



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

## DISEÑO Y CONSTRUCCION DE UN SISTEMA DE AFORO VEHICULAR.

**T E S I S**  
QUE PARA OBTENER EL TITULO DE  
INGENIERO ELECTRICO ELECTRONICO  
P R E S E N T A N  
ALEJANDRO CASTELLANOS DE LA TORRE  
PEDRO MORENO MUÑOZ  
MIGUEL ANGEL ROMERO MAGAÑA  
DIRECTOR DE TESIS: ING. EDUARDO RAMIREZ SANCHEZ



MEXICO, D.F.

2004

TESIS CON  
FALLA DE ORIGEN



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.

A la Facultad de Ingeniería y a la Universidad Nacional Autónoma de México por las enseñanzas dentro y fuera de las aulas, por hacernos sentir como en casa las innumerables horas que vivimos dentro de sus instalaciones.

A nuestros profesores que además de conocimientos nos inculcaron valor y conciencia para enfrentar nuestro futuro.

A los amigos y compañeros, por compartir la experiencia del conocimiento.

Al Ing. Eduardo Ramírez Sánchez por su invaluable dedicación y enseñanza , gracias por todo.

**Agradecimientos generales.**

A mi Madre Clara:

...porque me ha enseñado que la vida se disfruta mas cuando las cosas las busca uno mismo, porque la amo por ser la persona que me dio la vida y también porque en los momentos que he necesitado ayuda siempre ha estado a mi lado. Además este logro es por ella ya que con su amor y dedicación me motiva siempre a lograr lo que me propongo.

A mi Padre José:

...que sabe tanto de la vida y siempre se preocupó por enseñarme la nobleza de la vida, y también porque siempre me ha enseñado a ser responsable.

A mi Hermano Alberto:

...porque siempre hemos sido mas que hermanos, los mejores amigos, los mas leales y porque al convivir con él aumento mi gusto por la ingeniería.

A mi Abuelita y Tíos:

...que siempre me han mostrado su cariño y mostraron interés en mi desarrollo, también agradezco su consejo y apoyo.

A mis Primos:

...ya que crecimos juntos y con su ejemplo me motivaron siempre a seguir adelante.

A Compañeros y Amigos:

...porque la vivieron conmigo y juntos encontramos la manera de salir victoriosos.

A Pedro y Miguel:

...por el gran equipo que formamos, lo logramos!

A toda la gente que creyó en mi y en mi entusiasmo muchas gracias porque siempre han sido un aliciente para que yo pudiera ser lo que soy ahora.

A todos Gracias!

**Alejandro**

A mis Papas y a mi Hermana.

Por iluminar el sendero de mi tiempo e invocar en mi la persona que soy, e impulsar mi capacidad a un limite desconocido. Gracias por su ejemplo, por su cariño, por su apoyo, dedicación y paciencia incondicional.

A mi novia.

Reyna gracias por el camino que recorres día con día a mi lado, por el apoyo incondicional concedido durante sin fin de proyectos y situaciones, y por tu atención y dedicación.

A mis abuelitas, abuelito y familiares.

Gracias por sus enseñanzas que la experiencia de la vida otorga, y por su dedicación incondicional.

A mis amigos

Gracias por su invaluable amistad, por sus consejos y preceptos.

A mis compañeros de proyecto

Miguel y Alejandro que mas que compañeros son grandes amigos, gracias por todo

**Pedro**

A mi Madre, Delia Magaña por su apoyo incondicional a lo largo de toda mi vida.

A mi Padre, Miguel Romero que siempre esta conmigo.

A mi Esposa, Esperanza García por su amor, el cual me hace sentir vivo cada día.

A Joaquín Del Prado y Ricardo Magaña, mis hermanos.

A mi Hermana y sobrinos, Perla Xochitl, Edwin y Bryan.

A toda mi familia, la cual puedo contar con ellos en todo momento.

A mis compañeros de este proyecto, Alejandro y Pedro.

GRACIAS!

**Miguel**

## INDICE.

<b>INDICE.</b>	<b>1</b>
<b>INTRODUCCIÓN.</b>	<b>12</b>
<b>OBJETIVO</b>	<b>13</b>
<b>1. ANTECEDENTES Y ANÁLISIS DE NECESIDADES.</b>	<b>15</b>
<b>1.1. RED NACIONAL DE CAMINOS.</b>	<b>15</b>
<b>1.2. HISTORIA.</b>	<b>15</b>
1.2.1. SITUACIÓN ACTUAL.	20
1.2.2. RETOS FUTUROS.	24
1.2.2.1. Objetivos planteados para mejorar el sistema carretero nacional.	26
<b>1.3. NECESIDAD DE CONTAR CON UN MEDIDOR DE AFORO VEHICULAR PARA LAS AUTOPISTAS ESTATALES Y FEDERALES DEL PAÍS.</b>	<b>26</b>
1.3.1. ANÁLISIS DEL ESQUEMA GENERAL DE AUTOPISTAS 1989-1994.	27
1.3.1.1. Concesión de autopistas en México.	27
1.3.1.2. Deficiencias en el diseño del esquema de concesión.	27
1.3.1.3. Planeación del esquema.	28
1.3.1.4. El Programa Nacional de Autopistas.	29
1.3.2. HACIA UN NUEVO ESQUEMA DE CONCESIÓN DE AUTOPISTAS Y UTILIDAD DEL CONTADOR DE AFORO VEHICULAR.	30
<b>1.4. ESTADÍSTICAS DE AFORO VEHICULAR EN LAS PRINCIPALES AUTOPISTAS Y PUENTES DE CUOTA DEL PAÍS.</b>	<b>30</b>
<b>1.5. VEHÍCULOS MÁS COMUNES Y CARRETERAS FEDERALES EN MÉXICO.</b>	<b>31</b>
1.5.1. DIMENSIONES DE VEHÍCULOS.	31
1.5.2. DIMENSIONES DE NEUMÁTICOS.	37
1.5.3. TABLA DE CLASIFICACIÓN DE VEHÍCULOS SEGÚN CAPUFE.	38
1.5.4. CARRETERAS FEDERALES EN MÉXICO.	39
1.5.4.1. Carretera Tipo ET.	39
1.5.4.2. Carretera Tipo A.	39
1.5.4.3. Carreteras Tipo B.	39

1.5.4.4. Carreteras Tipo C.	40
1.5.4.5. Carreteras Tipo D.	40
<b>2. MARCO TEORICO.</b>	<b>43</b>
<b>2.1. FAMILIA DE MICROCONTROLADORES 68HCXX DE MOTOROLA.</b>	<b>43</b>
2.1.1. 68HC11.	43
2.1.2. 68HC05.	44
2.1.3. 68HC16.	44
2.1.4. EL 68HC16.	44
2.1.5. 68HC08.	44
2.1.6. 68HC12.	45
2.1.6.1. Familia 68HC12 A.	45
2.1.6.2. Familia 68HC12 B.	45
2.1.6.3. Familia 68HC12 D.	45
<b>2.2. MICROCONTROLADOR 68HC912B32.</b>	<b>45</b>
2.2.1. CARACTERÍSTICAS DEL 68HC912B32 .	46
2.2.1.1. Unidad Central de Procesos.	47
2.2.1.2. Características y Beneficios Adicionales incluyen:	47
2.2.1.3. Diseño Modular Flexible.	48
2.2.1.4. Bajo Consumo, Bajo Voltaje, Bajo Ruido.	48
2.2.1.5. Memoria FLASH EEPROM.	48
2.2.1.6. Memoria EEPROM borrrable por byte.	49
2.2.1.7. Interfaz de Comunicación Serial (SCI).	49
<b>2.3. LENGUAJES DE PROGRAMACIÓN.</b>	<b>50</b>
2.3.1.1. Lenguaje C.	51
2.3.1.2. Visual Basic.	51
2.3.2. COMPILADOR CRUZADO ICC12.	51
2.3.3. LENGUAJE DE ALTO NIVEL CONTRA LENGUAJE ENSAMBLADOR.	52
2.3.4. EL COMPILADOR DE C DE IMAGECRAFT.	52
2.3.4.1. Características del compilador.	52
<b>3. SISTEMA AFORADOR VEHICULAR.</b>	<b>55</b>
<b>3.1. SISTEMA AFORADOR VEHICULAR.</b>	<b>55</b>

3.1.1. ALIMENTACIÓN.	56
3.1.2. ALMACENAMIENTO DE DATOS.	56
3.1.3. TARJETA DE ADQUISICIÓN DE DATOS.	56
3.1.4. TRANSMISIÓN Y RECEPCIÓN DE DATOS.	56
3.1.5. MICROCONTROLADOR MC68HC912B32.	56
<b>3.2. DIAGRAMA DE FLUJOS DEL SISTEMA</b>	<b>56</b>
<b>3.3. SENSORES DE PISO Y SU CARACTERIZACIÓN.</b>	<b>57</b>
3.3.1. BOTONES (SENSORES DE PISO).	57
3.3.2. REBOTE.	58
3.3.3. CARACTERIZACIÓN DEL REBOTE.	60
3.3.4. VENTANA DE ACTIVACIÓN DE BOTONES.	62
<b>3.4. DISEÑO DE TARJETA DE ADQUISICIÓN DE DATOS.</b>	<b>63</b>
<b>3.5. ACOPLAMIENTO.</b>	<b>66</b>
3.5.1. CIRCUITO COMPLETO DE LA TARJETA DE ADQUISICIÓN DE DATOS.	67
<b>3.6. SIMULACIONES Y RESULTADOS.</b>	<b>68</b>
<b>3.7. CIRCUITO IMPRESO.</b>	<b>68</b>
<b>3.8. ADQUISICIÓN Y PROCESAMIENTO DE DATOS EN EL MICROCONTROLADOR MC68HC912B32.</b>	<b>70</b>
3.8.1. RELOJ INTERNO.	70
3.8.2. SISTEMA DE INTERRUPCIÓN EN TIEMPO REAL.	71
3.8.2.1. Funciones de Reloj.	71
3.8.2.2. Interrupción en tiempo real.	72
3.8.2.3. Registro de control de la interrupción en tiempo real.	72
3.8.2.4. Funciones del módulo RTI.	73
3.8.2.5. Archivo de encabezado Rti.h.	74
3.8.2.6. Función Iniciar_Rti().	74
3.8.2.7. Función Rai_Rti().	75
3.8.3. MEMORIA EEPROM.	77
3.8.3.1. Memoria EEPROM en el MC68HC912B32.	77
3.8.3.2. Algoritmo de programación.	78
3.8.3.3. El registro EEPROG.	79
3.8.4. FUNCIONES DEL MÓDULO EEPROM.	80
3.8.4.1. Archivo de Encabezado Eeprom.h.	80
3.8.4.2. Función Iniciar_EE().	81

3.8.4.3. Función EE_tprog().	82
3.8.4.4. Función Guardar(tipo, tiempo).	83
3.8.4.5. Función Borrar().	85
3.8.5. COMUNICACIÓN SERIAL (SCI).	86
3.8.5.1. Interfaz de comunicación serial.	87
3.8.5.2. Formato de Datos.	88
3.8.5.3. Generador de Tasa de transferencia de datos (Baud Rate Generator).	89
3.8.5.4. El registro SC0CR2.	89
3.8.5.5. Protocolo de comunicación.	91
3.8.5.6. Formato de las tramas.	91
3.8.5.7. Formato de los datos.	91
3.8.6. FUNCIONES DEL MÓDULO SCI.	93
3.8.6.1. Archivo de encabezados Sci.h.	94
3.8.6.2. Función Inicia_Sci().	94
<b>3.9. SISTEMA DE AFORO VEHICULAR A ESCALA Y TEORÍA PARA SU IMPLEMENTACIÓN.</b>	<b>126</b>
3.9.1. MODELO A ESCALA	126
3.9.2. PROPUESTA PARA LA IMPLEMENTACIÓN DEL SISTEMA.	127
3.9.3. TABLA COMPARATIVA	129
<b><u>4. PROGRAMA DE RECOLECCIÓN DE DATOS AFORO 1.0</u></b>	<b><u>132</u></b>
<b>4.1. MÓDULO ABRIR</b>	<b>136</b>
<b>4.2. MANUAL PARA EL USUARIO DEL PROGRAMA.</b>	<b>140</b>
4.2.1. INSTALACIÓN.	140
4.2.1.1. Requerimientos del Sistema	140
4.2.1.2. Como Instalar:	140
4.2.2. PRINCIPAL	141
4.2.2.1. Comunicación.	143
4.2.2.2. Abrir.	144
4.2.2.3. Gráficas.	145
<b>4.3. TUTORIAL.</b>	<b>147</b>
<b><u>5. PRUEBAS Y RESULTADOS.</u></b>	<b><u>150</u></b>
<b>5.1. PRUEBAS DE DEPURACIÓN.</b>	<b>150</b>

---

5.2. ANÁLISIS DE RESULTADOS.	152
5.3. PRUEBAS TERMINALES.	153
5.4. SIMULADOR DEL CONTADOR DE AFORO VEHICULAR.	153
5.5. RESULTADOS ARROJADOS POR EL CONTADOR DE AFORO VEHICULAR.	155
<b>CONCLUSIONES.</b>	<b>160</b>
<b>APÉNDICE A</b>	<b>162</b>
<b>APENDICE B.</b>	<b>183</b>
<b>BIBLIOGRAFÍA</b>	<b>194</b>
LIBROS	194
INTERNET	195
OTRAS FUENTES DE CONSULTA.	196

---

Tabla 1. Longitud de carreteras por kilómetro cuadrado de territorio .....	20
Tabla 2. Longitud de la red carretera en el 2000.....	21
Tabla 3. Red Troncal CAPUFE.....	22
Tabla 4. Características de las redes carreteras .....	24
Tabla 5. Aforo garantizado .....	29
Tabla 6. Aforo en autopistas de cuota por tipo de vehiculo.....	31
Tabla 7. Crecimiento de la red carretera.....	31
Tabla 8. Autos mas comunes en México.....	32
Tabla 9. Dimensiones del VW sedan.....	32
Tabla 10. Dimensiones del Pointer.....	33
Tabla 11 Dimensiones del Derby.....	33
Tabla 12 Dimensiones del Jetta .....	33
Tabla 13. Dimensiones del Tsuru.....	34
Tabla 14. Dimensiones del Atos .....	34
Tabla 15. Dimensiones de la Ram Van.....	34
Tabla 16 Dimensiones de la Ram. ....	35
Tabla 17 Dimensiones del Interntional 4300. ....	35
Tabla 18. Especificaciones del Internacional 4400. ....	36
Tabla 19. Especificaciones Aerobús FC 198-115. ....	36
Tabla 20 Especificaciones del Internacional 9200i.....	37
Tabla 21. Medidas de neumáticos de automóvil .....	37
Tabla 22. Medidas de neumáticos de camioneta.....	38
Tabla 23. Medidas de neumáticos de camión .....	38
Tabla 24. Medidas de neumáticos .....	38
Tabla 25. Clasificación de CAPUFE .....	39
Tabla 26. Clasificación de carreteras.....	40
Tabla 27. Lecturas .....	60

---

---

Tabla 28. RTICTL.....	72
Tabla 29. RTR .....	73
Tabla 30. EEPROG .....	79
Tabla 31. Byte, Row.....	79
Tabla 32. Generador de tasa de transferencia.....	89
Tabla 33. SCOCR2.....	89
Tabla 34. Vectores de Interrupción .....	102
Tabla 35. Tabla comparativa de los dos sistemas.....	130
Tabla 36. Requerimientos del sistema.....	140
Tabla 37. Descripción de uso .....	143
Tabla 38. Comunicación con el sistema .....	144
Tabla 39. Resultados .....	145
Tabla 40. Descripción de graficar.....	146
Tabla 41. Resultados de depuración .....	152
Tabla 42. Resultados de prueba.....	155
Tabla 43. Clasificación de prueba.....	155
Tabla 44. Resultados arrojados por el aforador .....	156
Tabla 45. Graficas de resultados.....	157
Tabla 46. Detalle.....	159

---

Ilustración 1. Dimensión de autopista.....	41
Ilustración 2. Registros 68HC12.....	47
Ilustración 3. SCI.....	49
Ilustración 4. Tarjeta de evaluación.....	50
Ilustración 5. Compilador ICC12.....	53
Ilustración 6. Diagrama de flujos del sistema de aforo vehicular.....	57
Ilustración 7. Dimensión de los sensores.....	58
Ilustración 8. Código para medir rebote.....	59
Ilustración 9. Caracterización del rebote.....	60
Ilustración 10. Muestra para sensor nuevo.....	61
Ilustración 11. Muestra para sensor viejo.....	61
Ilustración 12. Ventana de activación de sensores.....	62
Ilustración 13. Ventana Promedio.....	62
Ilustración 14. Diagrama de Bloques de la tarjeta de adquisición de datos.....	63
Ilustración 15. Puertos en la tarjeta de evaluación.....	64
Ilustración 16. Circuito de lectura.....	65
Ilustración 17. Compuerta and.....	66
Ilustración 18. Diagrama completo de la tarjeta de adquisición de datos.....	67
Ilustración 19. Circuito impreso.....	69
Ilustración 20. Circuito impreso cara superior.....	70
Ilustración 21. Circuito impreso cara inferior.....	70
Ilustración 22. Interrupción en tiempo real.....	72
Ilustración 23. Rti.h.....	74
Ilustración 24. Rti.....	75
Ilustración 25. Iniciar Rti.....	75
Ilustración 26. Función Rti.....	76
Ilustración 27. Código Rti.....	76

---

Ilustración 28. Mapa de memoria.....	78
Ilustración 29. eeprom.h.....	81
Ilustración 30. Diagrama eeprom.....	82
Ilustración 31. Código eeprom.....	82
Ilustración 32. Función ee_tprog.....	83
Ilustración 33. ee_tprog.....	83
Ilustración 34. diagrama guardar tiempo.....	84
Ilustración 35. Función guardar tiempo.....	85
Ilustración 36. Diagrama Borrar.....	86
Ilustración 37. borrar.....	86
Ilustración 38. Comunicación serial sci.....	88
Ilustración 39. Sci.h.....	94
Ilustración 40. Propuesta de ubicación del sistema.....	128
Ilustración 41. Ubicación de sensores.....	128
Ilustración 42. Comunicación.....	134
Ilustración 43. Programa en pc.....	141
Ilustración 44. Ambiente del programa aforo.....	142
Ilustración 45. Resumen.....	144
Ilustración 46. Graficas.....	146
Ilustración 47. Resultados.....	156
Ilustración 48. Detalle de resultados.....	158

---

Diagrama 1. Diagrama de bloques.....	55
Diagrama 2. Iniciar_Sci() .....	94
Diagrama 3. Rai_Sci() .....	96
Diagrama 4. Mandar().....	99
Diagrama 5. Reiniciar().....	100
Diagrama 6. IRQ_Cuenta(testigo) .....	105
Diagrama 7. Irq_Push(tipo_eje).....	106
Diagrama 8. Irq_Buscar().....	108
Diagrama 9. Selecciona(s).....	111
Diagrama 10. IRQ_F1() .....	112
Diagrama 11. IRQ_F2() .....	113
Diagrama 12. IRQ_F3() .....	114
Diagrama 13. IRQ_F4() .....	115
Diagrama 14. IRQ_F5().....	117
Diagrama 15. IRQ_F6().....	119
Diagrama 16. IRQ_F7().....	121
Diagrama 17. IRQ_F8().....	122
Diagrama 18. Rai_Irq().....	124

---

Código 1. Inicia_Sci() .....	95
Código 2. Rai_Sci() .....	97
Código 3. Mandar() .....	99
Código 4. Reiniciar() .....	100
Código 5. Irq.h.....	104
Código 6. Inicia_IRQ().....	104
Código 7. IRQ_Cuenta(testigo).....	106
Código 8. Irq_Push(tipo_eje) .....	107
Código 9. Irq_Buscar() .....	109
Código 10. Selecciona(s) .....	112
Código 11. IRQ_F2().....	113
Código 12. IRQ_F2() .....	114
Código 13. IRQ_F3() .....	115
Código 14. IRQ_F4() .....	116
Código 15. IRQ_F5() .....	118
Código 16. IRQ_F6() .....	120
Código 17. IRQ_F7() .....	122
Código 18. IRQ_F8() .....	123
Código 19. Rai_Irq() .....	125

---

## INTRODUCCIÓN.

Se conocen como medidores de aforo, a las instalaciones que se destinan a la medición de flujos. Este flujo puede ser de líquido (aguas de un río o líquido que circula por una conducción artificial cerrada o abierta) o de tráfico. Por ejemplo, se utilizan medidores de aforo portátiles o fijos para la medición del flujo de vehículos por un punto dado de la red viaria. Estas estaciones consisten en un sistema de equipos contadores y clasificadores que registran el paso de los diferentes tipos de vehículos que pasan por ahí.

Los medidores de aforo vehicular son de suma importancia en la actualidad, ya que se utilizan para planificar y gestionar una red carretera. Estos medidores deben de contar con información confiable y oportuna, además de funcionar automáticamente los 365 días de año. En la actualidad existen varios tipos de medidores de aforo vehicular o equipos contadores de tránsito usados en diversos países del mundo, algunos de ellos utilizan para detectar vehículos, espiras electromagnéticas y sensores piezoeléctricos, y otros mas sofisticados que cuentan con un arco metálico dotado de sensores que miden flujo, velocidad y peso de los vehículos que transitan bajo el, además también están dotados de detectores de proximidad y cámaras, todo enviado a una central por medio de comunicación inalámbrica. En ambos casos deben contar con una fuente ilimitada de poder y estar ubicados en una zona de fácil monitoreo.

## OBJETIVO

Diseñar e implementar un sistema contador de aforo vehicular prototipo que mida el paso de los vehículos clasificándolos por tipos, según el ancho de los neumáticos y el número de ejes, además de almacenar la fecha y hora en que ocurre el evento. Utilizando un microcontrolador de 16 bits, en este caso el MC68HC912B32 de Motorola. Así como implementar una aplicación en Visual Basic para la descarga de datos y la fácil interpretación de los mismos.

La finalidad de este proyecto es ubicar el sistema contador de aforo vehicular en una autopista federal o estatal, con el propósito de poder cuantificar el paso de vehículos y con ello proporcionar estadísticas confiables para la inversión de capital en las mismas, entre otros usos, dando además la teoría básica de su implementación.

## CAPITULO 1

### ANTECEDENTES Y ANÁLISIS DE NECESIDADES

## 1. ANTECEDENTES Y ANÁLISIS DE NECESIDADES.

### 1.1. Red Nacional de Caminos.

Las carreteras de México enlazan por igual tanto a los principales centros de población, como a los de producción y consumo del país. Este sistema carretero cuenta con 5,683 kilómetros de autopistas federales de cuota, y 147,456 kilómetros conforman la red de caminos rurales. En sus diferentes modalidades, por este sistema se movilizan anualmente más de 2,700 millones de personas y alrededor de 620 millones de toneladas de carga, lo que lo convierte en el principal medio de traslado en el país, equivalente al 60% del tonelaje total de carga que circula por el territorio, y al 98% de los pasajeros. El uso de la red carretera del país se ha quintuplicado de 1960 a la fecha, con un crecimiento anual del 9%.

Las autopistas y puentes de cuota han sido factor fundamental en el desarrollo económico, político, social y cultural de México durante los últimos 50 años, ya que han permitido la creación de infraestructuras tan importantes como la educativa, la hidráulica, la agrícola, la urbana y la de salud, entre otras, a la vez que han integrado y comunicado a diversas zonas y regiones, lo que ha facilitado su articulación con el resto de la República.

### 1.2. Historia.

En nuestro país la historia de las carreteras se remonta hasta la época prehispánica, caracterizada por la existencia de caminos amplios y bien hechos, como las calzadas empedradas, que cruzaban el lago de Texcoco para unir con tierra firme la isla de Tenochtitlán, capital del Imperio Azteca. Sin embargo, predominaban los senderos, algunos de considerable extensión.

A la llegada de los españoles a México, con la introducción de las bestias y de los vehículos para el transporte de personas y mercancías, los senderos resultaron inadecuados, por lo que fue necesario ampliarlos, cambiar las rutas y suavizar las pendientes. Se abrieron nuevos caminos y se extendieron los ya existentes: así, al final del Virreinato había en México 7 mil kilómetros de “caminos reales” por los que se podía transitar en carreteras y 19,000 kilómetros de caminos de herradura.

La orientación de los caminos obedeció, en ese entonces, a los intereses de los conquistadores españoles por nuestros recursos naturales y los de otros territorios cercanos, a fin de satisfacer las necesidades del Imperio y comerciar con el Oriente.

De esta manera, los principales caminos que se construyeron para unir a la ciudad de México con el resto del territorio, fueron los siguientes: el de Veracruz; el de Acapulco; el de Santa Fe del Nuevo México; el de Guadalajara y el de Guatemala vía Oaxaca. Estos caminos pasaban por varias ciudades intermedias como Puebla, Taxco y Durango y contaban con ramales que servían de comunicación con otras ciudades importantes.

En 1821 México se hace independiente y es hasta 1867, en que el Presidente Juárez destino por primera vez en la historia de México una parte de presupuesto a la apertura y conservación de caminos, no se hicieron obras importantes en este aspecto.

Durante el Porfiriato, a fines del siglo pasado, se hizo énfasis en la construcción de la red ferroviaria. Poco se realizó en materia de caminos y los que se construyeron, tenían por principal objetivo alimentar las estaciones de los ferrocarriles y en menor grado comunicar zonas que carecían de vías férreas.

La incorporación del automóvil marca otra época en la historia de las carreteras. En 1925 los automovilistas de México estaban limitados a transitar únicamente en calles y calzadas urbanas. El transporte de personas y mercancías de una ciudad a otra tenía que hacerse a través de los ferrocarriles.

Con el objeto de dotar al país de una amplia red de carreteras, se creó en 1925 la Comisión Nacional de Caminos que constituyó al antecedente de lo que hoy es la Secretaría de Comunicaciones y Transporte S.C.T.

La expansión económica experimentada a partir de 1925 determinó que el país contara para 1930 con 1,426 kilómetros de carreteras, cifra que se elevó a 9,929 para 1940. A partir de entonces, la construcción de carreteras ha ido en aumento hasta los últimos años, de tal manera que la longitud de la red para el 2003 llega a los 365,119 kilómetros. Con la construcción de las carreteras no solo han crecido las ciudades o se han creado nuevos centros de población, sino que también se ha propiciado el acercamiento de los centros de producción, tanto agrícolas como industriales, con los núcleos de consumo; la producción que antes se destinaba solo al consumo local ha encontrado nuevos mercados en beneficio del productor y de la misma

economía nacional. De esta forma, la red de caminos a la vez que favorece el desarrollo de la cultura entre la ciudadanía, asegura y fortalece la unidad nacional. De ahí que los auto transportes ocupen en México el primer lugar dentro de los medios de transporte.

**1821**, El Marqués Miguel La Grúa Talamanca y Branciforte emite la disposición para el pago del peaje, que a partir del 6 de julio se cobrará en el nuevo camino que va de la capital a la ciudad de Toluca, así como las reglas que se han de observar en el tránsito de dicho camino y las penas en que incurrir los contraventores a ellas.

Por decreto de la soberana Junta Provisional Gubernativa se establecen cuatro ministerios: de Relaciones Exteriores e Interiores; de Justicia y Negocios Eclesiásticos; de Hacienda Pública y de Guerra y Marina. Correspondió al primero lo inherente a las obras públicas y lo relativo a comunicaciones, se despachaba en otros ministerios.

**1857**, Se funda la Administración General de Caminos de Peajes, como un primer intento por consolidar las funciones encaminadas a satisfacer las necesidades de comunicación en el ámbito nacional.

**1861**, El presidente Benito Juárez establece la Secretaría de Fomento, Comunicaciones y Obras Públicas, integrando funciones de diversos ministerios.

**1917**, Las funciones de Obras Públicas se trasladan a la Secretaría de Gobernación y entonces queda solo como Secretaría de Comunicaciones.

Las funciones de Obras Públicas salen de la Secretaría de Gobernación y se restituye la Secretaría de Comunicaciones y Obras Públicas.

**1931**, Desaparece la Comisión Nacional de Caminos y con sus funciones se crea la Dirección Nacional de Caminos que se integra a la S.C.O.P.

**1932**, Se instala en cada entidad federativa un organismo denominado Junta Local de Caminos para la construcción de caminos alimentadores, bajo el sistema de cooperación bipartita; con la participación del gobierno federal y de los gobiernos de los estados. Se expide el Reglamento de Tránsito, de los Caminos Nacionales y de los Particulares de Concesión Federal.

**1946**, Acuerdo por el que se autoriza el libre tránsito de los camiones de carga por las carreteras federales.

**1949,** Se crea como empresa subsidiaria de Nacional Financiera, la "Compañía Constructora del Sur, S.A. de C.V. para construir carreteras de altas especificaciones.

Se establece el Comité Nacional de Caminos Vecinales para la construcción de caminos de la comunidad, bajo el sistema de participación tripartita, con la participación del gobierno federal, de los gobiernos de los estados y los particulares.

**1952,** Se inauguran las primeras dos carreteras de cuota del país: México-Cuernavaca y Amacuzac-Iguala, y su administración y operación se encomiendan a la misma compañía que las construyó.

Se publica el acuerdo que dicta las medidas de seguridad y vigilancia en las carreteras para el servicio público y particular de auto transporte de jurisdicción federal.

**1958,** Se crea el organismo público federal descentralizado del gobierno federal "Caminos Federales de Ingresos", adscrito a la Secretaría de Comunicaciones y Obras Públicas, de la que el primer presidente del consejo de administración fue el subsecretario de Obras Públicas.

**1959,** La Secretaría de Comunicaciones y Obras Públicas (SCOP) se transforma en la Secretaría de Comunicaciones y Transportes (SCT) y se crea la de Obras Públicas (SOP) a la que se asigna el organismo y preside su consejo.

Se crea el organismo público federal descentralizado "Caminos y Puentes Federales de Ingresos".

Se publica la Ley que crea el Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado, en lugar de la Dirección de Pensiones.

**1975,** Se publica el Reglamento de Tránsito en Carreteras Federales, que deroga el Reglamento de Tránsito del 21 diciembre de 1932.

**1977,** El organismo CAPUFE, se adscribe al sector Comunicaciones y Transportes, cuyo titular preside el consejo de administración.

**1983,** La Secretaría de Asentamientos Humanos y Obras Públicas (SAHOP) se transforma en Secretaría de Desarrollo Urbano y Ecología (SEDUE), con nuevas atribuciones, y las obras públicas se transfieren a la Secretaría de Comunicaciones y Transportes (SCT).

---

**1985,** Se publica el decreto por el que se reestructura la organización y funcionamiento del organismo público descentralizado "Caminos y Puentes Federales de Ingresos y Servicios Conexos".

Se trasladan a Cuernavaca, Mor. Las oficinas centrales de CAPUFE, que estaban en la ciudad de México.

**1987,** Por acuerdo de la Secretaría de Comunicaciones y Transportes, de Programación y Presupuesto y, de Hacienda y Crédito Público, las carreteras de cuota cuya construcción puede ser financiada sin recursos federales, serán canalizadas a fideicomisos privados constituidos por el Banco Nacional de Obras y Servicios Públicos como institución fiduciaria, previo otorgamiento de la concesión para la primera de las dependencias y que la operación y mantenimiento de las vías de cuota, serán encargadas a "Caminos y Puentes Federales de Ingresos y Servicios Conexos".

**1990,** Con la autorización de la Secretaría de Comunicaciones y Transportes, de Programación y Presupuesto y, Hacienda y Crédito Público, Caminos y Puentes inició la aportación económica al Programa de Carreteras Concesionadas para mejorar la vialidad financiera de distintos proyectos carreteros privados.

Entrega la SCT a CAPUFE el tramo Zacapalco-Rancho Viejo, de 17.3 Km. y la caseta de cobro de la carretera Amacuzac-Taxco. Se establece el seguro del usuario en las instalaciones carreteras de CAPUFE.

**1994,** Entra en vigor el Tratado de Libre Comercio de América del Norte, entre México, Canadá y Estados Unidos de América. (1 de enero) Se publica la norma oficial sobre el peso y dimensiones máximas que deben cumplir los vehículos de auto transporte que transitan en los caminos y puentes de jurisdicción federal.

**1995,** Se publica en el Diario oficial de la Federación el decreto que adiciona el diverso por el que se reestructura la organización y funcionamiento del organismo público descentralizado Caminos y Puentes Federales de Ingresos y Servicios Conexos.

**1997,** El Diario Oficial de la Federación publica el decreto por el que se declara de utilidad e interés público la concesión de 23 carreteras.

**1998,** CAPUFE recibe de BANOBRAS la transferencia de 23 concesiones rescatadas para su operación, conservación y mantenimiento mayor y menor.

### 1.2.1. Situación actual.

Por su importancia y características, la red carretera mexicana se clasifica en: red federal, redes estatales, caminos rurales y brechas mejoradas. Esta red se ha desarrollado de manera gradual a lo largo de varias décadas, comunica a casi todas las regiones y comunidades del país a través de más de 333 mil kilómetros de caminos de todos tipos.

La red federal de carreteras es atendida en su totalidad por el gobierno federal. Registra la mayor parte de los desplazamientos de pasajeros y carga entre ciudades y conduce los recorridos de largo itinerario, los relacionados con el comercio exterior y los producidos por los sectores más dinámicos de la economía nacional. Las redes estatales cumplen una función de gran relevancia para la comunicación regional, para enlazar las zonas de producción agrícola y ganadera y para asegurar la integración de extensas áreas en diversas regiones del país. Por su parte, los caminos rurales y las brechas mejoradas son vías modestas y en general no pavimentadas; su valor es más social que económico, pues proporcionan acceso a comunidades pequeñas que de otra manera estarían aisladas. Sin embargo, su efecto en las actividades y la calidad de vida de esas mismas comunidades es de gran trascendencia.

Si bien la red carretera posee una importancia de primer orden para nuestro país, que se destaca en el mundo por su vocación para construir caminos, tanto la extensión del territorio nacional como las limitaciones presupuestales que ha padecido y otros factores inciden en el hecho de que México posea una densidad carretera (es decir, la longitud de carreteras por kilómetro cuadrado de territorio) relativamente baja, como lo podemos ver en la Tabla 1.

México	EU	Canadá	Francia	Alemania	Italia	Japón	Reino Unido
0.14	0.64	0.10	1.62	1.77	1.04	3.04	1.61

**Tabla 1. Longitud de carreteras por kilómetro cuadrado de territorio**

En México, entre 1995 y 2000 se construyeron o modernizaron 10 mil 371.7 kilómetros, de los que 6 mil 521.8 fueron carreteras pavimentadas y 3 mil 849.9, caminos rurales.

Para finales del año 2000, la red carretera nacional tenía una longitud total de 333 mil 247.1 kilómetros, de los que 106 mil 571.5 corresponden a carreteras libres, 5 mil 933.1 a autopistas de cuota, 160 mil 185.1 a caminos rurales y alimentadores y 60 mil 557.4 a brechas.

De las carreteras libres, 41 mil 865.8 kilómetros pertenecen a la red federal, mientras que 64 mil 705.7 están distribuidos entre las 31 redes estatales. Por lo que se refiere a las autopistas de cuota, la red operada por CAPUFE, integrada por su red propia y la que opera por cuenta de terceros —incluida la red del Fideicomiso de Apoyo para el Rescate de Autopistas Concesionadas (FARAC), tiene una longitud de 4 mil 714.7 kilómetros, las concesionadas a particulares cuentan con 786 kilómetros y 432.4 son concesiones estatales de cuota (Tabla 2).

Longitud de la red carretera en el 2000.

CARACTERÍSTICAS	LONGITUD (Km.)
Red federal	47,366.5
Cuota	5,500.7
Libre	41,865.8
Redes estatales	65,138.1
Cuota	432.4
Libre	64,705.7
Red de caminos rurales	160,185.1
SCT	4,596.9
Estados	108,530.2
Otros	47,058.0

**Tabla 2. Longitud de la red carretera en el 2000**

Los caminos rurales en su gran mayoría se han transferido a los gobiernos de los estados, por lo que la SCT sólo mantiene la jurisdicción directa de 4 mil 596.9 kilómetros. El total de caminos rurales a cargo de los estados suma 108 mil 530.2 kilómetros, mientras que el total que es responsabilidad de municipios y otras dependencias es de 47 mil 58 kilómetros. Las brechas se reparten en todo el país y suman 60 mil 557.4 kilómetros.

Del total de kilómetros en servicio, 107 mil 822.4 están pavimentados, 145 mil 350 corresponden a carreteras revestidas, 19 mil 517.3 son de terrecerías y 60 mil 557.4 son brechas. Entre los caminos pavimentados, una longitud de 9 mil 872.6 kilómetros (9.2 por ciento del total) tiene cuatro o más carriles.

Los principales aspectos de la problemática de la red federal de carreteras son el deficiente estado físico en que se encuentra, las limitaciones geométricas y de capacidad de una longitud importante de los corredores que constituyen sus tramos más utilizados, la todavía insuficiente cobertura, el mal estado de los caminos rurales y la necesidad de consolidar el funcionamiento del sistema nacional de autopistas de cuota, tanto en lo que se refiere a sus

aspectos financieros como a cuestiones operativas y de conservación. Las condiciones físicas en que se encuentra la red federal de carreteras se clasifica en 25% buenas, 35% regulares y 40% malas.

Uno de los elementos más importantes de la red carretera nacional lo constituyen los 14 corredores que conectan las cinco mezo regiones con que cuenta el país y que proporcionan acceso y comunicación permanente a las principales ciudades, fronteras y puertos marítimos.

La longitud de estos corredores, que atienden poco más del 54 por ciento de los flujos carreteros interregionales, es de 19 mil 263 kilómetros. Actualmente se encuentran en proceso de modernización total, mismo que, hasta el año 2000, había avanzado en un 60.8 por ciento, lo que equivale a 11 mil 714 kilómetros. Así, se encuentran pendientes de modernizar otros 7 mil 549 kilómetros, es decir, el 39.2 por ciento de su longitud (Tabla 3). Por lo que se refiere a las autopistas de cuota, México cuenta con uno de los sistemas mas extensos del mundo con una longitud total e 5 mil 933 kilómetros integrado por 74 autopistas y 48 puentes.

La red a cargo de Caminos y Puentes Federales de Ingresos y Servicios Conexos (CAPUFE) está compuesta por vías construidas por el Estado que fueron entregadas a ese organismo para su administración y explotación. En general se trata de carreteras maduras con altos niveles de aforo e ingreso.

CORREDOR	LONGITUD TOTAL (Km.)
México-Nogales con ramal a Tijuana	3,074
México-Nuevo Laredo con ramal a Piedras Negras	1,735
Querétaro-Ciudad Juárez	1,770
Acapulco-Tuxpan	830
Mazatlán-Matamoros	1,245
Manzanillo-Tampico con ramal a Lázaro Cárdenas y Ecuandureo	1,856
Acapulco-Veracruz	851
Veracruz-Monterrey con ramal a Matamoros	1,297
Transpeninsular de Baja California	1,776
Altiplano	581
Puebla -Progreso	1,320
Puebla-Oaxaca-Ciudad Hidalgo	1,007
Transistmico	702
Peninsular de Yucatán	1,219
TOTAL	19,263

**Tabla 3. Red Troncal CAPUFE**

La red FARAC está integrada por las vías rescatadas en 1997, más las que han entrado en operación a partir de 1998. Se trata de carreteras nuevas concesionadas a BANOBRAS, institución nacional de crédito, cuyos flujos de vehículos e ingresos no han alcanzado todavía una madurez plena.

La red concesionada se forma por autopistas construidas entre 1989 y 1994 que fueron concesionadas a particulares, gobiernos estatales e instituciones financieras. También se trata de vías relativamente nuevas con niveles de aforo e ingreso variables, pues hay algunas maduras y otras que no lo son tanto.

En cuanto a su administración, presupuesto y operación, las tres redes son diferentes, como se observa en la Tabla 4.

CARACTERÍSTICA	CAPUFE	FARAC	CONCESIONADAS
Red	Autopistas construidas por el Estado y entregadas a Capufe para su administración y explotación.	Autopistas rescatadas en 1997 y las que sct ha puesto en operación a partir de 1998.	Autopistas construidas sobre todo entre 1989 y 1994; están concesionadas a particulares, gobiernos estatales o instituciones financieras.
Autopistas predominantes	Maduras, con altos niveles de ingresos.	Nuevas, construidas entre 1989 y 1994; en proceso de maduración.	Relativamente nuevas, con excepción de concesiones otorgadas para lograr objetivos financieros.
Propiedad	Patrimonio nacional.	Patrimonio nacional. Concesionadas a Banobras, institución fiduciaria, que contrata a Capufe para operar y administrar.	Patrimonio nacional. Concesionadas a terceros para pago de créditos y recuperación de inversiones.
Tarifas	Autorizadas por shcp.	Autorizadas por el Comité Técnico del farac.	Autorizadas por sct bajo reglas específicas.
Presupuesto	Autorizado por el H. Congreso de la Unión y controlado por la shcp.	Autorizado por el Comité Técnico del farac.	Autorizado por el Comité Técnico de cada fideicomiso de administración de la concesión.

Decisiones	Consejo de Administración de Capufe.	Comité Técnico del farac.	Comité Técnico de Fideicomisos de Administración.
------------	--------------------------------------	---------------------------	---

**Tabla 4. Características de las redes carreteras**

En materia de caminos rurales, el país cuenta con una red de 160 mil 185.1 kilómetros y comunica a 19 mil comunidades en las que habitan 14 millones de personas, es decir, el 60 por ciento de la población rural nacional. Esta red está compuesta por caminos de bajas especificaciones, que son transitables en toda época del año y desempeñan un papel fundamental en la comunicación e integración permanentes de las comunidades a las que sirven.

### **1.2.2. Retos futuros.**

El sub. sector carretero, durante los últimos años, se ha considerado como elemento principal del sistema de transporte nacional. El crecimiento de la economía y del ingreso, las tendencias demográficas, la necesidad de contar con servicios de transporte cada vez más eficientes y confiables en todo el territorio, permiten anticipar que la demanda nacional de transporte por carretera seguirá creciendo en todas las regiones del país, así como para toda clase de carreteras.

Por lo que toca al sistema de autopistas de cuota, se presentan las oportunidades de consolidar el manejo de la red cuya operación está a cargo directo del Estado; introducir nuevos esquemas de administración, operación y cobro de peajes; expandir la red mediante la ejecución de nuevos proyectos basados en esquemas financieros en los que participe el sector privado, y de depurar definitivamente las finanzas del sistema.

Independientemente de la naturaleza y magnitud de los desafíos que deberá superar el sistema carretero nacional durante el periodo 2000-2006, las condiciones del entorno y las modalidades de trabajo necesarias para lograrlo serán distintas a las del pasado reciente. En particular, se reconocen las necesidades de descentralizar recursos y responsabilidades, así como de establecer nuevos y más estrechos mecanismos de colaboración con los gobiernos de los estados.

La transparencia, el estricto apego al marco legal y normativo, la rendición de cuentas y la oportuna difusión de información serán valores esenciales en las tareas de la Secretaría, órgano encargado del manejo del sistema carretero.

En materia de conservación de la red federal de carreteras, las principales tareas a desarrollar estarán orientadas a preservar el importante activo nacional que representa, con objeto de propiciar una operación económica, eficiente y segura de los vehículos.

En lo que se refiere a construcción y modernización de carreteras federales, el principal reto por enfrentar será el de concluir la modernización integral de los 14 corredores a que se aludió anteriormente. A través de estas acciones se buscará asegurar una comunicación eficiente de las principales ciudades, puertos marítimos, cruces fronterizos y centros de producción y distribución ubicados en todo el territorio.

En los caminos rurales existe el doble desafío de seguir extendiendo la cobertura de la red, tan importante para la vida de las comunidades rurales y para la vinculación productiva regional, y asegurar, a la vez, la adecuada conservación del vasto número de caminos existentes.

El reto que se plantea para las autopistas de peaje consiste en consolidar su funcionamiento, homogeneizar sus condiciones de servicio, expandirla y extenderla tanto en cobertura como en tipo y calidad de servicios y sentar las bases para su pleno aprovechamiento como un activo productivo de importancia estratégica para la vida nacional.

Por otro lado, dadas las limitaciones presupuestales que afectan y seguirán afectando al sub. sector, será cada vez más importante diseñar e instrumentar mecanismos financieros de obras carreteras que, con la participación del sector privado y de gobiernos estatales y municipales, permitan aumentar los montos disponibles para inversión en carreteras y acelerar la puesta en operación de importantes proyectos en todo el país.

En síntesis, los años 2000-2006 se presentan como un periodo pleno de retos y oportunidades para el desarrollo y la consolidación del sistema carretero nacional. Como consecuencia, las decisiones que se tomen durante la presente administración y las acciones que se deriven de ellas habrán de influir trascendientemente en el funcionamiento de las carreteras de México por muchos años más allá del 2006.

La misión que la SCT se ha fijado en el ámbito de las carreteras es la de dotar al país de un sistema eficaz, seguro, respetuoso del medio ambiente y permanentemente accesible para



La transparencia, el estricto apego al marco legal y normativo, la rendición de cuentas y la oportuna difusión de información serán valores esenciales en las tareas de la Secretaría, órgano encargado del manejo del sistema carretero.

En materia de conservación de la red federal de carreteras, las principales tareas a desarrollar estarán orientadas a preservar el importante activo nacional que representa, con objeto de propiciar una operación económica, eficiente y segura de los vehículos.

En lo que se refiere a construcción y modernización de carreteras federales, el principal reto por enfrentar será el de concluir la modernización integral de los 14 corredores a que se aludió anteriormente. A través de estas acciones se buscará asegurar una comunicación eficiente de las principales ciudades, puertos marítimos, cruces fronterizos y centros de producción y distribución ubicados en todo el territorio.

En los caminos rurales existe el doble desafío de seguir extendiendo la cobertura de la red, tan importante para la vida de las comunidades rurales y para la vinculación productiva regional, y asegurar, a la vez, la adecuada conservación del vasto número de caminos existentes.

El reto que se plantea para las autopistas de peaje consiste en consolidar su funcionamiento, homogeneizar sus condiciones de servicio, expandirla y extenderla tanto en cobertura como en tipo y calidad de servicios y sentar las bases para su pleno aprovechamiento como un activo productivo de importancia estratégica para la vida nacional.

Por otro lado, dadas las limitaciones presupuestales que afectan y seguirán afectando al sub. sector, será cada vez más importante diseñar e instrumentar mecanismos financieros de obras carreteras que, con la participación del sector privado y de gobiernos estatales y municipales, permitan aumentar los montos disponibles para inversión en carreteras y acelerar la puesta en operación de importantes proyectos en todo el país.

En síntesis, los años 2000-2006 se presentan como un periodo pleno de retos y oportunidades para el desarrollo y la consolidación del sistema carretero nacional. Como consecuencia, las decisiones que se tomen durante la presente administración y las acciones que se deriven de ellas habrán de influir trascendientemente en el funcionamiento de las carreteras de México por muchos años más allá del 2006.

La misión que la SCT se ha fijado en el ámbito de las carreteras es la de dotar al país de un sistema eficaz, seguro, respetuoso del medio ambiente y permanentemente accesible para

**Falta página**

**N° 26**

### ***1.3.1. Análisis del esquema general de autopistas 1989-1994.***

#### ***1.3.1.1. Concesión de autopistas en México.***

El esquema general de concesión de carreteras en México no estableció una estructura de incentivos que lograra distribuir adecuadamente el riesgo. El discurso oficial señalaba que el objetivo central era construir autopistas de altas especificaciones hacia los principales centros productivos para disminuir los costos de comercialización de los productos y superar los rezagos del sector. Todo ello a través de criterios empresariales del manejo de recursos de la provisión privada de la infraestructura. Sin embargo, en la realidad el proceso de concesionamiento respondió más a exigencias políticas, de reputación y credibilidad que demandaba la incorporación de nuestra economía a la dinámica mundial.

Primero, el gobierno se ganó la confianza de los empresarios. A través de BANOBRAS llevó a cabo tres proyectos piloto de autopistas con financiamiento gubernamental, donde el sector privado participó de lleno en la construcción y parcialmente en el financiamiento. Con una señalización positiva sobre los proyectos, resultado de su viabilidad económica y financiera comprobada y las expectativas optimistas que generó; en febrero de 1989 se anunció el Programa Nacional de Autopistas.

#### ***1.3.1.2. Deficiencias en el diseño del esquema de concesión.***

El esquema general de concesiones no fue bien diseñado. No cumplió su principal objetivo y los contratos de concesionamiento sufrieron modificaciones e incluso algunos se renegociaron. Oficialmente estos contratos con término de concesión fija, contemplaban un sistema tarifario con actualización automática, estimaciones de aforo por categoría vehicular y crecimiento garantizado. A la postre, estos factores permitirían obtener los ingresos necesarios para cubrir el capital invertido, los créditos, el servicio de la deuda de los proyectos y un rendimiento adicional.

No obstante, los costos de construcción fueron superiores a los originalmente establecidos, el aforo vehicular resultó inferior al previsto y su estructura financiera no correspondió a la capacidad de generación de ingreso de las autopistas.

### ***1.3.1.3. Planeación del esquema.***

El diseño del contrato fue inadecuado debido entre otras cosas a la inexperiencia del gobierno en la elaboración de este tipo de esquemas lo que llevó a la construcción de un contrato tal que, las condiciones de operación parecían contradecir la propia lógica económica y de planeación de la concesión de autopistas.

Es un tanto contradictorio que, se planeara un esquema de concesión de infraestructura carretera con el objetivo incrementar la eficiencia operativa y económica, cuando para cada vía concesionada existía una vía alterna federal libre. Estas carreteras libres funcionaron como un bien sustituto, incrementaron la elasticidad de los precios de la demanda de la vía concesionada, quitándole su poder de monopolio natural. Si bien estas vías, en el mediano y largo plazo tienen un efecto positivo para los usuarios al presionar al concesionario para ofrecer tarifas más competitivas ante la amenaza de pérdida potencial de ingreso, también se tradujeron en factores que precipitarían una renegociación del contrato ante choques adversos de demanda, en general altamente correlacionados con la dinámica económica del país.

Por otro lado, “la premura del proceso provocó que los estudios presentaran problemas metodológicos, se excluyeron parámetros económicos determinantes (población, empleo, auto per cápita, ingreso disponible) y tampoco se realizaron análisis de tráfico con variaciones temporales, por tipo de vehículo, por origen, destino y frecuencia. Las diferencias en las elasticidades de las tarifas entre usuarios terminales no fueron consideradas, ni el ajuste de las cuotas hacia la optimización de los ingresos y mucho menos la planeación de la minimización de los costos de mantenimiento y construcción en el largo y corto plazo, todo ello en función del vencimiento de los acuerdos financieros”. De manera similar, no se consideró el impacto del uso de las autopistas de los programas de privatización y modernización de infraestructura en otros sectores.

La predisposición de la SCT a mostrar la conveniencia de esta infraestructura propició pronósticos muy optimistas de aforo al tiempo que, otros factores como el desconocimiento de la demanda que enfrentarían las vías, la mayoría de las concesiones otorgadas eran autopistas nuevas y la limitada información para los estudios de tráfico provocaron que la planeación de estos esquemas estuvieran poco apegada a la realidad.

Promedio diario del tráfico como porcentaje de aforo garantizado	Numero de concesiones
Arriba de 100	5
75-100	2
50-74	8
25-49	8
6-24	9

**Tabla 5. Aforo garantizado**

#### *1.3.1.4. El Programa Nacional de Autopistas.*

El Programa Nacional de Autopistas contemplaba 5,245 Km. de autopistas en 52 concesiones, 37 concesiones (71%) eran para la construcción, explotación y mantenimiento de autopistas nuevas; 14 concesiones (27%) para la modernización y conservación de infraestructura ya existente y sólo una concesión para su explotación y conservación.

El Programa Nacional de Autopistas se anuncia en febrero de 1989 y para agosto ya se habían otorgado dos concesiones. Dos meses entre la expedición del pliego general del concurso y la emisión del dictamen son insuficientes, entonces entre la entrega de carpetas con la información para elaborar la propuesta y la entrega de la oferta había entre dos y tres semanas. Así, la precipitación en la elaboración de las posturas, que incluían las proyecciones del aforo vehicular y el costo del proyecto, así como el diseño de las carreteras fueron resultado de un intento del gobierno por alcanzar sus objetivos antes que finalizara el sexenio vigente.

El esquema tarifario y la garantía de aforo fueron las principales condiciones en el esquema de incentivos, debido al término fijo de la concesión, la existencia de vías alternas obligatorias a cada autopista concesionada y la falta de acuerdos de utilización con sectores clave. Así, la viabilidad del proyecto se condicionó tanto a la realización efectiva de los pronósticos de aforo, como a la efectividad del esquema tarifario oficial y preestablecido para recolectar los ingresos necesarios. La recuperación de la inversión se condicionó en su totalidad a la realización del aforo estimado.

La mayoría de las autopistas de cuota resultaron excesivamente caras para el sector de auto transporte de carga. En promedio, a diciembre de 1994 las tarifas de automóviles fueron 157% superiores a las de Capufe, 154% en el caso de autobuses y 125% más elevadas para camiones. Los volúmenes de tránsito reales fueron en promedio inferiores a los previstos en

los títulos de concesión, 16% en el caso de automóviles, 27% en autobuses y 70% en camiones.

El gobierno absorbió gran parte del riesgo que debía enfrentar el concesionario y si bien no ofreció explícitamente macro garantías, si otorgó otras que tenían fuertes implicaciones con el desempeño económico. Como resultado, el contrato fue inminentemente renegociable y en el largo plazo las garantías se tradujeron en fuertes desembolsos de recursos presupuestales, sobre todo en situaciones adversas, cuando el gobierno estaba en condiciones menos propicias para hacerle frente a estos compromisos.

### ***1.3.2. Hacia un nuevo esquema de concesión de autopistas y utilidad del contador de aforo vehicular.***

La SCT y la Unidad de Autopistas de Cuota, así como CAPUFE deben colaborar en la elaboración de las proyecciones de aforo proporcionando información relevante y aportando experiencia sobre la operación de carreteras. Asimismo, se deben realizar consultas con sectores vinculados: auto transporte de carga y de pasajeros, ferroviario, portuario y aéreo. Incluso, la autoridad debe proporcionar facilidades para que cada concesionario individualmente o con auxilio de la entidad reguladora entable acuerdos con empresas de estos sectores sobre el uso de las vías. Con mayor información sobre los parámetros económicos determinantes del tráfico en los ejes carreteros, se evitará la sobrestimación de los pronósticos de aforo, además de utilizar un sistema confiable de estimación del aforo en las autopistas estatales o federales que tengan un destino similar al de la concesión, siendo como parámetro importante el número de camiones de carga que circulen por las mismas.

Otra utilidad que se le podría dar al medidor de aforo vehicular en las autopistas federales sería el siguiente: Modernizar las autopistas, seguridad adicional, gasolineras, hoteles, talleres, restaurantes, módulos de ambulancias, etc., en los tramos con mayor afluencia vehicular.

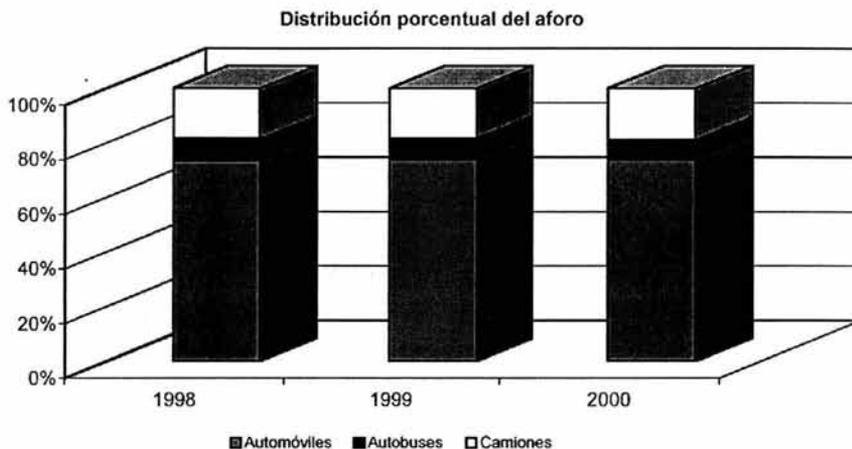
## **1.4. Estadísticas de aforo vehicular en las principales autopistas y puentes de cuota del país.**

Aforo en las autopistas y puentes de cuota por tipo de vehículo:

Tipo de vehículo	1998	1999	2000
Automóviles	430 184	468 597	507 392
Autobuses	47 530	48 229	49850

Camiones	109 514	117 205	130 385
Total	587 228	634 031	687 627

**Tabla 6. Aforo en autopistas de cuota por tipo de vehiculo**



Año	Concesionadas	CAPUFE	FARAC	Total
1990	44 765	283 985	-	328 750
1991	79 629	282 348	-	361 977
1992	94 966	282 026	-	376 992
1993	114 326	314 463	-	428 789
1994	234 510	298 478	-	532 988
1995	226 964	236 681	-	463 645
1996	181 060	247 697	66 040	494 797
1997	193 013	264 491	80 873	538 377
1998	203 824	282 478	100 926	587 228
1999	221 576	296 528	115 927	634 031
2000	236 474	307 404	143 749	687 627

**Tabla 7. Crecimiento de la red carretera**

## 1.5. Vehículos más comunes y carreteras federales en México.

### 1.5.1. Dimensiones de vehículos.

Los vehículos más comunes en México son los que se muestran en la tabla siguiente, estos datos son obtenidos del sector asegurador mexicano. Es un reflejo, que los autos más asegurados son los que mas hay.

En la siguiente tabla únicamente mostramos los 30 más comunes.

Vehículo	Armadora	Asegurados
TSURU	NISSAN	225,508
CHEVY	GENERAL MOTORS	164,621
V.W. SEDAN	VOLKSWAGEN	94,275
SENTRA	NISSAN	79,381
NEON	CHRYSLER	68,582
POINTER	VOLKSWAGEN	56,967
FOCUS	FORD	49,556
STRATUS Y BREEZE	CHRYSLER	47,773
NUEVO JETTA	VOLKSWAGEN	45,862
VOYAGER	CHRYSLER	41,277
WINDSTAR	FORD	37,959
ESCORT Y NUEVO ESCORT	FORD	34,216
NUEVO GOLF	VOLKSWAGEN	33,634
FAIRMONT, TOPAZ	FORD	33,606
JETTA GEN. 4	VOLKSWAGEN	30,343
CUTLASS	GENERAL MOTORS	27,028
FIESTA	FORD	26,576
CAVALIER	GENERAL MOTORS	26,356
SHADOW	CHRYSLER	25,295
SUBURBAN, CARRY ALL	GENERAL MOTORS	24,410
SPIRIT	CHRYSLER	23,237
CAVALIER (nueva generación)	GENERAL MOTORS	20,812
ATOS	CHRYSLER	20,802
STRATUS RT	CHRYSLER	20,181
FORD EXPLORER (IMPORT.)	FORD	19,904
GRAND MARQUIS, CROWN VIC.	FORD	19,866
IKON	FORD	19,259
HONDA CIVIC	HONDA	17,748
HONDA ACCORD	HONDA	17,448
DERBY Y NUEVO DERBY	VOLKSWAGEN	17,346

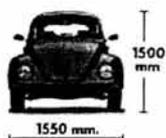
**Tabla 8. Autos mas comunes en México**

A continuación se muestran las dimensiones de anchura, altura, y distancia entre ejes de algunos de estos vehículos.

**VW Sedan.**

Dimensiones (mm)	
Largo:	4060
Ancho:	1,550
Alto:	1500
Distancia entre ejes:	2,400

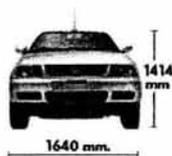
**Tabla 9. Dimensiones del VW sedan**

**Pointer.**

Dimensiones (mm)	
Largo:	3883
Ancho:	1630
Alto:	1415
Distancia entre ejes:	2468

**Tabla 10. Dimensiones del Pointer****Derby.**

Dimensiones (mm)	
Largo:	4137
Ancho:	1640
Alto:	1414
Distancia entre ejes:	2440

**Tabla 11 Dimensiones del Derby****Jetta.**

Dimensiones (mm)	
Largo:	4376
Ancho:	1735
Alto:	1446
Distancia entre ejes:	2513

**Tabla 12 Dimensiones del Jetta**

**Tsuru.**

Dimensiones (mm)	
Largo	4324
Ancho	1640
Alto	1369
Distancia entre ejes	2430

**Tabla 13. Dimensiones del Tsuru****Atos.**

Dimensiones (mm)	
Largo	3495
Ancho	1495
Alto	1580
Distancia entre ejes	2380

**Tabla 14. Dimensiones del Atos**

A continuación presentamos las dimensiones de camionetas, camiones y tractocamiones con remolque.

**Ram Van.**

Dimensiones (mm)	
Largo	4890
Ancho	2000
Alto	2030
Distancia entre ejes	2780

**Tabla 15. Dimensiones de la Ram Van**



**Ram.**

Dimensiones (mm)	
Largo	5276
Ancho	2030
Alto	1890
Distancia entre ejes	3061

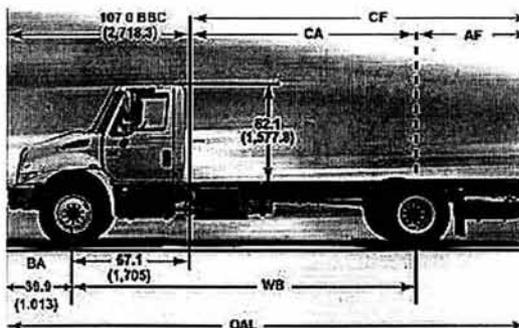
**Tabla 16 Dimensiones de la Ram.**



**International 4300 - 195 HP**

Dimensiones (mm)	
Distancia entre ejes (WB)	5,994.4
Cabina a eje trasero (CA)	4,290.1
Cabina a fin de chasis (CF)	6,779.3
Volado Trasero (AAF)	2,489.2
Largo total (OAL)	9,497.1

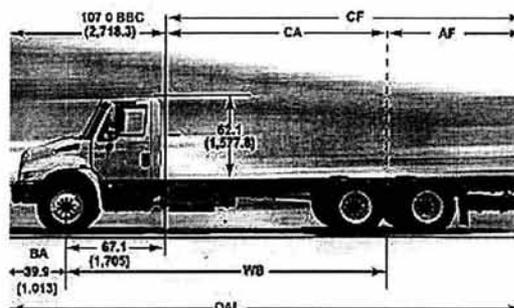
**Tabla 17 Dimensiones del Interntional 4300.**



**International 4400 - 215 HP**

Dimensiones (mm)	
Distancia entre ejes (WB)	5,562.6
Cabina a eje trasero (CA)	3,858.3
Cabina a fin de chasis (CF)	6,499.3
Volado Trasero (AAF)	2,641.6
Largo total (OAL)	9,217.7

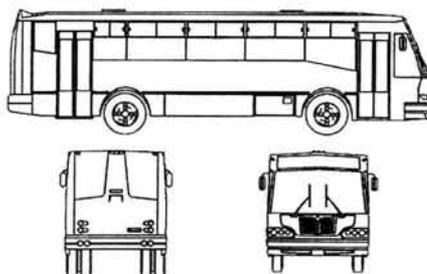
**Tabla 18. Especificaciones del Internacional 4400.**



**Aerobuss FC 198-115**

Largo total (LT)	10,485 Mts
Distancia entre ejes (DE)	5,029 Mts.
Volado Trasero (VT)	2,922 Mts.
Ancho	2,500 Mts.
Altura Total	3,212 Mts.
Altura Interior	2,000 Mts.

**Tabla 19. Especificaciones Aerobús FC 198-115.**

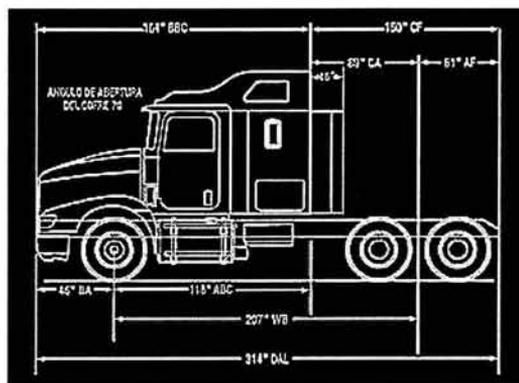


**International 9200i**

Dimensiones	mm
Distancia entre ejes	4,902.2
Largo Total	7,620.0
Ancho Total	2,428.2

Volado delantero	1,168.4
Volado trasero	1,549.4
Distancia cab.-intereje	2,108.2
Distancia cab.-ext. tras.	3,657.6
Distancia frente tras. cab.	3,962.4
Altura quinta rueda	1,226.8
Posición quinta rueda	101.6

Tabla 20 Especificaciones del Internacional 9200i.



### 1.5.2. Dimensiones de neumáticos.

A partir del catálogo de diversas llanteras se obtuvieron las dimensiones de ancho de neumáticos para cada tipo de vehículo y su respectiva medida de identificación. A continuación se presentan estas medidas en mínimo y máximo en cada caso y la tabla resumiendo la investigación.



	Ancho [mm]	Medida
Mínimo	152	145/65R15
Máximo	304	295/50R15

Tabla 21. Medidas de neumáticos de automóvil



	Ancho [mm]	Medida
Mínimo	180	7.00R14
Máximo	318	33X12.5R15

**Tabla 22. Medidas de neumáticos de camioneta**



	Ancho [mm]	Medida
Mínimo	226	225/70R19.5
Máximo	422	425/65R22.5

**Tabla 23. Medidas de neumáticos de camión**

	Mínimo	Máximo		
Vehículo	Medida	Ancho [mm]	Medida	Ancho [mm]
Automóvil	145/65R15	152	295/50R15	304
Camioneta	7.00R14	180	33X12.5R15	318
Camión	225/70R19.5	226	425/65R22.5	422
Remolque ligero	145/65R15	152	33X12.5R15	318
Remolque pesado	225/70R19.5	226	425/65R22.5	422

**Tabla 24. Medidas de neumáticos**

### 1.5.3. Tabla de clasificación de vehículos según CAPUFE.

Con base en las características de cada vehículo se integran las tarifas en las rutas de mayor utilización. Las estructuras se conforman de acuerdo al siguiente equipo:

Clave	Estructura Vehicular
A	Automóvil, Pick-Up, Panel y Motocicleta
B	Autobuses

C-2	Camión de 2 ejes
C-3	Camión de 3 ejes
C-4	Camión de 4 ejes
C-5	Camión de 5 ejes
C-6	Camión de 6 ejes
C-7	Camión de 7 ejes
C-8	Camión de 8 ejes
C-9	Camión de 9 ejes
R-1	Eje excedente ligero
R-2	Eje excedente pesado

Tabla 25. Clasificación de CAPUFE

#### **1.5.4. Carreteras federales en México.**

El artículo 6to. del reglamento sobre el peso, dimensiones y capacidad de los vehículos de auto transporte se refiere a la siguiente clasificación para el tránsito en los caminos y puentes de jurisdicción federal.

##### **1.5.4.1. Carretera Tipo ET.**

Son aquellas que forman parte de los ejes de transporte que establezca la Secretaría, cuyas características geométricas y estructurales permiten la operación de todos los vehículos autorizados con las máximas dimensiones de capacidad y peso, así como de otros que por interés general autorice la Secretaría, y que su tránsito se confine a este tipo de caminos.

##### **1.5.4.2. Carretera Tipo A.**

Son aquellas que por sus características geométricas y estructurales permiten la operación de todos los vehículos autorizados con las máximas dimensiones, capacidad y peso, excepto aquellos vehículos que por sus dimensiones y peso sólo se permitan en las carreteras tipo ET.

##### **1.5.4.3. Carreteras Tipo B.**

Son aquellas que conforman la red primaria y que atendiendo a sus características geométricas y estructurales prestan un servicio de comunicación interestatal, además de vincular el tránsito.

#### 1.5.4.4. Carreteras Tipo C.

Red secundaria.- Son carreteras que atendiendo a sus características prestan servicio dentro del ámbito estatal con longitudes medias, estableciendo conexiones con la red primaria.

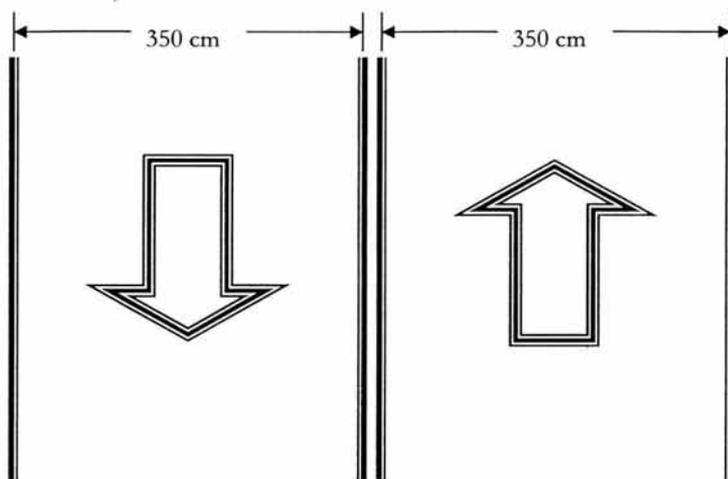
#### 1.5.4.5. Carreteras Tipo D.

Red alimentadora.- Son carreteras que atendiendo a sus características geométricas y estructurales principalmente prestan servicio dentro del ámbito municipal con longitudes relativamente cortas, estableciendo conexiones con la red secundaria. Atendiendo a sus características geométricas se tipifican en:

Tipo de Carretera	Nomenclatura
Carretera de 4 carriles, eje de transporte	ET 4
Carretera de 2 carriles, eje de transporte	ET 2
Carretera de 4 carriles	A4
Carretera de 2 carriles	A2
Carretera de 4 carriles, Red Privada	B4
Carretera de 2 carriles, Red Privada	B2
Carretera de 2 carriles, Red Secundaria	C
Carretera de 2 carriles, Red Alimentadora	D

Tabla 26. Clasificación de carreteras.

El siguiente esquema muestra las dimensiones en centímetros por carril de los caminos y puentes a lo que se refiere el artículo 6to. del reglamento sobre peso, dimensiones y capacidad de los vehículos de auto transporte que transitan en la jurisdicción federal.



**Ilustración 1. Dimensión de autopista**

## CAPITULO 2

### MARCO TEORICO

## 2. MARCO TEORICO.

Por definición, un controlador es un dispositivo que se emplea para controlar algún aspecto de un sistema: el controlador recibe información del estado del sistema y genera las señales de control necesarias para que el sistema cumpla su función. Hace algunos años los controladores se construían con componentes discretos, con lo que hasta los controladores más simples tenían la forma de pesadas y voluminosas cajas.

Con la aparición de los microprocesadores, los controladores pasaron a caber en una pequeña placa de circuito impreso. En la actualidad todos los componentes de un controlador están incluidos en un único chip, el microcontrolador. Un Microcontrolador típico incluye todos los elementos de un ordenador, pero además posee algunos dispositivos pensados para aplicaciones de control. El microcontrolador típico posee instrucciones de manipulación de bits, fácil acceso a la entrada / salida y una eficiente gestión de interrupciones. Todas estas características son apropiadas para el control de un sistema y al estar incluidas en el propio microcontrolador reducen en gran medida el costo y tamaño del sistema.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montarlo en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador embebido.

### 2.1. Familia de microcontroladores 68HCXX de Motorola.

#### 2.1.1. 68HC11.

En 1985 Motorola desarrollo el 68HC11, un microcontrolador que es altamente compatible con los microprocesadores 6800 aunque incluyendo más características. El 68CH11 utiliza tecnología HCMOS (semiconductores oxido-metal complementarios de alta densidad) que lo hace más pequeño y rápido, utiliza poca energía para alimentarse, y tiene alta tolerancia a señales de ruido lo cual en aplicaciones de control este hecho es de vital importancia.

Existen muchas versiones del 68HC11 y cada una de ellas tiene sus propias características y numero de identificación, algunos micros de esta familia pueden operar con

---

3.0 V para aplicaciones de bajo voltaje. También otras compañías como Toshiba y Mitsubishi han fabricado este micro con previa licencia de Motorola.

### ***2.1.2. 68HC05.***

Este es un micro muy popular por sus aplicaciones a muy bajo costo comparándolo con el 68HC11. Los micros de esta familia presentan variaciones significantes pero típicamente los puertos C y D son usados como de entrada/salida, algunos puertos tienen menos de 8 líneas. A su vez Motorola divide el 68HC05 en otras subfamilias y algunas son: Familia C, B, M, T, P/J, SC, L, K. Algunas de estas versiones están diseñadas para trabajar con 1.8 V.

### ***2.1.3. 68HC16.***

Motorola introdujo este microcontrolador en 1991 como un “puente” entre la familia de 8-bits 68HC11 y la de 32-bits 68300. En cuanto a hardware es compatible con la familia de los microprocesadores 68300 y 6800. En tanto que también es compatible en cuanto a software con la familia del 68HC11.

### ***2.1.4. El 68HC16.***

Utiliza la versión en 16 bits del CPU del 68HC11, tiene 3 nuevas multiplicaciones y 2 instrucciones adicionales para división, nuevos modos de direccionamiento y registros adicionales. Estas características hacen mas fácil para el MCU usar lenguaje de alto nivel y desarrollar funciones de de DSP.

### ***2.1.5. 68HC08.***

En 1993 Motorola introdujo la familia 68HC08 que resulta compatible con la familia 68HC05, esto significa que el software funciona igual para las dos familias. Mientras que el 68HC05 puede ser bastante costoso, algunos consumidores prefieren mantener las mismas posibilidades sin tener que cambiar de familia, como a la 68HC11.

### **2.1.6. 68HC12.**

#### **2.1.6.1. Familia 68HC12 A.**

La familia 68HC12 A ofrece una MCU con bus expandido que opera de 3.0 a 5.5V. Otras características de esta familia incluyen 4Kbytes de EEPROM, hasta 5 MBytes de capacidad de direccionamiento externo, PLL, y módulos SCI duales.

#### **2.1.6.2. Familia 68HC12 B.**

La Familia 68HC12 B es la primera MCU de 16 bits en el mundo que incluye tanto FLASH EEPROM como EEPROM borrrable a byte integradas en un chip.

Todos los dispositivos de esta familia incorporan un controlador de bus de comunicaciones seriales avanzadas multiplexadas operando con el módulo Byte Data Link Control (68HC912B32 y 68HC12BE32) o con el módulo Controller Area Network (68HC912BC32).

#### **2.1.6.3. Familia 68HC12 D.**

La Familia 68HC12 D es una de las MCUs de 16 bits con más integración Todos los dispositivos de esta familia incorporan módulos duales del SCI, un módulo Controller Area Network para comunicaciones seriales avanzadas, módulos duales del ADC de 10 bits, programa y arreglos de memoria de datos grandes usando tanto FLASH como EEPROM borrrable a byte, PLL, e interrupciones del teclado.

## **2.2. Microcontrolador 68HC912B32.**

EL microcontrolador MC68HC912B32 es un dispositivo de 16 bits compuesto por un on-chip estándar incluyendo una Unidad Central de Proceso (CPU) también de 16 bits, 32 kbytes en memoria EEPROM Flash, 1 Kbytes en memoria RAM, 768 bytes en EEPROM, una interfase de comunicación serial asíncrona, un timer de 8 canales y un acumulador de pulsos de 16 bits, un convertidor análogo-digital de 10 bits, un modulador de ancho de pulsos de 4 canales. Este chip es el primer microcontrolador de 16 bits en incluir la posibilidad de borrar datos en memoria en Flash y EEPROM, en el mismo dispositivo.

### 2.2.1. Características del 68HC912B32.

- CPU12 de 16 Bits
  - Set de Instrucciones compatible con la familia M68HC11.
  - Modelo de programación idéntico al M68HC11.
  - ALU de 20 bits.
  - Instrucción Queue.
  - Enhanced Indexed Addressing.
  - Instrucciones para Lógica Difusa.
- Bus Multiplexado.
  - Single Chip o Expandido.
- Operación de +3 a +5 Voltios DC.
- Interfase RS-232.
- 32Kbytes de EEPROM FLASH interna con programa residente DBug12 Monitor/Debugger para operación en los modos:
  - EVB.
  - JUMP-EEPROM.
  - POD-DBM.
  - BOOT-LOAD.
  - Conector para acceso a las líneas del Bus y de I/O del MCU.
  - Área para desarrollo de prototipos.
  - Posibilidad de Transferencia de archivos desde el PC permitiendo generación de código OFF-BOARD.
  - Dos modos para generar código usuario EVB.

Para pequeños programas o subrutinas, DBug12 se puede usar para colocar el código objeto directamente en memoria. Para programas largos, el ensamblador IASM12 se puede

utilizar en el PC para generar el archivo objeto S-Record. Dbug12 puede entonces emplearse el archivo S-Record en la memoria de EVB.

### 2.2.1.1. Unidad Central de Procesos.

El núcleo del 68HC912 es la CPU12, una evolución de 16 bits de alta velocidad de la arquitectura del 68HC11 que fue diseñada para mantener compatibilidad completa en el código fuente con el núcleo del 68HC11. El 68HC912 soporta completamente todos los registros internos, las instrucciones, los modos de direccionamiento, y los modos de operación del 68HC11.

7	A	0	7	B	0	ACUMULADORES DE 8 BITS A y B o ACUMULADOR DE 16 BITS D
15	D				0	
15	IX				0	REGISTRO INDICE X
15	IY				0	REGISTRO INDICE Y
15	SP				0	APUNTADOR DE LA PILA
15	PC				0	CONTADOR DE PROGRAMA
S X H I N Z Y C						REGISTRO DE CODIGOS DE CONDICION

**Ilustración 2. Registros 68HC12**

### 2.2.1.2. Características y Beneficios Adicionales incluyen:

- Hasta 8 MHz de velocidad del bus a 5 V.
- Hasta 5 MHz de velocidad del bus a 3 V.
- 64 instrucciones nuevas, unidad Aritmética Lógica de 20 bits (ALU), cola de instrucciones, y siete nuevos modos de direccionamiento indexado.
- Rutas de datos de 16 bits.
- 64 Kbytes de memoria lineal y capacidad de direccionamiento paginado, permitiendo el acceso a más de 4 Mbytes de espacio de programa y 1 Mbyte espacio de datos.
- Instrucciones Call y Return from Call (RTC) para direccionamiento paginado eficiente.
- Optimización para lenguaje de alto nivel (HLL).
- Capacidad de matemática rápida (16 x 16 MUL, 32 ÷ 16 DIV, EMACS).

- Instrucciones de lógica difusa (Fuzzy) para simplificar la programación, y reducir el tamaño del código, y ejecución más rápida del código.
- Modos de espera y de paro de bajo consumo de energía.

### ***2.2.1.3. Diseño Modular Flexible.***

Un bus estándar de interfaz, el Módulo Ligerero Bus (LMB) diseñado para bajo consumo, se usa para conectar la CPU12 a módulos periféricos especializados. El LMB es similar al bus Ínter módulos (IMB) presente en las familias 68HC16, 68300, y MPC500, permitiendo a Motorola apoyarse en módulos periféricos existentes, ya probados para desarrollar nuevas derivaciones del 68HC12.

### ***2.2.1.4. Bajo Consumo, Bajo Voltaje, Bajo Ruido.***

El 68HC12 corre hasta 8 MHz y fue diseñado para una operación de rango extendido de voltaje así como consumo de poca energía. Los Dispositivos operan desde 3.0 V hasta 5.5 V.

### ***2.2.1.5. Memoria FLASH EEPROM.***

Todas las derivaciones de la familia 68HC912 presentan en la pastilla memoria FLASH EEPROM no volátil que es borrable por secciones y soporta operaciones de byte o de palabra alineada.

Adicionalmente, la terminación rápida es asegurada con el acceso de un solo ciclo, y un bloque protegido opcional de 1 a 2 Kbytes está disponible. Este subsistema de memoria innovador, de 16 bits de ancho, ofrece muchos beneficios, incluyendo:

- Programación del prototipo a través del Background Debug Mode.
- Re programabilidad en campo.
- Programación y borrado rápidos.
- Pronta salida al mercado Producción de unidades que pueden ser detalladas al final del proceso.

### 2.2.1.6. Memoria EEPROM borrable por byte.

La Familia 68HC12 también presenta EEPROM en el chip borrable por byte para aumentar la flexibilidad de la programación sin requerir otra fuente separada de voltaje. Esta memoria integrada no volátil permite:

- Almacenar la información de calibración.
- Sistemas auto ajustables o auto adaptables.
- Bitácora de Datos para registro o seguridad de los datos.

### 2.2.1.7. Interfaz de Comunicación Serial (SCI).

El SCI es un sistema tipo UART de comunicación serial asíncrona. El SCI puede usarse para comunicaciones entre el microcontrolador y una Terminal, o una computadora, o en una red de microcontroladores. Una aplicación típica del SCI es una comunicación a media distancia (RS-232).

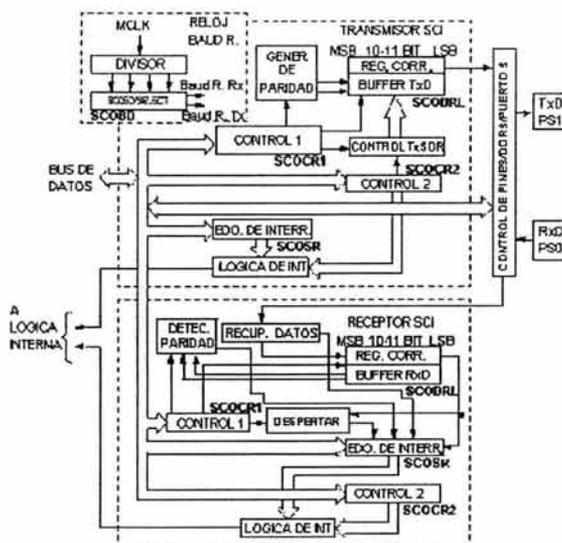
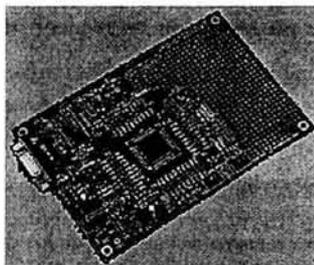


Ilustración 3. SCI

Motorola y varios proveedores independientes de herramientas de desarrollo ofrecen herramientas de desarrollo tanto de hardware como de software para la familia 68HC12. Motorola ofrece Kits de Desarrollo para el HC12 entre lo que se incluyen las tarjetas de

Evaluación para un dispositivo específico , la interfaz de depuración Serial (SDI), y el paquete de desarrollo MCUez.



**Ilustración 4. Tarjeta de evaluación**

Las tarjetas de evaluación para un dispositivo específico (EVB) son herramientas económicas para diseñar y depurar código para, y evaluar la operación de, los microcontroladores 68HC12 . Al proporcionar lo esencial de las señales de la MCU y la circuitería de E/S, las EVBs simplifican al usuario la evaluación de prototipos en hardware y software. Las EVBs tienen un área de prototipo, que permite al cliente interfazarse con la E/S del microcontrolador y con las líneas del bus. Algunas EVBs pueden también acomodar varios tipos y configuraciones de memoria externa para satisfacer los requisitos de una aplicación particular.

### 2.3. Lenguajes de programación.

Un lenguaje de programación es un código utilizado para la creación de programas. Lo utilizan los programadores (generalmente especializados en un lenguaje en particular) para la creación de diversas aplicaciones. Si no existieran los lenguajes, los programas serían escritos en lenguaje de máquina: ceros y unos. El objetivo de un programa es el de procesar información en forma de datos, para obtener un determinado resultado. Lo que diferencia un lenguaje computacional de otro es la forma de representar los datos y las herramientas que ofrece para procesarlos.

Es decir: Los tipos de datos que maneja, su declaración de Variables, las constantes, la forma de realizar asignaciones, la estructura general de un programa, los comentarios, sus operaciones de entrada / salida, las operaciones condicionales, las iteraciones. En la realización de este proyecto utilizamos lenguaje c para la programación de microcontrolador, el programa

que lo compila es el icc12 de imagecraft, también se uso Visual Basic para programar la aplicación que comunica el microcontrolador con la PC, además del despliegue de los resultados en pantalla. A continuación se describirá muy brevemente las funciones básicas de estos dos lenguajes de programación. Si se quiera mayor detalle de los mismos acudir a los manuales correspondientes.

### ***2.3.1.1. Lenguaje C.***

C es un lenguaje de programación de propósito general que ha sido estrechamente asociado con UNIX en donde fue desarrollado puesto que tanto el sistema como los programas que corren en él están escritos en lenguaje C, sin embargo, este lenguaje no está ligado a ningún sistema operativo y a ninguna máquina, y aunque se le llama “lenguaje de programación de sistemas” debido a su utilidad para escribir compiladores y sistemas operativos, se utiliza con igual eficacia para escribir importantes programas en diversas disciplinas.

Podemos decir que el lenguaje C es un lenguaje de nivel medio, ya que combina elementos de lenguaje de alto nivel con la funcionalidad del lenguaje ensamblador. Es un lenguaje estructurado, ya que permite crear procedimientos en bloques dentro de otros procedimientos. Hay que destacar que el C es un lenguaje portable, ya que permite utilizar el mismo código en diferentes equipos y sistemas informáticos: el lenguaje es independiente de la arquitectura de cualquier máquina en particular.

### ***2.3.1.2. Visual Basic.***

Visual Basic es un entorno de desarrollo diseñado para la creación de aplicaciones para los entornos de trabajo Microsoft Windows 95, 98, 2000, Me, XP y NT. Este lenguaje integra las facultades de un lenguaje de alto nivel con las herramientas de diseño gráfico, lo cual nos da acceso a todas las funciones de los sistemas anteriormente citados.

### ***2.3.2. Compilador Cruzado ICC12.***

Un compilador cruzado es un compilador que se ejecuta en una computadora, pero que genera código ejecutable para un CPU distinto a la de la propia computadora. Para el caso del

---

compilador cruzado en C que utilizamos en esta tesis, el compilador se ejecuta en una PC pero el código que se genera es para el microcontrolador 68HC12.

### ***2.3.3. Lenguaje de alto nivel contra lenguaje ensamblador.***

Normalmente los microcontroladores se programan en lenguaje ensamblador porque suelen ser programas cortos y además es necesario conocer con alto grado de detalle lo que ocupa el programa y la velocidad a que se va a ejecutar. Al emplear un lenguaje de alto nivel ganamos en comodidad, abstracción, portabilidad, legibilidad y se necesita menor tiempo de codificación, pero por el contrario se pierde un poco de noción de que es lo que está pasando dentro del microcontrolador. Además el volumen del código generado es mucho mayor que el que se genera en ensamblador.

Dentro de los lenguajes de alto nivel, el lenguaje C es más adecuado para la programación de microcontroladores puesto que es un lenguaje que permite estar muy en contacto con la máquina.

### ***2.3.4. El compilador de C de Imagecraft.***

#### ***2.3.4.1. Características del compilador.***

- **Compilador Exclusivo para el 68HC12:** Otros compiladores de C están diseñados para generar código para microcontroladores y microprocesadores muy diversos, lo que complica su utilización.
- **Fácil llamada a rutinas en ensamblador:** Es posible programar rutinas directamente en ensamblador y llamarlas desde los programas en C.
- **Inserción de instrucciones en ensamblador:** Es posible insertar rutinas directamente en el ensamblador en el propio código en C, utilizando la instrucción ASM.
- **Generación de ficheros en ensamblador:** Este compilador no genera archivos ejecutables directamente, sino que crea archivos en ensamblador que luego se convierten en ejecutables. Estos archivos en ensamblador pueden ser modificados por el usuario lo que permite optimizar el código.
- **Fichero ejecutable en formato .s19:**

- Fácil configuración de la situación del código, pila y datos.
- El lenguaje soportado en ANSI C.
- Funcionamiento del compilador.



**Ilustración 5. Compilador ICC12**

El funcionamiento del compilador es como se indica en la figura. Primero se toma un archivo en C, se compila y se obtiene un archivo .S en ensamblador. Este archivo se vuelve a compilar utilizando el ensamblador y se crea un ejecutable con extensión S19.

## CAPITULO 3

### SISTEMA AFORADOR VEHICULAR

### 3. SISTEMA AFORADOR VEHICULAR.

#### 3.1. Sistema aforador vehicular.

Este sistema esta diseñado para medir el paso de vehículos, clasificándolos por tipos, según el ancho de los neumáticos y número de ejes.

El sistema consta de cinco bloques como se muestra en el Diagrama 1, estos son: alimentación, microcontrolador MC68HC912B32 (programa), almacenamiento de datos (memoria), tarjeta de adquisición de datos (sensores), transmisión recepción e interpretación de datos (PC).

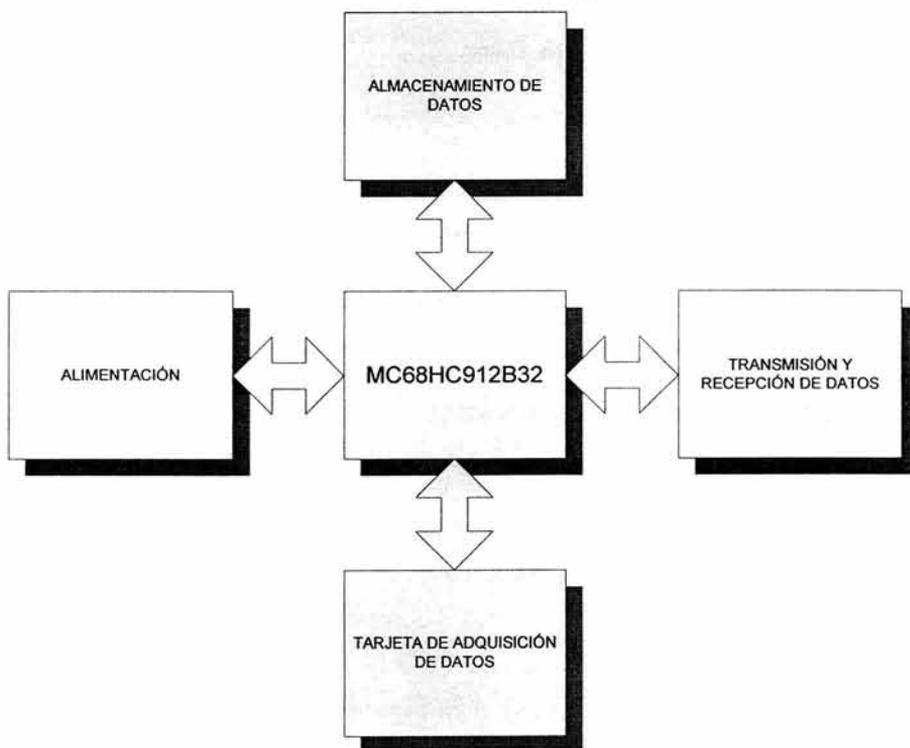


Diagrama 1. Diagrama de bloques del Aforador

### ***3.1.1. Alimentación.***

Este bloque se refiere al tipo de fuente que se utiliza, en el caso de tener alimentación de la red de energía eléctrica, se utiliza una fuente de 127 VAC a 5 VDC 1A, en caso contrario una batería.

### ***3.1.2. Almacenamiento de datos.***

Aquí es donde se estarán almacenando los resultados obtenidos en tiempo real al correr el programa. En este caso se almacenan en la memoria EEPROM y se trata de los datos: tipo de vehículo, hora y fecha en que sucedió el evento.

### ***3.1.3. Tarjeta de adquisición de Datos.***

Es donde se registra el paso de los vehículos así como la interrupción que serán enviados al microcontrolador.

### ***3.1.4. Transmisión y recepción de datos.***

Esta etapa es la que interactuara con el sistema de aforador, ya que los datos serán recibidos y procesados para ser presentados visualmente en una computadora.

### ***3.1.5. Microcontrolador MC68HC912B32.***

Se utiliza una tarjeta de evaluación M68EVB912B32, en donde se carga el programa en la memoria flash. Este programa obtiene los datos de la tarjeta de adquisición, los procesa, almacena y gestiona la comunicación con la PC.

## **3.2. Diagrama de flujos del sistema**

Aquí en este diagrama se ve el funcionamiento del sistema de aforo vehicular, mas adelante se explicara a detalle.

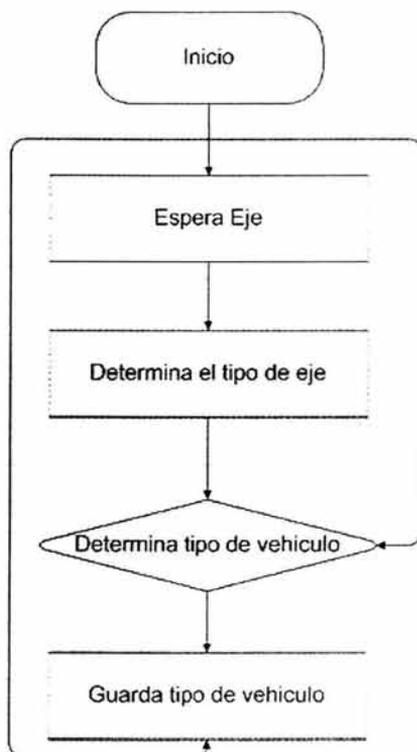


Ilustración 6. Diagrama de flujos del sistema de aforo vehicular

### 3.3. Sensores de piso y su caracterización.

#### 3.3.1. Botones (sensores de piso).

Fabricados de hule muy resistente capaces de soportar un promedio de un millón de opresiones bajo condiciones estándar de temperatura y uso. Estos botones son polarizados con positivo y negativo desde 5v hasta 127v. Se conectan junto con una resistencia y que al oprimirse cierran el circuito produciendo el flujo de corriente. Para su instalación es recomendable empotrarlos al pavimento sobre una base de hierro y lo más alineados posible; el número de botones que se pueden instalar va desde 1 hasta 64 que es el número de botones que contiene la hilera proporcionada por el fabricante; o más si fuera requerido.

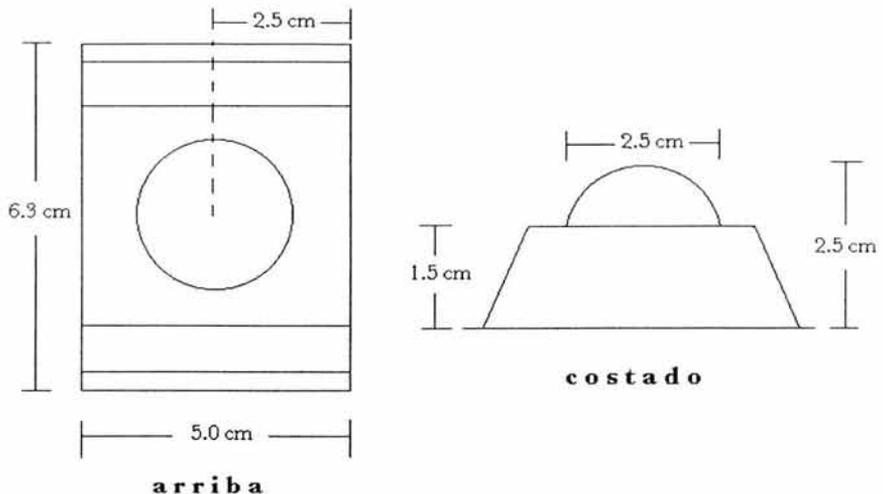


Ilustración 7. Dimensión de los sensores

### 3.3.2. Rebote.

El rebote es un fenómeno común en dispositivos de contacto debido a las fuerzas y velocidades a las que son sometidos, básicamente consiste en un mayor número de lecturas a las predichas ya que por las inercias de los elementos de dicho dispositivo se producen repeticiones no deseadas en las piezas en contacto y que resultan en un excedente a las lecturas esperadas. Dichas repeticiones de contactos transcurren en fracciones de segundo y quedan fuera del alcance de los sentidos humanos pero en dispositivos electrónicos de precisión resulta de crítica importancia.

En el microcontrolador M68EVB912B32 se programó una secuencia en ensamblador para determinar si en particular en este tipo de botones de piso se presentaba rebote al momento del paso del eje de un automóvil estándar; estas pruebas fueron hechas tanto para un botón nuevo como para uno que ya había cumplido su ciclo de uso.

```

;Cuenta pulsos de rebote del botón con entrada en puerto B pin 1
;Resultado en localidad $850

portb equ $01
ddrb equ $03

PROGRAMA equ $0800

org PROGRAMA

clr $0850

bset ddrb,$FF ;configura p/escritura portA
clr portb
bcir ddrb,$01 ;configura p/lectura pin 1

salto1 ldaa portb
beq salto1 ;regresa a salto1 si A=0

incb stab $850 ;incrementa localidad

salto2 ldaa portb
bne salto2 ;regresa a salto2 si A=1

bra salto1

```

Ilustración 8. Código para medir rebote.

Este programa es muy simple donde primero en la etiqueta “salto1” espera mientras en el pin 1 del puerto B se encuentre un cero lógico, una vez que aparezca un 1 lógico producto de una opresión en el botón, la localidad \$850 se incrementará en una unidad y el programa ahora esperara a un cero lógico en la etiqueta “salto 2”. En caso de que se presentara uno o más rebotes la secuencia se repetiría una o más veces respectivamente. La resolución es de 1.75 microsegundos, es decir, es capaz de registrar rebotes con tiempo mayor a este, en particular para en dispositivo esta resolución es por mucho suficiente para la identificación del rebote, ya que rebotes están del orden de milisegundos. Los resultados obtenidos de esta práctica fueron que estos botones si presenta rebote al momento de ser activados por un eje tanto en el botón nuevo como en el ya usado aunque la diferencia de lectura no varía mucho. Estos resultados de 10 lecturas para cada botón son los siguientes:

Botón	Lec 1	Lec 2	Lec 3	Lec 4	Lec 5	Lec 6	Lec 7	Lec 8	Lec 8	Lec 10
Nuevo	3	1	4	2	2	1	1	3	1	2
Usado	2	4	1	3	4	1	2	3	5	3.

Tabla 27. Lecturas

## 3.3.3. Caracterización del Rebote.

Después de haber observado la existencia del rebote en los botones se programo una nueva secuencia para determinar el lapso de tiempo en que este fenómeno transcurre y también donde es mas propenso que se presente, si al activar el botón o al soltarlo.

```

;Caracteriza rebote con entrada en puerto A pin 1
;Resultado a partir de localidad $850

; 8=$08 repeticiones =      = 1.0ms
; 16=$0A             = 7989= 2.0ms
; 20=$14             = 9985= 2.5ms   lseg = $850 a $9DF
; 40=$28             = 19965= 5.0ms   . $917
;100=$64             = 49905= 12.5ms  $89F
;240=$F0             =119765= 30.0ms  $872

porta equ $00
ddra  equ $02
inicio equ $800

      org inicio

      ldx #$850 ;indice x en loc $850

      bset ddra,$FF ;configura p/escritura porta
      clr  porta
      bclr ddra,$01 ;configura para lectura pin 1

      clr $00,x

salto1 ldaa porta
      beq salto1 ;regresa a salto1 si A=0

salto  LDAA #$14 ;EMPIEZA retardo y repite 20 dec.

retar2 ldab #$f8 ;repite 248
retar1 decb ;248*2+1
      bne retar1 ;497

      deca ;(497+2)*20+1
      bne retar2 ;TERMINA retardo (9,981)

      ldaa porta ;TIEMPO entre cada escritura de porta
      STAA $00,x ;2.5 ms que son 9,985 inst

      inx ;(9,984)
      bra salto ;(9,985)

```

Ilustración 9. Caracterización del rebote.

El objetivo del programa es almacenar en cada localidad, a partir de la \$850, el valor lógico que encuentre en el pin 1 del puerto A, este almacenamiento lo hace cada 2.5 milisegundos calculado mediante el ciclo de retardo y lo inicia a partir de que el botón sea presionado por primera vez.

Las muestras fueron sacadas para ambos botones y varias veces a cada uno dando como resultado el intervalo de tiempo del rebote de entre 2.5 a 12.5 milisegundos presentándose principalmente al activar el botón aunque en ciertos casos también fue observado al soltarlo.

A continuación se presenta dos muestras junto con sus graficas de lo mencionado anteriormente, en la primera muestra para el botón nuevo se observa un rebote con duración de 5 milisegundos al inicio de la activación y en la segunda para el botón usado se observan dos rebotes al inicio de su activación y un tercero al momento de soltar el botón.

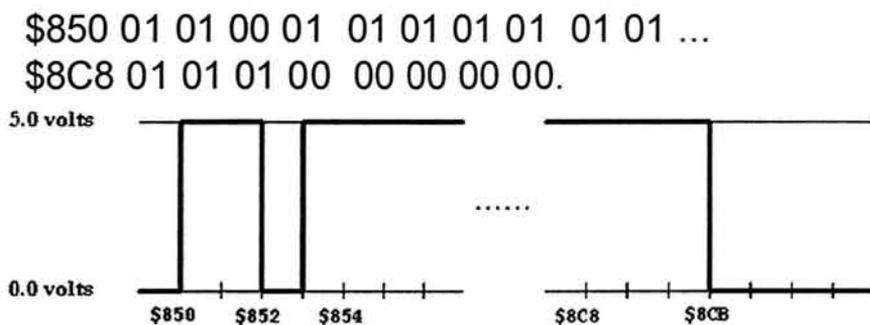


Ilustración 10. Muestra para sensor nuevo

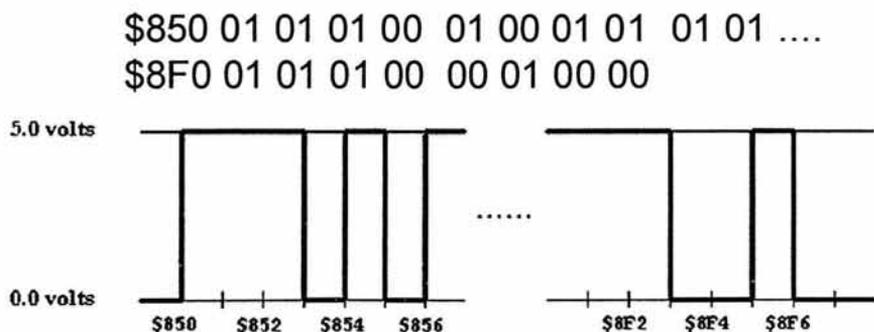


Ilustración 11. Muestra para sensor viejo

### 3.3.4. Ventana de activación de botones.

La ventana de activación es el tiempo que transcurre mientras el botón se encuentra oprimido por el neumático del eje que esta circulando en ese momento, este tiempo variará considerablemente dependiendo de la velocidad a la que rueda el vehículo en cuestión y a las dimensiones propias de sus neumáticos. Utilizando también la secuencia de instrucciones para caracterizar el rebote, se buscó esta ventana de activación para velocidades promedio en condiciones de rodamiento continuo previos al paso de un tope, esto con el objeto de determinar la ventana mínima y así tener conocimiento del tiempo con que se contará para el procesamiento de los datos que arroje el paso de este eje. Después de varias muestras el resultado para una ventana mínima de tiempo fue de 250 milisegundos, lo que indica que las ventanas de activación de la línea de botones instalados partirían de este tiempo hasta una ventana del orden de los segundos o inclusive minutos, como se menciono anteriormente esto dependería de la velocidad del vehículo y las dimensiones de sus neumáticos.

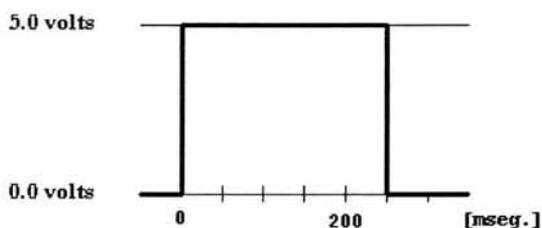


Ilustración 12. Ventana de activación de sensores

Además de la ventana mínima se determinó la ventana promedio que se presentaría en la mayoría de los ejes registrados, esta ventana es de alrededor de 1 segundo.

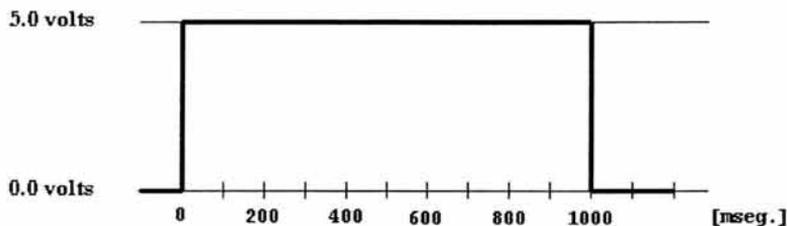


Ilustración 13. Ventana Promedio.

### 3.4. Diseño de tarjeta de adquisición de datos.

El fin de construir una tarjeta de adquisición de datos para el medidor de aforo vehicular, es el de hacer un sistema que lea la señal que emiten los sensores y que proteja a los puertos de entrada, así como proporcionar una señal de interrupción.

En esta tarjeta se colocaron botones mini switch normalmente abiertos, que sirven para simular los sensores que van al piso en un sistema a escala 1:1. Los sensores que se colocan en el piso tienen que ser 32, para abarcar medio carril, en esta tarjeta de adquisición se han puesto 16 botones para la prueba del sistema de contador de aforo vehicular considerando que ningún neumático de automóvil mide menos de 10cm.

Para la obtención de mejores resultados se probaron dos diseños de tarjetas, a continuación se presenta el diseño que fue elegido. Se muestran pruebas, simulaciones y resultados.

Esta tarjeta se diseñó a partir de módulos, estos módulos fueron probados individualmente, y después en conjunto. Los módulos son los siguientes. Lectura de puerto de entrada y señal de interrupción IRQ. A continuación se presenta el diagrama de bloques de la tarjeta de adquisición de datos.

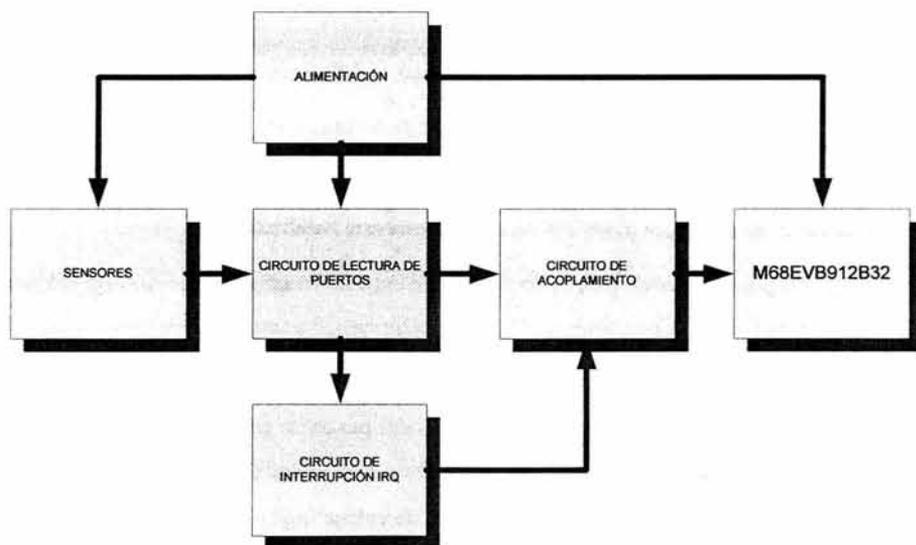


Ilustración 14. Diagrama de Bloques de la tarjeta de adquisición de datos

Los puertos utilizados para la lectura son el Puerto P y el Puerto T, se eligieron ya que son de entrada y salida, y además se encontraba colocados cerca uno del otro en la tarjeta entrenadora.

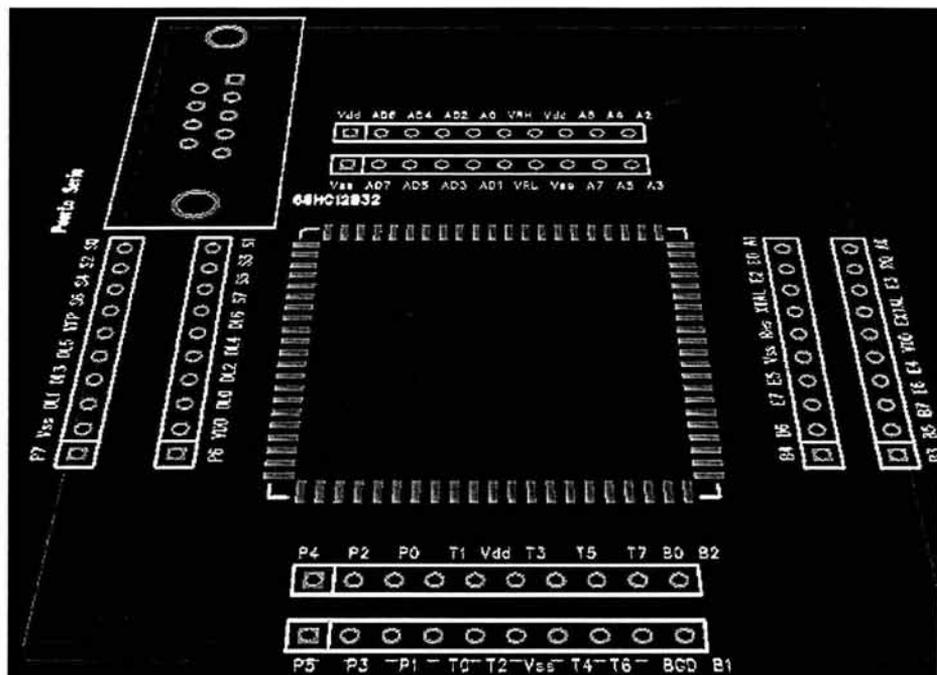


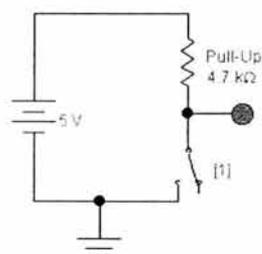
Ilustración 15. Puertos en la tarjeta de evaluación

El puerto P es de propósito general de entrada y salida, el PP0-PP3 son usados con el modulador de ancho de pulso PWM, cuando este está habilitado.

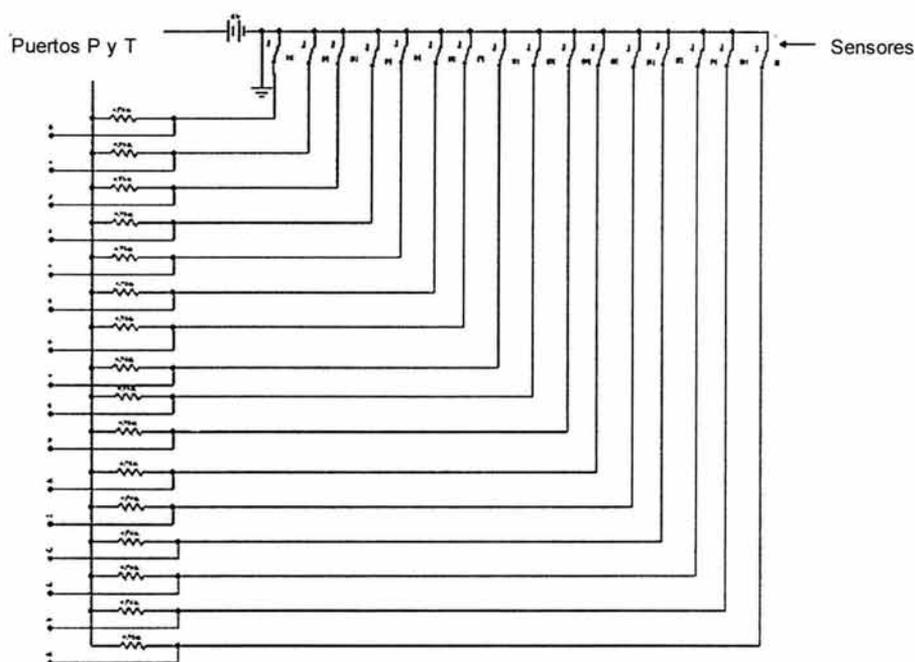
El puerto T es de propósito general de entrada y salida cuando no está habilitado para capturar datos y la salida compara el temporizador con el acumulador de pulsos del subsistema.

Para el diseño del circuito de entrada se tomó en cuenta que los pines de estos puertos son de colector abierto, esto quiere decir que cada pin es un transistor npn en colector abierto, como queremos que la entrada esté en voltaje alto cuando el interruptor esté abierto, es necesario colocar una resistencia de elevación de voltaje (pull-up) en este caso de  $4.7k\Omega$  y una fuente de voltaje externa.

En la siguiente figura se presenta el circuito antes mencionado, ejemplificando la señal de salida cuando esta el interruptor cerrado y después lo abrimos.



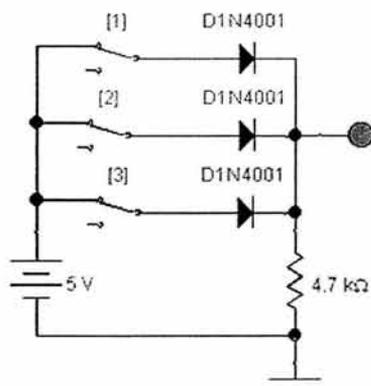
El circuito de lectura completo es el siguiente.



**Ilustración 16. Circuito de lectura**

El circuito de interrupción se diseña para activar la señal que se enviara al IRQ de microcontrolador cuando se active alguno de los 16 sensores de la tarjeta de adquisición. Esta señal tiene que ser un flanco de bajada, esto quiere decir que a oprimirse un interruptor, el que sea, se envíe una señal de 5v a 0v.

Para realizar lo anterior se utiliza una compuerta AND construida con diodos, que es mejor que usar un circuito integrado convencional, ya que se puede construir una AND de 16 entradas sin ningún problema, y utilizando elementos pasivos que consumen menos corriente que los CI.



A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Ilustración 17. Compuerta and

### 3.5. Acoplamiento.

El acoplamiento se incluye en el diseño de la tarjeta, se necesita este acoplamiento ya que los puertos P y T de la tarjeta no están en orden numérico, y requiere algunos cruces para unir el microcontrolador con la tarjeta de adquisición.

### 3.5.1. Circuito Completo de la tarjeta de adquisición de datos.

Este circuito es el resultado de la combinación de los módulos de lectura, interrupción y acoplamiento. Su funcionamiento es el siguiente; la señal que envían los mini switch a los puertos P y T del microcontrolador esta en alto, esto quiere decir que simplemente al conectar la tarjeta de adquisición se envían 5V a cada pin de entrada de los puertos P y T, al oprimir un mini switch, se envía una señal con flanco de bajada. Se opto por este modelo ya que se conecta la etapa de interrupción a cada nodo de salida de la etapa de lectura, y tomando en cuenta que la IRQ es una interrupción de modo asíncrono y que se puede seleccionar si la interrupción se provoca por un flanco de bajada o por un nivel lógico 0, entonces así nos ahorramos un modulo el cual tendría que invertir la señal.

A continuación se muestra el circuito de la tarjeta de adquisición de datos.

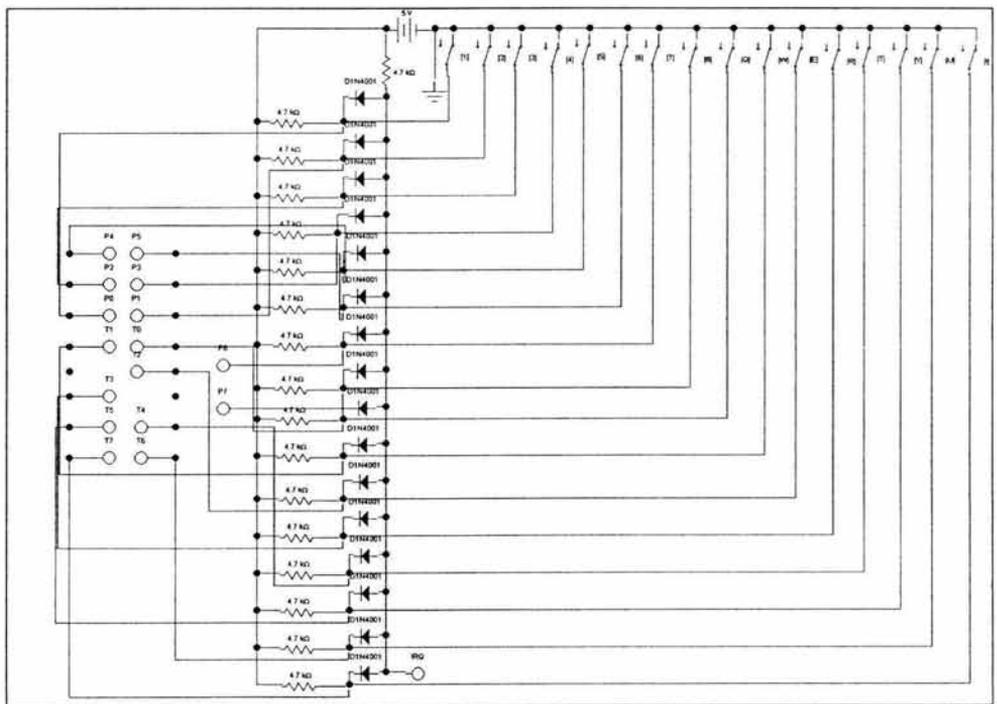


Ilustración 18. Diagrama completo de la tarjeta de adquisición de datos

### 3.6. Simulaciones y resultados.

Para comprobar el funcionamiento de la tarjeta de adquisición de datos, propusimos un pequeño ejemplo del funcionamiento, simulando en electronics workbench (ver Apéndice C.) Se supone que pasa un automóvil, se tiene que comprobar que se envían señales de flanco de bajada tanto en los sensores como en el IRQ.

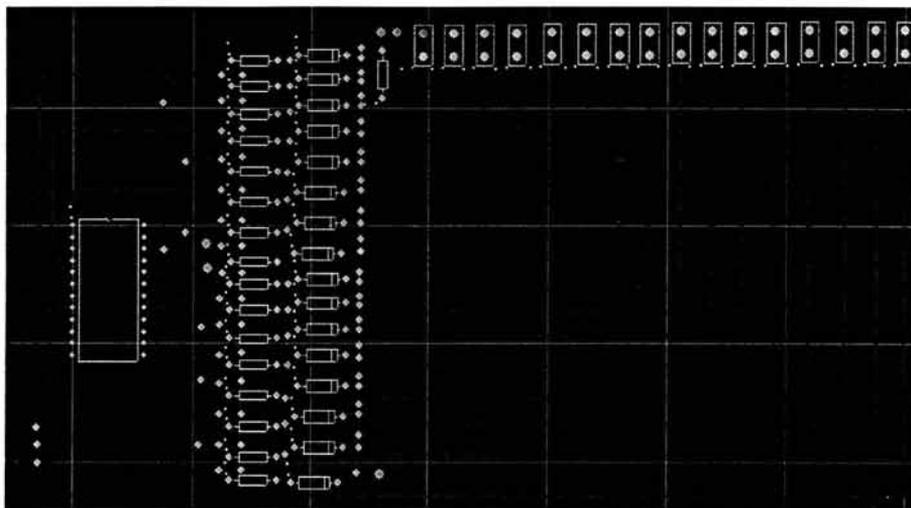
Los botones que debe oprimir el vehículo por el ancho de sus neumáticos son los siguientes, automóvil (rodada sencilla en ambos ejes); Al pasar el automóvil aprieta un botón. El IRQ se mantiene en alto, hasta que se oprime algún botón, que es cuando presenta un flanco de bajada.

En conclusión las pruebas resultaron favorables, por tanto se procede a la elaboración del circuito impreso.

### 3.7. Circuito Impreso.

Después de haber probado el circuito, se procedió a realizar el impreso, para hacerlo se utilizo el Protel99 se, únicamente utilizamos el PCB, ya que la posición de los elementos era específica, así como el de la pistas.

El circuito impreso quedo de la siguiente forma.



**Ilustración 19. Circuito impreso**

Los componentes se colocaron de tal forma que no estorben al paso de los vehículos, porque cabe recordar que este modelo simula al real, que esta colocado en la mitad de un carril de una carretera federal o estatal.

Se utilizo un diseño con un impreso de dos caras, así se evita poner puentes de alambre.

Se utilizo la técnica de planchado y cloruro ferrico, para hacer el impreso, las plantillas que utilizamos para el planchado son las siguientes.

Parte de abajo del circuito impreso:

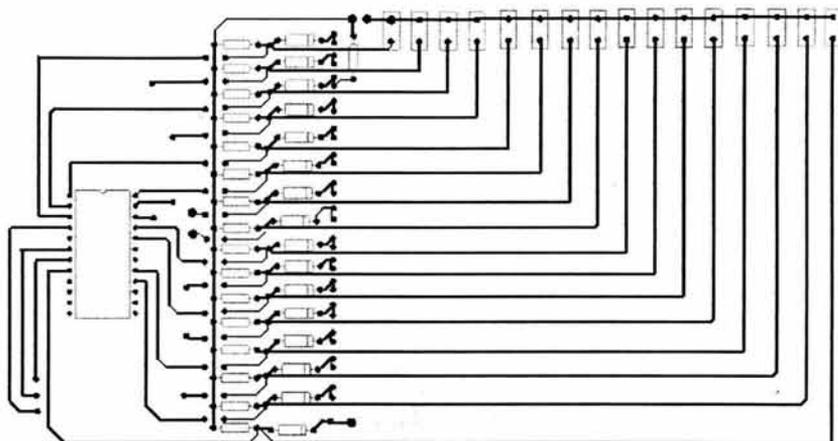


Ilustración 20. Circuito impreso cara superior

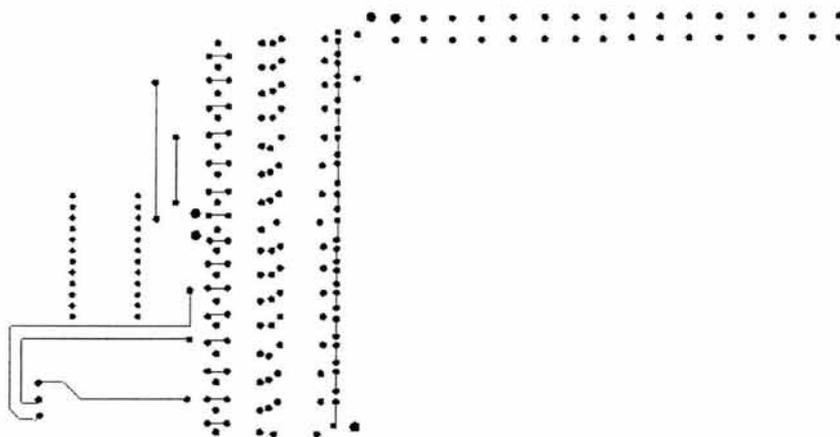


Ilustración 21. Circuito impreso cara inferior

### 3.8. Adquisición y procesamiento de datos en el microcontrolador MC68HC912B32.

#### 3.8.1. Reloj Interno.

El Objetivo de contar con un reloj interno es proporcionar una manera sencilla pero eficaz de mantener un registro del tiempo en el que suceden los eventos. Como describimos

anteriormente el sistema de aforo vehicular a diseñar, no solo cuantificará y calificará a los vehículos que pasen por encima de él, también debe llevar un registro de cuando suceden los eventos; de tal forma que se cuente con información suficiente para un análisis estadístico sobre el tráfico.

Por ejemplo, al contar con información estadística sobre cuando existe un menor tráfico en la vía de estudio, se podrá determinar en que momento es más oportuno el mantenimiento de la misma.

El reloj interno se fundamenta en contador llamado base que se incrementa periódicamente a una frecuencia conocida. Así, al conocer el valor de este contador y su frecuencia de actualización podremos, comparando su valor con una constante, incrementar la variable del reloj cada segundo. Para lograr la mayor exactitud posible en la frecuencia de actualización de la variable base se utiliza el sistema de interrupción en tiempo real del microcontrolador.

La variable del reloj, llamada tiempo, esta formada por cuatro bytes, de tal forma que tiene su rango es de 0 a 2.14 Gs. (69.04 años aproximadamente). Lo que nos da un holgado período de observación, limitado solamente por el suministro de energía al sistema.

### ***3.8.2. Sistema de Interrupción en Tiempo Real.***

#### ***3.8.2.1. Funciones de Reloj.***

En el microcontrolador, el circuito de generación de reloj proporciona señales de reloj E internas y externas, así como las señales de reloj interno utilizado por el CPU y los periféricos. Un circuito de monitoreo de reloj proporciona un vigilante de operación correcta de la computadora (watchdog COP), un circuito de interrupción periódica también está incorporado en el MC68HC912B321.

---

<sup>1</sup> MC68HC912B32 Technical Summary 16-Bit Microcontroler. Pág. 57. Motorola INC., 1997.  
Ref. MC68HC912B32TS/D

### 3.8.2.2. Interrupción en tiempo real

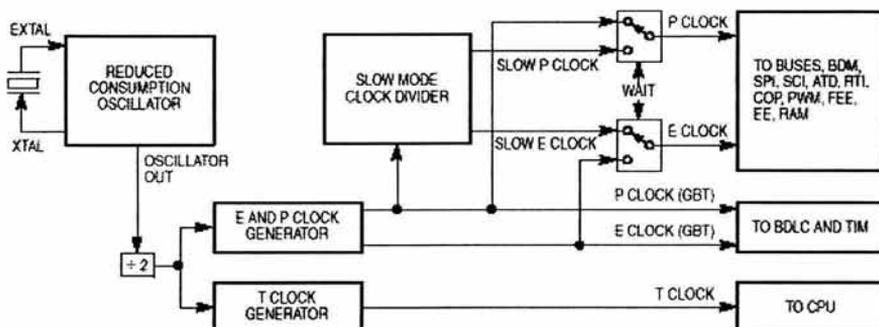


Ilustración 22. Interrupción en tiempo real

Existe un interruptor en tiempo real (periódico) disponible para el usuario. La interrupción ocurre a una de las siete frecuencias a seleccionar. Una bandera de interrupción y un bit de habilitación son asociados a esta función. Existen tres bits para seleccionar la frecuencia<sup>2</sup>.

### 3.8.2.3. Registro de control de la interrupción en tiempo real.

Existe un registro en el microcontrolador MC68HC912B32 que proporciona el control del sistema de interrupción en tiempo real, el RTICTL.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	RTIE	RSWAI	RSBCK	0	RTBYB	RTR2	RTR1	RTR0
Inicio:	0	0	0	0	0	0	0	0.

Tabla 28. RTICTL

En donde:

RTIE: Habilita la Interrupción en tiempo real.

0 = La solicitud de interrupción por el RTI es restringida.

1 = Se solicitará la interrupción cuando la bandera RTIFLG sea colocada.

RSWAI: RTI y COP son detenidos cuando se entra en ESPERA.

<sup>2</sup> Idem. Pág. 57

0 = Permite que los sistemas RTI y COP corran normalmente.

1 = Detiene ambos sistemas cuando se entra al estado EPERA.

RSBCK: Detiene RTI y COP en el modo de depuración.

0 = Permite los sistemas RTI y COP en el modo de depuración.

1 = Inhibe ambos sistemas cuando se encuentra en el modo de depuración, esto es útil para emulación.

RTBYP: Paso de la cadena de división del interruptor.

0 = La cadena de división funciona normalmente.

1 = La cadena de división es ignorada, permite pruebas más rápidas.

RTR2, RTR1, RTR0: Selección del período de interrupción.

RTR2	RTR1	RTR0	Divide E por:	Período de Interrupción @ 4MHz	Período de Interrupción @ 8MHz
0	0	0	-	-	-
0	0	1	213	2.48 ms.	1.024 ms.
0	1	0	214	4.096 ms.	2.48 ms.
0	1	1	215	8.196 ms.	4.096 ms.
1	0	0	216	16.384 ms.	8.196 ms.
1	0	1	217	32.768 ms.	16.384 ms.
1	1	0	218	65.536 ms.	32.768 ms.
1	1	1	219	131.72 ms.	65.536 ms.

Tabla 29. RTR

### 3.8.2.4. Funciones del módulo RTI.

A continuación se presentan las rutinas necesarias para lograr la funcionalidad del reloj del sistema de aforo vehicular. El módulo RT esta formado por el archivo Rti.c que contiene el código de las funciones y el archivo de encabezado Rti.h que contiene las declaraciones de las variables y funciones externas del módulo.

El código de las funciones está escrito en ANSI C3 y fue compilado satisfactoriamente en el compilador ImageCraft ICC12©4.

<sup>3</sup> C Programming in ICC12. Joshep Valvano

### 3.8.2.5. Archivo de encabezado Rti.h.

En el archivo Rti.h se encuentran las declaraciones de funciones y variables externas del módulo. La función `Inicia_Rti()` ejecuta los procedimientos de inicio del sistema RTI y es llamada en la función `main()` del módulo principal. La variable externa `rti_tiempo` es la variable que mantiene la cuenta de los segundos desde que se inicio el sistema.

```
/******  
/* Archivo Rti.h  
/* Encabezado del módulo de atención a la interrupción  
/* RTI  
/******  
/* Inicializa RTI con pre-escalador a 8.196 ms entre interrupción @ 8MHz */  
void Iniciar_Rti(void);  
extern long rti_tiempo;
```

### Ilustración 23. Rti.h

### 3.8.2.6. Función `Iniciar_Rti()`.

La función `Iniciar_Rti()` se encarga de configurar el sistema de interrupción periódica y las variables necesarias. La variable `tiempo` es de cuatro bytes y la variable `base` es de dos bytes.

---

<sup>4</sup> Capítulo 2.4 de esta Tesis y [www.imagecraft.com](http://www.imagecraft.com)

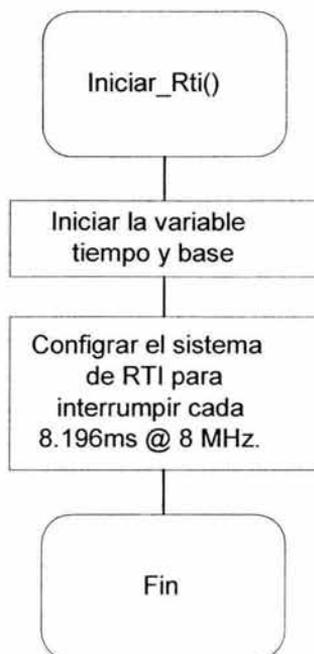


Ilustración 24. Rti

```

int rti_base;
long rti_tiempo; /* El tiempo */
void Iniciar_Rti(void)
{
    rti_tiempo = rti_base =0; /* Tiempo a cero */
    RTICTL = 132; /* Configurar Módulo RTI para 8.196 ms @ 8 MHz */
}
  
```

Ilustración 25. Iniciar Rti

### 3.8.2.7. Función Rai\_Rti().

Esta función se encarga de atender a la interrupción periódica. Esta interrupción ocurre cada 8.196 ms. Con lo que suceden 122.010 interrupciones cada segundo, al incrementar la variable base cada interrupción y después compararla con 122, sabremos cada cuando debemos incrementar la variable tiempo. El error entre el valor real y el valor utilizado es de 721.31ns. Con lo que tendremos un atraso de un segundo cada 16.046 días.

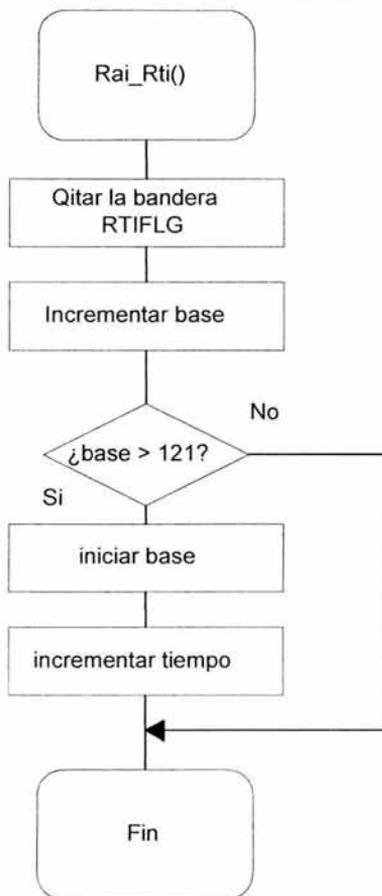


Ilustración 26. Función Rti

```

void rai_rti(void)
{
    RTIFLG = 128; /* Limpiar bandera RTIFLG */
    rti_base++; /* Incrementa la base*/
    if(rti_base > 121) /* ¿A pasado un segundo? */
    {
        rti_base = 0; /* Base a cero */
        rti_tiempo++; /* Incrementa el tiempo */
    }
}
  
```

Ilustración 27. Código Rti

También se incluyen en el módulo RTI instrucciones al preprocesador para indicarle que la función `Rai_Rti()` es una rutina de atención a interrupción y que debe colocarle la instrucción RTI al final de la misma. A su vez, se le indica que el vector de interrupción en tiempo real esta en la dirección `0xFFFF0`.

### ***3.8.3. Memoria EEPROM.***

El objetivo del módulo EEPROM es el de contener las funciones necesarias para el manejo de la memoria EEPROM. Los datos adquiridos por el sistema de aforo vehicular necesitan ser almacenados de alguna manera antes de que sean recolectados por el sistema de publicación. Decidimos utilizar como medio de almacenamiento la memoria de lectura/escritura y eléctricamente borrrable que esta incluida en el microcontrolador MC68HC912B32. Esta memoria tiene un tamaño de 768 bytes; esto proporciona espacio para almacenar aproximadamente 153 vehículos.

#### ***3.8.3.1. Memoria EEPROM en el MC68HC912B32.***

La memoria EEPROM del microcontrolador MC68HC912B32 es una memoria no volátil de 768 bytes que puede ser utilizada para acceso frecuente a datos estáticos ó para almacenar código de acceso rápido.

La memoria EEPROM del MC68HC912B32 es configurada en un arreglo de 16 bits. La memoria EEPROM puede ser leída por bytes, por palabra alineada o por palabra desalineada. El tiempo de acceso es de un ciclo para bytes y palabras alineadas y dos ciclos para palabras desalineadas.

La programación se ejecuta por byte o palabra alineada. Los intentos por programar o borrar palabras desalineadas fallarán. Solamente el menor byte será encadenado y programado o borrado. La programación y el borrado de la EEPROM pueden ser ejecutados en cualquier modo de operación.

Cada una de los bytes o palabras alineadas de la EEPROM deben ser borrados antes de ser programados. El módulo EEPROM soporta borrado de bytes, palabras alineadas, renglones de 32 bytes o borrado completo, todos utilizando una bomba de carga interna. También es posible borrar en modo prueba todos los renglones pares e impares; el estado borrado es `0xFF`. El módulo EEPROM contiene candados internos que protegen datos

almacenados contra corrupción de datos por colocar el voltaje de borrado/programado accidentalmente. Le voltaje de programación se deriva del voltaje VDD interno que contiene una bomba de carga interna. La EEPROM tiene una vida mínima de 10,000 ciclos de programación/borrado sobre el rango completo de temperatura de operación<sup>5</sup>.

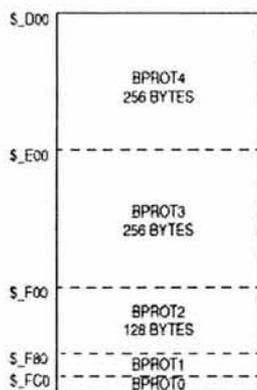


Ilustración 28. Mapa de memoria

### 3.8.3.2. Algoritmo de programación.

La programación o borrado de la memoria EEPROM debe seguir la siguiente secuencia:

1. Escribir el valor deseado de BYTE, ROW y ERASE; EELAT = 1.
2. Escribir el byte o la palabra alineada en la dirección EEPROM.
3. Escribir EEPGM = 1.
4. Esperar el tiempo de programación ( $t_{prog}$ ) o el tiempo de borrado ( $t_{erase}$ ).
5. Escribir EEPGM = 0.
6. Escribir EELAT = 0.

<sup>5</sup> MC68HC912B32 Technical Summary 16-Bit Microcontroler. Pág. 83. Motorola INC., 1997.  
Ref. MC68HC912B32TS/D

También es posible programar o borrar mas de un byte o palabra alineada saltando del paso 5 al paso 2.

### 3.8.3.3. El registro *EEPROM*.

El registro de control *EEPROM* se utiliza para configurar la memoria *EEPROM* y colocar el voltaje de programación en la memoria, consiste en:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	BULKP	0	0	BYTE	ROW	ERASE	EELAT	EEPGM
INICIO	1	0	0	0	0	0	0	0

**Tabla 30. EEPROM**

BULKP: Protección contra borrado completo.

0 = La *EEPROM* puede ser borrada completamente.

1 = La *EEPROM* es protegida contra el borrado completo y el borrado de renglones de 32 bytes.

BYTE: Borrado de byte o palabra alineada.

0 = El borrado completo y por renglones de 32 bytes es permitido.

1 = Solamente se permite el borrado de un byte o una palabra alineada.

ROW: Borrado completo o por renglones de 32 bytes.

0 = Borra la memoria *EEPROM* completa.

1 = Borra solamente un renglón de 32 bytes.

BYTE	ROW	Tamaño de bloque
0	0	Borra la memoria <i>EEPROM</i> entera
0	1	Borra un renglón de 32 bytes
1	0	Borra un byte o una palabra alineada.
1	1	Borra un byte o palabra alineada.

**Tabla 31. Byte, Row**

ERASE: Control de borrado.

0 = La *EEPROM* es configurada para programación o lectura.

1 = La EEPROM es configurada para borrado.

EELAT: Control de encadenamiento de la EEPROM

0 = La EEPROM es configurada para una lectura normal.

1 = El bus de dirección de la EEPROM y el bus de datos son encadenados para programación o borrado.

EEPGM: Inicia la programación y el borrado de la EEPROM.

0 = Corta el voltaje de programación a la EEPROM.

1 = Aplica el voltaje de programación a la EEPROM.

### ***3.8.4. Funciones del módulo EEPROM.***

A continuación se presentan las rutinas necesarias para lograr el almacenamiento de datos en la memoria EEPROM. El módulo EEPROM esta formado por el archivo Eeprom.c que contiene el código de las funciones y el archivo de encabezado Eeprom.h que contiene las declaraciones de las variables y funciones externas del módulo.

El código de las funciones está escrito en ANSI C6 y fue compilado satisfactoriamente en el compilador ImageCraft ICC12©7.

#### ***3.8.4.1. Archivo de Encabezado Eeprom.h.***

En el archivo Eeprom.h se encuentran las declaraciones de funciones y variables externas del módulo. La función Iniciar\_EE() coloca el un puntero a la próxima localidad disponible en la memoria EEPROM. La función Borrar() coloca en estado de borrado (0xFF) todas las localidades de la memoria EEPROM. La función guardar escribe en las localidades correspondientes los datos que se requiere almacenar, esta función inhibe las interrupciones cuando se aplica el voltaje de programación a la memoria como medida de protección de la memoria EEPROM. La estructura de datos str\_ee\_datos es una plantilla del formato de los datos a guardar.

---

<sup>6</sup> C Programming in ICC12. Joshep Valvano

<sup>7</sup> Capítulo 2.4 de esta Tesis y [www.imagecraft.com](http://www.imagecraft.com)

```
/* **** */
/* Archivo: Eeprom.h */
/* Encabezado de Eeprom.c */
/* **** */
/* Rango de EEPROM */
#ifndef EEINICIO
#define EEINICIO 0x0d00
#endif
#ifndef EEFIN
#define EEFIN 0x0ffd
#endif
void Iniciar_EE(void);
void Borrar(void);
void Guardar(char dato, long tiempo);
extern struct str_ee_datos
{
    char tipo;
    long tiempo;
}*ee_pdatos;
```

#### Ilustración 29. eeprom.h

##### 3.8.4.2. Función *Iniciar\_EE()*.

La función *Iniciar\_EE()* coloca el puntero *ee\_pdatos* en la próxima dirección disponible en la EEPROM.

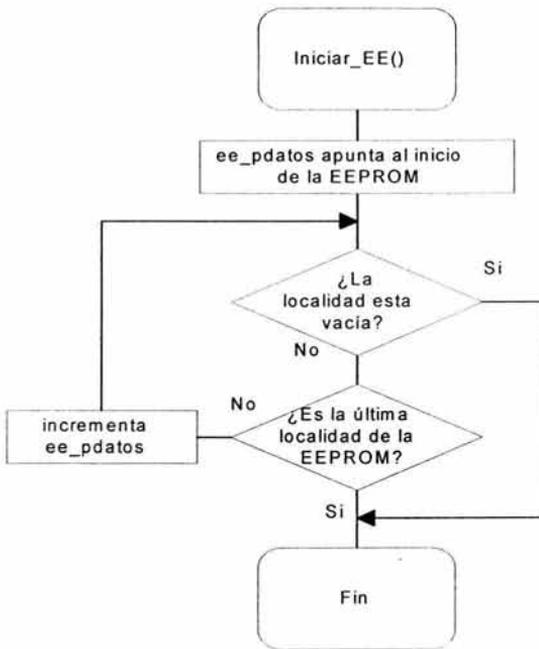


Ilustración 30. Diagrama eeprom

```

struct str_ee_datos
{
  char tipo;
  long tiempo;
} *ee_pdatos;
void Iniciar_EE(void)
{
  ee_pdatos = (struct str_ee_datos *) (EEINICIO); /* Apunta al inicio de EEPROM *
  /* Si la localidad apuntada esta vacia Y no se ha llegado al final de EEPROM *
  while (~ee_pdatos->tipo && (ee_pdatos <= (struct str_ee_datos *) (EEFIN)))
    ee_pdatos++; /* Apuntar a la siguiente localidad */
}
  
```

Ilustración 31. Código eeprom.

### 3.8.4.3. Función *EE\_tprog()*.

La rutina *EE\_tprog()* es la encargada de esperar el tiempo de programación establecido por el fabricante del microcontrolador MC68HC912B32; 10 milisegundos.

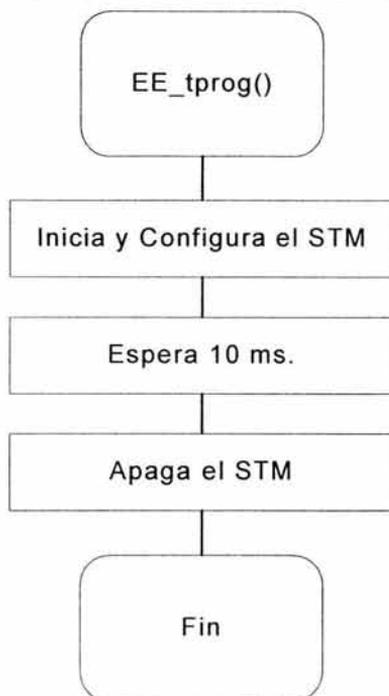


Ilustración 32. Función ee\_tprog

```

int ee_t;
void EE_tprog(void)
{
  TSCR = 128; /* Habilitar el Standar Timer Module */
  TMSK2 = 5; /* Configurar el STM a pasos de 16 us @ 8 MHz */
  ee_t = TCNT + 625; /* Espera 10 ms */
  while(ee_t > TCNT );
  TSCR = 0; /* Deshabilitar el Standar Timer Module */
}
  
```

Ilustración 33. ee\_tprog

#### 3.8.4.4. Función Guardar(tipo, tiempo).

La función Guardar(tipo, tiempo) es la que se encarga de escribir los datos que se necesitan almacenar en la memoria EEPROM. Debido a que la memoria no permite la escritura de palabras no alineadas, separamos los datos a escribir en bytes para poder

escribirlos. Al momento de aplicar el voltaje de programación a la memoria EEPROM las interrupciones de inhiben para proteger la memoria.

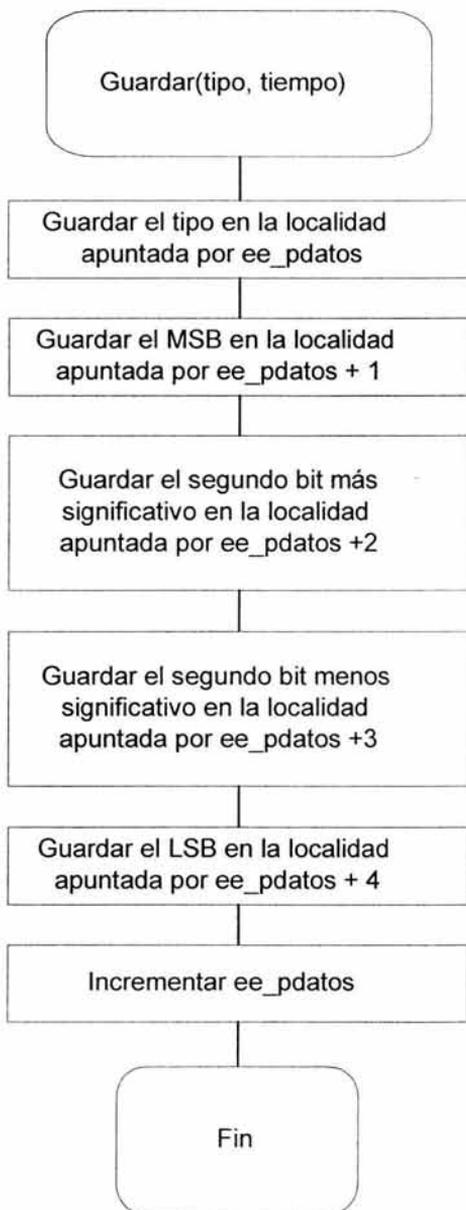


Ilustración 34. diagrama guardar tiempo

```

char *ee_caracter;
void Guardar(char dato, long tiempo)
{
  EEPROT=0; /* Deshabilitar protección de EEPROM */
  EEPROM = 130; /* Anclar el bus de datos al de dirección de EEPROM */
  ee_pdatos->tipo = dato; /* Escribir dato en la dirección correspondiente */
  INTR_OFF(); /* Inicia área crítica, deshabilitar interrupciones */
  EEPROM = 131; /* Aplicar el voltaje de programación a la EEPROM */
  EE_tprog(); /* EE_tprog */
  EEPROM = 130; /* Quitar el voltaje de programación de la EEPROM */
  INTR_ON(); /* Termina área crítica, habilitar interrupciones */

  ee_caracter = (char *) (ee_pdatos) + 1;
  *ee_caracter = (char) (tiempo >> 24); /* dato -> dir */
  INTR_OFF(); /* Inicia área crítica */
  EEPROM = 131; /* EEPROM */
  EE_tprog(); /* Espera EEtprog */
  EEPROM = 130; /* !EEPGM */
  INTR_ON(); /* Termina área crítica */

  ee_caracter = (char *) (ee_pdatos) + 2;
  *ee_caracter = (char) (tiempo >> 16); /* dato -> dir */
  INTR_OFF(); /* Inicia área crítica */
  EEPROM = 131; /* EEPROM */
  EE_tprog(); /* Espera EEtprog */
  EEPROM = 130; /* !EEPGM */
  INTR_ON(); /* Termina área crítica */

  ee_caracter = (char *) (ee_pdatos) + 3;
  *ee_caracter = (char) (tiempo >> 8); /* dato -> dir */
  INTR_OFF(); /* Inicia área crítica */
  EEPROM = 131; /* EEPROM */
  EE_tprog(); /* Espera EEtprog */
  EEPROM = 130; /* !EEPGM */
  INTR_ON(); /* Termina área crítica */

  ee_caracter = (char *) (ee_pdatos) + 4;
  *ee_caracter = (char) (tiempo); /* dato -> dir */
  INTR_OFF(); /* Inicia área crítica */
  EEPROM = 131; /* EEPROM */
  EE_tprog(); /* Espera EEtprog */
  EEPROM = 130; /* !EEPGM */
  INTR_ON(); /* Termina área crítica */
  EEPROM = 128; /* Valor de inicio */
  EEPROT = 255; /* Protección */
  ee_pdatos++;
}

```

Ilustración 35. Función guardar tiempo

#### 3.8.4.5. Función Borrar().

Esta función se encarga de colocar todas la localidades de la memoria EEPROM en el estado de borrado 0xFF.

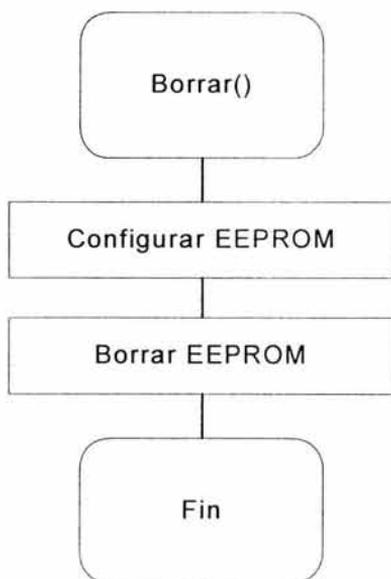


Ilustración 36. Diagrama Borrar

```

/* Función global para borrar EEPROM */
void Borrar(void)
{
    EEPROT = 0; /* Sin Protección */
    EEPROM = 6; /* !BULKP EERASE EELAT */
    INTR_OFF(); /* Empieza área crítica */
    EEPROM = 7; /* EEPGM */
    EE_tprog(); /* Espera EEtprog */
    EEPROM = 6; /* !EEPGM */
    INTR_ON(); /* Termina área crítica */
    EEPROM = 128; /* Valor de inicio */
    EEPROT = 255; /* Protección */
}
  
```

Ilustración 37. borrar

### 3.8.5. Comunicación Serial (SCI).

El objetivo del módulo de comunicación serial es intercambiar información entre el sistema de adquisición de datos y el sistema de exhibición de resultados. Esto es, los datos recaudados por el aforador deberán ser enviados al sistema de publicación que contenga el programa de presentación diseñado para dicho propósito. Típicamente, el programa estará

soportado por una computadora portátil, sin embargo, podrá encontrarse en cualquier dispositivo que permita una comunicación serial al aforador y que sea capaz de ejecutar programas WIN32.

En esta sección se presenta también el protocolo diseñado para el intercambio de datos entre los sistemas, de tal forma que, en caso de no contar con un sistema que pueda ejecutar el programa diseñado en esta tesis, se tengan las bases para programar otro adecuadamente. Por ejemplo, será necesario diseñar otro programa de presentación de datos si el dispositivo en el que se desea recolectar y publicar los datos es una agenda personal ó un sistema de transmisión de radio.

El procedimiento de comunicación serial de nuestro aforador vehicular utiliza la interfaz serial del MC68HC912B32; esta consiste en dos sub-sistemas de entrada/salida serie independientes, la interfaz de comunicación serial (SCI) y la interfaz periférica serial (SPI). Las terminales de cada una de las interfaces se encuentran en el puerto de propósito general S, con lo que dichas terminales son compartidas por los sistemas de comunicación y por el sistema de entrada/salida<sup>8</sup>.

#### ***3.8.5.1. Interfaz de comunicación serial.***

La interfaz de comunicación serial del MC68HC912B32 es un sistema de comunicación asíncrona del tipo no retorno a cero NRZ, con formato de un bit de inicio, ocho ó nueve bits de datos y un bit de paro. Contiene un circuito independiente para generar la tasa de transferencia de datos (baud rate), un transmisor y un receptor de datos serie. Puede ser configurado para ocho ó nueve bits de dato (uno de los cuales puede ser designado como bit de paridad, par ó impar). Si se habilita, la paridad de los datos enviados y recolectados puede ser revisada y señalizada por el Hardware. El generador de la tasa de transferencia de datos (baud rate) esta basado en un contador, lo que permite la flexibilidad al escoger la taza de transferencia. Existe un “despertador de recepción” (reciver wake-up), un detector de cambio en la línea (idle line detector), modo retroalimentación (loop-back mode) y varias características

---

<sup>8</sup> MC68HC912B32 *Technical Summary 16-Bit Microcontroler*. Pág. 83. Motorola INC., 1997.  
Ref. MC68HC912B32TS/D

para detección de errores. Dos terminales proveen la interfaz externa para transmitir (TXD) y recibir (RXD) datos<sup>9</sup>.

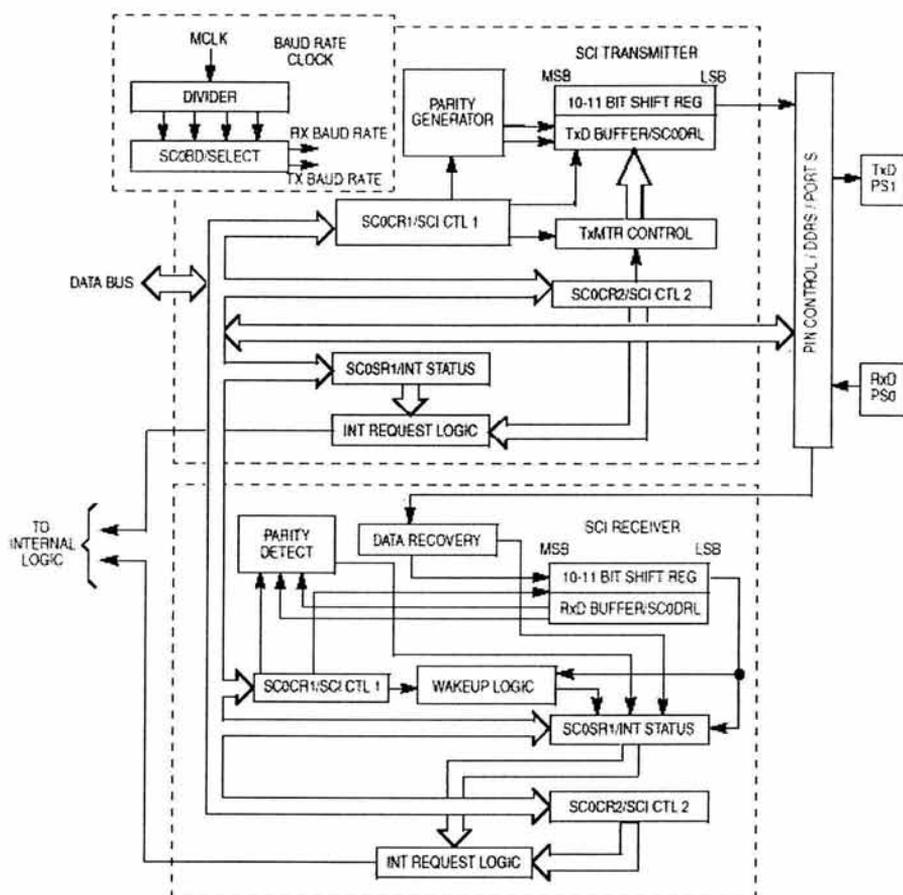


Ilustración 38. Comunicación serial sci

### 3.8.5.2. Formato de Datos.

El formato de datos requiere de las siguientes condiciones:

El estado en la línea debe ser alto antes de transmitir o recibir algún mensaje. Un bit de inicio (lógica baja) se utiliza para indicar el comienzo de cada carácter enviado ó recibido. Los

<sup>9</sup> Idem. Pág. 83

caracteres son transmitidos enviando ó recibiendo el bit menos significativo (LSB) primero. Un bit de paro (lógica alta) se utiliza para indicar el fin de una trama (una trama consiste en el bit de inicio, ocho ó nueve bits de datos y un bit de paro). Un paro (BREAK) se define como la transmisión o recepción de ceros durante una trama completa o más. Este SCI soporta paridad por hardware para la transmisión y la recepción<sup>10</sup>.

### 3.8.5.3. Generador de Tasa de transferencia de datos (Baud Rate Generator).

El generador de tasa de transferencia de datos (baud rate) del MC68HC912B32 se basa en un contador de 13 bits. Este contador le da al generador una flexibilidad para conseguir un razonable nivel de independencia con la frecuencia de operación del CPU, además es capaz de conseguir tasas de transferencia estándar con un mínimo de porcentaje de error. La fuente del reloj del generador es el reloj principal P11.

Baud Rate Deseable	Divisor @ P = 4 MHz	Divisor @ P = 8MHz
110	2273	4545
300	833	1667
600	417	833
1200	208	417
2400	104	208
4800	52	104
9600	26	52
14400	17	35
19200	13	26
38400	-	13

Tabla 32. Generador de tasa de transferencia

### 3.8.5.4. El registro SC0CR2.

En nuestro sistema de aforador vehicular sólo configuramos directamente el registro de control 2 del SCI. Este consiste en:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
INICIO:	0	0	0	0	0	0	0	0

Tabla 33. SCOCR2

<sup>10</sup> Idem. Pág. 84

<sup>11</sup> Idem. Pág. 85

---

TIE: Habilitar la interrupción en transmisión.

0 = interrupción TDRE deshabilitada.

1 = La interrupción SCI será requerida siempre y cuando la bandera TDRE sea colocada.

TCIE: Habilitar la interrupción de transmisión completa.

0 = Inhibe la interrupción por transmisión completa.

1 = La interrupción SCI será requerida siempre y cuando la bandera TC sea colocada.

RIE: Habilitar la interrupción por recepción.

0 = Inhibe las interrupciones por RDRF y OR.

1 = La interrupción SCI será requerida siempre y cuando sea colocada la bandera RDRF ó la bandera OR.

ILIE: Habilitar la interrupción por cambio en la línea.

0 = La interrupción por cambio en la línea es detenida.

1 = La interrupción SCI es requerida siempre y cuando la bandera IDLE sea colocada.

TE: Habilitar Transmisión.

0 = Inhibe la transmisión.

1 = Se habilita el circuito de transmisión del SCI y la terminal TXD (puerto S, bit 1) es asignada al transmisor. El bit TE puede ser utilizado para determinar el estado de la línea antes de habilitar la transmisión.

RE: Habilitar Receptor.

0 = Inhibe al Receptor.

1 = Habilita el circuito de recepción del SCI.

RWU: Control del despertado por recepción.

0 = El receptor SCI trabaja normalmente.

1 = Habilita la función de despertador por recepción e inhibe las interrupciones por recepción. Normalmente este bit es borrado para despertar el receptor.

SBK: Manda PARO (BREAK).

0 = El generador de paro está apagado.

1 = Genera el código de paro (10 ó 11 ceros consecutivos)<sup>12</sup>.

Ejemplo: si quisiéramos iniciar el SCI para poder transmitir y recibir datos y además que interrumpa al CPU cuando se reciba algún dato debe escribir en este registro el número decimal 44.

#### ***3.8.5.5. Protocolo de comunicación.***

Como hemos determinado, el sistema de aforo vehicular almacenará los datos que se refieren al tipo de vehículo y al tiempo en el que estos pasaron. Sin embargo, el que estén los datos en el microcontrolador no nos sirve de mucho, es necesario recolectar esos datos para poder hacer uso de la información que nos otorgan. Este es el propósito del módulo de comunicación serial.

El siguiente protocolo fue diseñado para facilitar la recolección de datos del sistema aforador por cualquier sistema con acceso a un puerto de comunicación serial del tipo NRZ RS-232.

#### ***3.8.5.6. Formato de las tramas.***

La estructura de las tramas enviadas y recibidas por el microcontrolador es la siguiente:

Tasa de transmisión de información: (baud rate) 9600 bps.

Un bit de inicio.

8 bits de datos.

Un bit de paro.

#### ***3.8.5.7. Formato de los datos.***

El sistema de aforo vehicular esta programado para recibir datos en formato ASCII estándar con la siguiente estructura.

---

<sup>12</sup> Idem. Pág. 87 y 88

---

<C><COM>

Los comandos del sistema se componen de dos caracteres, el primero de ellos se utiliza para establecer correctamente la interrupción por recepción y puede tener cualquier valor ASCII. El segundo determina el comando a ordenar, la lista de comandos disponibles es la siguiente:

'1' Ordena al microcontrolador que envíe los todos los datos por el puerto de comunicación.

'2' Ordena al microcontrolador borrar todos los datos que contenga y que inicie el sistema.

Ejemplo: Para recolectar los datos del microcontrolador será necesario enviar la siguiente cadena "\$1".

El sistema de aforo vehicular esta programado para enviar datos en formato ASCII estándar con la siguiente estructura:

<ENCABEZADO><lf><cr><T><lf><cr><ND><lf><cr><D1,T1,D2,T2,...Dn,  
Tn,><lf><cr>

en donde:

lf: Alimentador de línea (line feed 0x10).

cr: Retorno de carro (carry return 0x13).

ENCABEZADO: Cadena de caracteres con la versión del sistema ("Aforo 1.0").

T: Tiempo transcurrido desde la última iniciación del sistema, en segundos.

ND: Número de caracteres a enviar.

Dx: Tipo del vehículo n, los valores de Dx son los siguientes:

A vehículo de 2 ejes.

B vehículo de 3 ejes.

C vehículo de 4 ejes.

D vehículo de 5 ejes.

E vehículo de 6 ejes.

---

F vehículo de 7 ejes.

G vehículo de 8 ejes.

H vehículo de 9 ejes.

I eje pequeño.

J eje grande.

Tx: Tiempo en segundos transcurridos desde la iniciación del sistema hasta cuando pasó el vehículo n.

Ejemplo: Supongamos que el sistema ha sido iniciado hace una hora y ha registrado 2 vehículos, un automóvil a los 10 minutos y un camión de 5 ejes a los 25 minutos después de iniciado el sistema. Si recolectamos los datos entonces obtendremos el siguiente mensaje:

“Aforo 1.0

3600

2

A,600,D,1500,

“.

### ***3.8.6. Funciones del módulo SCI.***

A continuación se presentan las rutinas necesarias para lograr la comunicación entre el sistema de aforo vehicular y el sistema de presentación de datos cumpliendo con el protocolo propuesto. El módulo SCI esta formado por el archivo Sci.c que contiene el código de las funciones y el archivo de encabezado Sci.h que contiene las declaraciones de las variables y funciones externas del módulo.

El código de las funciones está escrito en ANSI C13 y fue compilado satisfactoriamente en el compilador ImageCraft ICC12©14.

---

<sup>13</sup> C Programming in ICC12. Joshep Valvano

<sup>14</sup> Capitulo 2.4 de esta Tesis y [www.imagecraft.com](http://www.imagecraft.com)

### 3.8.6.1. Archivo de encabezados Sci.h.

El archivo Sci.h contiene la declaración de las variables y funciones externas del módulo SCI. La función `Inicia_Sci()`, es necesaria para iniciar el sistema de comunicación serial y es requerida en la función `main()` del módulo principal.

```

/*****
/* Archivo Sci.h
/* Encabezado del módulo de atención a la interrupción
/* SCI
*****/
void Inicia_SCI(void);

```

Ilustración 39. Sci.h

### 3.8.6.2. Función `Inicia_Sci()`.

La función `Inicia_Sci()` inicia el sistema de comunicación serial para 9600,8,1y habilita la interrupción por recepción de datos. Utiliza la macro `setbaud(BAUDRATE)` del módulo `912b32.h` que esta incluida en el compilador ICC12.

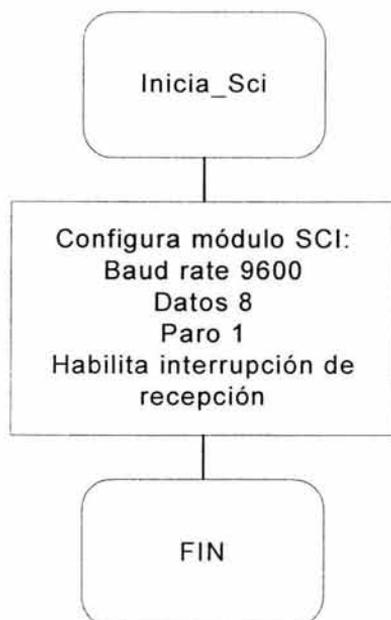


Diagrama 2. `Iniciar_Sci()`

---

```
/* Inicialización de la comunicación serial */
```

```
void Inicia_Sci(void)
```

```
{
```

```
  setbaud(BAUD9600);
```

```
  SC0CR2 = 44;
```

```
}
```

```
Código 1. Inicia_Sci()
```

```
3.4.3.3.3 Función Rai_Sci().
```

Esta rutina es la que se encarga de atender la interrupción por recepción de datos, aquí se evalúan los comandos y se ejecutan las funciones correspondientes. Al entrar a la rutina de interrupción, se cancelan las interrupciones posteriores por recepción de datos pero se permite las demás interrupciones, de esta manera, el sistema sigue contando con el reloj y con la atención de los eventos.

Se utiliza la función `getchar()` de la entrada y salida estándar proporcionada por el compilador ICC12 en `stdio.h`.

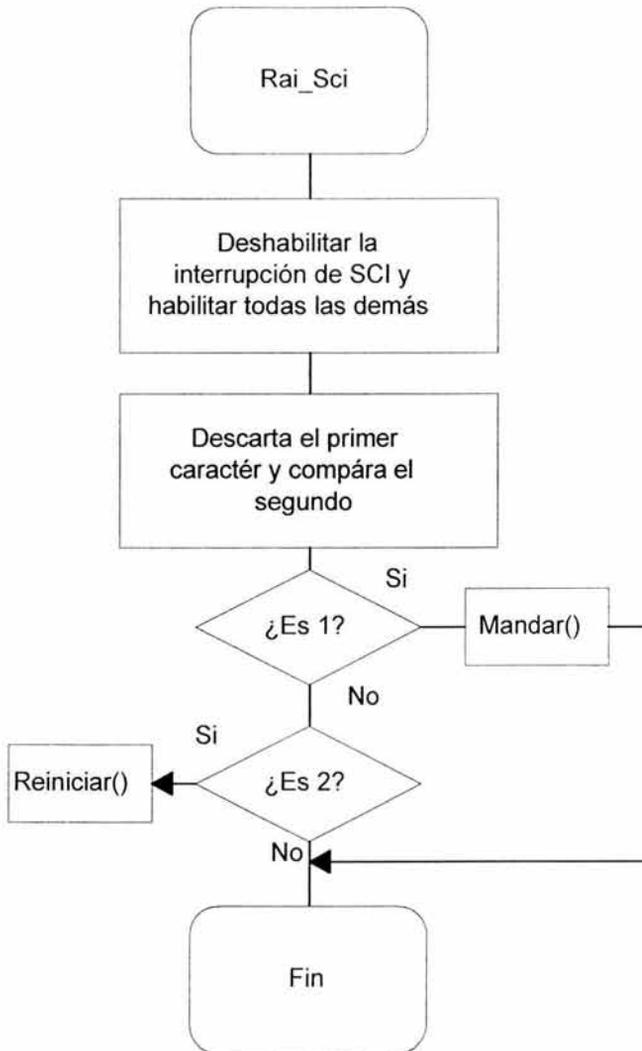


Diagrama 3. Rai\_Sci()

```
/* Rutina de atención a la interrupción SCI */
```

```
void rai_sci(void)
```

```
{
```

```
SC0CR2 = 12; /* No Interrupción de SCI */
```

```
INTR_ON(); /* Habilita Interrupciones */
```

---

```
    getchar()); /* Descartar primer caracter */  
    switch(getchar())  
    {  
    case REDPILL:  
        Mandar();  
        break;  
    case BLUEPILL:  
        Reiniciar();  
    default:  
        break;  
    }  
    SCOCR2 = 44;  
}
```

Código 2. Rai\_Sci()

#### 3.4.3.3.4 Función Mandar()

La función mandar se encarga de transmitir los datos recolectado por el sistema de aforo vehicular cumpliendo con el protocolo determinado. Después de mandar los datos el sistema se queda en el estado que estaba antes de ejecutar el comando. Para adquirir una serie de datos nuevos es necesario iniciar el sistema mediante el comando “\$2”.

En esta función se utilizan las funciones printf(), putchar() de la entrada y salida estándar que son proporcionadas por el compilador ICC12 en stdio.h.

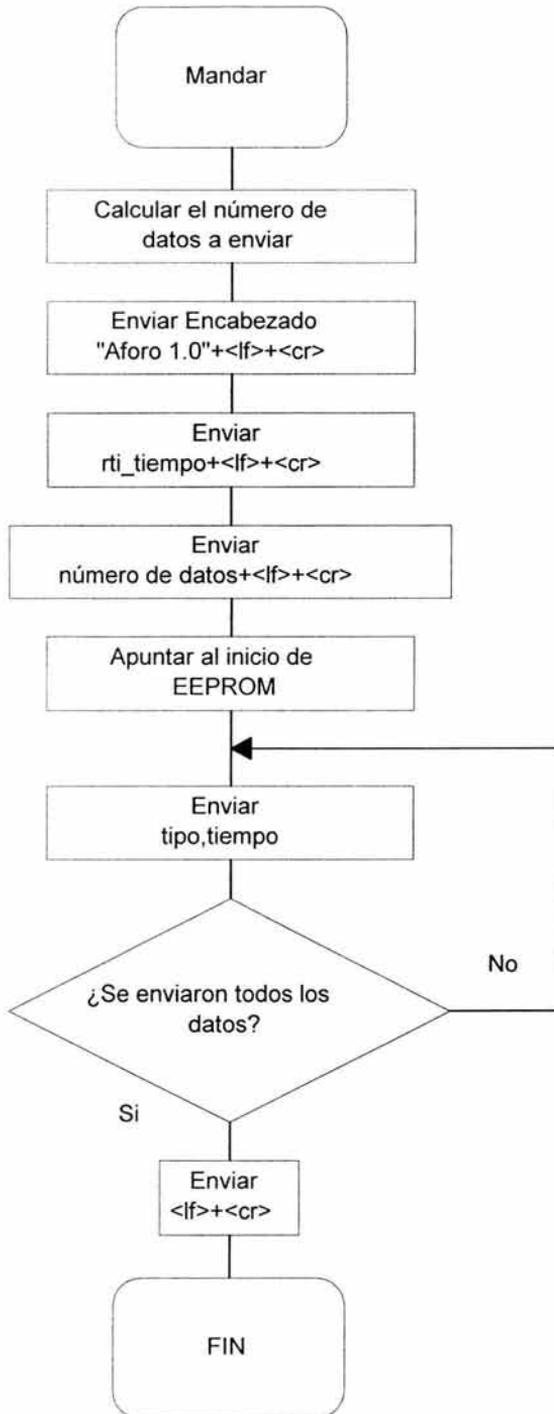


Diagrama 4. Mandar()

```

unsigned int num_datos;

struct str_ee_datos *sci_datos;

void Mandar(void)
{
    num_datos = ((unsigned int)(ee_pdata) - (unsigned int)(EEINICIO) + 1) / 5;
    printf("Aforo 1.0.\r\n");
    printf("%d\r\n", rti_tiempo);
    printf("%d\r\n", num_datos);
    sci_datos = (struct str_ee_datos)(EEINICIO);
    while(sci_datos++ < ee_datos)
        printf("%d,%d,", sci_datos->tipo, sci_datos->tiempo);
    putchar('\r');
    putchar('\n');
}

```

Código 3. Mandar()

#### 3.4.3.3.5 Función Reiniciar()

Esta función se encarga de iniciar el sistema de aforo vehicular, primero borra la memoria EEPROM, después escribe en el registro contador de programa (PC) de microcontrolador la dirección de inicio de programa.

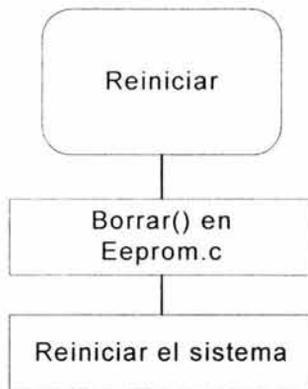


Diagrama 5. Reiniciar()

```

/* Reinicia Sistema */
void Reiniciar(void)
{
  Borrar();
  _start();
}
  
```

Código 4. Reiniciar()

En el programa también se incluyen instrucciones al preprocesador para indicarle que la función `Rai_Sci()` es una rutina de atención a interrupción y que debe colocar la instrucción `RTI` al final de la rutina. La otra instrucción escribe en el vector de atención de interrupción por `SCI` (la dirección `0xFFD6`) la dirección de `Rai_Sci()`.

### 3.3.4 Solicitud Externa de Interrupción.

El objetivo del módulo de solicitud externa de interrupción (`IRQ`) es gobernar los eventos que el sistema de aforo vehicular requiere evaluar. Este módulo se encarga de procesar la información referente a el paso de los vehículos por los sensores del sistema; da servicio a la interfaz del sistema.

La mayor parte del trabajo del microprocesador esta en este módulo debido a que, en él, se encuentran llamadas a varias funciones ubicadas en diversos módulos. El proceso de este módulo se basa en el sistema de solicitud externa de interrupción del microcontrolador

MC68HC912. Este sistema, en resumen, interrumpe el trabajo de la unidad central de procesamiento del microcontrolador debido al cambio en el estado de una terminal del mismo.

#### 3.4.4.1 Interrupciones Elegibles.

Las fuentes de interrupción elegible incluyen a los sistemas periféricos en el circuito integrado y a las solicitudes de servicio a interrupciones externas. Las interrupciones provenientes de estas fuentes son reconocidas cuando se ha borrado el bit de selección global de interrupción I, ubicado en el registro de control CCR. El estado predefinido de este bit es uno después de la inicialización del sistema, pero puede escribirse en cualquier momento.

La prioridad de las fuentes de interrupción esta predefinida, pero a cualquiera de ellas se le puede asignar la mayor prioridad mediante el registro de mayor prioridad HPRIOR.

Un interruptor al que se le asigno la mayor prioridad sigue subordinado al bit de interrupción global I y a los bits locales asociados. Los vectores de interrupción no son afectados por la asignación de prioridad. El registro HPRIOR puede ser escrito únicamente cuando las interrupciones son inhibidas, esto es cuando el bit global de interrupción esta en uno. La siguiente tabla muestra las fuentes de interrupción y los vectores asociados en el orden prioritario predefinido<sup>15</sup>.

Dirección del Vector	Fuente de Interrupción	Bit asociado en CCR	Registro(Bit) local de habilitación	Valor a escribir en HPRIOR para ascender
\$FFFE, \$FFFF	Reset	-	-	-
\$FFFC, \$FFFD	COP	-	COPCTL(CME, FCME)	-
\$FFFA, \$FFFB	Falla en COP	-	Selector de frecuencia de COP	-
\$FFF8, \$FFF9	Instrucción desconocida	-	-	-
\$FFF6, \$FFF7	SWI	-	-	-
\$FFF4, \$FFF5	XIRQ	X	-	-
\$FFF2, \$FFF3	IRQ	I	INTCR(IRQEN )	\$F2

<sup>15</sup> MC68HC912B32 Technical Summary 16-Bit Microcontroller. Pág. 52. Motorola INC., 1997. Ref. MC68HC912B32TS/D

\$FFF0, \$FFF1	RTI	I	RTICTL(RTIE)	\$F0
\$FFD6, \$FFD7	SCI 0	I	SC0CR2(TIE, TCIE, RIE, ILIE)	\$D6

Tabla 34. Vectores de Interrupción

## 3.4.4.1.1 Registro de Control de interrupción INTCR

El registro de control de interrupción tiene a su cargo el dominio de la solicitud externa de interrupción. En el se registran los comandos al módulo de interrupción.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	IRQE	IRQEN	DLY	0	0	0	0	0
INICIO:	0	1	1	0	0	0	0	0

## Registro 1. INTCR

En donde:

IRQE: Selecciona el flanco que activa la solicitud de interrupción

0 = La terminar IRQ es configurada para nivel bajo.

1 = La terminal IRQ es configurada para flanco de bajada.

IRQEN = Permite la solicitud externa de interrupción.

0 = No se permite la interrupción.

1 = Se permite la interrupción.

DLY : Habilita el oscilador de retraso en encendido.

0 = El microcontrolador regresa inmediatamente de un paro.

1 = Se impone un retraso para estabilización de 4096 ciclos después de retornar de un paro<sup>16</sup>.

## 3.4.4.1.2 El registro de mayor prioridad a interrupción HPRIOR.

El registro HPRIOR nos permite asignar una mayor prioridad a cualquier fuente de interrupción. En el sistema aforador vehicular necesitamos que la interrupción periódica tenga la mayor prioridad.

<sup>16</sup> Idem, Pág. 53

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	1	1	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0
INICIO:	1	1	1	1	0	0	1	0

Registro 2. HPRIOR

Para dar la mayor prioridad a una fuente de interrupción se debe escribir el menor byte de la dirección del vector de dicha fuente de interrupción en el HPRIOR. Por ejemplo, para otorgar la mayor prioridad a la interrupción en tiempo real debemos escribir \$F0 en el registro HPRIOR17.

### 3.3.4.2 Funciones del módulo IRQ.

A continuación se presentan las rutinas necesarias para lograr administrar los eventos externos del sistema de aforo vehicular. El módulo IRQ esta formado por el archivo Irq.c que contiene el código de las funciones y el archivo de encabezado Irq.h que contiene las declaraciones de las variables y funciones externas del módulo.

El código de las funciones está escrito en ANSI C18 y fue compilado satisfactoriamente en el compilador ImageCraft ICC12©19.

#### 3.3.4.2.1 El archivo de encabezados Irq.h

En este archivo se encuentran la declaración de funciones y variables globales del módulo.

```

/*****/
/* Archivo Irq.h */
/* Encabezado del módulo de atención a la interrupción */
/* IRQ */
/*****/

```

<sup>17</sup> Idem, Pág. 54

<sup>18</sup> *C Programming in ICC12*. Joshep Valvano

<sup>19</sup> Capítulo 2.4 de esta Tesis y [www.imagecraft.com](http://www.imagecraft.com)

---

```
void Inicia_Irq(void);
```

Código 5. Irq.h

#### 3.3.4.2.2 La función Inicia\_IRQ()

En esta función se escribe en el registro INTCR la palabra correspondiente al inicio del sistema de solicitud externa de interrupción.

```
/* Inicia IRQ */
```

```
void Inicia_Irq(void)
```

```
{
```

```
    INTCR = 192;
```

```
}
```

Código 6. Inicia\_IRQ()

#### 3.3.4.2.3 La función Irq\_Cuenta()

Esta función se encarga de determinar cuantos botones fueron apretados por el vehículo en un evento determinado, esto se logra mediante la disyunción de los datos obtenidos en un puerto y una constante de valor 1 recorrida a la izquierda 8 veces.

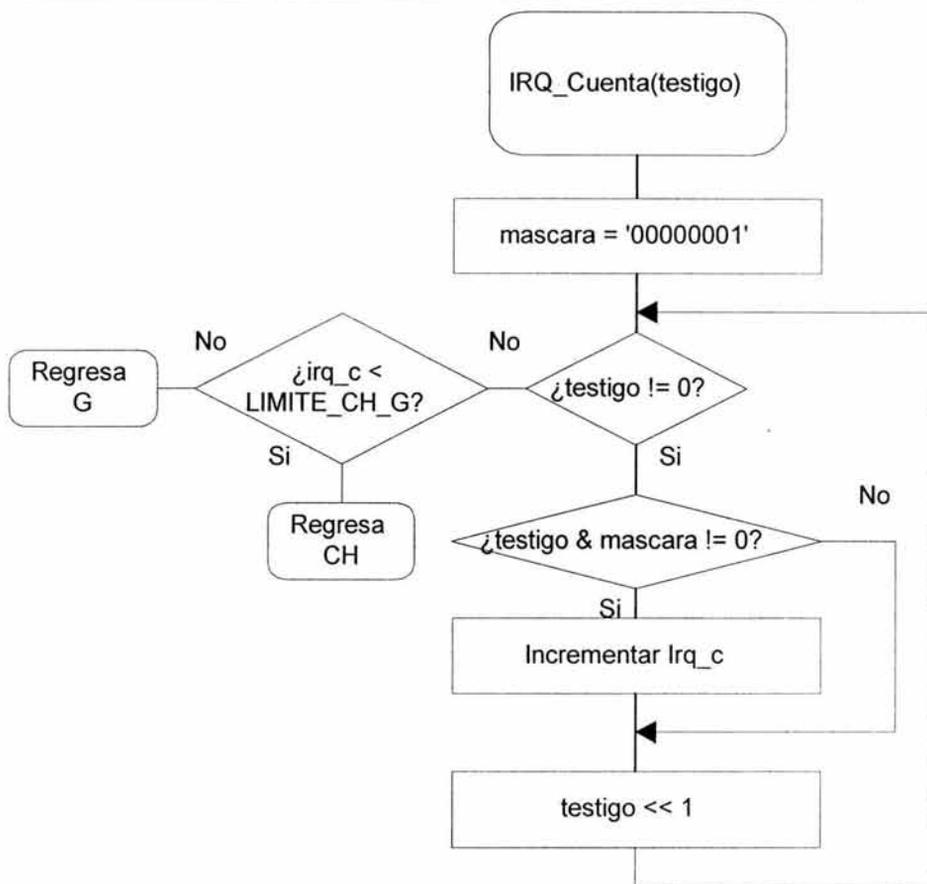


Diagrama 6. IRQ\_Cuenta(testigo)

```
/* Cuenta cuantos botones se apretaron */
```

```
int irq_c;
```

```
unsigned char irq_mascara;
```

```
int IRQ_Cuenta(unsigned char testigo)
```

```
{
```

```
    irq_c=0;
```

```
    irq_mascara=0x80;
```

```
    while(testigo)
```

```
{  
    if(testigo & irq_mascara) irq_c++;  
    testigo=testigo<<1;  
}  
if(irq_c < LIMITE_CH_G) return(CH);  
else return(G);  
}
```

Código 7. IRQ\_Cuenta(testigo)

#### 3.3.4.2.4 Función Irq\_Push()

La función Irq\_Push() se encarga de guardar ordenadamente en la memoria de acceso aleatorio, en el área de la tabla Datos, el dato recién colectado.

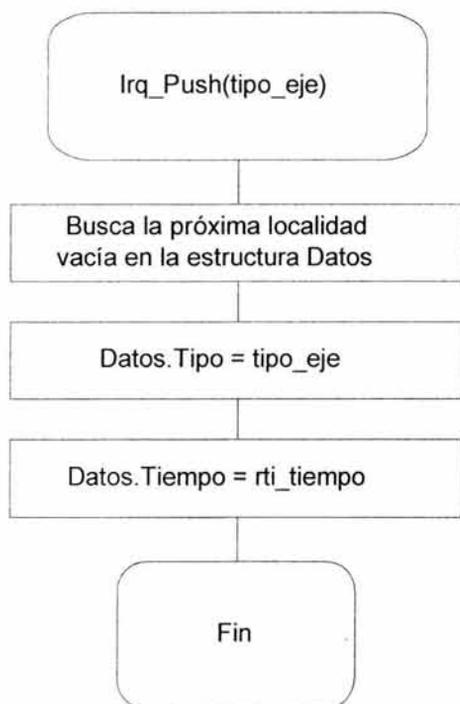


Diagrama 7. Irq\_Push(tipo\_eje)

```
int irq_i;

void IRQ_Push(char tipo_eje)

{

    irq_i = 0;

    while(Datos.Tipo[irq_i] && irq_i < 3) irq_i++;

    Datos.Tipo[irq_i] = tipo_eje;

    Datos.Tiempo[irq_i] = rti_tiempo;

}
```

Código 8. Irq\_Push(tipo\_eje)

#### 3.3.4.2.5 Función Buscar().

En esta rutina se encuentra el código necesario para localizar en la Tabla el correspondiente al dato actual. Regresa un índice que indica el número de renglon en el que los datos coinciden.

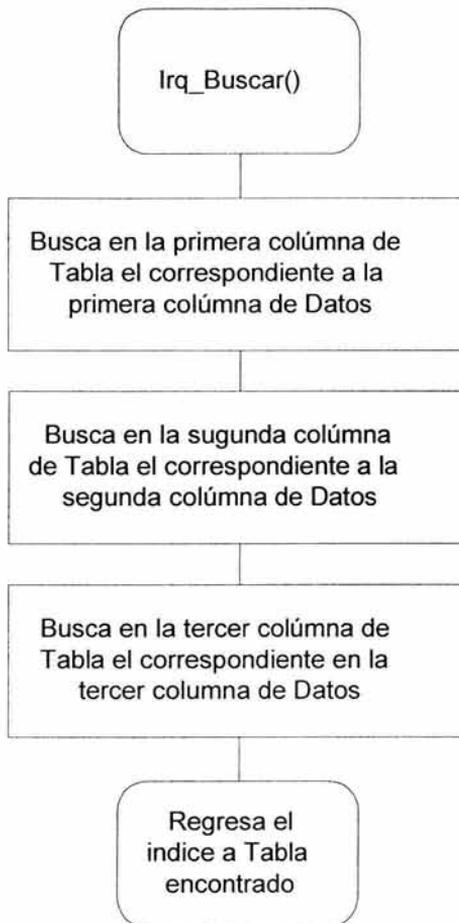


Diagrama 8. Irq\_Buscar()

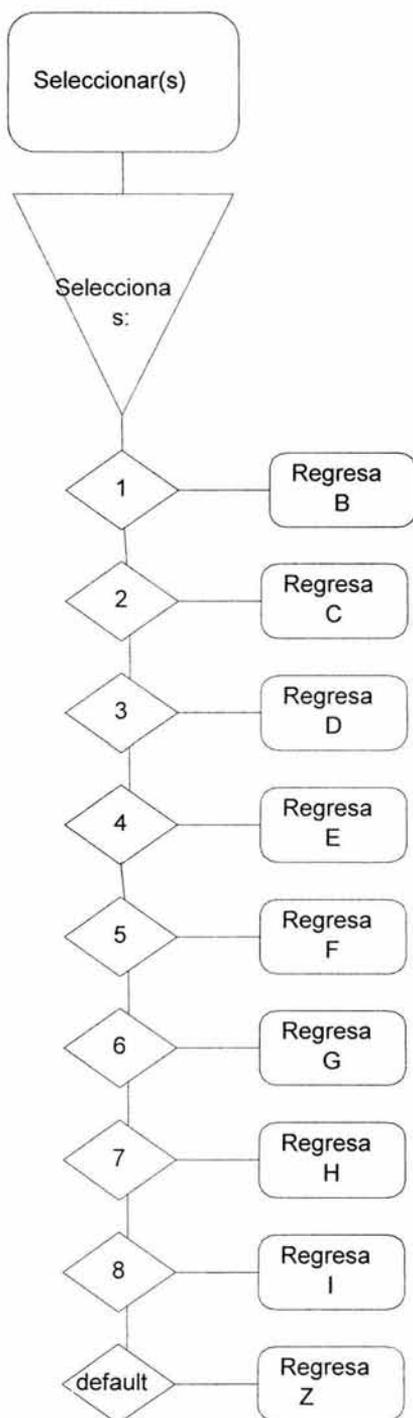
```
int irq_j;  
  
int IRQ_Buscar(void)  
{  
    irq_j = 0;  
    for(irq_i = 0; irq_i < 8; irq_i++)  
    {  
        while(Datos.Tipo[irq_j] == Tabla.Tipo[irq_i][irq_j]) irq_j++;  
    }  
}
```

```
if(irq_j > 2) return(irq_i + 1);  
irq_j = 0;  
}  
return(0);  
}
```

Código 9. Irq\_Buscar()

#### 3.3.4.2.6 Función Selecciona()

La rutina Selecciona() regresa un carácter que indica el tipo de vehiculo pesado que representa el número de ejes argumentados.



## Diagrama 9. Selecciona(s)

```
char Seleccionar(int s)
{
    switch(s)
    {
        case 1: return('B');
                break;
        case 2: return('C');
                break;
        case 3: return('D');
                break;
        case 4: return('E');
                break;
        case 5: return('F');
                break;
        case 6: return('G');
                break;
        case 7: return('H');
                break;
        case 8: return('I');
                break;
        default: return(s - 8);
                break;
    }
}
```

Código 10. Selecciona(s)

### 3.3.4.2.7 La Función IRQ\_F1()

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el primer renglón de la Tabla.

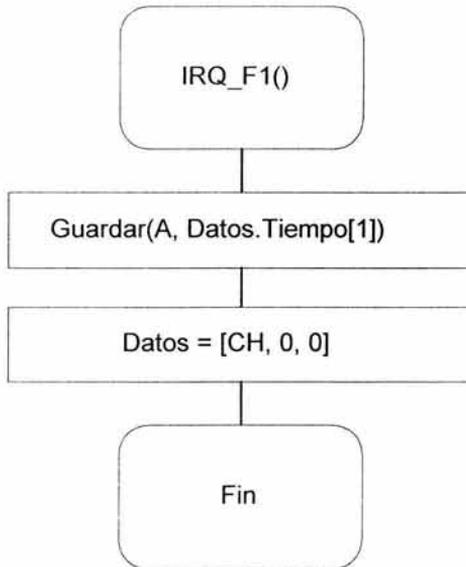


Diagrama 10. IRQ\_F1()

```
void IRQ_F1(void)
```

```
{
  Guardar(A, Datos.Tiempo[1]);
  Datos.Tipo[0] = CH;
  Datos.Tiempo[0] = Datos.Tiempo[2];
  Datos.Tipo[1] = 0;
  Datos.Tipo[2] = 0;
  Datos.Tiempo[1] = 0;
  Datos.Tiempo[2] = 0;
}
```

## Código 11. IRQ\_F2()

## 3.3.4.2.8 La Función IRQ\_F2().

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el segundo renglón de la Tabla.

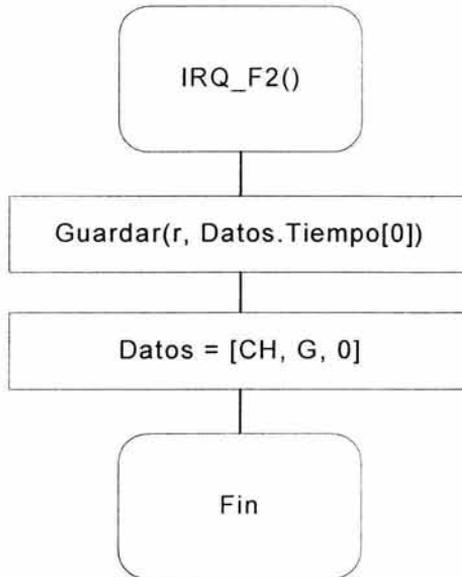


Diagrama 11. IRQ\_F2()

```

void IRQ_F2(void)
{
  Guardar('r', Datos.Tiempo[0]);
  Datos.Tipo[0] = CH;
  Datos.Tiempo[0] = Datos.Tiempo[1];
  Datos.Tipo[1] = G;
  Datos.Tiempo[1] = Datos.Tiempo[2];
  Datos.Tipo[2] = 0;
  Datos.Tiempo[2] = 0;
}
  
```

Código 12. IRQ\_F2()

3.3.4.2.9 La Función IRQ\_F3().

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el tercer renglón de la Tabla.

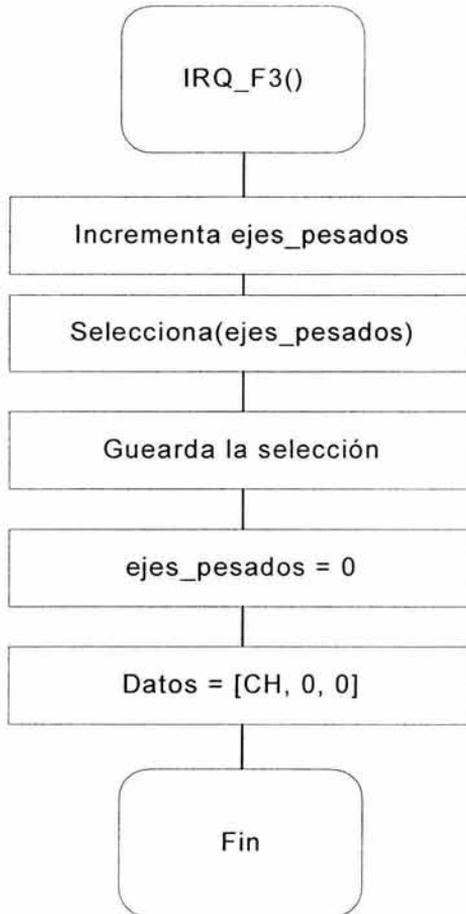


Diagrama 12. IRQ\_F3()

```
void IRQ_F3(void)
{
    irq_ejes_pesados++;
    irq_seleccion = Seleccionar(irq_ejes_pesados);
}
```

```
Guardar(irq_seleccion, Datos.Tiempo[1]);  
irq_ejes_pesados = 0;  
Datos.Tipo[0] = CH;  
Datos.Tiempo[0] = Datos.Tiempo[2];  
Datos.Tipo[1] = 0;  
Datos.Tipo[2] = 0;  
Datos.Tiempo[1] = 0;  
Datos.Tiempo[2] = 0;  
}
```

Código 13. IRQ\_F3()

#### 3.3.4.2.10 La Función IRQ\_F4().

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el cuarto renglón de la Tabla.

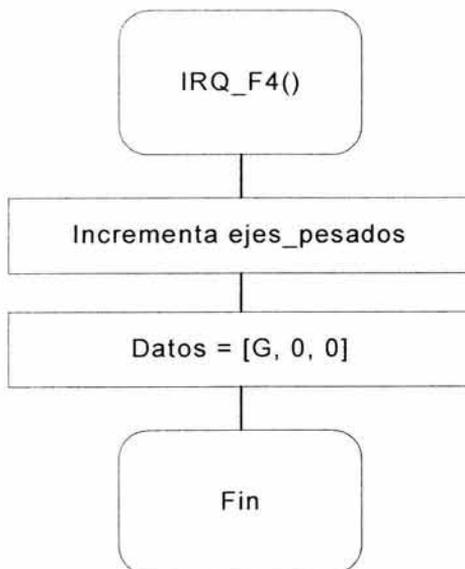


Diagrama 13. IRQ\_F4()

```
void IRQ_F4(void)
```

---

```
{  
    irq_ejes_pesados++;  
    Datos.Tipo[0] = G;  
    Datos.Tiempo[0] = Datos.Tiempo[2];  
    Datos.Tipo[1] = 0;  
    Datos.Tipo[2] = 0;  
    Datos.Tiempo[1] = 0;  
    Datos.Tiempo[2] = 0;  
}
```

Código 14. IRQ\_F4()

3.3.4.2.11 La Función IRQ\_F5().

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el quinto renglón de la Tabla.

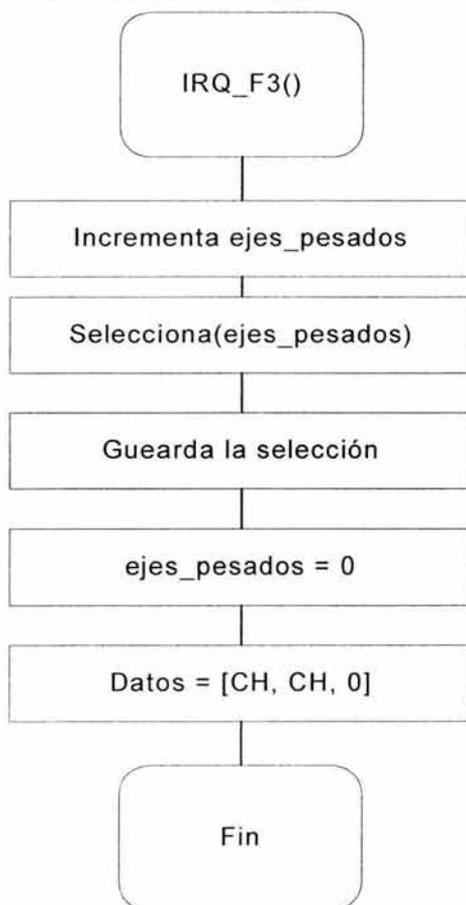


Diagrama 14. IRQ\_F5()

```
void IRQ_F5(void)
{
    irq_ejes_pesados++;
    irq_seleccion = Seleccionar(irq_ejes_pesados);
    Guardar(irq_seleccion, Datos.Tiempo[0]);
    irq_ejes_pesados = 0;
    Datos.Tipo[0] = CH;
```

---

```
Datos.Tiempo[0] = Datos.Tiempo[1];  
Datos.Tipo[1] = CH;  
Datos.Tiempo[1] = Datos.Tiempo[2];  
Datos.Tipo[2] = 0;  
Datos.Tiempo[2] = 0;  
}
```

Código 15. IRQ\_F5()

3.3.4.2.12 La Función IRQ\_F6()

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el sexto renglón de la Tabla.

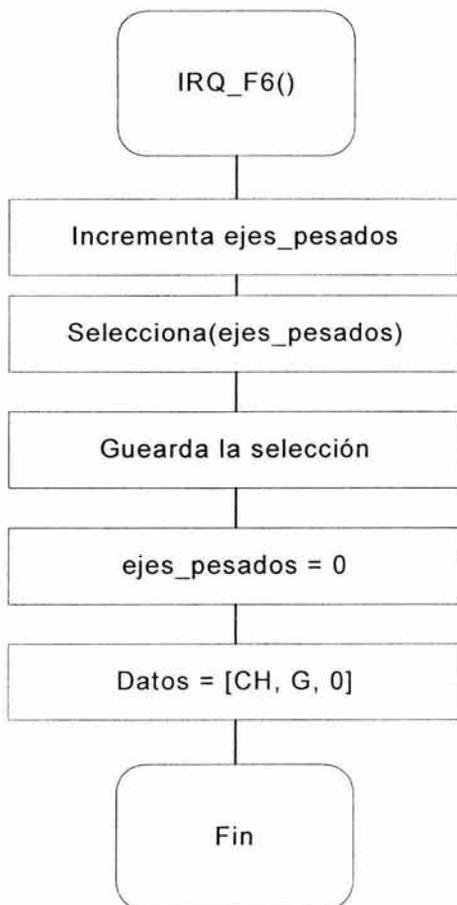


Diagrama 15. IRQ\_F6()

```
void IRQ_F6(void)
{
    irq_ejes_pesados++;
    irq_seleccion = Seleccionar(irq_ejes_pesados);
    Guardar(irq_seleccion, Datos.Tiempo[0]);
    irq_ejes_pesados = 0;
    Datos.Tipo[0] = CH;
```

---

```
Datos.Tiempo[0] = Datos.Tiempo[1];  
Datos.Tipo[1] = G;  
Datos.Tiempo[1] = Datos.Tiempo[2];  
Datos.Tipo[2] = 0;  
Datos.Tiempo[2] = 0;  
}
```

Código 16. IRQ\_F6()

3.3.4.2.13 La Función IRQ\_F7().

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el séptimo renglón de la Tabla.

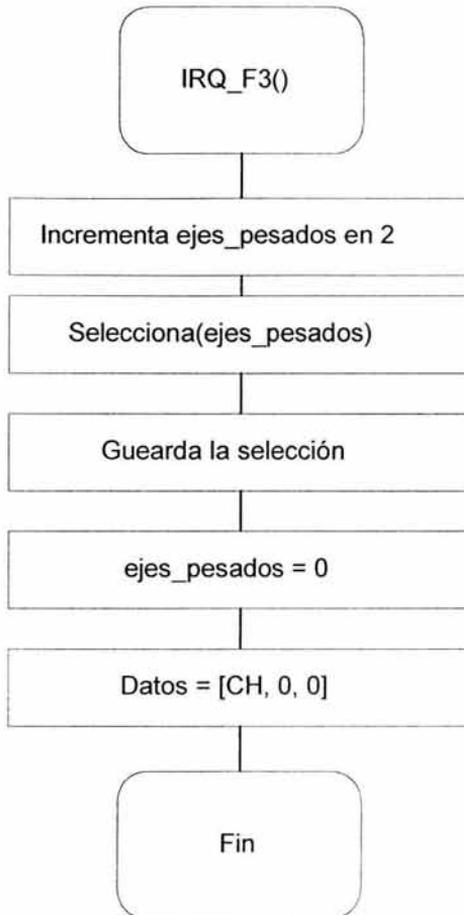


Diagrama 16. IRQ\_F7()

```
void IRQ_F7(void)
{
    irq_ejes_pesados+=2;
    irq_seleccion = Seleccionar(irq_ejes_pesados);
    Guardar(irq_seleccion, Datos.Tiempo[1]);
    irq_ejes_pesados = 0;
    Datos.Tipo[0] = CH;
```

```

Datos.Tiempo[0] = Datos.Tiempo[2];

Datos.Tipo[1] = 0;

Datos.Tipo[2] = 0;

Datos.Tiempo[1] = 0;

Datos.Tiempo[2] = 0;

}

```

Código 17. IRQ\_F7()

#### 3.3.4.2.14 La Función IRQ\_F8().

Esta es la función que se ejecutará cuando se el dato y el estado de la tabla Datos coincida con el último renglón de la Tabla.

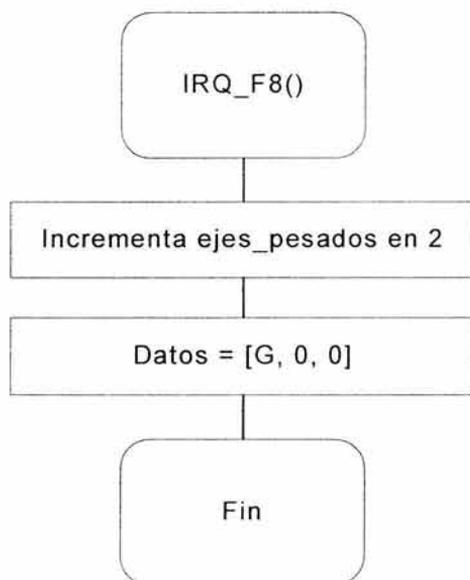


Diagrama 17. IRQ\_F8()

```

void IRQ_F8(void)
{
    irq_ejes_pesados+=2;

    Datos.Tipo[0] = G;
}

```

---

```
Datos.Tiempo[0] = Datos.Tiempo[2];  
Datos.Tipo[1] = 0;  
Datos.Tipo[2] = 0;  
Datos.Tiempo[1] = 0;  
Datos.Tiempo[2] = 0;  
}
```

Código 18. IRQ\_F8()

#### 3.3.4.2.15 Rutina de Atención a Interrupción por solicitud Externa.

Esta función gestiona la dirección de las acciones a tomar cuando es requerida una interrupción por solicitud externa. Cuando un vehículo es detectado por la interfaz externa del sistema de aforo vehicular, el módulo IRQ solicita al CPU del microcontrolador MC68HC912B32 ejecute esta rutina.

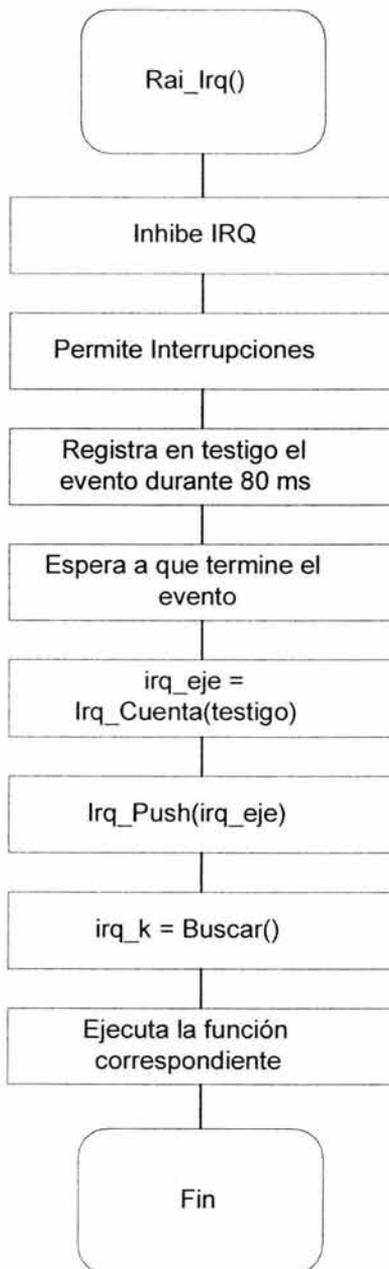


Diagrama 18. Rai\_Irq()

*/\* Rutina de atención de interrupción IRQ \*/*

```
int irq_t, irq_eje, irq_k;

unsigned char testigoT;

void rai_irq(void)
{
    INTCR = 0; /* Deshabilita IRQ */
    INTR_ON(); /* Permite interrupciones */
    TSCR = 128; /* TEN */
    TMSK2 = 5; /* Pre-escalador por 32 */
    testigoT = 0;
    irq_t = TCNT + 5000; /* Registra por 80 ms */
    while(irq_t > TCNT)
    {
        testigoT |= ~PORTT;
    }
    TSCR = 0; /* Valor de inicio */
    while(~PORTT); /* Espero a que suelte */
    irq_eje = IRQ_Cuenta(testigoT);
    IRQ_Push(irq_eje);
    irq_k = IRQ_Buscar();
    if(irq_k) Tabla.Funciones[irq_k - 1]();
    INTCR = 192;
}

Código 19. Rai_Irq()
```

En el programa también se incluyen instrucciones al preprocesador para indicarle que la función `Rai_Irq()` es una rutina de atención a interrupción y que debe colocar la instrucción `RTI` al final de la rutina. La otra instrucción escribe en el vector de atención de interrupción por `IRQ` (la dirección `0XFFF2`) la dirección de `Rai_Irq()`.

### 3.9. Sistema de aforo vehicular a escala y teoría para su implementación.

El sistema se ha diseñado para su aplicación en un arroyo vehicular. Sin embargo el alcance de el proyecto que aquí se presenta es únicamente un modelo a escala funcional.

En esta sección se dará la teoría básica para poderlo implementar en un sistema real, ya que su principio de funcionamiento es idéntico.

#### 3.9.1. Modelo a escala

Para probar el correcto funcionamiento de el contador de aforo vehicular, se diseña un sistema a escala del “sistema aforador vehicular”, que nos sirve para realizar diversas pruebas, sin necesidad de utilizar vehículos ni sensores al piso, la teoría de su funcionamiento en comparación al sistema real es idéntico, solo se diferencia en: La forma de capturar datos, ya que son mini switch. Los puertos de lectura son dos (P y T), en vez de cuatro(en el caso del sistema 1:1).

La relación de estos puertos es la siguiente: al paso de ejes de una sola rodada se oprimen cuatro botones, al paso de ejes con doble rodada se oprimen cinco o mas botones. Esta relación se obtuvo de la tabla de “Dimensiones de neumáticos” del capítulo uno, de las dimensiones de los sensores y mini switch de la siguiente forma:

Los sensores como ya vimos miden 50mm, y las dimensiones críticas de los neumáticos son; clasificación CH 152mm a 400mm, clasificación G 452mm o mas. Esto es que para la clasificación CH, tiene que oprimir 8 botones como máximo, y para la clasificación G tiene que oprimir 9 botones, en el sistema a escala como ya lo mencionamos utilizamos mini switch, sus dimensiones son de 5mm cada uno, pero a diferencia del modelo real que utiliza 32 botones únicamente usamos 16 botones, lo que nos hace tener la relación de que un mini switch, es equivalente a un sensor, y como sus medidas son múltiplos los podemos dividir, así que en el modelo a escala; CH son 4 botones, y G son 5 botones.

Lo referente a el rebote, ya esta considerado dentro del programa, ya que se caracterizaron los sensores al piso, y se elimino, los mini switch también tiene rebote, pero el menor que el de los sensores al piso. Como ya se menciona en el capitulo tres.

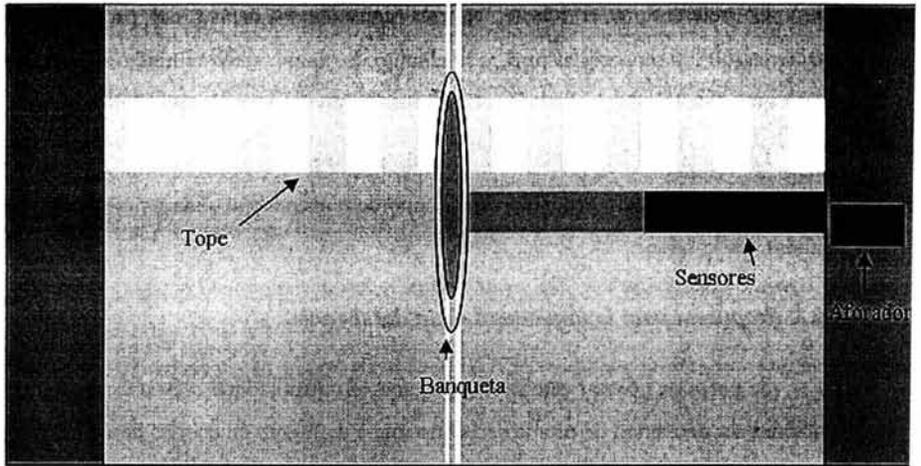
La tarjeta de adquisición de datos utilizada en este modelo a escala es idéntica en diseño al utilizado en el modelo real, solo varia el tipo de sensores que utiliza, y al numero de entradas y salidas que contiene.

### ***3.9.2. Propuesta para la implementación del sistema.***

Se propone colocar este sistema en una autopista federal o estatal, el sistema tiene que funcionar sin necesidad de monitoreo constante y de fuente de energía ilimitada, esto es que el sistema después de ser colocado solamente será monitoreado una vez al mes, donde se descargaran los datos almacenados y se reemplazara la batería que debe durar al menos 31 días.

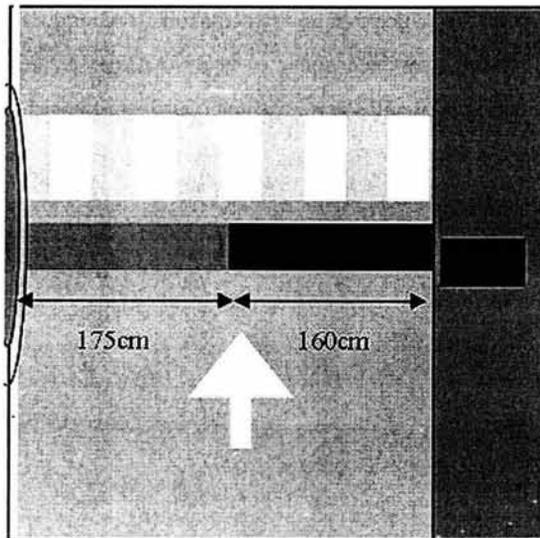
Este contador de aforo ocupara medio carril, con el fin de contar el numero de ejes y el ancho de los neumáticos de la mitad del vehículo.

Se propone poner un tope después de los sensores, para que el vehículo disminuya su velocidad, con esto preservar la vida útil de los botones como también registrar fielmente el ancho de los neumáticos, a su vez poner una banquetta que divida los carriles de contra flujo, para evitar dos cosas, una que rebasen los vehículos y no sean contados, y segundo, forzarlos a que un neumático pase por los sensores, también poner en la misma línea en que van los sensores unos reflejantes, para evitar que los autos mas angostos o incluso motocicletas eviten pasar por los sensores.



**Ilustración 40. Propuesta de ubicación del sistema**

Serán colocados 32 botones al piso sin separación entre cada uno de ellos y comenzando a partir del extremo derecho del carril tomando en cuenta el sentido del mismo. Con esta cantidad de botones se cubren 160cm de los 350cm de ancho que tiene cada carril. Las razones por las que se considero distribuir este numero de sensores son dos principalmente:



**Ilustración 41. Ubicación de sensores**

La primera por la capacidad del microcontrolador para procesar la información entrante a través de sus puertos, donde son utilizados cuatro de ellos; cada uno de 8 pines. Con este numero de puertos se garantiza la recolección de datos a tiempo, quedando listo el micro para registrar el siguiente evento y dejando los otros puertos restantes disponibles para otras aplicaciones.

La segunda razón se refiere principalmente a economizar la instalación de la línea de botones. En lugar de cubrir el ancho total de cada carril con 64 botones o mas, con la mitad de sensores se registran los datos proporcionados por los neumáticos de un solo extremo de cada eje y esto es suficiente para determinar el tipo de vehículo que se trata. Según investigación de campo se registro que solo el 4% de los vehículos en circulación tienen una diferencia significativa en sus neumáticos izquierdos referidos sus derechos, con esto se afirma que los neumáticos de un solo extremo son suficientes para determinar de que tipo se trata.

El microcontrolador a utilizar, se sugiere que sea de uso militar por los rangos de temperatura que tiene que resistir, el almacenamiento de datos se recomienda externo.

Ya que el sistema quedara sin monitoreo, su alimentación será por medio de baterías. Por ejemplo, el sistema actual consume alrededor de 65mA, 5V, dando una potencia de 325mW, si utilizamos una batería de 12v, 12Ah, nos da una potencia de 144Wh. Necesitamos 774h por  $325\text{mw} = 241.8\text{Wh}$ , de consumo para alimentar el sistema 31 días como mínimo. Con el consumo anterior se requieren dos baterías de 12v y 12Ah.

### 3.9.3. Tabla comparativa

Para entender mejor las diferencias y similitudes de los dos sistemas, ponemos esta tabla comparativa.

	Modelo Real	Modelo a escala.
Software		
IRQ, SCI, Reloj	√	√
Almacenamiento interno		√
Almacenamiento externo	√	
Aplicación en Visual Basic	√	√
Hardware		
68hc12b32		√
68hc12A4	√	
Tarjeta de Adquisición	√	√
Sensores Microswitch		16

Sensores	32	
Memoria Externa	√	
Complementos		
Sistema adaptado	√	
Prototipo		√
Alimentación		
Fuente		√
Batería	√	

Tabla 35. Tabla comparativa de los dos sistemas

**CAPITULO 4**  
**PROGRAMA DE RECOLECCIÓN DE DATOS**

#### 4. PROGRAMA DE RECOLECCIÓN DE DATOS AFORO 1.0

Como expresamos en su momento el objetivo principal del Sistema de Aforo Vehicular es proporcionar información confiable del flujo de vehículos en una vía de comunicación elegida. Sin embargo, el sistema no está completo sin un módulo que presente de los datos recolectados; de nada nos servirían dichos datos sino podemos interpretarlos.

El objetivo del Programa de Recolección de Datos Aforo 1.0 es precisamente complementar el Sistema de Aforo Vehicular, proporcionando una forma sencilla de mostrar la información recolectada por el sistema. Para lograrlo, el programa consta de dos módulos principales. Un módulo de comunicación con el Sistema de Aforo Vehicular, para extraer la información que nos proporciona este. Y otro módulo para desplegar los datos recolectados en forma agradable para el interesado.

A continuación explicamos como están formados dichos módulos, que es lo que hacen y cuál es su relación entre ellos.

Módulo de Comunicación serial PC-Microcontrolador.

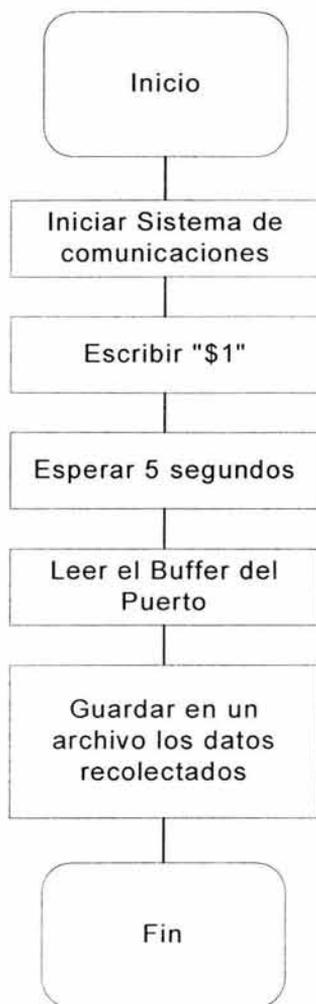
La función más importante del Programa de recolección de Datos Aforo 1.0 es lograr recolectar la información ubicada dentro del Sistema de Aforo Vehicular. Para ello se requiere un sistema que gestione la comunicación entre la Computadora Personal y el Microcontrolador. Visual Basic nos proporciona un complemento activo que nos facilita esta tarea; el MSComm. Utilizar este complemento es muy sencillo, solamente tenemos que informarle el puerto a utilizar, la velocidad de transmisión y recepción, el número de bits a transmitir y/o recibir, paridad, etc.

A continuación se presenta la hoja de propiedades de este complemento.



Figura 4.1, mscomm.

El algoritmo de comunicación que utilizaremos es el siguiente:



**Ilustración 42. Comunicación**

Comunicación con el sistema.

A continuación se presenta el código utilizado en el módulo de comunicación.

```
Private Sub lblconectar_Click()
```

```
Dim temp As Integer
```

```
temp = MsgBox("Conecte el sistema al puerto de comunicaciones seleccionado y haga click en Aceptar para continuar...", vbInformation)
```

---

```
MSComm1.PortOpen = True

ProgressBar1.Value = 0
ProgressBar1.Visible = True
tiempo = 0
Timer1.Enabled = True
Do
    DoEvents
    ProgressBar1.Value = tiempo
Loop While tiempo < 50
Timer1.Enabled = False
ProgressBar1.Visible = False
Buffer = MSComm1.Input
MSComm1.PortOpen = False
ChDir App.Path
On Error Resume Next
CommonDialog1.ShowSave
archivo = CommonDialog1.filename
If archivo Then
    Texto = CStr(Now) & ", " & StrConv(Buffer, vbUnicode)
    Open archivo For Output As #1
    Print #1, Texto
    Close
End If
End Sub
```

---

Código 4.1, Conectar()

#### 4.1. Módulo Abrir

Este módulo es el encargado de contener las funciones que ejecutan el despliegue en pantalla, con un formato legible, de los datos recolectados por el Sistema de Aforo Vehicular.

Al recuperar la información del Sistema de Aforo Vehicular, estos tienen el formato que elegimos para su transmisión, en formato ASCII y separados por comas, es decir de la forma:

<Aforo 1.0>,<Segundos Transcurridos desde su inicio>,<Número de Datos a enviar>,<T1>,<S1>,...,<Tn>,<Sn>,

Que como se observa forman una cadena de texto, la cual se almacena una variable 'Texto' en el programa y es esta con la que se trabaja.

El código del módulo Abrir del programa de adquisición de datos Aforo 1.0 es el siguiente:

```
Public Sub Compilar()
```

```
Dim i, j As Integer
```

```
Dim inicio As Long
```

```
Dim tipo As String
```

```
A = 0
```

```
B = 0
```

```
C3 = 0
```

```
C4 = 0
```

```
C5 = 0
```

```
C6 = 0
```

```
C7 = 0
```

C8 = 0

C9 = 0

R1 = 0

R2 = 0

inicio = InStr(Texto, ",")

If inicio Then

    fecha\_fin = Left(Texto, inicio - 1)

    Texto = Mid(Texto, inicio + 1)

End If

inicio = InStr(Texto, ",")

If inicio Then

    Version = Left(Texto, inicio - 1)

    Texto = Mid(Texto, inicio + 1)

End If

inicio = InStr(Texto, ",")

If inicio Then

    fecha\_inicio = DateAdd("s", -CDBl(Left(Texto, inicio - 1)), fecha\_fin)

    Texto = Mid(Texto, inicio + 1)

End If

inicio = InStr(Texto, ",")

If inicio Then

    elementos = CLng(Left(Texto, inicio - 1))

    Texto = Mid(Texto, inicio + 1)

End If

For i = 1 To elementos

```
inicio = InStr(Texto, ",")

If inicio Then

    tipo = Left(Texto, inicio - 1)

    Texto = Mid(Texto, inicio + 1)

    inicio = InStr(Texto, ",")

If inicio Then

    Select Case tipo

        Case "A"

            A = A + 1

            TipoA(A) = DateAdd("s", CDb(Left(Texto, inicio - 1)), fecha_inicio)

        Case "B"

            B = B + 1

            TipoB(B) = DateAdd("s", CDb(Left(Texto, inicio - 1)), fecha_inicio)

        Case "C"

            C3 = C3 + 1

            TipoC3(C3) = DateAdd("s", CDb(Left(Texto, inicio - 1)), fecha_inicio)

        Case "D"

            C4 = C4 + 1

            TipoC4(C4) = DateAdd("s", CDb(Left(Texto, inicio - 1)), fecha_inicio)

        Case "E"

            C5 = C5 + 1

            TipoC5(C5) = DateAdd("s", CDb(Left(Texto, inicio - 1)), fecha_inicio)

        Case "F"

            C6 = C6 + 1

            TipoC6(C6) = DateAdd("s", CDb(Left(Texto, inicio - 1)), fecha_inicio)
```

---

```
Case "G"

    C7 = C7 + 1

    TipoC7(C7) = DateAdd("s", CDbl(Left(Texto, inicio - 1)), fecha_inicio)

Case "H"

    C8 = C8 + 1

    TipoC8(C8) = DateAdd("s", CDbl(Left(Texto, inicio - 1)), fecha_inicio)

Case "I"

    C9 = C9 + 1

    TipoC9(C9) = DateAdd("s", CDbl(Left(Texto, inicio - 1)), fecha_inicio)

Case "r"

    R1 = R1 + 1

    TipoR1(R1) = DateAdd("s", CDbl(Left(Texto, inicio - 1)), fecha_inicio)

Case Else

    R2 = R2 + CLng(tipo)

    For j = R2 + 1 To R2 + CLng(tipo)

        TipoR2(j) = DateAdd("s", CDbl(Left(Texto, inicio - 1)), fecha_inicio)

    Next j

End Select

Texto = Mid(Texto, inicio + 1)

End If

End If

Next i

End Sub
```

## Código 4.2. Compilar()

El programa también incluye distintas funciones encargadas de la visualización en pantalla del programa y la gráfica de los datos.

## 4.2. Manual para el usuario del programa.

### 4.2.1. Instalación.

#### 4.2.1.1. Requerimientos del Sistema

El sistema Aforo 1.0 es una aplicación visual basada en eventos y objetos, soportada en sistema Win32, los requerimientos mínimos del sistema son los siguientes:

Sistema Operativo	Windows® 95/98/2000/XP/, Windows NT.
Procesador	Intel® Pentium I ó equivalente @ 100MHz.
Disco Duro	Al menos 10 Mb. de espacio disponible.
Memoria	16 Mb.
Comunicación	Puerto de Comunicación Serial RS232 estándar.

**Tabla 36. Requerimientos del sistema**

En esta sección se explica como instalar el sistema Aforo 1.0, sobre la documentación disponible y acerca de la página electrónica del proyecto.

#### 4.2.1.2. Como Instalar:

Adquiera una versión del sistema Aforo 1.0, se puede obtener una versión gratuita en la página del proyecto [www.aforador.com.mx](http://www.aforador.com.mx) o solicitar a los autores una copia en CD-ROM. [tesis@aforador.com.mx](mailto:tesis@aforador.com.mx)

Haga doble clic en el ícono Setup.exe y siga las instrucciones en la pantalla.

Ejecutar el Programa.

Haga clic en Menú Inicio.

Seleccione Programas, Facultad de Ingeniería UNAM.

Haga clic en Aforo 1.0.

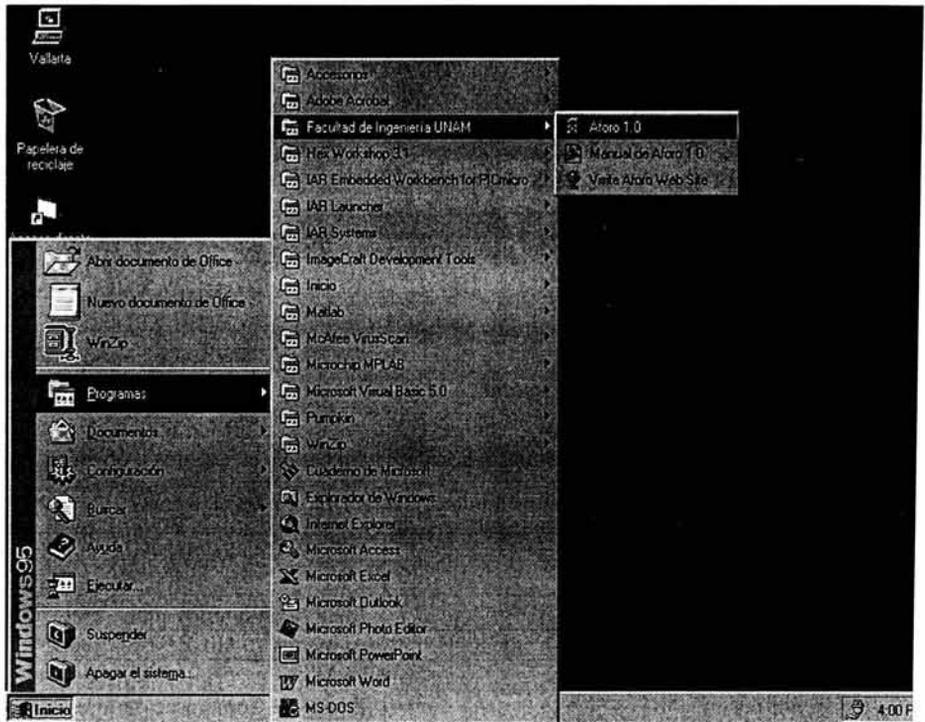


Ilustración 43. Programa en pc

Desinstalación del Programa, Haga clic en icono Inicio de la barra de tareas.

Seleccione Configuración, Panel de Control.

Haga doble clic en el icono Agregar o Quitar Programas.

Seleccione Sistema Aforo 1.0. Facultad de Ingeniería UNAM.

Haga clic en Agregar / quitar.

Siga las instrucciones en pantalla.

#### 4.2.2. Principal

En esta sección se presenta una referencia completa al sistema Aforo 1.0. Se presenta un Tutoría para la adquisición y reinicio del sistema de aforo vehicular.

Las siguientes ilustraciones muestran la introducción y los diferentes componentes de la ventana principal del sistema Aforo 1.0. respectivamente.

# Universidad Nacional Autónoma de México

## Facultad de Ingeniería



### Aforo 1.0.1 Sistema de Aforo Vehicular

Director de Tesis: Ing. Eduardo Ramírez

Tesistas:

Alejandro Castellanos

Pedro Moreno Muñoz

Miguel Angel Romero

Entrar



### Aforo 1.0.1 Sistema de Aforo Vehicular

**Comunicación**

Conectar al

**Abrir**

Abrir archivo de

**FI - UNAM**

**Salir**

Terminar el

Ilustración 44. Ambiente del programa aforo

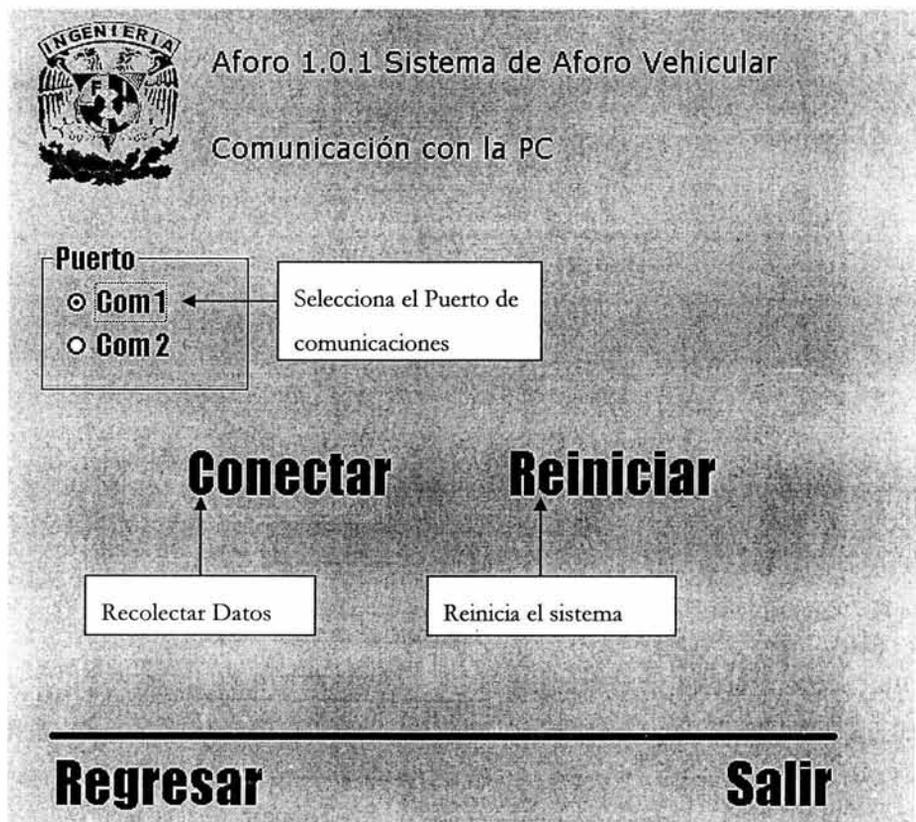
Estos tres elementos son explicados a continuación:

Elemento	Descripción.
Comunicación	Abre la ventana de comunicación. Es utilizado para recolectar datos del sistema de aforo vehicular y/o reiniciarlo.
Abrir	Abre la venta de archivos. Es utilizado para abrir un archivo con datos recolectado.
Salir	Termina el programa.

Tabla 37. Descripción de uso

#### 4.2.2.1. Comunicación.

La siguiente ilustración muestra la venta de comunicación del sistema Aforo 1.0.



Estos 5 elementos se explican a continuación.

Elemento	Descripción.
Puerto	Selecciona el Puerto de Comunicaciones en el que este conectado el sistema de aforo vehicular. El puerto Com 1 es seleccionado en forma predeterminada.
Conectar	Inicia la comunicación con el Sistema y recolecta los datos.
Reiniciar	Se comunica con el Sistema de Aforo Vehicular y le ordena reiniciarse.
Regresar	Regresa a la ventana Principal.
Salir	Termina el Programa.

Tabla 38. Comunicación con el sistema

#### 4.2.2.2. Abrir.

La siguiente ilustración muestra la venta Abrir del sistema Aforo 1.0. (Figura 4.6)



Ilustración 45. Resumen

Estos 5 elementos se explican a continuación.

Elemento	Descripción.
Ventana de Texto	Presenta el archivo seleccionado. En esta ventana se exhibe un resumen del archivo seleccionado.
Imprimir	Imprime el resumen del archivo seleccionado.
Gráficas	Accesa a la ventana de Gráficas.
Regresar	Regresa a la ventana Principal.
Salir	Termina el Programa.

**Tabla 39. Resultados**

#### **4.2.2.3. Gráficas.**

La siguiente ilustración muestra la ventana Gráficas del sistema Aforo 1.0. ( Figura )

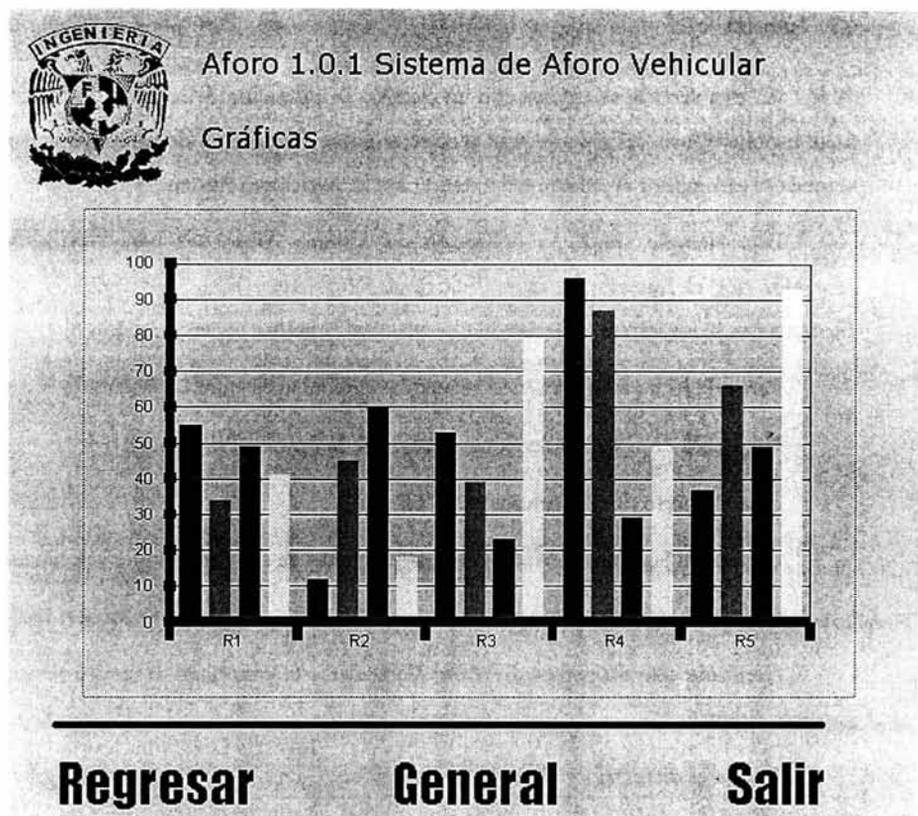


Ilustración 46. Graficas

Estos 3 elementos se presentan a continuación.

Elemento	Descripción.
General	Genera el gráfico correspondiente al archivo seleccionado anteriormente.
Regresar	Regresa a la venta Abrir.
Salir	Termina el Programa.

Tabla 40. Descripción de graficar

### 4.3. Tutorial.

En esta sección se explica con un ejemplo la utilización del sistema aforo. La primera parte expone el uso del sistema para recolectar datos del Sistema de aforo vehicular, y en la segunda se ejemplifica el reinicio del sistema para adquisiciones futuras.

Este ejemplo simula la utilización del sistema Aforo 1.0 para analizar los datos recogidos por el Sistema de aforo Vehicular. Estos datos serán guardados en el archivo Ejemplo.afo, se imprimirá un resumen y se analizará la gráfica correspondiente.

Para recolectar la información requerida se deberán ejecutar las siguientes acciones.

Haga click en el comando Comunicación de la ventana Principal.

Elija el puerto de comunicaciones a usar.

Conecte el Sistema de Aforo Vehicular al puerto de comunicaciones seleccionado.

Haga click en Conectar.

Confirme que el Sistema de Aforo Vehicular esta conectado al puerto seleccionado y presione Aceptar.

Escoja el directorio y el nombre del archivo (Ejemplo.afo) en donde se guardarán los datos.

Haga click en Regresar.

Para visualizar los datos recolectados ejecute los siguientes pasos:

Presione el comando Abrir de la ventana Principal.

Escoja el archivo en donde se guardaron los datos recolectados anteriormente (Ejemplo.afo).

Si quiere imprimir el resumen presione el comando Imprimir.

Para ver la gráfica relacionada:

Haga clic en el comando Gráficas de la ventana Abrir.

Presione el comando General.

Presione el comando Salir en cualquier ventana para terminar el programa.

Para dejar listo el Sistema de Aforo Vehicular para posteriores recolecciones siga los siguientes pasos

Abra el sistema Aforo 1.0.

Presione el comando Comunicación de la ventana Principal.

Presione el comando reiniciar en la ventana Comunicaciones.

Verifique su intención de reiniciar el sistema haciendo click en Aceptar.

## CAPITULO 5

### PRUEBAS Y RESULTADOS

## 5. PRUEBAS Y RESULTADOS.

Para realizar las pruebas se utilizó la tarjeta de adquisición de datos mencionada en el capítulo tres. Las pruebas finales del sistema constan en pruebas de depuración y pruebas terminales, en ambas se fabrica una tabla de pruebas. Estos datos son introducidos en la tarjeta de adquisición donde a escala se pretende simular el paso de los ejes de cada vehículo. La lista está integrada por diversos tipos de vehículos tratando de abarcar el total de posibles combinaciones y cubrir cada una de las categorías de la clasificación según CAPUFE. Así mismo se encuentra la comparación entre el número de botones reales que un neumático oprimiría y la escala que la tarjeta registra. Recordemos que el neumático más ancho es de 422mm, siendo este nuestra frontera para el eje Ch, y mayor que 422mm que son de doble rodada es nuestro eje G. Sincronizando el tiempo con el inicio de la prueba se comenzarán a simular los eventos uno a uno tomando nota del tiempo y tipo de vehículo que se obtenga como respuesta.

### 5.1. Pruebas de depuración.

Para probar el correcto funcionamiento del sistema de aforo vehicular se empleó la tabla de pruebas que contiene los posibles casos que se pueden presentar durante el funcionamiento del medidor de aforo vehicular, considerando el ancho de sus neumáticos según catálogos de fabricantes y deduciendo por escritorio el número de botones que cada eje oprimirá.

Tipo	Ancho llantas	Rodada	Botones reales	Botones tarjeta	Tipo de eje	Prueba Tiempo	Clas. Real	Resultado tiempo	clasificación
Automóvil	18.9	1	4	1	Ch				
	18.9	1	4	1	Ch	12	A	12	A
Autobús	29.3	1	6	1	Ch				
	29.3	2	12	2	G				
	29.3	2	12	2	G	26	C3	26	C
Autobús	32	1	7	1	Ch				
	32	2	14	2	G	35	B	35	B
Automóvil	19.4	1	4	1	Ch				
	19.4	1	4	1	Ch	42	A	42	A
R Ligero	17	1	4	1	Ch	48	R1	48	R
Camión	29.8	1	6	1	Ch				
	29.8	2	12	2	G				

	29.8	2	12	2	G				
	29.8	2	12	2	G				
	29.8	2	12	2	G				
	29.8	2	12	2	G				
	29.8	2	12	2	G				
	29.8	2	12	2	G	01:14	C8	74	H
Camión	27.9	1	6	1	Ch				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G	01:39	C5	99	E
Automóvil	22.9	1	5	1	Ch				
	22.9	1	5	1	Ch	01:51	A	111	A
R Ligero	20.1	1	5	1	Ch	01:58	R1		
Automóvil	24	1	5	1	Ch			118	A
	24	1	5	1	Ch	02:10	A	130	R
Camión	27.9	1	6	1	Ch				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G	02:43	C5	163	E
R Ligero	20.1	1	5	1	Ch	02:59	R1		
Camioneta	18.9	1	4	1	Ch			179	A
	18.9	1	4	1	Ch	03:10	A		
R Ligero	17	1	4	1	Ch	03:18	R1	198	A
R Ligero	17	1	4	1	Ch	03:24	R1		
Automóvil	19.4	1	4	1	Ch			204	A
	19.4	1	4	1	Ch	03:35	A	215	R
Camión	24	1	5	1	Ch				
	24	2	10	2	G	03:46	B	226	B
R Ligero	17.7	1	4	1	Ch	03:52	R1	232	R
Camión	27.9	1	6	1	Ch				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G				
	27.9	2	12	2	G		C9		
R Pesado	27.9	2	12	2	G		R2		
R Pesado	27.9	2	12	2	G	04:26	R2	266	K
Automóvil	22	1	5	1	Ch				
	22	1	5	1	Ch	04:39	A	279	A
R Ligero	18.8	1	4	1	Ch	04:47	R1		
Automóvil	18.9	1	4	1	Ch			287	A

	18.9	1	4	1	Ch	04:57	A		
Camioneta	22	1	5	1	Ch			297	A
	22	1	5	1	Ch	05:06	A	306	r

Tabla 41. Resultados de depuración

## 5.2. Análisis de resultados.

Los resultados obtenidos concuerdan perfectamente con los suministrados de la lista fabricada previamente. El tiempo registrado en cada opresión empata con satisfactoria precisión al tiempo en que sucedió el evento permitiendo así que cada vehículo quede registrado con fecha y hora al momento de su paso. Con respecto a la clasificación de cada tipo de vehículo resulto exacta conforme lo esperado y lo introducido a través de la tarjeta de adquisición.

Sobre la existencia de remolques ligeros se presentaran diversos casos dependiendo del numero de ejes que este remolque contenga. Si es el caso de que un remolque ligero presenta dos ejes o algún otro numero par entonces no serán almacenados como tales, es decir, dos ejes de un remolque equivale a un vehículo tipo A y así es como queda registrado; Ahora veremos mas detalladamente lo anterior.

El primer vehículo que pasa es un automóvil con dos ejes ch, y pasa a los 12 segundos que se inicio el sistema de aforo, como resultado nos da un vehículo tipo A y 12 segundos de ocurrido el evento, esto es correcto, tanto en tiempo como en clasificación. En los siguientes casos sucede lo mismo, es correcta la clasificación y tiempo.

Cuando llegamos a un automóvil que pasa a los 111 seg. (1:51), que lleva un remolque ligero (de rodada sencilla) a los 118 seg.(1:58), y después pasa otro automóvil a los 130 seg. (2:10). En los resultados vemos que el remolque se lo adjudica al segundo vehículo.

El caso en el que pasa un camión con cinco ejes (163 seg./2:43 min.) que trae un remolque sencillo (179seg./2:59 min.), siguiéndolo un automóvil (190seg./3:10 min.) con dos remolques sencillos (198 seg./3:18min. y 204seg./3:24min. respectivamente) y un automóvil sin remolque (215seg./3:35min.). nos da como resultado el paso de un camión de cinco ejes, tres automóviles consecutivos y un remolque sencillo. Esto es debido a que considera a el remolque sencillo del camión como el eje delantero del primer automóvil que además trae dos

remolques con ejes sencillos, que se van agrupando en pares, dándonos como resultado siete ejes de rodada sencilla.

El caso en que pasa un camión de dos ejes (226 seg./3:46min.) que trae un remolque sencillo (232seg./3:56min.) y después pasa un camión con nueve ejes y dos remolques grandes (226seg./4:26min.). Nos da como resultado un vehículo tipo B (1 eje Ch, y un eje G), y un remolque sencillo, que es en realidad lo que paso, ya que los remolques se acumulan hasta que pase un camión (eje Ch, y eje G), y como en este caso se da, por eso se muestra. El camión de nueve ejes al traer dos remolques grandes, nos da como resultado un camión con once ejes, ya que los remolques pesados se distinguirán a partir del décimo eje (grande).

Estos son los casos mas sobresalientes de la prueba de depuración, que nos sirvieron para corregir y mejorar el programa del sistema de aforo vehicular.

### 5.3. Pruebas terminales.

Se decidió realizar estas pruebas con el paso de cincuenta vehículos simulando una autopista federal, para facilitar estas pruebas se elaboró un programa que simula los resultados del contador de aforo vehicular, con el fin de tener una referencia para los resultados del mismo. Este simulador se hizo en Excel, ya que cuenta con las herramientas suficientes para construirlo.

### 5.4. Simulador del contador de aforo vehicular.

Las variables a ingresar en este simulador son: ancho de los neumáticos, número de ejes, tipo de rodada trasera (doble o sencilla), número de remolque y tipo de remolque (doble o sencillo). Nos va a arrojar los siguientes resultados: botones a oprimir en el modelo real, clasificación según capufe (A,B,C3,etc) y numero y tipo de remolque. La variable tiempo se introduce manualmente, pero hasta que se realizan las pruebas en el modelo a escala.

Las pruebas se programaron de la siguiente manera, para obtener los botones a oprimir en el sistema real delanteros(botones R/D) se dividió la parte entera de el ancho de los neumáticos mas uno(=ENTERO((B3/5)+1)), para obtener los botones a oprimir en el sistema real traseros(botones R/T), se multiplico el número de rodada por botones R/D. Para obtener la clasificación se utilizo un programa en visual Basic para aplicaciones.

La tabla de referencia es la siguiente:

Se probaran estos vehículos en el contador de aforo vehicular, el objetivo de la tabla siguiente es para comparar los resultados que se introducen, con los que son arrojados en el sistema aforador.

Lista de vehículos para prueba:

Tipo	Ancho Llantas	Ejes	Rodada	Botones R/D	Botones R/T	Tiempo	Clasificación	Num. Rem	Tipo de Rem.	Remolques
Automóvil	20	2	1	5	5	0:10	A	0	0	0
Autobús	32	3	2	7	14	0:17	C3	0	0	0
Autobús	30	2	2	7	14	0:22	B	0	0	0
Automóvil	19.5	2	1	4	4	0:31	A	0	0	0
Camión	34	9	2	7	14	0:41	C9	0	0	0
Camión	32	5	2	7	14	1:10	C5	0	0	0
Automóvil	18	2	1	4	4	1:35	A	0	0	0
Camión	34	9	2	7	14	1:47	C9	0	0	0
Automóvil	21	2	1	5	5	2:05	A	0	0	0
Automóvil	26	2	1	6	6	2:22	A	0	0	0
Automóvil	29	2	1	6	6	2:39	A	0	0	0
Camión	32	6	2	7	14	2:57	C6	0	0	0
Autobús	35	2	2	8	16	3:30	B	0	0	0
Automóvil	19	2	1	4	4	3:47	A	0	0	0
Automóvil	20	2	1	5	5	4:29	A	2	1	2R1
Autobús	30	3	2	7	14	4:52	C3	0	0	0
Camión	38	5	2	8	16	5:12	C5	0	0	0
Autobús	33	3	2	7	14	5:36	C3	0	0	0
Automóvil	27	2	1	6	6	5:51	A	1	1	1R1
Automóvil	25	2	1	6	6	6:21	A	0	0	0
Camión	34	8	2	7	14	6:57	C8	0	0	0
Camión	36	9	2	8	16	8:12	C9	2	2	2R2
Camión	34	9	2	7	14	8:52	C9	2	2	2R2
Camión	32	2	2	7	14	9:30	B	0	0	0
Automóvil	21	2	1	5	5	9:56	A	0	0	0
Camión	28	2	2	6	12	10:11	B	0	0	0
Camión	32	7	2	7	14	10:52	C7	0	0	0
Camión	30	4	2	7	14	12:20	C4	0	0	0
Automóvil	26	2	1	6	6	12:42	A	0	0	0
Automóvil	24	2	1	5	5	13:21	A	0	0	0
Camión	34	6	2	7	14	13:45	C6	0	0	0
Autobús	35	3	2	8	16	15:10	C3	0	0	0
Camión	34	3	2	7	14	15:42	C3	1	1	1R1
Camión	35	8	2	8	16	16:15	C8	0	0	0
Automóvil	23	2	1	5	5	17:21	A	1	1	1R1
Automóvil	20	2	1	5	5	18:10	A	0	0	0

Camión	30	5	2	7	14	19:21	C5	0	0	0
Autobús	33	3	2	7	14	19:47	C3	0	0	0
Automóvil	27	2	1	6	6	20:12	A	0	0	0
Automóvil	25	2	1	6	6	0:25:13	A	0	0	0
Camión	34	8	2	7	14	00:25:51	C8	0	0	0
Camión	36	9	2	8	16	00:27:10	C9	0	0	0
Camión	34	9	2	7	14	00:28:21	C9	0	0	0
Camión	32	2	2	7	14	00:30:05	B	0	0	0
Automóvil	21	2	1	5	5	00:32:40	A	1	1	1R1
Camión	28	2	2	6	12	00:35:10	B	0	0	0
Camión	32	7	2	7	14	00:36:09	C7	0	0	0
Camión	30	4	2	7	14	00:36:40	C4	0	0	0
Automóvil	26	2	1	6	6	00:37:20	A	0	0	0
Automóvil	24	2	1	5	5	00:39:50	A	0	0	0

Tabla 42. Resultados de prueba

Conteo	
A =	20
B =	6
C3 =	6
C4 =	2
C5 =	3
C6 =	2
C7 =	2
C8 =	3
C9 =	6
R1 =	5
R2 =	2
Total:	50

Tabla 43. Clasificación de prueba

### 5.5. Resultados arrojados por el contador de aforo vehicular.

Los datos anteriores son introducidos al sistema de aforo vehicular, y los resultados son los siguientes:

Versión: Aforo 1.0

Fecha de Inicio de Sistema: 09/11/2003 01:05:05 p.m.  
 Fecha de Recolección del Datos: 09/11/2003 01:45:00 p.m.  
 Número de Datos Recolectados: 55  
 Tipos A: 21  
 Tipos B: 6  
 Tipos C3: 6  
 Tipos C4: 2  
 Tipos C5: 3  
 Tipos C6: 2  
 Tipos C7: 2  
 Tipos C8: 3  
 Tipos C9: 6  
 Tipos Remolque Ligero: 4  
 Tipos Remolque Pesado: 2.

Tabla 44. Resultados arrojados por el aforador




**Aforo 1.0.1 Sistema de Aforo Vehicular**  
**Resultados**

Versión: Aforo 1.0  
 Fecha de Inicio de Sistema: 09/11/2003 01:05:05 p.m.  
 Fecha de Recolección del Datos: 09/11/2003 01:45:00 p.m.  
 Número de Datos Recolectados: 55  
 Tipos A: 21  
 Tipos B: 6  
 Tipos C3: 6  
 Tipos C4: 2  
 Tipos C5: 3  
 Tipos C6: 2  
 Tipos C7: 2  
 Tipos C8: 3  
 Tipos C9: 6

**Regresar Imprimir Gráficas Salir**  
**Detalles**

Ilustración 47. Resultados

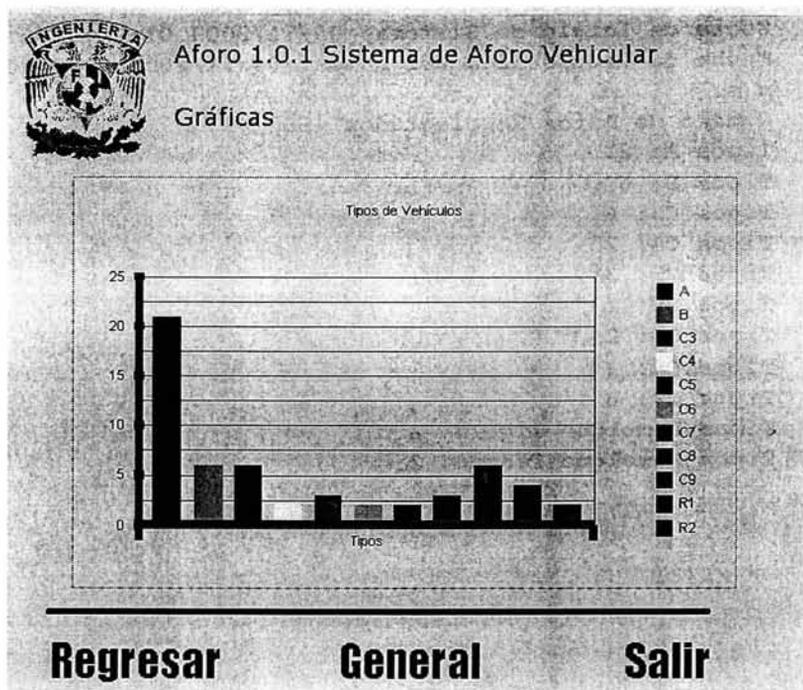


Tabla 45. Graficas de resultados.

Resultados arrojados por el contador de aforo

---

A=10	G=652
C=17	D=740
B=22	A=762
A=31	A=801
I=45	F=852
E=70	C=910
A=95	C=933
I=107	r=941
A=125	H=975
A=142	A=1041
A=159	A=1010
F=177	r=1025
B=210	E=1161
A=227	C=1187
A=242	A=1212
A=269	A=1513
C=292	H=1551
E=312	I=1630
C=336	I=1701
A=351	B=1805
A=362	A=1930
r=381	r=1960
H=417	B=2110
K=492	G=2169
K=532	D=2200
B=570	A=2240
A=596	A=2390 .
B=611	

#### Ilustración 48. Detalle de resultados

La tabla anterior nos muestra el tipo de vehículo, y el tiempo en que paso en segundos.

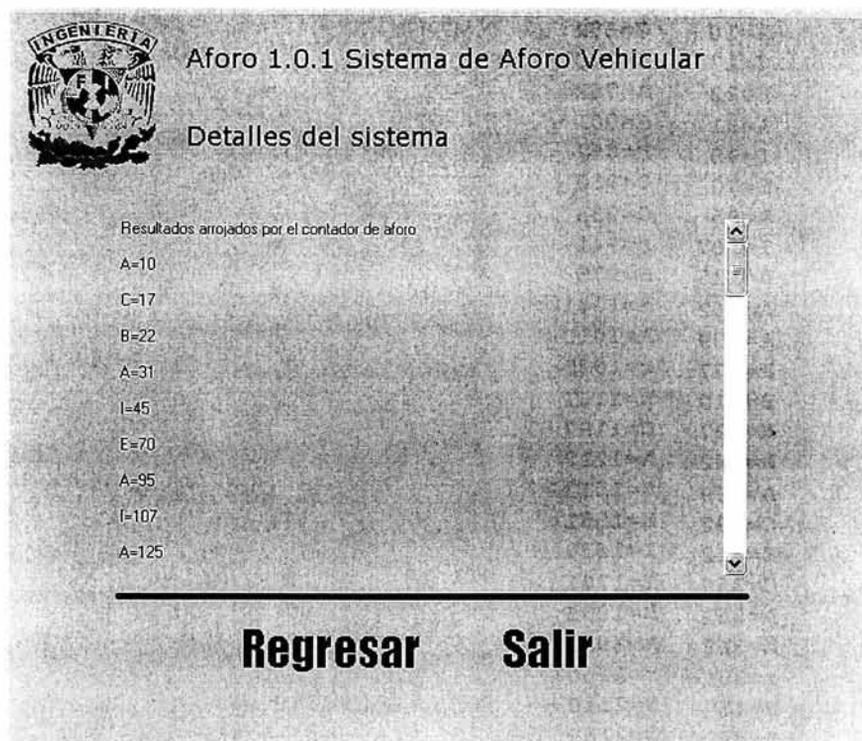


Tabla 46. Detalle

Como vemos, los resultados son los esperados, como ya lo habíamos comentado en las pruebas de depuración, en este caso tiene un comportamiento idéntico, también vemos que las graficas tienen la misma pauta, así concluimos en estas pruebas el correcto funcionamiento del contador de aforo vehicular.

---

## CONCLUSIONES.

El diseño del sistema de aforo vehicular ha sido un éxito debido a su respuesta en pruebas de escritorio y en pruebas terminales, tanto en clasificación, hora, fecha, y almacenamiento de datos.

El diseño, desarrollo y depuración del programa que requiere el Sistema de Aforo Vehicular para funcionar se facilitó mucho gracias a la utilización del lenguaje C. El uso de las rutinas programadas y puestas a disposición en las librerías estándares del lenguaje C, simplificaron considerablemente el diseño de los módulos que conforman el sistema. Esto confirma nuestras expectativas de eficacia, factibilidad y confianza en el código fuente del Sistema de Aforo Vehicular. Además, la utilización de este lenguaje de programación añadió portabilidad a nuestro algoritmo; ya que sin modificaciones significativas al código se podría utilizar en alguna otra plataforma.

Resultó también satisfactoria la decisión de practicar el estilo de programación “foreground / background” como alternativa a la utilización de la técnica convencional “super - loop”; Ya que con el uso de rutinas ejecutadas bajo instrucción de alguna de las interrupciones empleadas, logramos un mayor control sobre que acciones emprender para determinado evento. Logramos también obtener una mayor ventaja en la creación y desempeño de rutinas de tiempo.

El gran reto entonces, en lo que al diseño del programa de captura se refiere, resultó ser, el de aprender, practicar, desarrollar y realizar un código en el lenguaje de programación C capaz de utilizar las ventajas proporcionadas por el manejo de interrupciones, sin tener la necesidad de utilizar un sistema operativo en tiempo real; logrando así la optimización en el uso del microcontrolador empleado, utilizando solamente un compilador. Al conseguir este objetivo originamos a su vez, documentación relacionada que servirá a futuras aplicaciones que requieran optimizar el uso de un microcontrolador embebido, con herramientas de bajo costo.

La implementación fue realizada en un sistema a escala, debido a que no se contó con la infraestructura necesaria para la implementación del sistema real, las pruebas terminales se llevaron a cabo en este sistema, simulando el paso de algunos vehículos, tomando como patrón algunas autopistas de México, obteniendo resultados similares a los esperados.

La implementación de la aplicación en Visual Basic nos da un ambiente amigable, nos facilita la interpretación del sistema, y nos muestra estadísticas confiables. Esta aplicación es una parte importante de la implementación del proyecto, ya que es la interfase del sistema embebido y el usuario.

Consideramos que este sistema aporta una solución al problema en cuestión de medición de aforo vehicular, que es el principal objetivo de este proyecto, ya que muestra información confiable, del aforo en autopistas que no son de cuota, para la adecuada inversión en la red carretera nacional, y por ende su modernización.

---

## APÉNDICE A

### **Infraestructura Carretera de México.**

#### **Sistema Nacional de Autopistas.**

El Sistema Nacional de Autopistas (SNA) cuenta con 71 autopistas y 41 puentes, los cuales integran una red de 5,910 km de longitud, que es la quinta más grande del mundo. La mayoría de estas autopistas y puentes se ubican dentro de 14 grandes ejes carreteros que dan servicio a los principales flujos de tránsito entre las diversas regiones del país. En general, las autopistas cuentan con mejores características que el resto de las carreteras federales: carriles más anchos, accesos controlados, mejor señalamiento, pendientes y curvas más suaves, sin entronques a nivel, etc.

El gobierno federal está a cargo de la operación, explotación y mantenimiento del 77% de dicha red: Una parte es administrada directamente por Caminos y Puentes Federales (11 autopistas y 30 puentes, con una longitud de 805 km) y otra parte está concesionada al Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas (35 autopistas y 4 puentes, con una longitud de 3,770 km). El 23% restante de la longitud del SNA está a cargo de concesionarios y está repartido entre empresas privadas (13 autopistas y un puente, con una longitud total de 768 km), gobiernos estatales (9 autopistas y 6 puentes, con 329 km) y organismos financieros (3 autopistas con longitud de 237 km).

Si bien es cierto que durante el inicio del programa de concesiones de autopistas y puentes se presentaron problemas, debidos a diversos factores, a partir de 1997 el SNA tuvo un repunte que lo ha llevado a participar cada vez en mayor medida en la atención de los flujos del transporte terrestre en todo el país. Basta citar que entre las 22 obras que el gobierno rescató en 1997, el aforo ha crecido a un ritmo de 13% anual hasta 2002, lo que representa un crecimiento bruto de 55%. En la totalidad del Sistema, durante el período 1997-2002 el aforo ha crecido al 8% anual. Además, se han incorporado 12 nuevas autopistas y 4 puentes. En total, hoy circulan por el SNA en promedio casi 765,800 vehículos diariamente, contra 526,970 que circularon durante 1997.

El crecimiento observado se ha debido a diversos factores, entre los que destacan la creciente aceptación de las autopistas por parte de los usuarios, la estabilidad de la economía y las medidas de reducción y compactación tarifaria que el gobierno federal ha implantado. Quitando la inflación, las tarifas para automóviles, autobuses y camiones se han reducido en el período citado en un 11%, 8% y 19%, respectivamente. Este año, se estima que los ingresos totales de las autopistas y puentes a cargo del gobierno federal serán de alrededor de 15,000 millones de pesos y que los correspondientes al resto del Sistema rebasarán los 4,000 mdp.

En conclusión, en la actualidad se avanza con el propósito de ofrecer una red troncal de autopistas estructurada, con capacidad y cobertura suficientes para hacerla accesible a todos los mexicanos y para multiplicar las oportunidades de desarrollo de la población.

Corredores Troncales de la Red Carretera.



## 14 CORREDORES TRONCALES

México - Guadalajara - Tepic - Mazatlán- Guaymas - Hermosillo - Nogales, con ramal a Tijuana.	—————
México - Querétaro - San Luis Potosí- Saltillo - Monterrey - Nuevo Laredo, con ramales a Piedras Negras.	—————
Querétaro - Irapuato - León - Lagos de More- no - Aguascalientes - Zacatecas - Torreón- Chihuahua - Cd. Juárez.	—————
Acapulco - Cuernavaca - México -Tuxpam.	—————
Puebla - Progreso.	—————
Mazatlán - Durango - Torreón - Saltillo- Monterrey - Reynosa - Matamoros.	—————
Manzanillo - Guadalajara - Lagos de Moreno- San Luis Potosí - Tampico con ramal a L. Cárdenas.	—————
Acapulco - Cuernavaca - Puebla - Veracruz	—————
Veracruz - Monterrey con ramal a Matamoros.	—————
Transpeninsular de Baja California	—————
Altiplano.	—————
Puebla - Cd. Hidalgo	—————
Circuito Transísmico	—————
Circuito Turístico de la Península	—————

### Longitud y características de la red de carreteras por entidad federativa (kilómetros)

Entidad	Brechas mejoradas	Terracería	Revestidas	Pavimentadas			Total
				Dos carriles	Cuatro o más carriles	Total	
Aguascalientes	327	-	823	1 020	95	1 115	2 265
Baja California	4 283	415	4 167	2 444	400	2 844	11 709
Baja California Sur	1 056	726	1 820	1 578	49	1 627	5 229
Campeche	608	337	462	3 305	57	3 362	4 769
Coahuila	-	-	4 507	3 221	639	3 860	8 367
Colima	141	-	1 055	760	151	911	2 107
Chiapas	-	1 055	15 864	4 639	72	4 711	21 630
Chihuahua	-	1 063	6 417	4 473	798	5 271	12 751
Distrito Federal	-	-	-	79	71	150	150
Durango	1 618	-	7 581	3 536	386	3 922	13 121
Guanajuato	2 006	-	5 027	3 934	489	4 423	11 456
Guerrero	3 196	-	8 314	4 087	277	4 364	15 874
Hidalgo	171	180	5 758	2 916	222	3 138	9 247
Jalisco	14 156	185	5 297	4 920	646	5 566	25 204
México	-	-	4 116	5 384	501	5 885	10 001
Michoacán	4 000	-	3 970	4 972	406	5 378	13 348
Morelos	-	-	477	1 350	200	1 550	2 027
Nayarit	2 291	-	1 830	1 329	152	1 481	5 602
Nuevo León	-	-	2 902	3 568	747	4 315	7 217
Oaxaca	3 620	-	8 252	4 255	45	4 300	16 172
Puebla	154	-	4 500	3 867	216	4 083	8 737
Querétaro	-	-	1 817	1 186	201	1 387	3 204
Quintana Roo	-	-	3 092	1 785	233	2 018	5 110
San Luis Potosí	-	86	7 091	3 524	306	3 830	11 007
Sinaloa	5 560	1 677	5 800	2 736	618	3 354	16 391
Sonora	13 476	-	4 412	5 186	753	5 939	23 827
Tabasco	-	550	4 097	3 855	118	3 973	8 620
Tamaulipas	-	419	8 945	3 787	306	4 093	13 457
Tlaxcala	-	-	1 238	1 180	75	1 255	2 493
Veracruz	7 164	-	9 572	5 305	508	5 813	22 549
Yucatán	3 669	-	2 623	5 699	277	5 976	12 268
Zacatecas	1 268	-	6 760	3 105	126	3 231	11 259
<b>Total</b>	<b>68 764</b>	<b>6 693</b>	<b>148 586</b>	<b>102 985</b>	<b>10 140</b>	<b>113 125</b>	<b>337 168</b>

Fuente: Subsecretaría de Infraestructura.

### Longitud y características de la red de carreteras troncales de cuota por entidad federativa (kilómetros)

Entidad	Pavimentadas		Total <sup>1/</sup>
	Dos carriles	Cuatro o más carriles	
Aguascalientes	-	-	-
Baja California	-	227	227
Baja California Sur	20	-	-
Campeche	-	40	40
Coahuila	-	221	221
Colima	37	47	84
Chiapas	204	-	204
Chihuahua	167	397	564
Distrito Federal	-	49	49
Durango	-	331	331
Guanajuato	-	139	139
Guerrero	70	222	292
Hidalgo	41	36	77
Jalisco	74	451	525
México	62	299	361
Michoacán	276	232	508
Morelos	44	118	162
Nayarit	36	109	145
Nuevo León	7	357	364
Oaxaca	136	23	159
Puebla	106	179	285
Querétaro	-	109	109
Quintana Roo	15	73	88
San Luis Potosí	179	34	213
Sinaloa	34	378	412
Sonora	-	481	481
Tabasco	-	54	54
Tamaulipas	15	44	59
Tlaxcala	-	25	25
Veracruz	125	444	569
Yucatán	-	154	154
Zacatecas	37	29	66
<b>Total</b>	<b>1 685</b>	<b>5 302</b>	<b>6 987</b>

<sup>1/</sup> Incluye carreteras propias de CAPUFE, contratadas, administradas por CAPUFE, FARAC, a cargo de instituciones financieras concesionadas a la iniciativa privada, concesionadas a los gobiernos de los estados y estatales de cuota.

Fuente: Subsecretaría de Infraestructura.

**Longitud y características de la red de carreteras troncales libres por entidad federativa (kilómetros)**

Entidad	Pavimentadas		Total
	Dos camiles	Cuatro o más camiles	
Aguascalientes	280	95	375
Baja California	1 359	116	1 475
Baja California Sur	1 151	49	1 200
Campeche	1 242	17	1 259
Coahuila	1 193	354	1 547
Colima	233	49	282
Chiapas	2 112	72	2 184
Chihuahua	1 811	355	2 166
Distrito Federal	79	22	101
Durango	1 850	55	1 905
Guanajuato	905	210	1 115
Guerrero	1 860	40	1 900
Hidalgo	708	133	841
Jalisco	1 912	106	2 018
México	667	113	780
Michoacán	2 151	141	2 292
Morelos	227	29	256
Nayarit	777	23	800
Nuevo León	771	373	1 144
Oaxaca	2 771	22	2 793
Puebla	1 097	37	1 134
Querétaro	417	35	452
Quintana Roo	798	65	863
San Luis Potosí	1 349	272	1 621
Sinaloa	607	141	748
Sonora	1 541	221	1 762
Tabasco	503	55	558
Tamaulipas	1 987	237	2 224
Tlaxcala	503	50	553
Veracruz	2 378	56	2 434
Yucatán	1 201	56	1 257
Zacatecas	1 425	73	1 498
<b>Total</b>	<b>37 865</b>	<b>3 672</b>	<b>41 537</b>

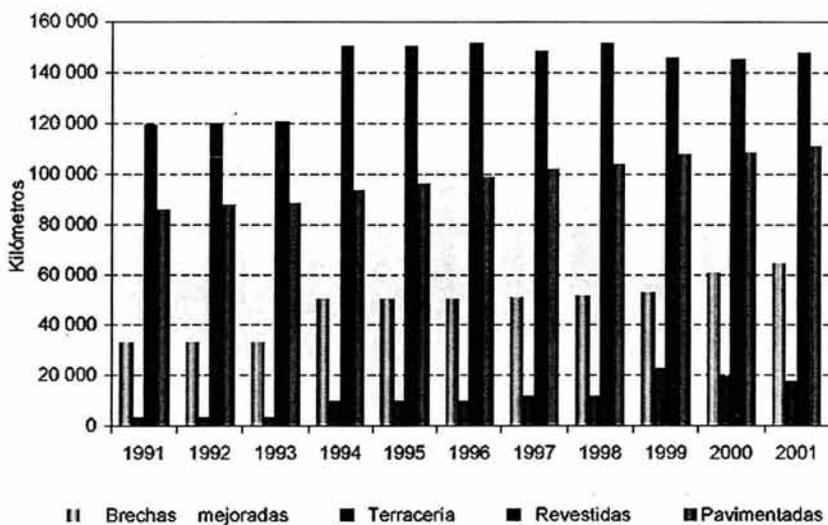
Fuente: Subsecretaría de Infraestructura.

## Longitud y características de la red de carreteras (kilómetros)

Año	Brechas mejoradas	Terracería	Revestidas	Pavimentadas			Total
				Dos carriles	Cuatro o más carriles	Total	
1992	33 120	3 058	120 245	79 826	7 607	87 433	243 856
1993	33 120	3 026	120 666	80 416	7 955	88 371	245 183
1994	50 536	9 751	150 437	85 605	8 263	93 868	304 592
1995	50 602	9 786	150 100	87 467	8 449	95 916	306 404
1996	50 432	9 778	151 664	89 805	8 912	98 717	310 591
1997	51 231	11 787	148 336	92 955	9 295	102 250	313 604
1998	52 416	11 812	151 541	94 589	9 434	104 023	319 792
1999	52 992	22 547	145 907	98 031	10 055	108 086	329 532
2000	60 557	19 588	145 279	98 241	10 247	108 488	333 912
2001	64 736	17 337	147 474	100 562	10 348	110 910	340 457
2002	68 764	6 693	148 586	102 985	10 140	113 125	337 168

Fuente: Dirección General de Evaluación y Subsecretaría de Infraestructura.

Longitud de la red de carreteras



## Longitud y características de la red troncal federal (kilómetros) <sup>1/</sup>

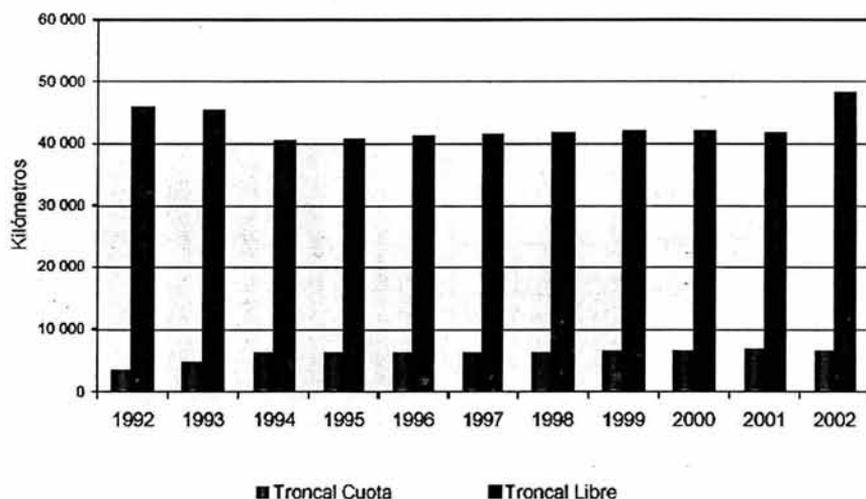
Año	Cuota <sup>2/</sup>			Libres					Total
	Pavimentados			Pavimentados		Revestidas	Terracería	Total	
	Dos cariles	Cuatro cariles	Total	Dos cariles	Cuatro Carriles				
1992	330	3 140	3 470	40 323	3 505	1 795	185	45 808	49 278
1993	573	4 095	4 668	40 227	3 079	1 827	153	45 286	49 954
1994	1 222	5 072	6 294	37 929	2 420	-	-	40 349	46 643
1995	1 226	5 082	6 308	38 117	2 534	-	-	40 651	46 959
1996	1 206	5 150	6 356	38 213	2 801	-	-	41 014	47 370
1997	1 183	5 211	6 394	38 457	2 954	-	-	41 411	47 805
1998	1 200	5 188	6 388	38 537	3 116	-	-	41 653	48 041
1999	1 116	5 313	6 429	38 413	3 352	-	-	41 765	48 194
2000	1 337	5 261	6 598	38 223	3 643	-	-	41 866	48 464
2001	1 485	5 274	6 759	37 938	3 707	-	-	41 645	48 404
2002	1 685	5 302	6 987	37 865	3 672	-	-	41 537	48 524

<sup>1/</sup> Las cifras de 1994 no son comparables con años anteriores debido al cambio de metodología en la captación de información a cargo del Sector Comunicaciones y Transportes. De 1989 a 1993 se consignaron cifras proporcionadas por las unidades administrativas de la SCT. A partir de 1994 se reportan datos de los 31 Centros SCT.

<sup>2/</sup> Incluye estatales de cuota.

Fuente: Subsecretaría de Infraestructura.

### Longitud de la red troncal federal



### Longitud de la red de autopistas operada por Caminos y Puentes Federales de Ingresos y Servicios Conexos

Autopista	Entidad federativa	Longitud (km)		Total
		4 y/o más carriles	Dos carriles	
<b>Red propia</b>				
Puebla-Orizaba	Puebla	104.6	-	104.6
	Veracruz	30.6	-	30.6
Tijuana-Ensenada	Baja California	89.5	-	89.5
Cd. Mendoza-Córdoba	Veracruz	30.7	-	30.7
Chapalilla-Compostela	Nayarit	-	35.5	35.5
Cosoleacaque-Nuevo Teapa	Veracruz	34.0	-	34.0
Rancho Viejo-Taxco	Guerrero	-	8.3	8.3
Tehuacán-Oaxaca	Oaxaca	27.0	132.0	159.0
	Puebla	-	84.0	84.0
Arriaga-Huixtla	Chiapas	-	204.5	204.5
La Rumorosa-Tecate	Baja California	55.5	-	55.5
Aeropuerto-Los Cabos-San José del Cabo <sup>1/</sup>	Baja California Sur	-	20.2	20.2
Ent. Cuauhtémoc-Ent. Osiris <sup>1/</sup>	Zacatecas	-	32.0	32.0
Tuxtla Gutiérrez- San Cristóbal de las Casas <sup>1/</sup>	Chiapas	-	21.0	21.0
<b>Total</b>		<b>371.9</b>	<b>537.5</b>	<b>909.4</b>
<b>Red contratada <sup>2/</sup></b>				
México-Tizayuca	Hidalgo	11.4	-	11.4
	México	34.4	-	34.4
Guadalajara-Colima	Colima	-	16.0	16.0
	Jalisco	69.0	61.0	130.0
Tepic-Ent. San Blas	Nayarit	20.5	-	20.5
Atzacomulco-Matavatio	México	-	22.4	22.4
	Michoacán	-	42.0	42.0
Culiacán-Las Brisas	Sinaloa	125.8	-	125.8
Asunción-Tejocotal <sup>3/</sup>	Hidalgo	-	38.5	38.5
<b>Total</b>		<b>261.1</b>	<b>179.9</b>	<b>441.0</b>

<sup>1/</sup> Inició operaciones durante 2002.

<sup>2/</sup> Autopistas concesionadas operadas por contrato.

<sup>3/</sup> Incluye el Libramiento de Tutancingo.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

Autopista	(Continuación)			
	Entidad federativa	Longitud (km)		Total
		4 y/o más carriles	Dos carriles	
<b>Red FARAC<sup>4</sup></b>				
México-Cuernavaca	Distrito Federal	28.3	-	28.3
	Morelos	33.2	-	33.2
Puente de Ixtla-Iguala	Guerrero	-	44.7	44.7
	Morelos	-	18.9	18.9
La Pera-Cuautla	Morelos	-	34.2	34.2
Zacapaico-Rancho Viejo	Guerrero	-	17.3	17.3
Campeche-Champotón	Campeche	39.5	-	39.5
Cuernavaca-Acapulco	Guerrero	201.6	-	201.6
	Morelos	61.0	-	61.0
Monterrey-Nuevo Laredo	Nuevo León	123.1	-	123.1
Lib. Pon. de Tampico	Tamaulipas	-	14.5	14.5
Cadereyta-Reynosa	Nuevo León	132.0	-	132.0
Zapotlanejo-Lagos de Moreno	Jalisco	118.5	-	118.5
León-Lagos de Moreno-Aguascalientes	Guanajuato	28.0	-	28.0
	Jalisco	75.9	-	75.9
Chamapa-Lechería	México	27.3	-	27.3
Lib. Noreste de Querétaro	Querétaro	37.5	-	37.5
Pátzcuaro-Uruapan	Michoacán	-	56.5	56.5
Maravatio-Zapotlanejo	Jalisco	78.0	-	78.0
	Michoacán	231.7	-	231.7
Guadalajara-Zapotlanejo	Jalisco	26.0	-	26.0
Guadalajara-Tepic	Jalisco	86.6	-	86.6
	Nayarit	82.0	-	82.0
Córdoba-Veracruz	Veracruz	98.0	-	98.0
La Tinaja-Cosoleacaque	Veracruz	228.0	-	228.0
Mazatlán-Culiacán	Sinaloa	181.5	-	181.5
Estación Don-Nogales	Sonora	468.5	-	468.5
Lib. Oriente de Saltillo	Coahuila	21.0	-	21.0
La Carbonera-Puerto México	Coahuila	32.0	-	32.0
Torreón-Satillo	Coahuila	115.0	-	115.0
Gómez Palacio-Corralitos	Chihuahua	42.0	-	42.0
	Durango	109.3	-	109.3
Reynosa-Matamoros	Tamaulipas	44.0	-	44.0
<b>Subtotal</b>		<b>2 749.5</b>	<b>188.1</b>	<b>2 935.6</b>

<sup>4</sup>FARAC Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Autopistas administradas por CAPUFE a partir de septiembre de 1998

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos

(Continuación)

Autopista	Entidad federativa	Longitud (km)		Total
		4 y/o más carriles	Dos carriles	
<b>Red FARAC<sup>4</sup></b>				
Lib. Escuinapa-Rosario	Sinaloa	37.0	-	37.0
Uruapan-Nueva Italia	Michoacán	-	59.3	59.3
Aguadulce-Cárdenas	Tabasco	53.3	-	53.3
Tihuatlán-Gutiérrez Zamora	Veracruz	26.1	-	26.1
Nueva Italia-Lázaro Cárdenas	Michoacán	-	22.0	22.0
México-Querétaro <sup>5</sup>	Hidalgo	24.4	-	24.4
	México	90.7	-	90.7
	Querétaro	60.4	-	60.4
Querétaro- Celaya <sup>5</sup>	Guanajuato	36.4	-	36.4
	Querétaro	8.1	-	8.1
Celaya-Irapuato <sup>5</sup>	Guanajuato	59.3	-	59.3
México- Puebla <sup>5</sup>	Distrito Federal	0.2	-	0.2
	México	46.6	-	46.6
	Puebla	64.1	-	64.1
Las Choapas-Ocozocoautla <sup>1</sup>	Veracruz	-	17.0	17.0
<b>Total</b>		<b>3 256.1</b>	<b>284.4</b>	<b>3 540.5</b>

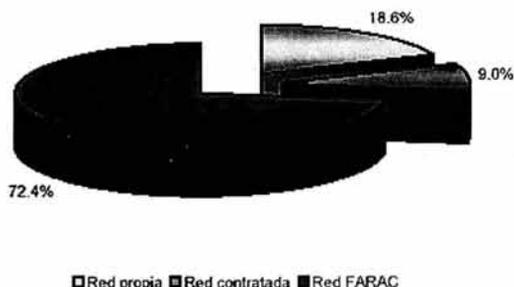
<sup>4</sup> FARAC: Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Autopistas administradas por CAPUFE a partir de septiembre de 1996.

<sup>5</sup> A partir del mes de octubre del 2002 las autopistas México-Querétaro y México-Puebla pasaron de la Red Propia a la red Farac y en diciembre del mismo año la autopista Querétaro-Irapuato.

<sup>1</sup> Inicie operaciones durante 2002.

Fuente: Caminos y Puertos Federales de Ingresos y Servicios Conexos.

Distribución porcentual de la longitud de la red carretera de CAPUFE



### Longitud de puentes operados por Caminos y Puentes Federales de Ingresos y Servicios Conexos por entidad federativa

Entidad	Puentes nacionales	Longitud (metros)	Puentes internacionales <sup>1/</sup>	Longitud (metros)
<b>Red propia</b>				
Coahuila	-	-	Acuña	129
	-	-	Piedras Negras	113
Chiapas	-	-	Suchiate II	3 694
	-	-	Dr. Rodolfo Robles	189
Chihuahua	-	-	Paso del Norte <sup>2/</sup>	216
	-	-	Ojinaga	121
Michoacán	La Piedad	92	-	-
Nuevo León	Cadereyta	179	-	-
Oaxaca	Caracol	164	-	-
	Papaloapan	288	-	-
Sinaloa	Culiacán	433	-	-
	Sinaloa	327	-	-
Tabasco	Grijalva	254	-	-
	Usumacinta	347	-	-
Tamaulipas	San Juan	175	Camargo	116
	Tampico	1 543	Laredo	108
	-	-	Juárez-Lincoln <sup>3/</sup>	159
	-	-	Las Flores	83
	-	-	Matamoros	155
	-	-	Miguel Alemán	155
Veracruz	-	-	Reynosa	112
	Alvarado	530	-	-
	Coatzacoalcos	985	-	-
	Ing. Antonio Dovalí	1 268	-	-
	Nautla	214	-	-
	Pánuco	179	-	-
	Tecolutla	368	-	-
Tlacotalpan	597	-	-	
<b>Total</b>	<b>17 puentes</b>	<b>7 943</b>	<b>13 puentes</b>	<b>5 350</b>

1/ La longitud de los puentes internacionales corresponde a la parte mexicana administrada por CAPUFE.

2/ El puente Paso del Norte se le conoce también como: Benito Juárez y/o Santa Fe.

3/ El puente Juárez-Lincoln se le conoce también como: Laredo II.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

Entidad	Puentes nacionales	Longitud (metros)	(Continuación)	
			Puentes internacionales <sup>1/</sup>	Longitud (metros)
<b>Red contratada <sup>4/</sup></b>				
Coahuila	-	-	Piedras Negras II	102
Chihuahua	-	-	Zaragoza-Ysleta	155
Nuevo León	-	-	Solidaridad-Colombia	180
Sinaloa	San Miguel	70	-	-
Tamaulipas	-	-	Libre Comercio	140
Veracruz	Tuxpan	457	-	-
<b>Total</b>	<b>2 puentes</b>	<b>527</b>	<b>4 puentes</b>	<b>577</b>
<b>Red FARAC <sup>5/</sup></b>				
Campeche	Zacatal	3 861	-	-
Michoacán	Ignacio Chávez	4 033	-	-
Tamaulipas	-	-	Ignacio Zaragoza	810
	-	-	Nuevo Amanecer <sup>6/</sup>	2 629
<b>Total</b>	<b>2 puentes</b>	<b>7 894</b>	<b>2 puentes</b>	<b>3 439</b>

<sup>4/</sup> Puentes concesionados operados por contrato.

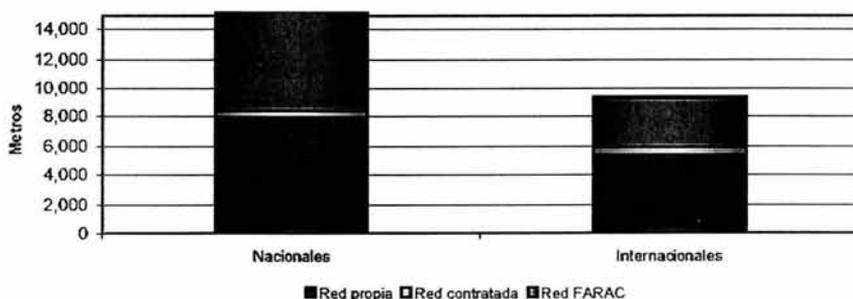
<sup>5/</sup> Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Puentes administrados por CAPUFE a partir de septiembre de 1998.

<sup>6/</sup> El puente Nuevo Amanecer se le conoce también como Reynosa Pharr.

<sup>1/</sup> La longitud de los puentes internacionales corresponde a la parte mexicana administrada por CAPUFE.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

Longitud de puentes administrados por CAPUFE



**Tránsito de vehículos en las autopistas de Caminos  
y Puentes Federales de Ingresos y Servicios Conexos  
(miles de vehículos)**

Camino	Vehículo	%	Camino	Vehículo	%
<b>Red propia</b>			<b>Red FARAC <sup>2</sup></b>		
Puebla-Cd. Mendoza	12 258	29.7	México-Cuernavaca	10 078	4.8
Tijuana-Ensenada	10 906	26.5	Puente de Ixtla-Iguala	1 517	0.7
Cd. Mendoza-Córdoba	8 034	19.5	La Pera-Cuautla	5 118	2.4
Chapallilla-Compostela	1 218	3.0	Zacapalco-Rancho Viejo	476	0.2
Rancho Viejo-Taxco	815	2.0	Campeche-Champotón	1 090	0.5
Tehuacán-Oaxaca	5 545	13.5	Cuernavaca- Acapulco	13 547	6.4
Arriaga-Huixtla	545	1.3	Monterrey-Nuevo Laredo	2 301	1.1
La Rumorosa-Tecate	1 781	4.3	Lib. Pon. de Tampico	2 390	1.1
Aeropuerto-Los Cabos-San	55	0.1	Cadereyta-Reynosa	1 741	0.8
José del Cabo <sup>4</sup>			Zapotlanejo-Lagos de Moreno	5 832	2.8
Tuxtla Gutiérrez-San Cristóbal	25	0.1	León-Lagos de Moreno-Aguascalientes	5 435	2.6
de las Casas <sup>4</sup>			Chamapa-Lechería	14 508	6.9
Cuauhtémoc-Osiris <sup>4</sup>	37	0.1	Lib. Noreste de Querétaro	3 336	1.6
<b>Total</b>	<b>41 219</b>	<b>100.0</b>	Pátzcuaro-Uruapan	3 984	1.9
<b>Red contratada <sup>1</sup></b>			Maravatio-Zapotlanejo	9 459	4.5
México-Tizayuca	27 396	64.2	Guadalajara-Zapotlanejo	9 201	4.4
Guadalajara-Colima	4 105	9.6	Guadalajara-Tepic	7 327	3.5
Tepic-San Blas	2 337	5.5	Córdoba-Veracruz	5 361	2.5
Atzacmulco-Maravatio	3 767	8.8	La Tinaja-Cosoleacaque	3 776	1.8
Culiacán-Las Brisas	3 255	7.6	Mazatlán-Culiacán	2 575	1.2
Asunción-Tejocotal	1 795	4.2	Estación Don-Nogales	8 766	4.2
<b>Total</b>	<b>42 655</b>	<b>100.0</b>	Lib. Oriente de Saltillo	3 067	1.5
			La Carbonera-Puerto México	3 031	1.4
			Torreón-Saltillo	1 697	0.8
			Gómez Palacio-Corralitos-Autop. Unión	1 396	0.7
			Reynosa-Matamoros	1 144	0.5
			Escuinapa-Rosario	1 007	0.5
			Aguadulce-Cárdenas	1 333	0.6
			Uruapan-Nueva Italia	2 674	1.3
			Tihuatlán-Gutiérrez Zamora	448	0.2
			Nueva Italia-Lázaro Cárdenas	746	0.4
			México-Querétaro <sup>3</sup>	26 354	12.5
			Querétaro-Irapuato <sup>3</sup>	12 997	6.2
			México-Puebla <sup>3</sup>	36 777	17.5
			Las Choapas-Ocozocuatla <sup>4</sup>	126	0.1
			<b>Total</b>	<b>210 615</b>	<b>100.0</b>

<sup>1/</sup> Autopistas concesionadas operadas por contrato.

<sup>2/</sup> Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Autopistas administradas por CAPUFE a partir de septiembre de 1998.

<sup>3/</sup> A partir de octubre de 2002 las autopistas México-Querétaro y México-Puebla pasaron de la Red propia a la red FARAC y en diciembre la autopista Querétaro-Irapuato.

<sup>4/</sup> Iniciaron operaciones durante 2002.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

**Tránsito de vehículos en las autopistas de Caminos y Puentes Federales de Ingresos y Servicios Conexos por tipo de vehículo (miles de vehículos)**

Clase de vehículo	Red de caminos						Total	%
	Propia	%	Contratada <sup>1/</sup>	%	FARAC <sup>2/</sup>	%		
Automóviles <sup>3/</sup>	30 898	75.0	34 373	80.6	152 258	72.3	217 529	73.9
Autobuses	3 143	7.6	1 751	4.1	14 845	7.0	19 739	6.7
Camiones de carga de dos ejes	2 165	5.3	3 660	8.6	11 384	5.4	17 209	5.8
Camiones de carga de tres ejes	1 627	3.9	1 001	2.3	8 180	3.9	10 808	3.7
Camiones de carga de cuatro ejes	62	0.2	42	0.1	454	0.2	558	0.2
Camiones de carga de cinco ejes o más	3 324	8.1	1 828	4.3	23 494	11.2	28 646	9.7
<b>Total</b>	<b>41 219</b>	<b>100.0</b>	<b>42 655</b>	<b>100.0</b>	<b>210 615</b>	<b>100.0</b>	<b>294 489</b>	<b>100.0</b>

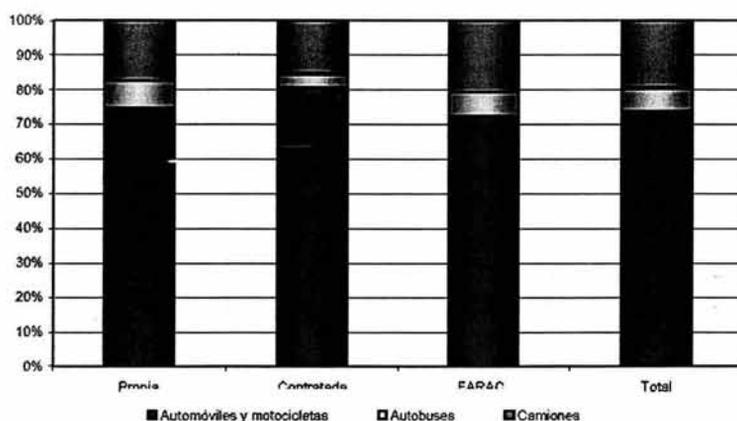
1/ Autopistas concesionadas operadas por contrato.

2/ Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Autopistas administradas por CAPUFE a partir de septiembre de 1998.

3/ Incluye automóviles con remolque y motocicletas.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

Distribución porcentual del tránsito vehicular en puentes por tipo de vehículo



**Tránsito de vehículos en los puentes de Caminos  
y Puentes Federales de Ingresos y Servicios Conexos  
(miles de vehículos)**

Puente	Red de puentes					
	Propia	%	Contratada <sup>1/</sup>	%	FARAC <sup>2/</sup>	%
Acuña, Coah. <sup>3/</sup>	2 124	3.5	-	-	-	-
Alvarado, Ver.	1 396	2.3	-	-	-	-
Cadereyta, N. L.	1 641	2.7	-	-	-	-
Camargo, Tamps. <sup>3/</sup>	726	1.2	-	-	-	-
Caracol, Oax.	1 672	2.8	-	-	-	-
Coatzacoalcos, Ver.	6 098	10.1	-	-	-	-
Culiacán, Sin.	1 718	2.9	-	-	-	-
Dr. Rodolfo Robles, Chis. <sup>3/</sup>	272	0.5	-	-	-	-
Grijalva, Tab.	3 191	5.3	-	-	-	-
Ignacio Chávez, Mich.	-	-	-	-	941	13.3
Ignacio Zaragoza, Tamps. <sup>3/</sup>	-	-	-	-	2 438	34.5
Ing. A. Dovalí Jaime Bis, Ver.	193	0.3	-	-	-	-
Ing. A. Dovalí Jaime, Ver.	2 244	3.7	-	-	-	-
Juárez-Lincoln, Tamps. <sup>3/</sup>	5 278	8.8	-	-	-	-
La Piedad, Mich.	2 574	4.3	-	-	-	-
Laredo, Tamps. <sup>3/</sup>	1 610	2.7	-	-	-	-
Las Flores, Tamps. <sup>3/</sup>	1 219	2.0	-	-	-	-
Libre Comercio, Tamps. <sup>3/</sup>	-	-	863	7.5	-	-
Matamoros, Tamps. <sup>3/</sup>	2 392	4.0	-	-	-	-
Miguel Alemán, Tamps. <sup>3/</sup>	1 231	2.0	-	-	-	-
Nautla, Ver.	1 519	2.5	-	-	-	-
Nuevo Amanecer, Tamps. <sup>3/</sup>	-	-	-	-	2 703	38.3
Ojinaga, Chih. <sup>3/</sup>	748	1.3	-	-	-	-
Pánuco, Ver.	1 436	2.4	-	-	-	-
Papaloapan, Oax.	1 386	2.3	-	-	-	-
Paso del Norte, Chih. <sup>3/</sup>	3 656	6.1	-	-	-	-
Piedras Negras II, Coah. <sup>3/</sup>	-	-	2 202	19.1	-	-
Piedras Negras, Coah. <sup>3/</sup>	1 347	2.3	-	-	-	-
Reynosa, Tamps. <sup>3/</sup>	5 779	9.6	-	-	-	-
San Juan, Tamps.	836	1.4	-	-	-	-
San Miguel, Sin.	-	-	2 233	19.3	-	-
Sinaloa, Sin.	2 692	4.5	-	-	-	-
Solidaridad-Colombia, N.L. <sup>3/</sup>	-	-	520	4.5	-	-
Suchiate II, Chis. <sup>3/</sup>	58	n.s.	-	-	-	-
Tampico, Tamps.	2 279	3.8	-	-	-	-
Tecolutía, Ver.	1 464	2.4	-	-	-	-
Tlacoalpan, Ver.	350	0.6	-	-	-	-
Tuxpan, Ver.	-	-	1 757	15.2	-	-
Usumacinta, Tab.	1 028	1.7	-	-	-	-
Zacatal, Camp.	-	-	-	-	981	13.9
Zaragoza-Ysleta, Chih. <sup>3/</sup>	-	-	3 965	34.4	-	-
<b>Total</b>	<b>60 157</b>	<b>100.0</b>	<b>11 540</b>	<b>100.0</b>	<b>7 063</b>	<b>100.0</b>

<sup>1/</sup> Puentes concesionados operados por contrato.

<sup>2/</sup> Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Puentes administrados por CAPUFE a partir de septiembre de 1998.

<sup>3/</sup> El registro vehicular corresponde en el norte al sentido del tránsito México-Estados Unidos y en el sureste al sentido del tránsito México-Guatemala.

n.s. - No significativo.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

### Tránsito de vehículos en los puentes de Caminos y Puentes Federales de Ingresos y Servicios Conexos por tipo de vehículo (miles de vehículos)

Clase de vehículo	Red de puentes							
	Propia	%	Contratada <sup>1/</sup>	%	FARAC <sup>2/</sup>	%	Total	%
Automóviles <sup>3/</sup>	48 433	80.5	9 366	81.2	6 139	86.9	63 938	81.2
Autobuses	2 511	4.2	332	2.9	186	2.6	3 029	3.8
Camiones de carga de dos ejes	2 639	4.4	368	3.2	174	2.5	3 181	4.0
Camiones de carga de tres ejes	1 734	2.9	233	2.0	95	1.3	2 062	2.6
Camiones de carga de cuatro ejes	82	0.1	21	0.2	14	0.2	117	0.2
Camiones de carga de cinco ejes o más	4 537	7.5	1 220	10.5	455	6.5	6 212	7.9
Triciclos <sup>4/</sup>	221	0.4	0	-	-	-	221	0.3
Total	60 157	100.0	11 540	100.0	7 063	100.0	78 760	100.0
Peatones	20 521	96.6	716	3.4	-	-	21 237	100.0

1/ Puentes concesionados operados por contrato.

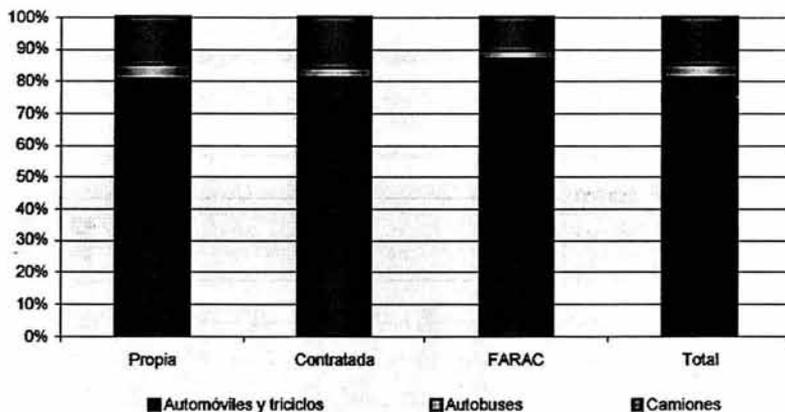
2/ Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Puentes administrados por CAPUFE a partir de septiembre de 1998.

3/ Incluye automóviles con remolque y motocicletas.

4/ Bicicletas con canastilla al frente para transportar carga o un pasajero; se registran en el puente Dr. Rodolfo Robles.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

Distribución porcentual del tránsito vehicular en puentes por tipo de vehículo



### Tránsito de vehículos en las casetas de cobro de Caminos y Puentes Federales de Ingresos y Servicios Conexos (miles de vehículos)

Año	Número de vehículos en caminos				Número de vehículos en puentes			
	Propios	Contratados <sup>1/</sup>	FARAC <sup>2/</sup>	Total	Propios	Contratados <sup>1/</sup>	FARAC <sup>2/</sup>	Total
1991	95 747	8 165	-	103 912	53 248	2 210	-	55 458
1992	94 501	8 379	-	102 880	54 324	2 452	-	56 776
1993	109 941	8 283	-	118 224	56 002	4 370	-	60 372
1994	95 281	25 555	-	120 836	54 760	5 135	-	59 895
1995	67 797 <sup>3/</sup>	46 175	-	113 972	51 002	7 364	-	58 366
1996	72 003	46 317	-	118 320	52 759	8 328	-	61 087
1997	80 523	47 932	-	128 455	54 389	9 545	-	63 934
1998	86 584	49 900	25 109	161 593	55 912	8 166 <sup>*</sup>	905	64 983
1999	93 911	51 468	80 210	225 589	58 751	9 420	4 346	72 517
2000	102 037	53 869	94 578	250 484	59 832	11 057	6 104	76 993
2001	109 120	40 003	124 068	273 191	59 206	11 089	6 563	76 858
2002	41 219	42 655	210 615	294 489	60 157	11 540	7 063	78 760

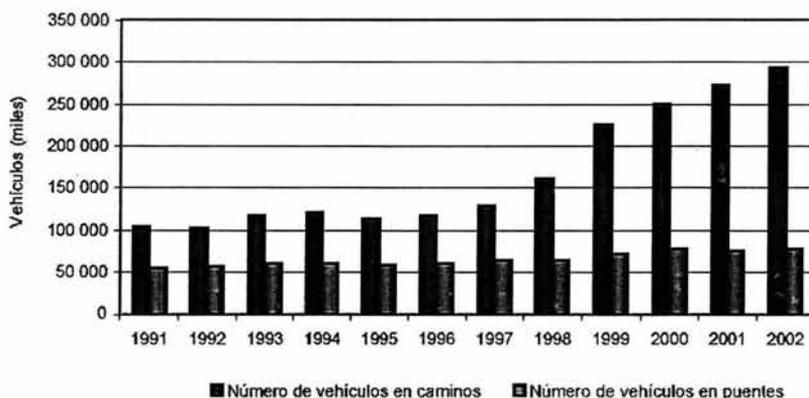
1/ Se refiere a autopistas o puentes concesionados operados por contrato.

2/ Fideicomiso de Apoyo al Rescate de Autopistas Concesionadas. Autopistas o puentes administrados por CAPUFE a partir de septiembre de 1998.

3/ Cifra modificada por la dependencia generadora de la información.

Fuente: Caminos y Puentes Federales de Ingresos y Servicios Conexos.

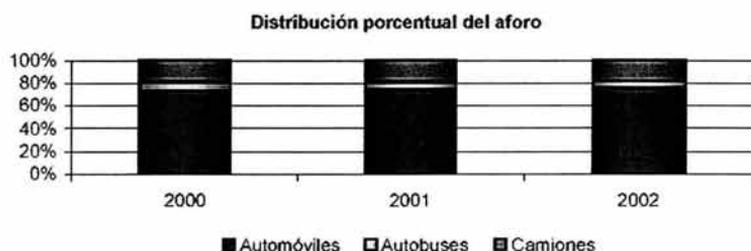
#### Tránsito de vehículos en las casetas de cobro de CAPUFE



### Aforo en las autopistas y puentes de cuota por tipo de vehículo (tránsito diario promedio anual)

Tipo de vehículo	2000		2001		2002		Variación	Variación
		%		%		%	00/01 (%)	01/02 (%)
Automóviles	508 753	73.7	546 901	74.2	579 153	75.6	7.5	5.9
Autobuses	49 850	7.2	51 184	7.0	46 801	6.1	2.7	-8.6
Camiones	131 464	19.1	138 878	18.8	139 835	18.3	5.6	0.7
<b>Total</b>	<b>690 067</b>	<b>100.0</b>	<b>736 963</b>	<b>100.0</b>	<b>765 789</b>	<b>100.0</b>	<b>6.8</b>	<b>3.9</b>

Fuente: Unidad de Autopistas de Cuota.



### Aforo en las autopistas y puentes de cuota (tránsito diario promedio anual)

Año	Concesionadas	CAPUFE	FARAC	Total	Variación anual (%)
1992	94 966	282 026	-	376 992	-
1993	114 326	314 463	-	428 789	13.7
1994	178 176	250 423	55 808	484 407	13.0
1995	173 789	236 679	52 385	462 853	-4.4
1996	181 060	247 698	66 865	495 623	7.1
1997	193 015	264 490	80 872	538 377	8.6
1998	203 825	282 478	100 925	587 228	9.1
1999	223 823	296 529	115 927	636 279	8.4
2000	238 914	307 404	143 749	690 067	8.5
2001	252 434 <sup>1/</sup>	318 978	165 551	736 963	6.8
2002	225 878	236 443	303 468	765 789	3.9

<sup>1/</sup> Cifras modificadas por la dependencia generadora de la información.

Nota: De 1994 a 1998, las cifras fueron modificadas por la dependencia generadora de la información.

Fuente: Unidad de Autopistas de Cuota.

**Aforo en las autopistas y puentes de cuota con mayor tránsito  
(tránsito diario promedio anual)**

Autopista	2001				2002			
	Automóviles	Autobuses	Camiones	Total	Automóviles	Autobuses	Camiones	Total
México-Pachuca	28 577	3 708	2 852	35 137	32 600	2 679	3 958	39 237
México-Querétaro	18 823	3 043	11 579	33 445	20 927	2 877	12 056	35 860
México-Puebla	22 184	3 251	5 491	30 926	24 177	3 266	5 626	33 069
México-Cuernavaca	22 062	1 895	2 125	26 082	23 178	768	3 044	26 990
Guadalajara-Zapotlanejo	17 710	1 993	3 658	23 361	19 168	1 531	4 510	25 209
Constituyentes-La Marquesa	19 580	2 311	808	22 699	20 812	2 302	830	23 944
Orizaba-Córdoba	15 204	1 739	5 650	22 593	16 317	1 822	5 657	23 796
Querétaro-Celaya <sup>1/</sup>	13 089	1 785	4 271	19 145				
Peñón-Texcoco	14 240	702	1 959	16 901	15 420	814	1 971	18 205
Querétaro-Irapuato <sup>1/</sup>					11 706	1 422	3 846	16 974
Ecatepec-Pirámides	11 988	1 687	1 382	15 057	12 997	1 726	1 360	16 083
Chamapa-Lechería	12 760	-	1 589	14 349	13 729	-	1 411	15 140
Celaya-Irapuato <sup>1/</sup>	9 545	1 209	3 476	14 230				
Cardel-Veracruz	7 957	772	2 322	11 051	8 768	728	2 427	11 923
Puebla-Orizaba	5 878	1 214	2 502	9 594	6 888	1 242	2 965	11 095
Tijuana-Ensenada	8 664	132	575	9 371	9 216	137	607	9 960
Lib. N. de Querétaro <sup>2/</sup>					3 270	446	5 423	9 139
Subtotal	228 261	25 441	50 239	303 941	239 173	21 760	55 691	316 624
Subtotal otras autopistas	141 015	17 330	54 511	212 856	156 675	16 534	50 533	223 742
Total	369 276	42 771	104 750	516 797	395 848	38 294	106 224	540 366

<sup>1/</sup> A partir del año 2002 las autopistas Querétaro-Celaya y Celaya-Irapuato se consideran dentro de la autopista Querétaro-Irapuato

<sup>2/</sup> Para el año 2001 no se considera dentro de las quince autopistas con mayor tránsito

Fuente: Unidad de Autopistas de Cuota.

Puentes	2001				2002			
	Automóviles	Autobuses	Camiones	Total	Automóviles	Autobuses	Camiones	Total
Coatzacoalcos	13 525	1 011	1 904	16 440	14 776	1 077	1 882	17 735
Reynosa	14 743	68	15	14 826	15 788	70	14	15 872
Juárez-Lincoln	14 705	105	22	14 832	14 337	109	24	14 470
Puente Int. Zaragoza-Ysleta	9 336	-	1 209	10 545	9 801	-	1 100	10 901
Paso del Norte	12 420	28	43	12 491	9 999	53	12	10 064
Grijalva	6 345	665	1 745	8 755	6 912	674	1 724	9 310
Sinaloa	5 084	647	2 209	7 940	5 035	642	2 102	7 779
Puente Int. Reynosa-Pharr	6 108	13	856	6 977	6 499	2	903	7 404
La Piedad	3 530	612	3 608	7 750	3 154	546	3 548	7 248
Puente Int. Los Tomates <sup>1/</sup>					6 132	151	398	6 681
Puente Colorado	5 732	558	1 150	7 440	4 954	519	1 194	6 667
Matamoros	6 368	2	4	6 374	6 549	2	4	6 555
Tampico	5 197	356	555	6 108	5 631	365	477	6 473
Puente San Miguel	3 562	516	1 982	6 060	3 604	596	2 020	6 220
Dovall Jaime	3 116	264	2 554	5 934	3 427	268	2 517	6 212
Puente Piedras Negras II <sup>2/</sup>	5 829	2	278	6 109				
Subtotal	115 600	4 847	18 134	138 581	116 598	5 074	17 919	139 591
Subtotal otros puentes	62 025	3 566	15 994	81 585	66 707	3 433	15 692	85 832
Total	177 625	8 413	34 128	220 166	183 305	8 507	33 611	225 423

<sup>1/</sup> Para el año 2001, no se considera dentro de los 15 puentes con mayor tránsito

<sup>2/</sup> Para el año 2002 no se considera dentro de los quince puentes con mayor tránsito

Fuente: Unidad de Autopistas de Cuota.

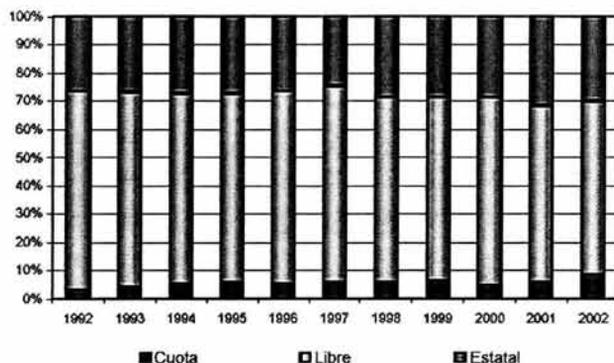
### Longitud aforada de la red nacional de carreteras por tipo de camino (kilómetros)

Año	Longitud aforada de la red carretera						Total	Long. aforada Long. pavimentada <sup>U</sup> (%)
	Cuota	%	Libre	%	Estatal	%		
1992	2 005.3	3.4	41 236.5	70.4	15 357.8	26.2	58 599.6	69.0
1993	2 607.7	4.5	40 241.0	68.8	15 599.9	26.7	58 448.6	68.1
1994	3 268.0	5.6	39 245.6	67.3	15 842.1	27.1	58 355.7	65.4
1995	3 509.0	5.9	39 565.7	66.9	16 083.2	27.2	59 157.9	65.0
1996	3 137.3	5.4	40 155.6	68.4	15 392.3	26.2	58 685.2	62.7
1997	3 434.7	6.1	39 114.9	69.4	13 785.5	24.5	56 335.1	58.3
1998	3 649.7	5.9	40 366.9	65.5	17 632.0	28.6	61 648.6	63.1
1999	3 957.3	6.4	40 355.9	65.2	17 558.7	28.4	61 871.9	62.3
2000	2 932.1	4.8	41 020.5	66.7	17 567.9	28.5	61 520.5	60.5
2001	3 811.3	5.8	41 392.9	62.7	20 799.1	31.5	66 003.3	63.4
2002	6 145.6	8.8	42 627.7	61.3	20 815.3	29.9	69 588.7	65.9

<sup>U</sup> No incluye la longitud pavimentada de la red rural.

Fuente: Dirección General de Servicios Técnicos.

Porcentaje de longitud aforada por tipo de camino

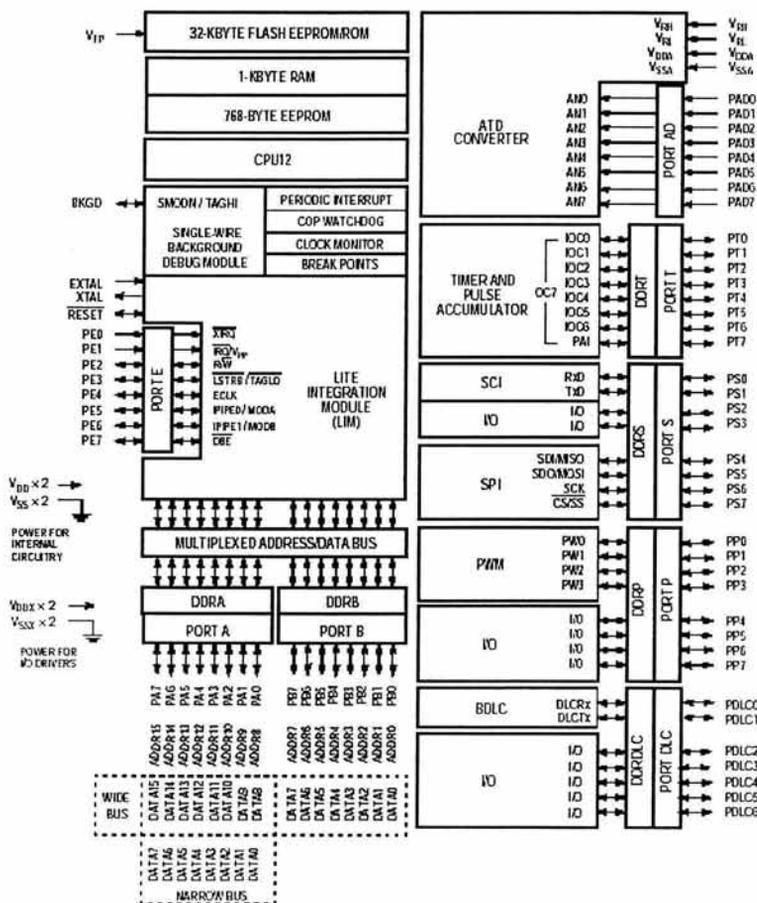


APENDICE B.

El microcontrolador MC68HC912B32.

Diagrama de bloques del Microcontrolador MC68HC912B32.

El microcontrolador MC68HC912B32 es un dispositivo de 16 bits compuesto por un on-chip estandard incluyendo una Unidad Central de Proceso (CPU) también de 16 bits, 32 kbytes en memoria EEPROM Flash, 1 Kbytes en memoria RAM, 768 bytes en EEPROM, una interfase de comunicación serial asincrona, un timer de 8 canales y un acumulador de pulsos de 16 bits, un convertidor análogo-digital de 10 bits, un modulador de ancho de pulsos de 4



canales.

### Pines de polarización.

Mnemonic	Pin Number	Description
V <sub>DD</sub>	10, 47	Internal power and ground
V <sub>SS</sub>	11, 48	
V <sub>DDX</sub>	31, 78	External power and ground supply to pin drivers
V <sub>SSX</sub>	30, 77	
V <sub>DDA</sub>	59	Operating voltage and ground for the ATD; allows the supply voltage to be bypassed independently
V <sub>SSA</sub>	60	
V <sub>RH</sub>	49	Reference voltages for the analog-to-digital converter
V <sub>RL</sub>	50	
V <sub>FP</sub>	69	Programming voltage for the FLASH EEPROM and required supply for normal operation — MC68HC912B32 and MC68HC912BC32 only. Pin 69 is a no connect (NC) on the MC68HC12BE32 and MC68HC12BC32.

### Modo de operación.

MODA y MODB son pines de selección. El estado que tengan durante el inicio determina el modo de operación del micro. El modo single-chip es usado para productos finales mientras que en modo expandido se obtiene un bus de datos multiplexado.

### RESET

Es una entrada para inicializar al microcontrolador a un estado conocido de inicio, cuando no esta presionado se encuentra en nivel bajo (cero volts) y cuando se presiona llega a nivel alto (5 volts).

### IRQ

Es un pin de atención de interrupción externa-mascarable, es controlado por el registro INTCR y puede ser programado para responder a flancos de subida o bajada de polarización. Este pin es siempre de entrada y puede ser visto en cualquier momento.

### Puertos.

El microcontrolador tiene 8 puertos que son usados para control y acceso a diversos subsistemas. También los pines de los puertos pueden ser usados para propósito general de

entrada/salida. Cada puerto consiste de un registro que puede ser leído o escrito en cualquier momento, también un registro que controla la dirección de cada pin. Con Reset todos los puertos son configurados para entrada.

Port Name	Pin Numbers	Data Direction DD Register (Address)	Description
Port A PA7-PA0	46-39	IrVOut DDRA (\$0002)	Port A and port B pins are used for address and data in expanded modes. The port data registers are not in the address map during expanded and peripheral mode operation. When in the map, port A and port B can be read or written anytime.  DDRA and DDRB are not in the address map in expanded or peripheral modes.
Port B PB7-PB0	25-18	IrVOut DDRB (\$0003)	
Port AD PAD7-PAD0	56-51	In	Analog-to-digital converter and general-purpose I/O
Port DLC/PCAN <sup>(1)</sup> PDLC6-PDLC0 PCAN6-PCAN2	70-76	IrVOut DDRDL (\$00FF)	Byte data link communication (BDLC) subsystem and general-purpose I/O
Port E PE7-PE0	26-29, 35-36	PE1-PE0 In PE7-PE2 IrVOut DDRE (\$0009)	Mode selection, bus control signals, and interrupt service request signals; or general-purpose I/O
Port P PP7-PP0	79, 80, 1-6	IrVOut DDRP (\$0057)	General-purpose I/O. PP3-PP0 are used with the pulse-width modulator when enabled.
Port S PS7-PS0	66-61	IrVOut DDRS (\$00D7)	Serial communications interface and serial peripheral interface subsystems and general-purpose I/O
Port T PT7-PT0	16-12, 9-7	IrVOut DDRT (\$00AF)	General-purpose I/O when not enabled for input capture and output compare in the timer and pulse accumulator subsystem

1. Port DLC applies to the MC68HC912B32 and MC68HC12BE32 and PCAN to the MC68HC(9)12BC32.

### Modelo de programación.

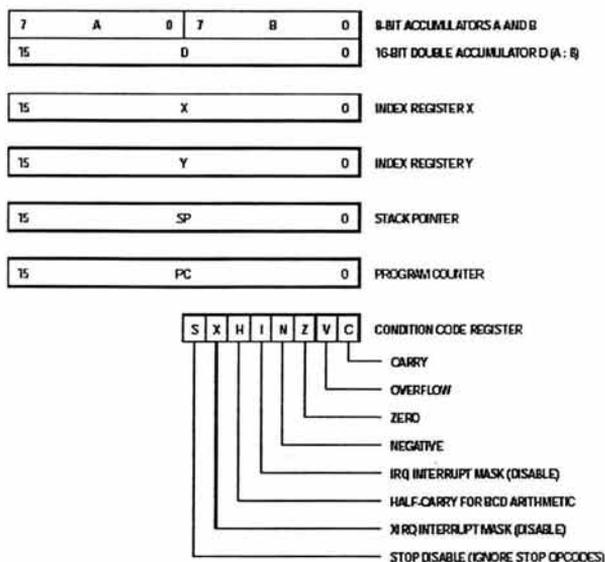
Su CPU tiene 2 acumuladores de 8-bits (A y B) de propósito general, que pueden ser concatenados a un solo acumulador de 16-bits (D) para ciertas instrucciones. También cuenta con:

Dos registros de índice (X y Y)

Apuntador de pila de 16-bits (SP)

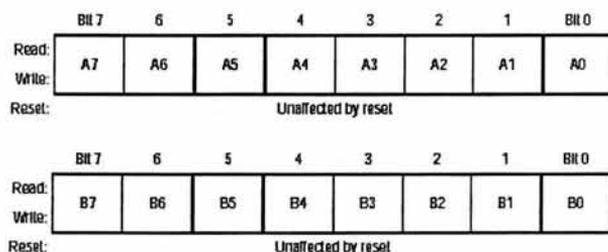
Contador de programa de 16-bits (PC)

Registro de códigos de condición (CCR)



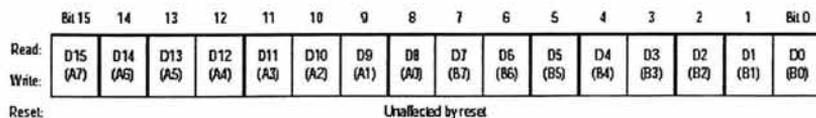
### Acumuladores A y B.

Los acumuladores A y B son de propósito general, 8-bits cada uno, contienen operándos y resultados de cálculos aritméticos o manipulaciones de datos.



### Acumulador D.

Este acumulador es la concatenación de los acumuladores A y B. Algunas instrucciones procesan la combinación de estos 2 acumuladores de 8-bits como uno doble de 16-bits.



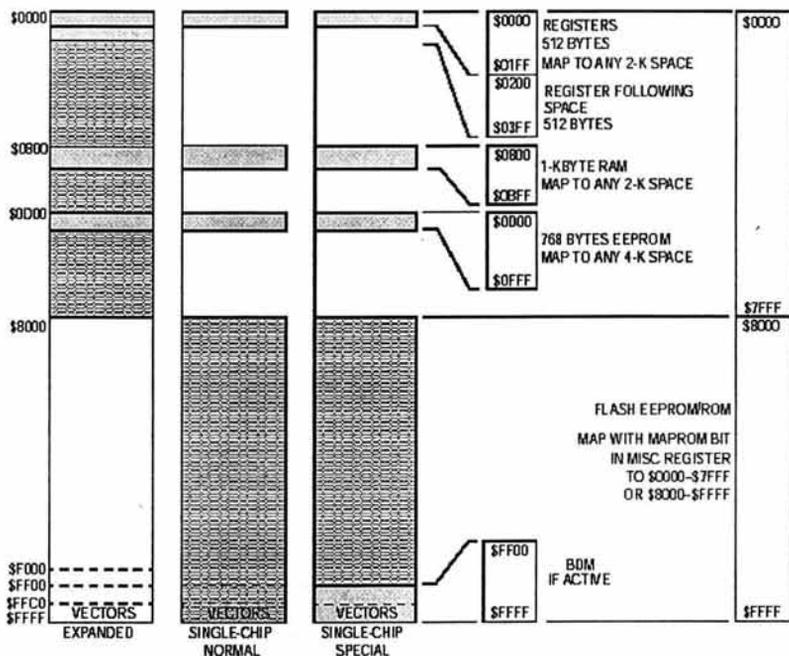
### Modos de direccionamiento.

Los modos de direccionamiento determinan como la unidad central de Proceso (CPU) accederá a las localidades de memoria que serán utilizadas. Estos modos de direccionamiento son una parte implícita de las instrucciones del CPU12. A continuación se presentan todos los modos de direccionamiento.

Addressing Mode	Source Format	Abbreviation	Description
Inherent	INST (no externally supplied operands)	INH	Operands (if any) are in CPU registers
Immediate	INST #opr <i>n</i> i or INST #opr16 <i>i</i>	IMM	Operand is included in instruction stream 8- or 16-bit size implied by context
Direct	INST opr <i>8</i> a	DIR	Operand is the lower 8 bits of an address in the range \$0000-\$00FF
Extended	INST opr16 <i>a</i>	EXT	Operand is a 16-bit address
Relative	INST rel <i>8</i> or INST rel16	REL	An 8-bit or 16-bit relative offset from the current pc is supplied in the instruction
Indexed (5-bit offset)	INST opr <i>5</i> ,xy <i>sp</i>	IDX	5-bit signed constant offset from X, Y, SP, or PC
Indexed (pre-decrement)	INST opr <i>3</i> -,xy <i>sp</i>	IDX	Auto pre-decrement x, y, or sp by 1 – 8
Indexed (pre-increment)	INST opr <i>3</i> +,xy <i>sp</i>	IDX	Auto pre-increment x, y, or sp by 1 – 8
Indexed (post-decrement)	INST opr <i>3</i> ,xy <i>sp</i> -	IDX	Auto post-decrement x, y, or sp by 1 – 8
Indexed (post-increment)	INST opr <i>3</i> ,xy <i>sp</i> +	IDX	Auto post-increment x, y, or sp by 1 – 8
Indexed (accumulator offset)	INST abd,xy <i>sp</i>	IDX	Indexed with 8-bit (A or B) or 16-bit (D) accumulator offset from X, Y, SP, or PC
Indexed (9-bit offset)	INST opr <i>9</i> ,xy <i>sp</i>	IDX1	9-bit signed constant offset from X, Y, SP, or PC (lower 8 bits of offset in one extension byte)
Indexed (16-bit offset)	INST opr <i>16</i> ,xy <i>sp</i>	IDX2	16-bit constant offset from X, Y, SP, or PC (16-bit offset in two extension bytes)
Indexed-Indirect (16-bit offset)	INST [opr <i>16</i> ,xy <i>sp</i> ]	[IDX2]	Pointer to operand is found at... 16-bit constant offset from X, Y, SP, or PC (16-bit offset in two extension bytes)
Indexed-Indirect (D accumulator offset)	INST [D,xy <i>sp</i> ]	[D,IDX]	Pointer to operand is found at... X, Y, SP, or PC plus the value in D

### Mapa de memoria.

Este es el mapa de memoria para cada modo de operación inmediatamente después del Reset.



Set de instrucciones del microcontrolador MC68HC912B32.

Source Form	Operation	Addr. Mode	Machine Coding (hex)	HCS12	Access Detail	SX HI	H Z VC
ABA	(A) + (B) ⇒ A Add Accumulators A and B	INH	18 04	00	00	--A-	Δ Δ Δ Δ
ABX	(B) + (X) ⇒ X Transfers to LEAX,B,X	IOX	1A 85	0c	0P	----	----
ABY	(B) + (Y) ⇒ Y Transfers to LEAY,B,Y	IOX	19 8D	0c	0P	----	----
ADCA #opnd ADCA opndB ADCA opndA ADCA opnd0_xysp ADCA opnd8_xysp ADCA opnd16_xysp ADCA [D_xysp] ADCA [opnd16_xysp]	(A) + (M) + C ⇒ A Add with Carry to A	RAM DIR EXT IOX IOX1 IOX2 [D_IOX] [IOX2]	85 11 89 0d 89 hh ll 88 xb 88 xb ff 88 xb ee ff 88 xb ee ff	0 rPC rPO rPC rPO rPCP rPCP rPCP	0 rCP rCP rCP rPO rCP rPCP rPCP	--A-	Δ Δ Δ Δ
ADCB #opnd ADCB opndB ADCB opndA ADCB opnd0_xysp ADCB opnd8_xysp ADCB opnd16_xysp ADCB [D_xysp] ADCB [opnd16_xysp]	(B) + (M) + C ⇒ B Add with Carry to B	RAM DIR EXT IOX IOX1 IOX2 [D_IOX] [IOX2]	C9 11 D9 0d D9 hh ll D8 xb D8 xb ff D8 xb ee ff D8 xb ee ff	0 rPC rPO rPC rPO rPCP rPCP rPCP	0 rCP rCP rCP rPO rCP rPCP rPCP	--A-	Δ Δ Δ Δ
ADDA #opnd ADDA opndB ADDA opndA ADDA opnd0_xysp ADDA opnd8_xysp ADDA opnd16_xysp ADDA [D_xysp] ADDA [opnd16_xysp]	(A) + (M) ⇒ A Add without Carry to A	RAM DIR EXT IOX IOX1 IOX2 [D_IOX] [IOX2]	88 11 88 0d 88 hh ll 88 xb 88 xb ff 88 xb ee ff 88 xb ee ff	0 rPC rPO rPC rPO rPCP rPCP rPCP	0 rCP rCP rCP rPO rCP rPCP rPCP	--A-	Δ Δ Δ Δ
ADDB #opnd ADDB opndB ADDB opndA ADDB opnd0_xysp ADDB opnd8_xysp ADDB opnd16_xysp ADDB [D_xysp] ADDB [opnd16_xysp]	(B) + (M) ⇒ B Add without Carry to B	RAM DIR EXT IOX IOX1 IOX2 [D_IOX] [IOX2]	C8 11 D8 0d D8 hh ll D8 xb D8 xb ff D8 xb ee ff D8 xb ee ff	0 rPC rPO rPC rPO rPCP rPCP rPCP	0 rCP rCP rCP rPO rCP rPCP rPCP	--A-	Δ Δ Δ Δ
ADDD #opnd ADDD opndB ADDD opndA ADDD opnd0_xysp ADDD opnd8_xysp ADDD opnd16_xysp ADDD [D_xysp] ADDD [opnd16_xysp]	(A,B) + (MM+1) ⇒ AB Add 16-Bit to D (A,B)	RAM DIR EXT IOX IOX1 IOX2 [D_IOX] [IOX2]	C3 11 kx D3 0d D3 hh ll D3 xb D3 xb ff D3 xb ee ff D3 xb ee ff	00 RDC RPO RDC RPO RDCP RDCP RDCP	00 RCP RCP RCP RPO RCP RDCP RDCP	----	Δ Δ Δ Δ
ANDA #opnd ANDA opndB ANDA opndA ANDA opnd0_xysp ANDA opnd8_xysp ANDA opnd16_xysp ANDA [D_xysp] ANDA [opnd16_xysp]	(A) & (M) ⇒ A Logical AND A with Memory	RAM DIR EXT IOX IOX1 IOX2 [D_IOX] [IOX2]	84 11 84 0d 84 hh ll 84 xb 84 xb ff 84 xb ee ff 84 xb ee ff	0 rPC rPO rPC rPO rPCP rPCP rPCP	0 rCP rCP rCP rPO rCP rPCP rPCP	----	Δ Δ 0-
ANDB #opnd ANDB opndB ANDB opndA ANDB opnd0_xysp ANDB opnd8_xysp ANDB opnd16_xysp ANDB [D_xysp] ANDB [opnd16_xysp]	(B) & (M) ⇒ B Logical AND B with Memory	RAM DIR EXT IOX IOX1 IOX2 [D_IOX] [IOX2]	C4 11 D4 0d D4 hh ll D4 xb D4 xb ff D4 xb ee ff D4 xb ee ff	0 rPC rPO rPC rPO rPCP rPCP rPCP	0 rCP rCP rCP rPO rCP rPCP rPCP	----	Δ Δ 0-
ANDCC #opnd	(CCR) & (M) ⇒ CCR Logical AND CCR with Memory	RAM	10 11	0	0	UUUU	UUUU

Note 1. Due to internal CPU requirements, the program word fetch is performed twice to the same address during this instruction.

Source Form	Operation	Addr. Mode	Machine Coding (hex)	HCS12 Access Detail	H68HC12	SXHI	NZVC
ASL <i>oprt6a</i> ASL <i>oprt0_xysp</i> ASL <i>oprt8_xysp</i> ASL <i>oprt16_xysp</i> ASL [D, <i>xysp</i> ] ASL [ <i>oprt6_xysp</i> ] ASIA ASIB	 Arithmetic Shift Left	EXT IDX IDX1 IDX2 [D,IDX] [IDX] INH	78 hh 11 68 xb 68 xb ff 68 xb ee ff 68 xb 68 xb ee ff 48 58	rPw0 rPw rPw0 rPwP rPwP rPwP rPwP rPwP 0 0	rPw rPw rPw rPw rPw rPw rPw 0 0	----	Δ Δ Δ Δ
ASLD	 Arithmetic Shift Left Double	INH	59	0	0	----	Δ Δ Δ Δ
ASR <i>oprt6a</i> ASR <i>oprt0_xysp</i> ASR <i>oprt8_xysp</i> ASR <i>oprt16_xysp</i> ASR [D, <i>xysp</i> ] ASR [ <i>oprt6_xysp</i> ] ASRA ASRB	 Arithmetic Shift Right	EXT IDX IDX1 IDX2 [D,IDX] [IDX] INH	77 hh 11 67 xb 67 xb ff 67 xb ee ff 67 xb 67 xb ee ff 47 57	rPw0 rPw rPw0 rPwP rPwP rPwP rPwP 0 0	rPw rPw rPw rPw rPw rPw rPw 0 0	----	Δ Δ Δ Δ
BCC <i>rel6</i>	Branch if Carry Clear (if C = 0)	REL	24 xx	PPP/P'	PPD/P'	----	----
BCLR <i>oprt6a, msk6</i> BCLR <i>oprt6a, msk8</i> BCLR <i>oprt0_xysp, msk6</i> BCLR <i>oprt8_xysp, msk6</i> BCLR <i>oprt16_xysp, msk6</i>	$(M \oplus (mm)) \oplus M$ Clear Bit(s) in Memory	DIR EXT IDX IDX1 IDX2	40 dd mm 10 hh 11 mm 00 xb mm 00 xb ee mm 00 xb ee ff mm	rPw0 rPwP rPw rPwP rPwP	rPw rPw rPw rPw rPwP	----	Δ Δ 0-
BCS <i>rel6</i>	Branch if Carry Set (if C = 1)	REL	25 xx	PPP/P'	PPD/P'	----	----
BEQ <i>rel6</i>	Branch if Equal (if Z = 1)	REL	27 xx	PPP/P'	PPD/P'	----	----
BGE <i>rel6</i>	Branch if Greater Than or Equal (if N ⊕ V = 0) (signed)	REL	2C xx	PPP/P'	PPD/P'	----	----
BGND	Place CPU in Background Mode see <i>CPU12 Reference Manual</i>	INH	00	VCPDP	VCPDP	----	----
BGT <i>rel6</i>	Branch if Greater Than (if Z + (N ⊕ V) = 0) (signed)	REL	2E xx	PPP/P'	PPD/P'	----	----
BHI <i>rel6</i>	Branch if Higher (if C + Z = 0) (unsigned)	REL	22 xx	PPP/P'	PPD/P'	----	----
BHS <i>rel6</i>	Branch if Higher or Same (if C = 0) (unsigned) same function as BCC	REL	24 xx	PPP/P'	PPD/P'	----	----
BITA <i>oprt6a</i> BITA <i>oprt6a</i> BITA <i>oprt6a</i> BITA <i>oprt0_xysp</i> BITA <i>oprt8_xysp</i> BITA <i>oprt16_xysp</i> BITA [D, <i>xysp</i> ] BITA [ <i>oprt6_xysp</i> ]	(A) ⊕ (M) Logical AND A with Memory Does not change Accumulator or Memory	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX]	85 ii 95 dd B5 hh 11 A5 xb A5 xb ff A5 xb ee ff A5 xb A5 xb ee ff	D rPc rP0 rPc rP0 rPwP rPwP rPwP	rP rP rP rP rP rP rP rP	----	Δ Δ 0-
BITB <i>oprt6a</i> BITB <i>oprt6a</i> BITB <i>oprt6a</i> BITB <i>oprt0_xysp</i> BITB <i>oprt8_xysp</i> BITB <i>oprt16_xysp</i> BITB [D, <i>xysp</i> ] BITB [ <i>oprt6_xysp</i> ]	(B) ⊕ (M) Logical AND B with Memory Does not change Accumulator or Memory	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX]	C5 ii D5 dd F5 hh 11 E5 xb E5 xb ff E5 xb ee ff E5 xb E5 xb ee ff	D rPc rP0 rPc rP0 rPwP rPwP rPwP	rP rP rP rP rP rP rP rP	----	Δ Δ 0-
BLE <i>rel6</i>	Branch if Less Than or Equal (if Z + (N ⊕ V) = 1) (signed)	REL	2F xx	PPP/P'	PPD/P'	----	----
BLO <i>rel6</i>	Branch if Lower (if C = 1) (unsigned) same function as BCS	REL	25 xx	PPP/P'	PPD/P'	----	----

Note 1. PPPP indicates this instruction takes three cycles to refill the instruction queue if the branch is taken and one program fetch cycle if the branch is not taken.

Source Form	Operation	Addr. Mode	Machine Coding (hex)	Access Detail			
				HCS12	M68HC12	SXH1	NZVC
BLS <i>rel</i>	Branch if Lower or Same (if C + Z = 1) (unsigned)	REL	23 rr	PPP/P <sup>1</sup>	PPD/P <sup>1</sup>	----	----
BLT <i>rel</i>	Branch if Less Than (if N ⊕ V = 1) (signed)	REL	2D rr	PPP/P <sup>1</sup>	PPD/P <sup>1</sup>	----	----
BM <i>rel</i>	Branch if Minus (if N = 1)	REL	2B rr	PPP/P <sup>1</sup>	PPD/P <sup>1</sup>	----	----
BNE <i>rel</i>	Branch if Not Equal (if Z = 0)	REL	34 rr	PPP/P <sup>1</sup>	PPD/P <sup>1</sup>	----	----
BPL <i>rel</i>	Branch if Plus (if N = 0)	REL	2A rr	PPP/P <sup>1</sup>	PPD/P <sup>1</sup>	----	----
BRA <i>rel</i>	Branch Always (if 1 = 1)	REL	20 rr	PPP	PPP	----	----
BRCLR <i>oprtéa, mask, rel</i> BRCLR <i>oprtéa, mask, rel</i> BRCLR <i>oprtéa, yisp, mask, rel</i> BRCLR <i>oprtéa, yisp, mask, rel</i> BRCLR <i>oprtéa, yisp, mask, rel</i>	Branch if (M) * (imm) = 0 (if All Selected Bit(s) Clear)	DIR EXT IDX IDX1 IDX2	4F dd nn rr 1F bh ll nn rr 0F xb nn rr 0F xb ff nn rr 0F xb ee ff nn rr	rPPP rPPP rPPP rPPP rPPP	rPPP rPPP rPPP rPPP rPPP	----	----
BRN <i>rel</i>	Branch Never (if 1 = 0)	REL	21 rr	D	D	----	----
BRSET <i>oprtéa, mask, rel</i> BRSET <i>oprtéa, mask, rel</i> BRSET <i>oprtéa, yisp, mask, rel</i> BRSET <i>oprtéa, yisp, mask, rel</i> BRSET <i>oprtéa, yisp, mask, rel</i>	Branch if (M) * (imm) = 0 (if All Selected Bit(s) Set)	DIR EXT IDX IDX1 IDX2	4E dd nn rr 1E bh ll nn rr 0E xb nn rr 0E xb ff nn rr 0E xb ee ff nn rr	rPPP rPPP rPPP rPPP rPPP	rPPP rPPP rPPP rPPP rPPP	----	----
BSET <i>oprtéa, mask</i> BSET <i>oprtéa, mask</i> BSET <i>oprtéa, yisp, mask</i> BSET <i>oprtéa, yisp, mask</i> BSET <i>oprtéa, yisp, mask</i>	(M) + (imm) ⇒ M Set Bit(s) in Memory	DIR EXT IDX IDX1 IDX2	4C dd nn 1C bh ll nn 0C xb nn 0C xb ff nn 0C xb ee ff nn	rPw rPw rPw rPw rPw	rPw rPw rPw rPw rPw	----	Δ Δ 0-
BSR <i>rel</i>	(SP) - 2 ⇒ SP; RTN <sub>h</sub> , RTN <sub>l</sub> ⇒ M <sub>(SP)</sub> M <sub>(SP+1)</sub> Subroutine address ⇒ PC Branch to Subroutine	REL	07 rr	BPPP	PPPS	----	----
BVC <i>rel</i>	Branch if Overflow Bit Clear (if V = 0)	REL	2F rr	PPP/P <sup>1</sup>	PPD/P <sup>1</sup>	----	----
BVS <i>rel</i>	Branch if Overflow Bit Set (if V = 1)	REL	29 rr	PPP/P <sup>1</sup>	PPD/P <sup>1</sup>	----	----
CALL <i>oprtéa, page</i> CALL <i>oprtéa, yisp, page</i> CALL <i>oprtéa, yisp, page</i> CALL <i>oprtéa, yisp, page</i> CALL [D, yisp] CALL [oprtéa, yisp]	(SP) - 2 ⇒ SP; RTN <sub>h</sub> , RTN <sub>l</sub> ⇒ M <sub>(SP)</sub> M <sub>(SP+1)</sub> (SP) - 1 ⇒ SP; (PPG) ⇒ M <sub>(SP)</sub> pg ⇒ PPAGE register; Program address ⇒ PC  Call subroutine in extended memory (Program may be located on another expansion memory page.)  Indirect modes get program address and new pg value based on pointer.	EXT IDX IDX1 IDX2 [D, IDX] [IDX2]	4A bh ll pg 4B xb pg 4B xb ff pg 4B xb ee ff pg 4B xb ee ff 4B xb ee ff	gnSgPPP gnSgPPP gnSgPPP gnSgPPP f11gnSgPPP f11gnSgPPP	gnSgPPP gnSgPPP gnSgPPP gnSgPPP f11gnSgPPP f11gnSgPPP	----	----
CBA	(A) - (B) Compare 8-Bit Accumulators	INH	1E 17	OO	OO	----	Δ Δ Δ Δ
CLC	0 ⇒ C Translate to ANDCC #5FE	IMM	10 FE	D	D	----	---0
CLI	0 ⇒ I Translate to ANDCC #5EF (enables 1-bit interrupts)	IMM	10 EF	D	D	----	---0
CLR <i>oprtéa</i> CLR <i>oprtéa, yisp</i> CLR <i>oprtéa, yisp</i> CLR <i>oprtéa, yisp</i> CLR [D, yisp] CLR [oprtéa, yisp]	0 ⇒ M Clear Memory Location	EXT IDX IDX1 IDX2 [D, IDX] [IDX2]	7F bh ll 6F xb 6F xb ff 6F xb ee ff 6F xb ee ff 6F xb ee ff	Dw Dw Dw Dw D1rw D1rw	wD Dw Dw Dw P1rw P1rw	----	0 1 0 0
CLRA CLRB	0 ⇒ A Clear Accumulator A 0 ⇒ B Clear Accumulator B	INH INH	87 C7	O O	O O	----	----
CLV	0 ⇒ V Translate to ANDCC #5FD	IMM	1E FD	D	D	----	---0-

Note 1. PPPP indicates this instruction takes three cycles to refill the instruction queue if the branch is taken and one program fetch cycle if the branch is not taken.

Source Form	Operation	Addr. Mode	Machine Coding (hex)	HCS12	Access Detail M68HC12	SXHI	HZVC
CMPA #oprt6i CMPA opria CMPA oprt6a CMPA opara_yisp CMPA opara_yisp CMPA opara_r6_yisp CMPA [D_yisp] CMPA [opara_r6_yisp]	(A) - (M) Compare Accumulator A with Memory	IMM DIR EXT IDX IDX1 IDX2 [D_IDX] [IDX2]	81 11 91 04 B1 hh 11 A1 xb A1 xb ff A1 xb 00 ff A1 xb 00 ff A1 xb 00 ff	0 rPC rPO rPC rPO rPCPP rPCrPC rPCrPC	9 rCP rCP rCP rCP rPCPP rPCrPC rPCrPC	---- Δ Δ Δ Δ	
CMPB #oprt6i CMPB opria CMPB oprt6a CMPB opara_yisp CMPB opara_yisp CMPB opara_r6_yisp CMPB [D_yisp] CMPB [opara_r6_yisp]	(B) - (M) Compare Accumulator B with Memory	IMM DIR EXT IDX IDX1 IDX2 [D_IDX] [IDX2]	C1 11 D1 04 F1 hh 11 B1 xb B1 xb ff B1 xb 00 ff B1 xb 00 ff B1 xb 00 ff	0 rPC rPO rPC rPO rPCPP rPCrPC rPCrPC	9 rCP rCP rCP rCP rPCPP rPCrPC rPCrPC	---- Δ Δ Δ Δ	
COM opria COM opara_yisp COM opara_yisp COM opara_r6_yisp COM [D_yisp] COM [opara_r6_yisp] COMA COMB	(M) ⇒ M equivalent to \$FF - (M) ⇒ M 1's Complement Memory Location  (A) ⇒ A Complement Accumulator A (B) ⇒ B Complement Accumulator B	EXT IDX IDX1 IDX2 [D_IDX] [IDX2] IMM IMM	71 hh 11 61 xb 61 xb ff 61 xb 00 ff 61 xb 00 ff 61 xb 00 ff 41 51	rPO rPW rPO rPCPP rPCrPW rPCrPW 0 0	rOPM rPM rPO rPCPP rPCrPW rPCrPW 0 0	---- Δ Δ 0 1	
CPD #oprt6i CPD opria CPD oprt6a CPD opara_yisp CPD opara_yisp CPD opara_r6_yisp CPD [D_yisp] CPD [opara_r6_yisp]	(A-B) - (M+1) Compare D to Memory (16-Bit)	IMM DIR EXT IDX IDX1 IDX2 [D_IDX] [IDX2]	B0 JJ kk 9C 04 B0 hh 11 AC xb AC xb ff AC xb 00 ff AC xb 00 ff AC xb 00 ff	00 rPC rPO rPC rPO rPCPP rPCrPC rPCrPC	00 rCP rCP rCP rCP rPCPP rPCrPC rPCrPC	---- Δ Δ Δ Δ	
CPS #oprt6i CPS opria CPS oprt6a CPS opara_yisp CPS opara_yisp CPS opara_r6_yisp CPS [D_yisp] CPS [opara_r6_yisp]	(SP) - (M+1) Compare SP to Memory (16-Bit)	IMM DIR EXT IDX IDX1 IDX2 [D_IDX] [IDX2]	B0 JJ kk 9F 04 B0 hh 11 AP xb AP xb ff AP xb 00 ff AP xb 00 ff AP xb 00 ff	00 rPC rPO rPC rPO rPCPP rPCrPC rPCrPC	00 rCP rCP rCP rCP rPCPP rPCrPC rPCrPC	---- Δ Δ Δ Δ	
CPX #oprt6i CPX opria CPX oprt6a CPX opara_yisp CPX opara_yisp CPX opara_r6_yisp CPX [D_yisp] CPX [opara_r6_yisp]	(X) - (M+1) Compare X to Memory (16-Bit)	IMM DIR EXT IDX IDX1 IDX2 [D_IDX] [IDX2]	B0 JJ kk 9E 04 B0 hh 11 AX xb AX xb ff AX xb 00 ff AX xb 00 ff AX xb 00 ff	00 rPC rPO rPC rPO rPCPP rPCrPC rPCrPC	00 rCP rCP rCP rCP rPCPP rPCrPC rPCrPC	---- Δ Δ Δ Δ	
CPY #oprt6i CPY opria CPY oprt6a CPY opara_yisp CPY opara_yisp CPY opara_r6_yisp CPY [D_yisp] CPY [opara_r6_yisp]	(Y) - (M+1) Compare Y to Memory (16-Bit)	IMM DIR EXT IDX IDX1 IDX2 [D_IDX] [IDX2]	B0 JJ kk 9D 04 B0 hh 11 AY xb AY xb ff AY xb 00 ff AY xb 00 ff AY xb 00 ff	00 rPC rPO rPC rPO rPCPP rPCrPC rPCrPC	00 rCP rCP rCP rCP rPCPP rPCrPC rPCrPC	---- Δ Δ Δ Δ	
DAA	Adjust Sum to BCD Decimal Adjust Accumulator A	IMM	18 07	000	000	----	Δ Δ Δ Δ
DBEQ abcdrys_rsl6	(ctrl) - 1 ⇒ ctrl if (ctrl) = 0, then Branch else Continue to next instruction  Decrement Counter and Branch if = 0 (ctrl = A, B, D, X, Y, or SP)	REL (9-bit)	04 1b rr	PPP (branch) PPO (no branch)	PPP	----	----



---

**BIBLIOGRAFÍA****Libros**

Brian W. Kernighan, Dennis M. Ritchie. El lenguaje de programación C. Prentice Hall. 1991. 294 paginas.

José Felipe Ramírez. Aprenda Visual Basic Practicando. Prentice Hall. 2001. 820 paginas.

Nathan Gurewich, Ori Gurewich. Visual Basic 5 en 21 días. Prentice Hall 1997. 798 paginas.

M. Morris Mano. Diseño Digital. Prentice Hall 1987. 491 paginas

Robert L. Boylestad, Louis Nashelsky. Electrónica: Teoría de Circuitos. . Prentice Hall 1997. 948 paginas.

Secretaria de Obras Publicas, Documentos para la historia de las carreteras en México, 1965.

Torres Portero Manuel. Diseño e Ingeniería Electrónica asistida sobre Protel. RAMA 2000, 287 paginas.

Sutaner Hands. Fabricación de circuitos impresos. Marcombo 1974. 128 paginas.

**Internet**

- Secretaria de Comunicaciones y Transportes. [www.sct.gob.mx](http://www.sct.gob.mx)
- Caminos y Puentes Federales, CAPUFE [www.capufe.gob.mx](http://www.capufe.gob.mx)
- Instituto Nacional de Estadística Geografía e Informática. INEGI [www.inegi.gob.mx](http://www.inegi.gob.mx)
- Motorola [www.motorola.com](http://www.motorola.com)
- Microcontrolador 68HC12 [www.cyc.dfis.ull.es/asignaturas/Leyfa/68hc12/](http://www.cyc.dfis.ull.es/asignaturas/Leyfa/68hc12/)
- Simulador del 68HC12 <http://almy.us/68hc12.html>
- ICC12 <http://www.imagecraft.com/software/mdevtools.html>
- Electronics Workbench  
[http://www.softinteligente.com.ar/productos/electronicworkbench/soporte\\_EWB.htm](http://www.softinteligente.com.ar/productos/electronicworkbench/soporte_EWB.htm)
- Tutorial de Protel  
[http://www.uhu.es/manuel\\_sanchez/tutoriales/protel99/protel99.html](http://www.uhu.es/manuel_sanchez/tutoriales/protel99/protel99.html)
- Manual de puerto serial en Visual Basic.  
<http://www.abcdatos.com/tutoriales/tutorial/g205.html>
- Remolques [www.thalman.net](http://www.thalman.net) , [www.altamirano.com.mx](http://www.altamirano.com.mx)
- Neumáticos [www.firestone.com.mx](http://www.firestone.com.mx)
- Dimensiones de autos Volkswagen [www.vw.com.mx](http://www.vw.com.mx)
- Dimensiones de autos Nissan [www.nissan.com.mx](http://www.nissan.com.mx)
- Dimensiones de autos Dodge [www.dodge.com.mx](http://www.dodge.com.mx)
- Lenguaje C <http://www.iespana.es/mundolinux/progr/manc/lengc.htm>
- Programa Nacional de carreteras <http://www.sct.gob.mx/acuota/punto1.htm>

**Otras fuentes de consulta.**

- Autopista de cuota México-Cuernavaca
- Autopista de cuota México-Querétaro
- Autopista de Cuota México-Puebla

Evaluation Board User Manual M68HC12B32EVB, Motorola.

M68HC12 Microcontroller M68ch12B Family Data Sheet

CPU68HC12 y Periféricos, Agosto 2001

Compilador cruzado de C, Imagecraft, Microrobotica.