BY)

MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

Order this document
by P2N2222A/D

Amplifier Transistors
NPN Silicon



P2N2222A



CASE 29-04, STYLE 17
TO-18 (TO-226AA)

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector-Emitter Voltage	V_{CE0}	40	Vdc
Collector-Base Voltage	V_{CBO}	75	Vdc
Emitter-Base Voltage	V_{EBO}	6.0	Vdc
Collector Current — Continuous	I_C	600	mA dc
Total Device Dissipation $\ominus T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	625 5.0	mW mW/ $^\circ\text{C}$
Total Device Dissipation $\ominus T_C = 25^\circ\text{C}$ Derate above 25°C	P_D	1.5 12	Watts mW/ $^\circ\text{C}$
Operating and Storage Junction Temperature Range	T_J, T_{stg}	-55 to +150	$^\circ\text{C}$

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	200	$^\circ\text{C/W}$
Thermal Resistance, Junction to Case	$R_{\theta JC}$	83.3	$^\circ\text{C/W}$

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector-Emitter Breakdown Voltage ($I_C = 10 \text{ mA dc}, I_E = 0$)	$V_{(BR)CEO}$	40	—	Vdc
Collector-Base Breakdown Voltage ($I_C = 10 \mu\text{A dc}, I_E = 0$)	$V_{(BR)CBO}$	75	—	Vdc
Emitter-Base Breakdown Voltage ($I_E = 10 \mu\text{A dc}, I_C = 0$)	$V_{(BR)EBO}$	6.0	—	Vdc
Collector Cutoff Current ($V_{CE} = 60 \text{ Vdc}, V_{EB(off)} = 3.0 \text{ Vdc}$)	I_{CEX}	—	10	nA dc
Collector Cutoff Current ($V_{CB} = 60 \text{ Vdc}, I_E = 0$) ($V_{CB} = 60 \text{ Vdc}, I_E = 0, T_A = 150^\circ\text{C}$)	I_{CBO}	— —	0.01 10	$\mu\text{A dc}$
Emitter Cutoff Current ($V_{EB} = 3.0 \text{ Vdc}, I_C = 0$)	I_{EBO}	—	10	nA dc
Collector Cutoff Current ($V_{CE} = 10 \text{ V}$)	I_{CEO}	—	10	nA dc
Base Cutoff Current ($V_{CE} = 60 \text{ Vdc}, V_{EB(off)} = 3.0 \text{ Vdc}$)	I_{BEX}	—	20	nA dc



P2N2222A

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted) (Continued)

Characteristic	Symbol	Min	Max	Unit
ON CHARACTERISTICS				
DC Current Gain ($I_C = 0.1\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$) ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$) ($I_C = 10\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$) ($I_C = 10\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $T_A = -55^\circ\text{C}$) ($I_C = 150\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$) ⁽¹⁾ ($I_C = 150\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$) ⁽¹⁾ ($I_C = 500\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$) ⁽¹⁾	h_{FE}	35 50 75 35 100 50 40	— — — — 300 — —	—
Collector-Emitter Saturation Voltage ⁽¹⁾ ($I_C = 150\text{ mAdc}$, $I_B = 15\text{ mAdc}$) ($I_C = 500\text{ mAdc}$, $I_B = 50\text{ mAdc}$)	$V_{CE(sat)}$	— —	0.3 1.0	Vdc
Base-Emitter Saturation Voltage ⁽¹⁾ ($I_C = 150\text{ mAdc}$, $I_B = 15\text{ mAdc}$) ($I_C = 500\text{ mAdc}$, $I_B = 50\text{ mAdc}$)	$V_{BE(sat)}$	0.6 —	1.2 2.0	Vdc

SMALL-SIGNAL CHARACTERISTICS

Current-Gain -- Bandwidth Product ⁽²⁾ ($I_C = 20\text{ mAdc}$, $V_{CE} = 20\text{ Vdc}$, $f = 100\text{ MHz}$)	f_T	300	—	MHz
Output Capacitance ($V_{CB} = 10\text{ Vdc}$, $I_E = 0$, $f = 1.0\text{ MHz}$)	C_{obo}	—	8.0	pF
Input Capacitance ($V_{EB} = 0.5\text{ Vdc}$, $I_C = 0$, $f = 1.0\text{ MHz}$)	C_{ibo}	—	25	pF
Input Impedance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$) ($I_C = 10\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{ie}	2.0 0.25	8.0 1.25	k Ω
Voltage Feedback Ratio ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$) ($I_C = 10\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{re}	— —	8.0 4.0	$\times 10^{-4}$
Small-Signal Current Gain ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$) ($I_C = 10\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{fe}	50 75	300 375	—
Output Admittance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$) ($I_C = 10\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{oe}	5.0 25	35 200	μmhos
Collector Base Time Constant ($I_E = 20\text{ mAdc}$, $V_{CB} = 20\text{ Vdc}$, $f = 31.8\text{ MHz}$)	τ_b/C_C	—	150	ps
Noise Figure ($I_C = 100\text{ }\mu\text{A}$, $V_{CE} = 10\text{ Vdc}$, $R_S = 1.0\text{ k}\Omega$, $f = 1.0\text{ kHz}$)	NF	—	4.0	dB

SWITCHING CHARACTERISTICS

Delay Time	(VCC = 30 Vdc, VBE(off) = -2.0 Vdc, I _C = 150 mAdc, I _{B1} = 15 mAdc) (Figure 1)	t_d	—	10	ns
Rise Time		t_r	—	25	ns
Storage Time	(VCC = 30 Vdc, I _C = 150 mAdc, I _{B1} = I _{B2} = 15 mAdc) (Figure 2)	t_s	—	225	ns
Fall Time		t_f	—	60	ns

1. Pulse Test: Pulse Width $\leq 300\text{ }\mu\text{s}$, Duty Cycle $\leq 2.0\%$.
2. f_T is defined as the frequency at which $|h_{fe}|$ extrapolates to unity.

P2N2222A

SWITCHING TIME EQUIVALENT TEST CIRCUITS

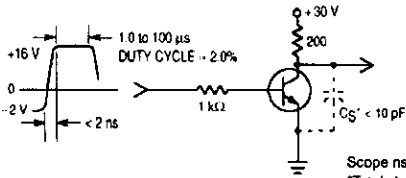


Figure 1. Turn-On Time

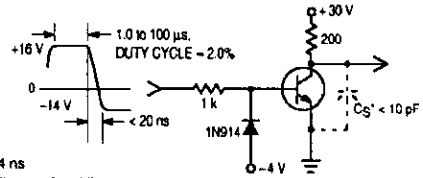


Figure 2. Turn-Off Time

Scope rise time <math>< 4\text{ ns}</math>
 *Total shunt capacitance of test jig,
 connectors, and oscilloscope.

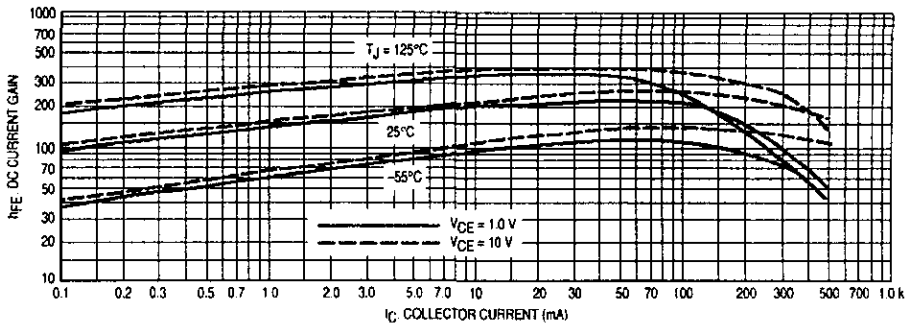


Figure 3. DC Current Gain

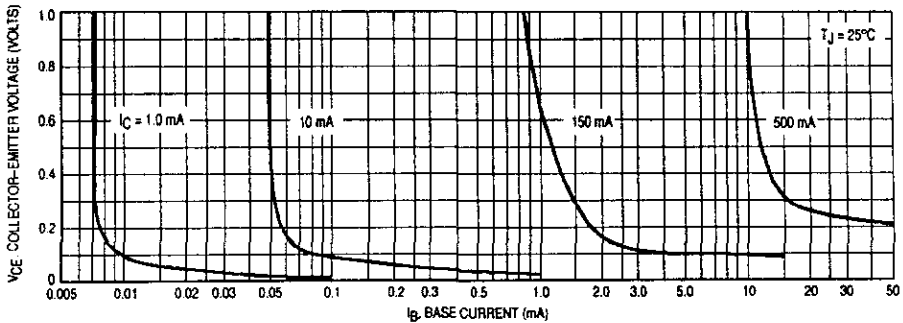


Figure 4. Collector Saturation Region

P2N2222A

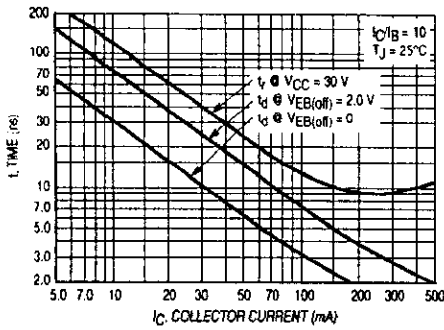


Figure 5. Turn-On Time

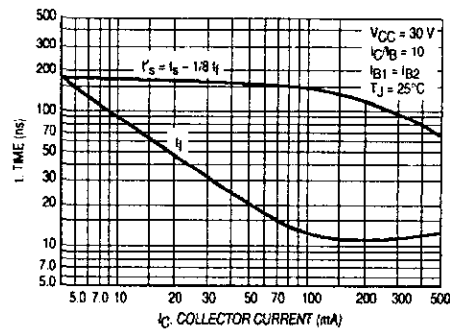


Figure 6. Turn-Off Time

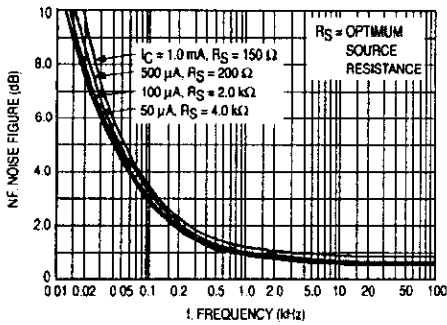


Figure 7. Frequency Effects

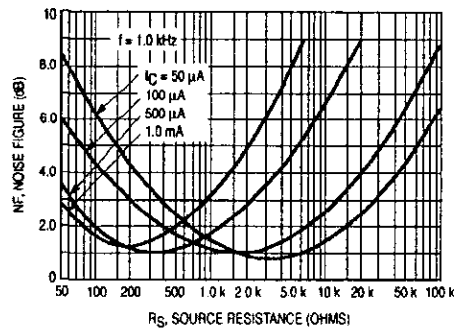


Figure 8. Source Resistance Effects

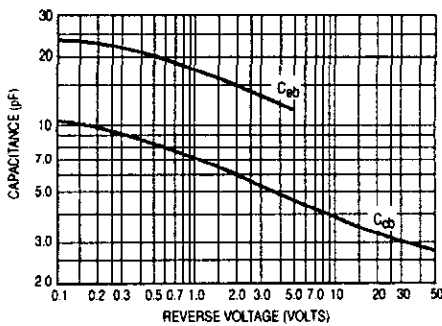


Figure 9. Capacitances

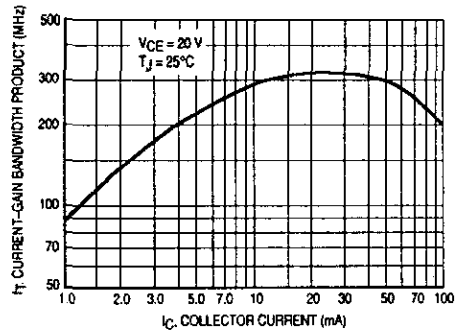
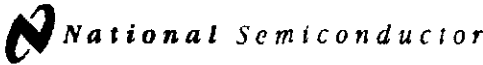


Figure 10. Current-Gain Bandwidth Product



PRELIMINARY
August 1997

COP8SAA7/COP8SAB7/COP8SAC7 8-Bit One-Time Programmable (OTP) Microcontroller

COP8SAA7/COP8SAB7/COP8SAC7 8-Bit One-Time Programmable (OTP) Microcontroller

General Description

The COP8Ax7 OTP microcontrollers are members of the COP8™ feature family using an 8-bit single chip core architecture. These devices are fabricated in National Semiconductor's high-density EPROM process, and offered on a variety of packages, temperature ranges and voltage ranges to satisfy a wide variety of applications.

Key Features

- Low cost 8-bit OTP microcontroller
- OTP program space with read/write protection (fully secured)
- Quiet Design (low radiated emissions)
- Multi-Input Wakeup pins with optional interrupts (4 to 8 pins)
- 8 bytes of user storage space in EPROM
- User selectable clock options
 - Crystal/Resonator options
 - Crystal/Resonator option with on-chip bias resistor
 - External oscillator
 - Internal R/C oscillator
- Internal Power-On Reset — user selectable
- WATCHDOG and Clock Monitor Logic — user selectable
- Up to 12 high current outputs

Device	EPROM	RAM	Package and I/O	
			Package Types	Number of I/O
COP8SAC7	4k	128	20 DIP/SO	16
			28 DIP/SO	24
			40 DIP	36
			44 PLCC	40
COP8SAB7	2k	128	20 DIP/SO	16
			28 DIP/SO	24
COP8SAA7	1k	64	16 DIP/SO	12
			20 DIP/SO	16
			28 DIP/SO	24

CPU Features

- Versatile easy to use instruction set
- 1 μ s instruction cycle time
- Eight multi-source vectored interrupts servicing
 - External interrupt
 - Idle Timer T0

- One Timer (with 2 interrupts)
- MICROWIRE/PLUS™ Serial Interface
- Multi-Input Wake Up
- Software Trap
- Default VIS (default interrupt)
- 8-bit Stack Pointer SP (stack in RAM)
- Two 8-bit Register Indirect Data Memory Pointers
- True bit manipulation
- Memory mapped I/O
- BCD arithmetic instructions

Peripheral Features

- Multi-Input Wakeup Logic
- One 16-bit timer with two 16-bit registers supporting:
 - Processor Independent PWM mode
 - External Event counter mode
 - Input Capture mode
- Idle Timer
- MICROWIRE/PLUS Serial Interface (SPI Compatible)

I/O Features

- Software selectable I/O options
 - TRI-STATE® Output
 - Push-Pull Output
 - Weak Pull Up Input
 - High Impedance Input
- Schmitt trigger inputs on ports G and L
- Up to 12 high current outputs
- Pin efficient (i.e., 40 pins in 44-pin package are devoted to useful I/O)

Fully Static CMOS Design

- Low current drain (typically < 4 μ A)
- Single supply operation: 2.7V to 5.5V
- Two power saving modes: HALT and IDLE

Temperature Ranges

0°C to +70°C, -40°C to +85°C, and -40°C to +125°C

Development Support

- Windowed packages for DIP and PLCC
- Real time emulation and full program debug offered by MetaLink Development System

TRI-STATE® is a registered trademark of National Semiconductor Corporation.
MICROWIRE/PLUS™, COP8™, MICROWIRE™ and WATCHDOG™ are trademarks of National Semiconductor Corporation.
META-MASTER™ is a trademark of MetaLink Corporation.
R/C™ is a registered trademark of International Business Machines Corporation.

Ordering Information

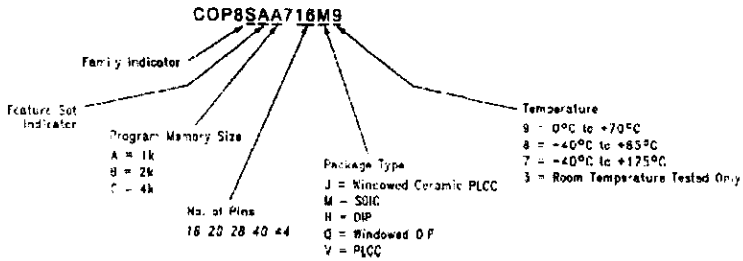


FIGURE 3. Part Numbering Scheme

DS01738B-8

Temperature	1k EPROM		2k EPROM		4k EPROM		4k EPROM	
	Order Number	Package	Order Number	Package	Order Number	Package	Windowed Device Order Number	Package
0°C to +70°C	COP8SAA716M9	16M						
	COP8SAA720M9	20M	COP8SAB720M9	20M	COP8SAC720M9	20M		
	COP8SAA728M9	28M	COP8SAB728M9	28M	COP8SAC728M9	28M		
	COP8SAA716N9	16N						
	COP8SAA720N9	20N	COP8SAB720N9	20N	COP8SAC720N9	20N	COP8SAC720Q3	20Q
	COP8SAA728N9	28N	COP8SAB728N9	28N	COP8SAC728N9	28N	COP8SAC728Q3	28Q
					COP8SAC740N9	40N	COP8SAC740Q3	40Q
-40°C to +85°C					COP8SAC744V9	44V	COP8SAC744J3	44J
	COP8SAA716M8	16M						
	COP8SAA720M8	20M	COP8SAB720M8	20M	COP8SAC720M8	20M		
	COP8SAA728M8	28M	COP8SAB728M8	28M	COP8SAC728M8	28M		
	COP8SAA716N8	16N						
	COP8SAA720N8	20N	COP8SAB720N8	20N	COP8SAC720N8	20N		
	COP8SAA728N8	28N	COP8SAB728N8	28N	COP8SAC728N8	28N		
-40°C to +125°C					COP8SAC740N8	40N		
					COP8SAC744V8	44V		
					COP8SAC720M7	20M		
					COP8SAC728M7	28M		
					COP8SAC720N7	20N		
					COP8SAC728N7	28N		
					COP8SAC740N7	40N		
				COP8SAC744V7	44V			

4.0 Electrical Characteristics

ESD Protection Level 2 kV
(Human Body Model)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Total Current into V_{CC} Pin (Source) 80 mA
 Total Current out of GND Pin (Sink) 100 mA
 Storage Temperature Range -65°C to +140°C

Supply Voltage (V_{CC}) 7V
 Voltage at Any Pin -0.6V to V_{CC} +0.6V

Note 1: Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics

0°C ≤ T_A ≤ +70°C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Operating Voltage	(Note 6)	2.7		5.5	V
Power Supply Rise Time from 0.0V (On-Chip Power-On Reset Selected)		10 ns		50 ms	
V _{CC} Start Voltage to Guarantee POR				0.25	V
Power Supply Ripple (Note 3)	Peak-to-Peak			0.1 V _{CC}	V
Supply Current (Note 4)					
CKI = 10 MHz	V _{CC} = 5.5V, t _C = 1 μs			6	mA
CKI = 4 MHz	V _{CC} = 4.5V, t _C = 2.5 μs			2.1	mA
HALT Current (Note 5) — WATCHDOG Disabled	V _{CC} = 5.5V, CKI = 0 MHz		<4	8	μA
IDLE Current (Note 4)					
CKI = 10 MHz	V _{CC} = 5.5V, t _C = 1 μs			1.5	mA
CKI = 4 MHz	V _{CC} = 4.5V, t _C = 2.5 μs			0.8	mA
Input Levels (V _{IH} , V _{IL})					
RESET					
Logic High		0.8 V _{CC}			V
Logic Low				0.2 V _{CC}	V
CKI, All Other Inputs					
Logic High		0.7 V _{CC}			V
Logic Low				0.2 V _{CC}	V
Value of the Internal Bias Resistor for the Crystal/Resonator Oscillator		0.5	1.0	2.0	MΩ
CKI Resistance to V _{CC} or GND when R/C Oscillator is Selected	V _{CC} = 5.5V	5	8	11	kΩ
Hi-Z Input Leakage (same as TRI-STATE output)	V _{CC} = 5.5V	-2		+2	μA
Input Pullup Current	V _{CC} = 5.5V, V _{IN} = 0V	-40		-250	μA
G and L Port Input Hysteresis		0.25 V _{CC}			V

DC Electrical Characteristics (Continued)					
0°C ≤ T _A ≤ +70°C unless otherwise specified.					
Parameter	Conditions	Min	Typ	Max	Units
Output Current Levels					
D Outputs					
Source	V _{CC} = 4.5V, V _{OH} = 3.3V	-0.4			mA
	V _{CC} = 2.7V, V _{OH} = 1.8V	-0.2			mA
Sink	V _{CC} = 4.5V, V _{OL} = 1.0V	10			mA
	V _{CC} = 2.7V, V _{OL} = 0.4V	2			mA
L Port					
Source (Weak Pull-Up)	V _{CC} = 4.5V, V _{OH} = 2.7V	-10		-110	µA
	V _{CC} = 2.7V, V _{OH} = 1.8V	-2.5		-33	µA
Source (Push-Pull Mode)	V _{CC} = 4.5V, V _{OH} = 3.3V	-0.4			mA
	V _{CC} = 2.7V, V _{OH} = 1.8V	-0.2			mA
Sink (L0-L3, Push-Pull Mode)	V _{CC} = 4.5V, V _{OL} = 1.0V	10			mA
	V _{CC} = 2.7V, V _{OL} = 0.4V	2			mA
Sink (L4-L7, Push-Pull Mode)	V _{CC} = 4.5V, V _{OL} = 0.4V	1.6			mA
	V _{CC} = 2.7V, V _{OL} = 0.4V	0.7			mA
All Others					
Source (Weak Pull-Up Mode)	V _{CC} = 4.5V, V _{OH} = 2.7V	-10		-110	µA
	V _{CC} = 2.7V, V _{OH} = 1.8V	-2.5		-33	µA
Source (Push-Pull Mode)	V _{CC} = 4.5V, V _{OH} = 3.3V	-0.4			mA
	V _{CC} = 2.7V, V _{OH} = 1.8V	-0.2			mA
Sink (Push-Pull Mode)	V _{CC} = 4.5V, V _{OL} = 0.4V	1.6			mA
	V _{CC} = 2.7V, V _{OL} = 0.4V	0.7			mA
	15				mA
	3				mA
Maximum Input Current without Latchup (Note 6)				±200	mA
RAM Retention Voltage, V _r		2.0			V
V _{CC} Rise Time from a V _{CC} ≥ 2.0V	(Note 9)	12			µs
Input Capacitance	(Note 8)			7	pF
Load Capacitance on D2	(Note 6)			1000	pF

AC Electrical Characteristics

0°C ≤ T_A ≤ +70°C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Instruction Cycle Time (t _C)					
Crystal/Resonator, External	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V	1.0 2.0		DC DC	μs μs
Internal R/C Oscillator	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V		2.0 TBD		μs μs
R/C Oscillator Frequency Variation (Note 8)	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V			±35 TBD	% %
External CKI Clock Duty Cycle (Note 8)	f _r = Max	45		55	%
Rise Time (Note 8)	f _r = 10 MHz Ext Clock			12	ns
Fall Time (Note 8)	f _r = 10 MHz Ext Clock			8	ns
Inputs					
t _{SETUP}	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V	200 500			ns ns
t _{HOLD}	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V	60 150			ns ns
Output Propagation Delay (Note 7)	R _L = 2.2k, C _L = 100 pF				
t _{PD1} , t _{PD0}	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V			0.7 1.75	μs μs
SO, SK	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V			1.0 2.5	μs μs
All Others	4.5V ≤ V _{CC} ≤ 5.5V 2.7V ≤ V _{CC} < 4.5V				
MICROWIRE Setup Time (t _{UNV3}) (Note 7)		20			ns
MICROWIRE Hold Time (t _{UWH}) (Note 7)		56			ns
MICROWIRE Output Propagation Delay (t _{UPD})				220	ns
MICROWIRE Maximum Shift Clock					
Master Mode				500	kHz
Slave Mode				1	MHz
Input Pulse Width (Note 7)					
Interrupt Input High Time		1			t _C
Interrupt Input Low Time		1			t _C
Timer 1 Input High Time		1			t _C
Timer 1 Input Low Time		1			t _C
Reset Pulse Width		1			μs

Note 2: t_C = Instruction cycle time (Clock input frequency divided by 10).

Note 3: Maximum rate of voltage change must be < 0.5 V/ns.

Note 4: Supply and IDLE currents are measured with CKI driven with a square wave Oscillator, CKD driven 180° out of phase with CKI, inputs connected to V_{CC} and outputs driven low but not connected to a load.

Note 5: The HALT mode will stop CKI from oscillating in the R/C and the Crystal configurations. In the R/C configuration, CKI is forced high internally. In the crystal or external configuration, CKI is TRI-STATE. Measurement of I_{DD HALT} is done with device neither sourcing nor sinking current, with L, F, C, G0, and G2-G5 programmed as low outputs and not driving a load; all outputs programmed low and not driving a load; all inputs tied to V_{CC}; WATCHDOG and clock monitor disabled. Parameter refers to HALT mode entered via setting bit 7 of the G Port data register.

Note 6: Pins G6 and RESET are designed with a high voltage input network. These pins allow input voltages > V_{CC} and the pins will have sink current to V_{CC} when biased at voltages > V_{CC} (the pins do not have source current when biased at a voltage below V_{CC}). The effective resistance to V_{CC} is 750Ω (typical). These two pins will not latch up. The voltage at the pins must be limited to < 14V. WARNING: Voltages in excess of 14V will cause damage to the pins. This warning excludes ESD transients.

Note 7: The output propagation delay is referenced to the end of the instruction cycle where the output change occurs.

Note 8: Parameter characterized but not tested

Note 9: Rise times faster than this specification may reset the device if POR is enabled and may affect the value of Idle Timer T0 if POR is not enabled.

APÉNDICE C
LISTADO PARCIAL DE
PROGRAMAS

Programa en ensamblador

```
.TITLE PERSEO15
.SECT MAIN,ROM
.SET DIG0,0X20
.SET DIG1,0X21
.SET DIG2,0X22
.SET DIG3,0X23
.SET DIG4,0X24
.SET DIG5,0X25
.SET DIG6,0X26
.SET DIG7,0X27
.SET DIG8,0X28
.SET DIG9,0X29
.SET LETa,0X2A
.SET LETb,0X2B
.SET LETc,0X2C
.SET LETd,0X2D
.SET LETe,0X2E
.SET LETf,0X2F
.SET LETg,0X30
.SET LETh,0X31
.SET LETj,0X32
.SET LETl,0X33
.SET LETp,0X34
.SET LETr,0X35
.SET LETs,0X36
.SET LETt,0X37
.SET LET?,0X38
.SET GUION,0X39
.SET LETn,0X40
.SET CONT,0XF1
```

```
.ORG 0X00
```

```
PRIN???:LD    DIG0,#0XEB
           LD    DIG1,#0X21
           LD    DIG2,#0XBA
           LD    DIG3,#0XB3
           LD    DIG4,#0X71
           LD    DIG5,#0XD3
           LD    DIG6,#0X5B
           LD    DIG7,#0XA1
           LD    DIG8,#0XFB
           LD    DIG9,#0XF1
           LD    LETa,#0XF9
           LD    LETb,#0X5B
           LD    LETc,#0XCA
           LD    LETd,#0X3B
           LD    LETe,#0XDA
           LD    LETf,#0XD8
           LD    LETg,#0XDB
           LD    LETh,#0X79
           LD    LETj,#0X2B
           LD    LETl,#0X4A
           LD    LETn,#0X19
           LD    LETp,#0XF8
           LD    LETr,#0X18
           LD    LETs,#0XD3
           LD    LETt,#0X5A
           LD    LET?,#0XB8
           LD    GUION,#0X10
           LD    PORTD,#0X00
           LD    PORTLD,#0XFF
           LD    CNTRL,#0XC0
           LD    PSW,#0X10
           LD    A,0X00
```

```
*****
*****
;* SUBROUTINA PARA EL BARRIDO DEL TECLADC.
*
*****
*****
```

```
PRIN???:LD    PORTFC,#0X03
           LD    PORTLC,#0XFF
           LD    PORTFD,#0X01
           LD    PORTLD,#0X00
           JSR   LIMPIA
           SBIT  1,PORTLD
           LD    PORTFC,#0X03
           LD    PORTFD,#0X01
           LD    A,LETL
           X     A,PORTD
           JSR   RETAR2
           RBIT  1,PORTLD
           JSR   LIMPIA
           SBIT  0,PORTLD
           JSR   TECLADO
           JSR   FIN
           LD    A,DIG1
           X     A,PORTD
           JSR   RETAR2
           RBIT  0,PORTLD
           LD    PORTD,#0X01
           JSR   HABLATCH
           JSR   PULSOS
           JSR   TECLADO
           JSR   FIN
           JSR   DESLATCH
           JSR   LIMPIA
           LD    PORTD,#0X01
           JSR   HABLATCH
           JSR   PULSOS
           LD    A,LETb
           X     A,PORTD
           JSR   RETAR2
           JSR   LIMPIA
           JSR   PULSOS
           JSR   DESLATCH
           LD    PORTD,#0X80
           JSR   HABLATCH
           JSR   PULSOS
           JSR   TECLADO
           JSR   PLACAS
           JSR   DESLATCH
           JSR   LIMPIA
           LD    PORTD,#0X80
           JSR   HABLATCH
           JSR   PULSOS
           LD    A,LETr
           X     A,PORTD
           JSR   RETAR2
           JSR   LIMPIA
           JSR   PULSOS
           JSR   DESLATCH
           SBIT  6,PORTD
           JSR   HABLATCH
           JSR   PULSOS
           JSR   TECLADO
           JSR   RELOJ
           JSR   DESLATCH
           JSR   LIMPIA
           LD    PORTD,#0X40
           JSR   HABLATCH
           JSR   PULSOS
           LD    A,LETE
           X     A,PORTD
           JSR   RETAR2
           JSR   LIMPIA
```



```

LD PORTLD,#0X00
RET
;*****
PRIN???: JSR BUZER
LD PORTFC,#0X03
LD CONT,#0X64
PRIN???:JSR LIMPIA
SBIT 1,PORTLD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 1,PORTLD
JSR PRIN???.
SBIT 0,PORTLD
LD A,DIG1
X A,PORTD
JSR PRIN???.
RBIT 0,PORTLD
JSR LIMPIA
SBIT 0,PORTD
JSR HABLATCH
JSR PRIN???.
LD A,LETF
X A,PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
DRSZ CONT
JMP PRIN???.
JMP PRIN???.
;*****
PRIN???:JSR PRIN???.
LD PORTFC,#0X03
LD CONT,#0X64
PRIN???:JSR PRIN???.
SBIT 1,PORTLD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 1,PORTLD
JSR LIMPIA
SBIT 0,PORTLD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 0,PORTLD
JSR PRIN???.
SBIT 0,PORTD
JSR PRIN???.
JSR PULSOS
LD A,LETF
X A,PORTD
JSR PRIN???.
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
LD PORTD,#0X80
JSR HABLATCH
JSR PULSOS
LD A,DIGO
X A,PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
LD PORTD,#0X40
JSR HABLATCH
JSR PULSOS
LD A,LETF
X A,PORTD
JSR RETAR2
JSR PRIN???.
JSR PRIN???.
JSR PRIN???.
LD PORTD,#0X02
JSR HABLATCH
JSR PULSOS
LD A,LETH
X A,PORTD
JSR RETAR2
JSR PRIN???.
JSR PULSOS
JSR DESLATCH
DRSZ CONT
JMP PRIN???.
;*****
PRIN???: LD PORTFC,#0X03
JSR LIMPIA
LD PORTLD,#0X03
JSR TECLADO
JSR FIN
LD A,DIGO
X A,PORTD
JSR RETAR2
LD PORTLD,#0X00
JSR PRIN???.
SBIT 0,PORTD
JSR HABLATCH
JSR PRIN???.
LD A,GUION
X A,PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
SBIT 6,PORTD
JSR HABLATCH
JSR PULSOS
JSR TECLADO
JSR DATOS
JSR DESLATCH
JSR LIMPIA
LD PORTD,#0XC0
JSR HABLATCH
JSR PULSOS
LD A,DIGO
X A,PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
LD PORTD,#0X02
JSR HABLATCH
JSR PULSOS
LD A,LETH
X A,PORTD
JSR RETAR2
JSR PRIN???.
JSR PULSOS
JSR PRIN???.
JMP PRIN???.
;*****
PRIN???:JSR BUZER
LD PORTFC,#0X03
LD CONT,#0X64
PRIN???:JSR LIMPIA
SBIT 1,PORTLD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 1,PORTLD
JSR LIMPIA
SBIT 0,PORTLD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 1,PORTLD
JSR LIMPIA
SBIT 0,PORTD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 0,PORTLD
JSR LIMPIA
SBIT 0,PORTD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 0,PORTLD
JSR LIMPIA
SBIT 0,PORTD
LD A,LETF
X A,PORTD
JSR RETAR2
RBIT 0,PORTLD

```

```

JSR LIMPIA
SBIT 0, PORTD
JSR HABLATCH
JSR PULSOS
LD A, LETT
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
LD PORTD, #0X80
JSR HABLATCH
JSR PULSOS
LD A, DIGO
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
LD PORTD, #0X40
JSR HABLATCH
JSR PULSOS
LD A, LETS
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
DRSZ CONT
JMP DEN
JMP CEROS

;*****
PRIN???: JSR BUZER
LD PORTFC, #0X03
LD CONT, #0X64
DE PRIN???: JSR LIMPIA
SBIT 1, PORTLD
LD A, LETP
X A, PORTD
JSR RETAR2
RBIT 1, PORTLD
JSR PRIN???
SBIT 0, PORTLD
LD A, LETL
X A, PORTD
JSR RETAR2
RBIT 0, PORTLD
JSR LIMPIA
SBIT 0, PORTD
JSR HABLATCH
JSR PULSOS
LD A, LETA
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
LD PORTD, #0X80
JSR HABLATCH
JSR PULSOS
LD A, LETC
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
LD PORTD, #0X40
JSR HABLATCH
JSR PULSOS
LD A, LETA
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH

LD PORTD, #0X20
JSR HABLATCH
JSR PULSOS
LD A, LETS
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
DRSZ CONT
JMP DE
JMP CERO

;*****
PRIN???: JSR BUZER
LD PORTFC, #0X03
LD CONT, #0X64
PRIN???: JSR LIMPIA
SBIT 1, PORTLD
LD A, LETS
X A, PORTD
JSR RETAR2
RBIT 1, PORTLD
JSR LIMPIA
SBIT 0, PORTLD
LD A, LETE
X A, PORTD
JSR RETAR2
RBIT 0, PORTLD
JSR LIMPIA
SBIT 0, PORTD
JSR PRIN???
JSR PULSOS
LD A, LETI
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PRIN???
JSR DESLATCH
LD PORTD, #0X80
JSR HABLATCH
JSR PULSOS
LD A, DIG1
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR PRIN???
LD PORTD, #0X40
JSR HABLATCH
JSR PULSOS
LD A, LETE
X A, PORTD
JSR RETAR2
JSR LIMPIA
JSR PULSOS
JSR DESLATCH
DRSZ CONT
JMP D

;*****
PRIN???: LD PORTFC, #0X03
JSR LIMPIA
LD PORTLD, #0X03
JSR TECLADO
JSR FIN
LD A, DIGO
X A, PORTD
JSR RETAR2
LD PORTLD, #0X00
JSR LIMPIA
SBIT 0, PORTD
JSR PRIN???
JSR PULSOS
LD A, GUION
LDA A, PORTD

```

JSR RETAR2	JSR RETAR2
JSR LIMPIA	JSR LIMPIA
JSR PULSOS	JSR PULSOS
JSR DESLATCH	JSR DESLATCH
SBIT 6, PORTD	LD PORTD, #0X02
JSR HABLATCH	JSR HABLATCH
JSR PULSOS	JSR PULSOS
JSR TECLADO	LD A, DIG1
JSR SERIE	X A, PORTD
JSR DESLATCH	JSR RETAR2
JSR LIMPIA	JSR LIMPIA
LD PORTD, #0XCO	JSR PULSOS
JSR HABLATCH	JSR DESLATCH
JSR PULSOS	JMP CERO
LD A, DIG0	.END PRINCIPIO
X A, PORTD	

PROGRAMA EN VISUAL BASIC

```

Private Sub RECIBIR_Click()
    Dim informacion As String
    Dim auxiliar As Integer
    Dim valor As Integer
    mdat0 = ""
enviar:
    For i = 0 To 20
        mcontrol = Chr(Asc("+~TY<")) ' 'Dato de control, para recibir los datos
        Tax_Presentación.Comuni.Output = mcontrol
        Next i

    Sleep (1200)
    Do
        mdat0 = mdat0 + Tax_Presentación.Comuni.Input
        If 0 = Tax_Presentación.Comuni.InBufferCount Then
            Exit Do
        End If
    Loop
    MsgBox Str(Len(mdat0)) + " dato(s) recibido(s)", 0, "Recepcion de Datos"
    ' Muestra mdat0 (mensaje recibido)
    Leer_Datos.Text = mdat0 + Chr(64) + Str(Len(mdat0))

    Placas.Text = Mid(mdat0, 1, 6)
    Folio.Text = Mid(mdat0, 7, 4)
    Tar1.Text = Mid(mdat0, 11, 4)
    Tar2.Text = Mid(mdat0, 15, 4)
    Cte W.Text = Mid(mdat0, 19, 4)
    Pulsos.Text = Mid(mdat0, 45, 4)
    Tot3.Text = Mid(mdat0, 23, 6)
    Tot4.Text = Mid(mdat0, 33, 4)
    Tot5.Text = Mid(mdat0, 65, 6)
    Tot6.Text = Mid(mdat0, 71, 4)
    Tot7.Text = Mid(mdat0, 37, 4)
    Tot8.Text = Mid(mdat0, 75, 4)
    Touch1.Text = Mid(mdat0, 125, 2)
    Touch2.Text = Mid(mdat0, 127, 2)
End Sub

Private Sub Tar1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    informacion.Caption = "Valor de la Tarifa de Dia."
End Sub

Private Sub Tar2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    informacion.Caption = "Valor de la Tarifa de Noche."
End Sub

```

```

Private Sub Tot4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

End Sub

Private Sub Probe_Click()

    Dim valor As Integer
    mdata = ""
enviar:
    For i = 0 To 20
        mcontrol = Chr(Asc("%&>")) 'Dato de control, para regresar a touch
        Tax Presentación.Comuni.Output = mcontrol
    Next i

End Sub

Private Sub Pulsos_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
información.Caption = "No. de Pulsos para la conversión de kilometros."

End Sub

Private Sub enviar_Click()
    Dim auxiliar As Integer
    Dim valor As Integer
    mdata = ""
enviar:
    For i = 0 To 3
        mcontrol = Chr(Asc("?") (154fg="))
        Tax Presentación.Comuni.Output = mcontrol
    Next i

    For i = 0 To 5
        mcontrol = mdata
        Tax Presentación.Comuni.Output = mcontrol
    Next i
    MsgBox msg1, 64, "Fin de Envio"

End Sub

Private Sub Form_Load()
    Dim counter As Integer
    Dim respuesta As String
    Dim P As Integer
    Dim cadenita As Variant
    Dim patron As Variant
    Dim cad As String
    Dim chars As Variant
    Dim aim As Integer
    Dim ecm As Integer
    Dim cn As String
    Dim c1 As Variant
    Dim c2 As Variant

    'On Error GoTo errorhandler

    ' FORMATO DE LA PANTALLA

    Grid1.Rows = 1
    Grid1.ColWidth(0) = 250
    Grid1.ColWidth(1) = 2050
    Grid1.ColWidth(2) = 1000
    Grid1.Row = 0
    Grid1.Col = 0
    Grid1.Text = ""
    Grid1.Col = 1
    Grid1.Text = "VARIABLE"
    Grid1.Col = 2
    Grid1.Text = "VALOR"

    'MNombre.Enabled = False
    'MClave.Enabled = False
    'Imprimir1.Enabled = True
    'Imprimir2.Enabled = False

```



```

Consul_Datos.Top = 1200
Consul_Datos.Left = 1200

```

```

' LIMPIA GRID

```

```

counter = 1
Do Until 15 = counter
  Gridl.Rows = Gridl.Rows + 1
  Gridl.Col = 0 ' En la columna 0 del registro counter
  Gridl.Row = counter
  Gridl.Text = counter ' Coloca el counter (# de registro)
  Gridl.Col = 1 ' En la columna 1
  Gridl.Text = " " & dt("") ' Coloca la clave
  Gridl.Col = 2 ' En la columna 2
  Gridl.Text = " " & dt("") ' Coloca el nombre
  counter = counter + 1 ' Incrementa el counter para nuevo renglon
Loop

```

```

' CARGA BD SI NO SE ABRE PUERTO (DE C:\Taxímetros)

```

```

continuar:

```

```

'acceso.mdb
Set db = OpenDatabase("", False, False, ";")
Set dt = db.OpenRecordset("")

```

```

dt.Edit
dt.MoveFirst
Do ' En bucle hasta fin de tabla
  Gridl.Col = 0 ' En la columna 0 del registro counter
  Gridl.Row = counter
  Gridl.Text = counter ' Coloca el counter (# de registro)
  Gridl.Col = 1 ' En la columna 1
  Gridl.Text = dt("dbvariable") ' Coloca la variable
  Gridl.Col = 2 ' En la columna 2
  Gridl.Text = dt("dbvalor") ' Coloca el valor
  counter = counter + 1 ' Incrementa el counter para nuevo renglon
  If counter > 14 Then Exit Do Else dt.MoveNext
Loop

```

```

dt.Close ' Cierra uso de tabla
db.Close ' Cierra uso de BD
P = 1000 ' Advitrario (proteger de abrir puertos)

```

```

'Imprimir1.Enabled = True
'Imprimir2.Enabled = False

```

```

Exit Sub

```

```

' EN CASO DE QUE OCURRA UN ERROR AL INCIAR LA CORRIDA DE LA APLICACION

```

```

errorhandler:

```

```

Select Case (Err)
  Case 3011, 3024
    msg = ""
  Case 76, 3044
    msg = ""
  'Case 8000 To 8020
  ' If 1 = P Then GoTo RET_PUERTO1
  ' If 2 = P Then GoTo RET_PUERTO2
  'If 3 = P Then GoTo RET_PUERTO3 Else msg = "ERROR " + Str$(Err)
  Case 3043
    msg = "NO SE ENCUENTRA DISCO DISPONIBLE"
  Case Else
    msg = "ERROR " + Str$(Err)
End Select

```

```

MsgBox msg, 16, "ERROR FATAL"
End

```

```

Exit Sub

```

```

' CARGA BD SI SE ABRE PUERTO (DE MODULO)

```

```

Carga_BD:

```

```

Set db = OpenDatabase("c:\ ", False, False, ";")

```

```

Set dt = db.OpenRecordset
counter = 1
c2 = 1
dt.Edit

Do Until counter > 14
c1 = Mid(, c2, 4)
dt.

Do

    If dt.EOF And Not (dt("dbvariable") = c1) Then
        Grid1.Col = 0
        Grid1.Text = counter
        Grid1.Col = 1
        Grid1.Text = c1
        Grid1.Col = 2
        Grid1.Text = "NO REG"
        Exit Do
    End If
    If "FFFF" = c1 Then Exit Do
    If dt("dbvariable") = c1 Then
        Grid1.Col = 0
        Grid1.Text = counter
        Grid1.Col = 1
        Grid1.Text = c1
        Grid1.Col = 2
        Grid1.Text = dt("dbvalor")
        Exit Do
    End If
    dt.MoveNext
    If dt.EOF Then
        Grid1.Col = 0
        Grid1.Text = counter
        Grid1.Col = 1
        Grid1.Text = c1
        Grid1.Col = 2
        Grid1.Text = "NO REG"
        Exit Do
    End If

    Loop
    counter = counter + 1          ' Incrementa el counter para nuevo renglon
    c2 = c2 + 5
Loop

cerrar:
dt.Close          ' Cierra uso de tabla
db.Close          ' Cierra uso de BD
P = 1000          ' Advitrario (proteger de abrir puertos)

Imprimir1.Enabled = True
Imprimir2.Enabled = True

Exit Sub

End Sub

```