

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA,

AUTOMATIZACIÓN Y CONTROL

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL

TÍTULO EN INGENIERÍA ELECTRÓNICA

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
PRECLASIFICADO Y AUDITORÍA PARA EL CONTROL**

VEHICULAR DE UNA ESTACIÓN DE PEAJE”

WILSON FERNANDO CHILUISA ESPÍN

SANGOLQUI – ECUADOR

2008

CERTIFICACIÓN

Por medio de la presente certificamos que el proyecto de grado para la obtención del título en Ingeniería Electrónica titulado **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE PRECLASIFICADO Y AUDITORÍA PARA EL CONTROL VEHICULAR DE UNA ESTACIÓN DE PEAJE”** fue desarrollado en su totalidad por el señor WILSON FERNANDO CHILUISA ESPÍN.

Atentamente,

Ing. Rodolfo Gordillo
DIRECTOR

Ing. Wilson Yépez
CODIRECTOR

AGRADECIMIENTO

A Dios por brindarme cada día de salud y alegría, y a mis padres por ser el pilar fundamental de mi vida y estar siempre presentes en todos los momentos.

Al Ing. Marco Zurita y Manuel Zurita gerentes de MZ Sistemas Eléctricos y Electrónicos quienes hicieron posible el desarrollo de este proyecto, y a los trabajadores quienes colaboraron incondicionalmente para sacar adelante el mismo.

Al director y codirector de tesis quienes apoyaron y fueron respaldo constante para el desarrollo del mismo.

Especial agradecimiento a mi incondicional amigo Fernando Carrillo quien se mostro siempre presto al desarrollo del presente trabajo.

A Paola querida amiga y compañera en estos años de mi vida.

A todos mis buenos amigos, a mis tios Fausto y Julio y todas las personas que no han sido nombradas y que forman parte importante de mi vida. Gracias por el apoyo constante.

DEDICATORIA

Este trabajo está dedicado a las personas más importantes de mi vida, mis padres en agradecimiento a su esfuerzo, cariño y sacrificio, de quienes me llevo el mejor ejemplo de esfuerzo y superación. A mis hermanos por su solidaridad y apoyo en todos los buenos y malos momentos.

PRÓLOGO

Actualmente sistemas de peaje tradicionales utilizan sistemas de posclasificación, en los que el operario es el que decide la clase a la que pertenece el vehículo y la tarifa a cancelarse. Estos sistemas presentan muchos inconvenientes, porque existe evasión y falta de control sobre la recaudación, además está expuesto a fallas de tipo humano.

De este modo, el presente proyecto ofrece una solución general, a partir del desarrollo de un sistema de preclasificado y auditoría para el control vehicular de una estación de peaje, cuyo diseño se basa en la utilización de software versátil y la utilización de dispositivos que brindan un mínimo impacto ambiental.

El sistema de preclasificación automática propuesto tiene como alcance la implementación de un prototipo electrónico para automatizar el proceso de detección de vehículos en una estación de peaje.

Ofrece una solución operativa y funcional ante el sistema posclasificador. Como características el sistema permitirá la auditoría permanente como un medio para lograr la aclaración de cualquier evento que ocurriera en la operación diaria. El diseño responde ampliamente a las necesidades de control vehicular dándole mayor confiabilidad al sistema haciéndolo más robusto, permitiendo un margen de error mínimo, para poder utilizar un método de pago rápido.

Así, se pudo realizar un sistema fiable, altamente flexible, con capacidad de expansión, bajo el cumplimiento de normas técnicas, constituido de dispositivos que ofrecen un gran desempeño, alto rendimiento y larga vida útil. La implementación del sistema requiere de un costo de inversión mínimo y reduce gastos correspondientes al consumo de energía eléctrica.

ÍNDICE DE CONTENIDO

CAPITULO I – FUNDAMENTACIÓN TEÓRICA

1.1	Introducción	1
1.2	Características de Funcionamiento del Sistema.....	2
1.3	Componentes del Sistema.....	3
1.4.	Descripción de los Elementos del Sistema	4
1.4.1	Elementos de Carril	4
1.4.1.1	Barrera Óptica	4
1.4.1.2	Tarjeta de Control Embebido Rabbit.....	5
1.4.1.3	Bandas Detectoras y Contadoras de Ejes.....	7
1.4.1.4	Loop (Lazo Magnético)	9
1.4.1.4	Barrera de Control Vehicular	11
1.4.1.5	Display de Usuario	12
1.4.1.6	Semáforo de Carril.....	13
1.4.2	Elementos de Cabina	14
1.4.3	Elementos de Video.....	14
1.4.4	Elementos de Comunicación.....	15
1.4.5	Elementos de Auditoria	15
1.5.	Condiciones Especiales del Sistema	16

CAPITULO II – SISTEMA CONTROLADOR

2.1	Introducción	18
2.2	Tarjeta Electrónica Rabbit.....	20
2.2.1	Características Generales	21
2.2.2	Características Eléctricas	21
2.2.3	Configuración Estandar	22

2.2.4 Ventajas	23
2.2.5 Características Mecánicas	23
2.2.6 Subsistemas Principales	24
2.2.7 Disposición de Puertos de Entrada y Salida Digitales.....	25
2.2.8 Disposición de Puertos y Memoria	25
2.2.9 Descripción de Recursos.....	26
2.2.9.1 Buses Internos y Externos.....	26
2.2.9.2 Comunicación Serial.....	27
2.2.9.3 Puertos Seriales	27
2.2.9.4 Puerto de Programación	28
2.2.9.5 Memoria.....	28
2.2.9.6 Sistema de Reloj.....	29
2.2.9.7 Entrada de Oscilador de 32.768 Khz	29
2.2.9.8 Generador Rampa.....	29
2.3 Protocolos de Comunicación	30
2.3.1 TCP/IP	31
2.3.1.1 Capas.....	32
2.3.2 Ethernet.....	34
2.3.2.1 Formato de la Trama de Ethernet	35
2.3.3 Estandar RJ-45	38
2.4 Comunicación Serial	39
2.4.1 Tipos de Comunicaciones Seriales.....	40
2.4.1.1 Simplex	40
2.4.1.2 Half Duplex	40
2.4.1.3 Full Duplex.....	40
2.4.2 RS-232	40
2.4.3 RS-422	43
2.4.4 RS-485	43
2.5 Comunicación con la Tarjeta Electrónica Rabbit	43
2.5.1 Red Ethernet.....	44
2.5.2 Socket TCP	45
2.5.3 Apertura del Socket.....	45
2.5.4 Funciones TCP Socket.....	46
2.5.5 Comunicación Serial	47

2.6 Software.....	47
-------------------	----

CAPITULO III – DISEÑO DEL SISTEMA DE PRECLASIFICACIÓN VEHICULAR Y AUDITORIA

3.1 Diseño Integral del Sistema	49
3.2 Disposición de los Elementos Sensores	49
3.3 Disposición de la Cabina de Cobro y Barrera de Acceso Y/O Salida	54
3.4 Disposición de Señales Luminosas y Visualizador de la Tarifa.....	56
3.5 Diseño del Controlador	59
3.5.1 Programación del Controlador	60
3.5.1.1 Consideraciones y Dimensionamiento.....	60
3.5.2 Dynamic C	61
3.5.3 Diagrama de Bloques.....	62
3.5.4 Diagrama de Flujo	63
3.5.5 Programación de la Tarjeta	65
3.5.5.1 Análisis del Número de Ejes de un Vehículo.....	67
3.5.5.2 Conexión TCP/IP.....	70
3.6 Diseño de la Interfaz Gráfica	72
3.6.1 Consideraciones	73
3.6.2 Base de Datos.....	73
3.6.2.1 Diseño de la Base de Datos	73
3.7 Diseño HMI	76
3.7.1 Menú Principal.....	77
a) Peaje	78
b) Procesos.....	82
c) Reportes	84
d) Personal	86
3.7.2 Desarrollo del Software	87
a) Ingreso de Usuarios.....	89
b) Operación	90
c) Envío de Datos al Controlador.....	93
d) Almacenamiento de Datos.....	94

CAPITULO IV – INSTALACIÓN DEL SISTEMA

4.1 Descripción	96
4.2 Sistema Electrico.....	96
4.2.1 Red Eléctrica y Fuente de Alimentación	96
4.2.2 Sistema de Protección.....	97
4.3 Integración Controlador – Elementos de Detección	98
4.4 Pruebas y Resultados.....	99
4.5 Procedimiento	100
4.5.1 Circuito de Adaptación de Señales.....	100
4.5.2 Lógica de Programación del Controlador.....	101
4.5.3 Conexión de la Tarjeta Rabbit con el Display	101
4.5.4 Conexión de la Tarjeta a la Interfaz Gráfica	101
4.5.5 Captación Automática de Imágenes	102
4.5.6 Almacenamiento de Datos a la Base	102
4.5.7 Integración del Controlador con Elementos de Detección	102

CAPITULO V – ANÁLISIS ECONÓMICO

5.1 Introducción.....	103
5.2 Inversión Inicial	104

CAPITULO VI - CONCLUSIONES Y RECOMENDACIONES

6.1 Introducción.....	108
6.2 Conclusiones.....	108
6.3 Recomendaciones.....	110

REFERENCIAS BIBLIOGRÁFICAS	111
---	------------

ANEXOS	114
---------------------	------------

CAPITULO I

FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN

Habitualmente se asocia el concepto de peaje a la tasa que se cobra a los medios de transporte por utilizar infraestructuras de comunicación vial, por ejemplo, automóviles para poder circular por autopistas que permiten un viaje mucho más rápido que por las carreteras normales.

El peaje se utiliza para financiar la construcción de infraestructuras de transporte como carreteras o puentes, sin utilizar dinero del Estado que ha sido recaudado como impuestos, de modo que solamente pagan aquellos que los usan y no el resto de los contribuyentes.

Normalmente el peaje se paga en las estaciones ubicadas en la autopista que se esté utilizando. Actualmente sistemas de peaje tradicionales utilizan sistemas de posclasificación, en los que el operario es el que decide la clase a la que pertenece el vehículo y la tarifa a cancelarse, y el equipo opera solo como un verificador que puede ser burlado muy fácilmente.

La mayor parte de sistemas de posclasificación presentan muchos inconvenientes, porque existe evasión y falta de control sobre la recaudación, debido a que el operador es el que decide la clase a la que pertenece el vehículo y está expuesto a fallas de tipo humano.

Al observarse las deficiencias que presenta el sistema posclasificador donde es el cajero quien determina la clase a la que pertenece el vehículo y la tarifa que está pagando, se admite una serie de vicios y problemas operativos.

El sistema de preclasificación automática propuesto ofrece una solución operativa y funcional ante el sistema posclasificador, ya que supera con mucho a los sistemas tradicionales que en muchos casos son altamente ineficientes.

Este prototipo electrónico tiene como alcance automatizar el proceso de detección de vehículos en una estación de peaje, como característica el sistema permitirá la auditoría permanente y el uso de sistemas de videograbación, como un medio para lograr la aclaración de cualquier evento que ocurriera en la operación diaria. El diseño responde ampliamente a las necesidades de control vehicular, permitiendo que sea el sistema quien clasifique en forma automática los vehículos, para poder utilizar un método de pago rápido que depende de saber con precisión la tarifa a pagar conjuntamente con el funcionamiento de todas las señales luminosas. Además, ofrece el desarrollo de un software aplicativo creando una herramienta de trabajo altamente eficaz que permitirá hacer mucho más eficiente su operación.

En colaboración con la empresa MZ Sistemas Eléctricos & Electrónicos se proyecta que este prototipo sea utilizado con fines comerciales en los sistemas de peaje que trabajan con la empresa, luego de realizar las pruebas y corrección de errores que se presentaren en su desarrollo.

1.2 CARACTERÍSTICAS DE FUNCIONAMIENTO DEL SISTEMA

Este proyecto tiene como alcance la implementación de un prototipo electrónico para automatizar el proceso de detección de vehículos en una estación de peaje. La detección será realizada mediante la discriminación del número de ejes de los automotores a través de bandas detectoras y clasificadoras de ejes “peanas”, sensores ópticos e inductivos acoplados a un controlador. Para efectos de éste último se utilizará una tarjeta electrónica de control embebido, la cual permite clasificar el tipo de vehículo según el número de ejes a través de las señales que recibe de los sensores, por medio de un programa local.

Además a través de esta tarjeta se realizará la comunicación con la PC que se encuentra en la cabina y enviará la información actualizada, requerida por el software para poder utilizar un sistema de pago rápido que depende de conocer con precisión la tarifa a pagar para la recaudación eficiente del operador. La información será almacenada en la base de datos para control de auditoría.

El software utilizado está caracterizado por un interfaz gráfica de usuario que servirá para incrementar el nivel de automatización del sistema permitiendo, entre otras posibilidades, el manejo de tarifas, la generación de reportes, la validación de usuarios y la interconexión con el sistema de grabación de video para la captación automática de imágenes de los vehículos en la estación de cobro con el fin de controlar mediante el sistema auditor, todos los eventos que ocurren en el carril y en forma particular aquellos en que hay descuento en la tarifa o el vehículo no paga [1].

1.3 COMPONENTES DEL SISTEMA

Los componentes del sistema se dividen en tres equipos los cuales se mencionan a continuación:

Ü Equipo de preclasificado

Este equipo realiza una clasificación inicial del vehículo de manera automática, esto se consigue al pasar el auto por los sistemas detectores que componen el sistema preclasificador y estos son:

- Separador Vehicular.
- Dispositivo clasificador y contador de ejes.

A continuación se realiza una confirmación de pago por parte del cajero receptor el cual se encargará de emitir los comprobantes de pago para el usuario, además el sistema preclasificador emite la liquidación de cada turno de trabajo y compara lo marcado por el cobrador con lo que detecto el equipo.

Ü Equipo de auditoría permanente

El equipo auditor es el encargado de controlar todos los eventos que ocurren en el carril y en forma particular aquellos en que hay descuento en la tarifa o el vehículo está exento de pago. Además está diseñado para emitir un reporte sumario por turno de lo registrado con el fin de realizar cierres de turno ágiles y eficientes, con una inversión mínima de tiempo.

Ü Equipo de videgrabación

Cuando el vehículo abandona la zona de cobro, es detectado por un sistema de videgrabación que permite visualizar las características generales del vehículo para la captación automática de imágenes de los vehículos en la estación de cobro, con el fin de confirmar la clasificación realizada por el equipo de preclasificación y la confirmación del cajero receptor a fin de verificar si existió algún error del cobrador o del equipo.

1.4. DESCRIPCIÓN DE LOS ELEMENTOS DEL SISTEMA

El sistema automatizado de peaje con video, está integrado por elementos mecánicos y electrónicos y por sistemas de funcionamiento y procesamiento de información, como se indica a continuación.

1.4.1 ELEMENTOS DE CARRIL

1.4.1.1 BARRERA ÓPTICA

Es un separador vehicular, su función es crucial para el funcionamiento del sistema de preclasificación, debido a que permite detectar y dividir cada uno de los vehículos que ingresan al carril. Este dispositivo mejora la precisión del sistema en forma muy notable, está diseñada para no ser interferida por lluvia, neblina, polvo, humo o insectos nocturnos [4].



Figura 1.1 Fotocélulas

Característica Técnicas:

- Ü La fotocélula no requiere alineamiento
- Ü Frecuencia infrarrojo: 1000Hz
- Ü Alcance: Se garantizan 18m con cualquier condición atmosférica
- Ü Alimentación: 12/24 VAC
- Ü Alcance contactos relés: 1A máximo a 24V
- Ü Temperatura de funcionamiento: de -20° a + 70°C
- Ü Absorbencia: 60mA
- Ü Dimensiones DOC I: 70 x 70 mm (parte empotrable Ø60 mm; profundidad 75mm)
- Ü Dimensiones DOC E: 70 x 70 mm; profundidad 34mm
- Ü Materiales de las cajas: nylon reforzado con vidrio

1.4.1.2 TARJETA DE CONTROL EMBEBIDO RABBIT

La función de la tarjeta electrónica de control embebido Rabbit, es discriminar el paso de ejes de rodada sencilla o rodada doble de los vehículos automotores. El complemento de esta tarjeta son las bandas detectoras y clasificadoras de ejes “peanas”, y los sensores ópticos los cuales se encuentran conectados a esta tarjeta.

La operación de sensado se efectúa en el momento en que el vehículo de rodada doble o sencilla pasa sobre las bandas detectoras generando ciertas señales que son interpretadas por la tarjeta de control embebido Rabbit. Esta información es captada y procesada por la tarjeta, y clasifica dichos vehículos dependiendo del tipo de rodada. De esta forma se lleva un estricto control de cobro, en función del número de ejes [8].

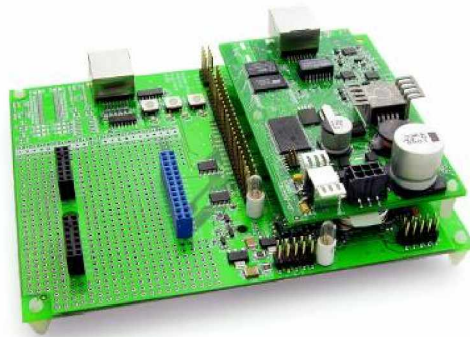


Figura 1.2 Módulo PowerCore Flex 3800 colocada sobre la placa.

Característica Técnicas:

- ü 39 líneas de entrada/salida de propósito general configurables.
- ü 3 entradas digitales adicionales.
- ü 2 salidas digitales adicionales.
- ü 5 puertos seriales, 3.3V CMOS compatible, con una tasa asíncrona superior a 6.45 Mbps.
- ü 3 puertos son configurables como puerto serial de reloj (SPI), 2 puertos como puerto serial HDLC, y 1 puerto como puerto serial SDLC.
- ü Uno de los puertos seriales está destinado a la programación.
- ü 512K de memoria flash para almacenamiento de instrucciones.
- ü 256K de memoria RAM estática para datos.
- ü Batería de respaldo.

Ü Alimentación:

Ü DC: no regulada de 8 a 43V DC (potencia 13.3W)

Ü AC: 24-60V AC con transformador con tap central (potencia 13.3W)

12-36V AC con transformador estándar (potencia 13.3W)

Corriente límite para placa a 5V DC regulados: 2A.

Corriente promedio para circuitos en placa: 400mA.

Ü Salidas:

Ü AC/DC: +3.45V DC: $I_{3VoutM} = 350mA$.

+5V DC: $I_{5VoutM} = [1600mA - I_{3VoutM}]$

No regulada AC/DC:

$I_{unregM} = 1600mA - (I_{5VoutA} + I_{3VoutA}) \times (6.7V/V_{in})$

Ü Temperatura de operación: $-40^{\circ}C$ a $+70^{\circ}C$

1.4.1.3 BANDAS DETECTORAS Y CONTADORAS DE EJES

La función de los sensores de contacto es detectar, contar y clasificar vehículos. Estos sensores son aptos para ser instalados en sistemas de preclasificación y posclasificación en la industria del peaje.

Según la disposición de los sensores, éstos pueden detectar el número de ejes, número de ruedas, ruedas duales y velocidad del vehículo. Estos sensores trabajan en combinación con dispositivos integrados en una placa de control.

La detección será realizada mediante la discriminación del número de ejes de los automotores a través de bandas detectoras y clasificadoras de ejes “peanas”. El principio de funcionamiento de los sensores de contacto (peanas) es el de un switch electrónico accionado mecánicamente por presión dirigida, y están dentro de las normas para trabajar con optoacopladores o microrelay [7].

Constructivamente el sensor está formado por dos láminas de acero armónico, electrodos normalmente separados, vulcanizados en el interior de un material de deformación elástica controlada [7].

Cuando un cuerpo en movimiento, en este caso la rueda de un vehículo, transita sobre un sensor, el mismo es elásticamente deformado, produciendo el cierre del circuito contador al hacer contacto ambos electrodos, una vez que la rueda ha dejado de hacer presión, el circuito eléctrico se abre por reacción mecánica de los laterales de goma tal como se muestra en la figura.

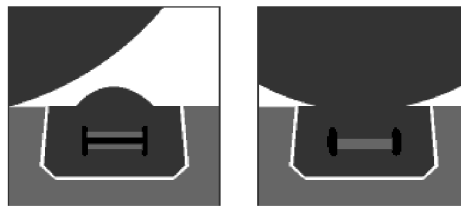


Figura 1.3 Funcionamiento del Sensor de Contacto

Consecuentemente, la resistencia eléctrica medida entre los electrodos varía de acuerdo al estado mecánico del sensor, abierto ó cerrado si está accionado por el paso de un vehículo o no; estas señales son interpretadas por el controlador para la preclasificación [7]. A continuación se muestra el perfil del sensor y sus partes.

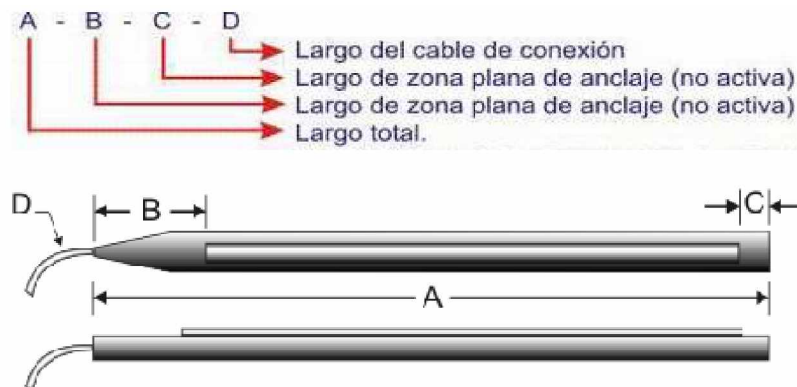


Figura 1.4 Banda Detectora de Ejes.

Característica Técnicas:

- Ü Velocidad: 0 -150Km/h
- Ü Temperatura: -40 a +65°C
- Ü Esfuerzo de Activación: >25 Kg
- Ü Tiempo de activación: <10 μ s
- Ü Tiempo de desactivación: <10 μ s
- Ü Peso: 1,33 Kg/m
- Ü Color: Negro
- Ü Largo del cable de conexión 5m
- Ü Resistencia eléctrica del sensor inactivo (*) >20m Ω
- Ü Resistencia eléctrica del sensor activo (*) <3,5 Ω
- Ü Tensión máxima 500 Volt

1.4.1.4 LOOP (LAZO MAGNÉTICO)

Este dispositivo se denomina (loop) o espira, montado en el piso forma un lazo inductivo y detecta la presencia de vehículos por su masa metálica a través de la medición de la inductancia de la espira detectora.

El principio de funcionamiento del equipo consiste en la detección de una masa metálica por medio de la medición de la inductancia de una espira detectora. La espira es parte de un circuito oscilador que genera un campo magnético. Al ser atravesado por el vehículo se produce un cambio en la frecuencia del oscilador. Esta variación es detectada por un circuito con microprocesador que determina, en base a la configuración del equipo, si debe producir o no la señal de salida [5].

Este dispositivo está diseñado para detectar la presencia de vehículos en la zona de cobro, esto permite que mientras el vehículo se encuentra pasando la barrera de acceso esta no caiga ni cause ningún daño al auto.



Figura 1.5 Detector de Masa Metálica (loop).

Características técnicas:

- Ü Tecnología: Campo magnético
- Ü Ajuste de frecuencia: automática
- Ü Modo de detección: presencia
- Ü Tiempo de presencia: 1 min. A lo infinito (presencia permanente) en 250 pasos
- Ü Duración de la impulsión de salida: 100ms o 500 ms
- Ü Gama de inductancia: 20uH a 1000uH
- Ü Gama de frecuencia: 20 khz a 130 khz
- Ü Opciones de frecuencia: 4 para el campo simple, 2 para el campo doble por canal
- Ü Sensibilidad: 0.005% a 0.5% en 250 pasos
- Ü Tiempo de reacción: 25ms para el campo simple, 50 ms para el campo doble (por canal)
- Ü Salto automático de sensibilidad.

- Û Tiempo de ajuste a la conexión: 8s máximo por canal
- Û Tensión de alimentación: 12/24 AC/DC
- Û Consumo: <2.5W
- Û Temperatura de operación: -30° a +70°C
- Û Peso: <200g
- Û 2 salidas a relé o optoaisladas
- Û Conexión: conector redondo estándar de 11 conectores de tipo 86 CP11
- Û Protecciones: Transformador de aislamiento de canales, diodos zener, descargador de gas contra las sobretensiones
- Û Dimensiones: 77mm (Alto) x 40mm (Ancho) x 75m (P)

1.4.1.4 BARRERA DE CONTROL VEHICULAR

Esta barrera está diseñada para trabajar 24 horas/días y 365 días/año, bajo las condiciones de trabajo más duras y adversas. Sus características de fabricación la hacen ideal para trabajar indistintamente a la intemperie o en recintos cerrados [3].



Figura 1.6 Barrera de Acceso.

Características Técnicas:

- Û Tensión nominal: 120 VAC

- Û Frecuencia: 60 Hz
- Û Corriente consumida 3 A
- Û Potencia consumida: 200 W
- Û Control interno: PLC SIEMENS LOGO
- Û Capacitor: 33 Uf
- Û Velocidad apertura/cierre: 0.6 segundos
- Û Longitud mástil: 4m
- Û Pintura: Polyester
- Û Grado de protección: IP-44
- Û Peso barrera: 62 kg
- Û Dimensiones del gabinete: 365 x 992 x 225[mm]
- Û Temperatura de funcionamiento: -20 a +55°C
- Û Termoprotección: 140°C
- Û Fin de carrera incorporado: si
- Û Velocidad de rotación del motor: 1400 RPM
- Û Ventilación forzada: rejilla (opcional)
- Û Medida para 50 Hz
- Û Resistencia Mecánica (con mantenimiento normal): 5000000 ciclos

1.4.1.5 DISPLAY DE USUARIO

Es un dispositivo electrónico permite la visualización de la tarifa a pagar a los usuarios de la autopista. Puede ser visto a cualquier hora del día y en circunstancias de escasa visibilidad.

Características Técnicas:

- Û Comunicación RS-485.
- Û Alta luminosidad.
- Û Visibilidad garantizada a 50[m] a plena luz del día.
- Û Tamaño de cada dígito: 12[cm] x 5[cm]
- Û Voltaje de alimentación: 24[V].
- Û Potencia máxima: 9[W].
- Û Color: blanco



Figura 1.7 Display del Usuario.

1.4.1.6 SEMÁFORO DE CARRIL

Dispositivo eléctrico que se utiliza para ordenar y regular el tránsito de vehículos y peatones por un carril de cobro utilizando colores rojo y verde.



Figura 1.8 Semáforo de Carril.

1.4.2 ELEMENTOS DE CABINA

Este gabinete contiene el servidor de la estación. La base de este sistema es una PC con procesador de tipo Intel Pentium IV, con 521 MB de memoria RAM y un disco duro de 40 GB mínimo. Con lo que se garantiza, que la capacidad de proceso y almacenamiento, no será en ninguna forma rebasada y no se tendrá problemas de manejo o pérdida de información por un periodo mínimo de un año. Aquí también están alojados el sistema de detección de presencia vehicular, las fuentes de alimentación de todos los elementos de sensado, el control del teclado para operación en modo manual, la impresora de comprobantes de pago y el monitor el cuál es de tipo VGA de 14" a color que es visible en cualquier condición de iluminación y servirá al peajista para verificar en todo momento el funcionamiento global del sistema [2].

La impresora de comprobantes es de tipo matriz de punto, lo que permite imprimir un comprobante cada cuatro segundos, aunque se puede mejorar este tiempo usando sistemas pre impresos [2].

El sistema opera con sus propias fuentes de alimentación y cuenta también con su propio respaldo de energía eléctrica (UPS), de tal forma que en casos de fallas graves o interrupción prolongada de energía se mantiene funcionando.

El gabinete cuenta además con chapas de seguridad para restringir el acceso a personal de operación.

1.4.3 ELEMENTOS DE VIDEO

El sistema incluye una cámara con visión nocturna, a color de foco fijo y con autoiris y su respectivo portacámara, lo que permite tomar registros en cualquier condición de luz, gracias a esto se permitirá tener siempre a la mano una fotografía del vehículo en buenas condiciones y evitar errores por inconsistencia.

Para que la revisión de las fotos pueda ser efectiva, cada uno de los eventos registrados se guarda en una base de datos la cuál archiva los eventos con hora, fecha, si la clasificación fue automática o manual y el tipo de tarifa que el auto cancelo.

1.4.4 ELEMENTOS DE COMUNICACIÓN

El sistema mediante el cual existe la comunicación entre el carril y la máquina de operación en la caseta de peaje, está basado en la integración de una red de tipo local, con un protocolo de comunicaciones estándar tipo TCP/IP usando puertos de comunicaciones y cableado tipo par trenzado, y salida de video desde la PC para el sistema de videograbación. Toda la información del carril desde la tarjeta de control embebido hacia la PC ubicada en la caseta de operación es verificada a fin de que cada uno de los eventos llegue completo y si no es así se vuelve a solicitar el envío de la información correspondiente al carril, hasta que se logre su recuperación total. En esta forma la posibilidad de pérdida de eventos es muy reducida.

1.4.5 ELEMENTOS DE AUDITORIA

La base de los elementos de administración y auditoría es la PC con procesador de tipo Intel Pentium IV, con 521 MB de memoria RAM y un disco duro de 40 GB mínimo. Estas especificaciones pueden variar en función de los equipos disponibles en el mercado, pero siempre serán equipos de la más reciente tecnología. Como complemento el sistema de auditoría cuenta con una impresora color con sistema de inyección de tinta, para la impresión de reportes.

El equipo de administración y auditoría será instalado en la caseta de operación del peaje para ser operado directamente por el jefe de turno, para realiza comprobaciones con las imágenes y con ellas proceder a efectuar los ajustes necesarios; tendrá las siguientes funciones:

- Ü Entrada al sistema (con permisos de acceso y restringido para operadores)
- Ü Acceso al menú
- Ü Configuración del sistema
- Ü Usuarios

- Ü Tarifas
- Ü Cambio de contraseñas
- Ü Salida del sistema
- Ü Preliquidación de turno
- Ü Reportes por fecha
- Ü Impresión de reportes

1.5. CONDICIONES ESPECIALES DEL SISTEMA

Una de las mayores preocupaciones es que no hay una experiencia previa al grado de precisión del sistema en su conjunto es por eso que se emiten los siguientes comentarios:

El sistema no opera como el método de posclasificación, en el cual el peajista es quien decide, la clase vehicular que va a cobrar y en la que el equipo en el caso de tener deficiencias de funcionamiento no podrá en forma alguna verificar. En el sistema preclasificador obligará siempre al peajista a asignar una forma de pago que podrá ser auditada en todo momento y que en caso de que el sistema se equivoque o cometa algún error, lo obligará a efectuar una función de reclasificación, la cual se guardara en la base de datos para la auditoría correspondiente [1].

El sistema está diseñado para clasificar en forma automática el mayor número posible de los vehículos existentes comercialmente en el mercado, pero es prácticamente imposible cubrir todas las posibilidades, porque siempre habrá algunos que fueron modificados en sus características originales por sus propietarios u que son de diseño y construcción especiales, como es el caso de transportes de maquinaria, unidades utilizadas en la agricultura, remolques, etc. Entonces los posibles errores de interpretación del sistema obligarán al peajista a utilizar el mecanismo de reclasificación y el error quedará eliminado [2].

Los componentes detectores del sistema han sido desarrollados con la más alta tecnología y con la calidad necesaria, para un rendimiento óptimo, sin embargo existen algunos defectos en el diseño de los accesos a las estaciones de peaje, que pueden afectar su eficiencia, están a una distancia considerable de la cabina de cobro y en ciertas circunstancias el vehículo que ya ha sido preclasificado puede efectuar maniobras de retorno dentro de la zona de detección que afectan al sistema automático.

Un claro ejemplo de esto es cuando los vehículos efectúan maniobras de avance y retroceso sobre los elementos de detección provocando la emisión de ejes inexistentes o vehículos “fantasmas”, que no existen, pero se han generado por problemas de índole operativo.

Esto se debe a que un conductor puede traer un auto estándar y cuando hace el cambio de velocidad lento, el auto por la inclinación de la autopista se tiende a regresar. Otras veces se debe al exceso de tránsito, puesto que los vehículos van sumamente pegados y por no pegarse tanto llegan a retroceder.

El sistema también puede fallar en el momento en que uno de los elementos de detección es interferido por elementos extraños al vehículo, como por ejemplo, cuando vendedores o personas se cruzan a través de la barrera óptica o pisan los sensores de detección y presencia vehicular.

En todos los casos anteriores, existirá un vehículo que es de una tarifa superior al vehículo que se tiene que cobrar, y en este caso se obliga al peajista a reclasificar, porque no recibirá en modo alguno el importe correcto del vehículo.

CAPITULO II

SISTEMA CONTROLADOR

2.1 INTRODUCCIÓN

La automatización es un sistema donde se transfieren tareas de producción realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.

La innovación tecnológica de los últimos años ha permitido interactuar hardware y software a través de dispositivos electrónicos para realizar el control de procesos, comunicación, automatización, es así que cada dispositivo ofrece diferentes tipos de soluciones, optimizando recursos, brindando además alta confiabilidad y eficiencia.

Este dispositivo electrónico es usado para controlar. El control se realiza sobre la base de una lógica, definida a través de un programa, son conocidos como Controladores, los cuales a través de su lógica programable se encargan de recibir las señales eléctricas en sus entradas procesan la información y activan órdenes de salida que se envían hacia los correspondientes actuadores; es decir actúa como un intermediario entre los dispositivos de hardware y software, su finalidad es la de permitir extraer el máximo de las funcionalidades del dispositivo para el cual ha sido diseñado.

Actualmente existe gran diversidad de controladores, y es común encontrar más de un controlador para cada aplicación, cada uno ofreciendo un nivel distinto de funcionalidades, los cuales ofrecen beneficios y soluciones prácticas a sistemas electrónicos, los mismos que están en función de la aplicación a desarrollar.

Entre los autómatas programables más usados se encuentran, los Controladores Lógicos Programables (PLC), microcontroladores, tarjetas de adquisición, y las tarjetas electrónicas de control embebido, las cuales ofrecen amplias ventajas en el desarrollo de aplicaciones a un costo de inversión relativamente bajo.

Una vez analizados los diferentes tipos de controladores se realiza una rigurosa selección entre los dispositivos que ofrecen los recursos específicos para el desarrollo del proyecto, es necesario considerar el costo del producto en el mercado, además de tomar en cuenta parámetros tales como:

- Consumo de energía eléctrica.
- Niveles de tensión y corriente.
- Velocidad de respuesta.
- Número y tipo de entradas y salidas.
- Tipo de comunicación.
- Número de puertos.
- Capacidad de expansión.
- Tiempo de vida útil.
- Sistema de protección.
- Inmunidad al ruido e interferencias.

El presente proyecto requiere un controlador que permita discriminar el paso de ejes de rodada sencilla o rodada doble de vehículos automotores, configuración de puertos bidireccionales para señales digitales de entrada y salida, establecer transmisión de datos mediante Ethernet y comunicación serial, que sea inmune a interferencia y al ruido, que tenga una velocidad de respuesta inmediata y que incluya fuente de alimentación propia.

Para el diseño e implementación de este proyecto se ha determinado la utilización de una tarjeta electrónica de control embebido. El dispositivo de control seleccionado para el desarrollo de este proyecto es la Tarjeta Rabbit con módulo PowerCore Flex 3800, la cual posee múltiples recursos y las características necesarias para la presente aplicación.

2.2 TARJETA ELECTRÓNICA RABBIT

Esta tarjeta electrónica de control embebido es un dispositivo fácil de usar ofrece amplias ventajas y múltiples recursos; posee un módulo, PowerCore, que incluye un sistema microprocesador que permite trabajar en red, el módulo está diseñado para conectar la fuente de alimentación en la misma placa [1]. Consta de 50 pines correspondientes a entradas, salidas y a las fuentes de alimentación, además tiene soporte mecánico robusto.

El desarrollo de la programación se realiza bajo la plataforma del poderoso Dynamic C, el mismo que incluye extensas librerías que garantizan el rápido manejo de recursos del controlador y la creación de redes y aplicaciones específicas a partir de los programas base.

Los módulos se programan en Dynamic C y se descargan a través de puerto de comunicación RS-232, existe la posibilidad de hacerlo mediante el puerto USB con la utilización del conversor RS-232/USB, o directamente mediante el enlace Ethernet utilizando el manejador de descarga Dynamic C con el enlace Rabbit [1]. La figura 2.1 muestra al módulo PowerCore Flex 3800.

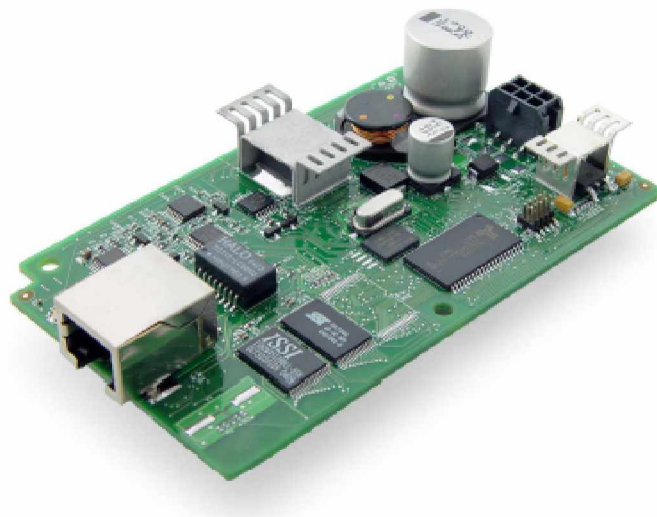


Figura 2.1. Módulo PowerCore Flex 3800

2.2.1 CARACTERÍSTICAS GENERALES

- Tamaño pequeño (60 mm × 102 mm × 28 mm)
- 39 líneas de entrada/salida (5V) de propósito general configurables.
- 3 entradas digitales adicionales.
- 2 salidas digitales adicionales.
- 5 puertos seriales, 3.3V CMOS compatible, con una tasa asíncrona superior a 6.45 Mbps.
- 3 puertos son configurables como puerto serial de reloj (SPI), 2 puertos como puerto serial HDLC y 1 puerto como puerto serial SDLC.
- Uno de los puertos seriales está normalmente destinado a la programación.
- 512K de memoria flash para almacenamiento de instrucciones.
- 256K de memoria RAM estática para datos.
- Procesador Rabbit 3000 corre a 25.8 MHz
- Conector de 50 pines de entrada y salida tipo bus
- Batería de respaldo.

2.2.2 CARACTERÍSTICAS ELÉCTRICAS

Alimentación

DC: no regulada de 8 a 43V DC (potencia 13.3W)

AC: 24-60V AC con transformador con derivación central (potencia 13.3W)

12-36V AC con transformador estándar (potencia 13.3W)

Corriente límite para placa a 5V DC regulados: 2A. [2]

Corriente promedio para circuitos en placa: 400mA. [2]

Salidas

AC/DC: +3.45V DC: $I_{3VoutM} = 350mA$.

+5V DC: $I_{5VoutM} = [1600mA - I_{3VoutM}]$

No regulada AC/DC:

$I_{unregM} = 1600mA - (I_{5VoutA} + I_{3VoutA}) \times (6.7V/V_{in})$

Temperatura de operación: -40°C a $+70^{\circ}\text{C}$
Humedad: 5% a 95% no condensado

Indicaciones:

- La fuente de alimentación de +3.45V DC tiene una tolerancia de $\pm 150\text{mA}$; la fuente de +5V de $\pm 250\text{mA}$.
- I_{3VoutM} = corriente máxima de salida disponible para el circuito del usuario correspondiente a la alimentación de +3.45V.
- I_{3VoutA} = corriente actual de salida utilizada por el circuito del usuario correspondiente a la alimentación de +3.45V.
- I_{5VoutM} = corriente máxima de salida disponible para el circuito del usuario correspondiente a la alimentación de +5V.
- I_{5VoutA} = corriente actual de salida utilizada por el circuito del usuario correspondiente a la alimentación de +5V.
- I_{unregM} = máxima corriente de salida AC/DC no regulada disponible para el circuito del usuario.
- Los 6.7 V constantes para el cálculo de corriente de I_{unregM} es derivado del 75% de la eficiencia de los 5V del regulador.

2.2.3 CONFIGURACIÓN ESTANDAR

CARACTERÍSTICA	MODULO POWER CORE 3800
Microprocesador	Rabbit 3000 corriendo a 51.6 MHz
Ethernet	10/100 compatible 10 Base-T interface
SRAM	512K Programación + 512 Datos
Memoria Flash (programación)	512K
Memoria Flash (almacenamiento de datos)	1 Mbyte (serial flash)
Limite de corriente para el regulador de 5V	2A

Tabla 2.1. Configuración Estándar.

2.2.4 VENTAJAS

- Rapidez de procesamiento
- Costo relativamente bajo
- Desarrollo de programación sencillo a través de lenguaje C depurado con Dynamic C
- Inmunes a interferencia y ruido.
- Comunicación Ethernet.
- Alta capacidad de almacenamiento en memoria.
- Comunicación serial.
- Conversor Análogo Digital.
- Fuentes de alimentación incluidas en placa.
- Manejable con fuentes AC o DC
- Batería de respaldo.
- Manejo de señales análogas y digitales.
- Generador de rampa

2.2.5 CARACTERÍSTICAS MECÁNICAS

El módulo PowerCore posee las siguientes medidas, 60 mm × 102 mm × 28 mm [12].

Su disposición física se muestra en las figuras 2.2, 2.3 y 2.4.

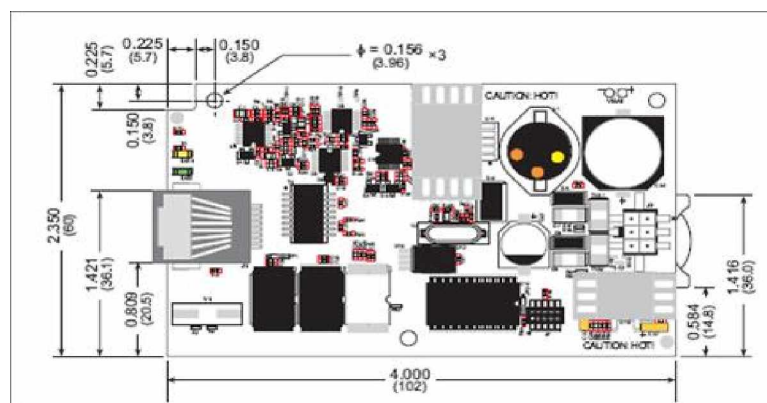


Figura 2.2. Vista superior del módulo PowerCore

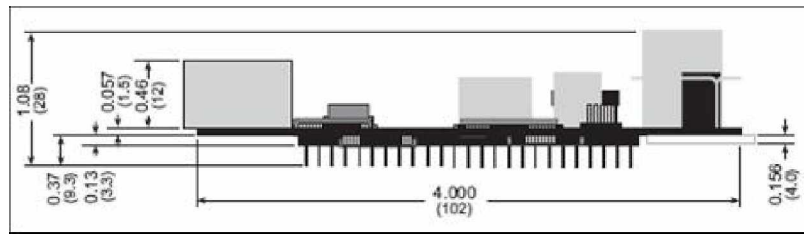


Figura 2.3. Vista frontal del módulo PowerCore

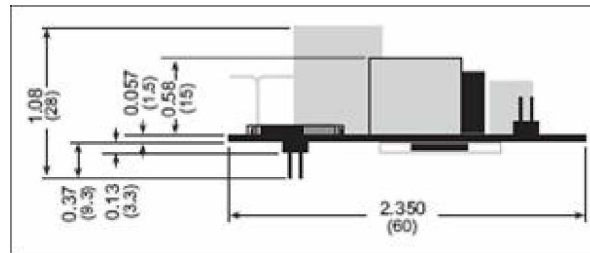


Figura 2.4. Vista lateral del módulo PowerCore

2.2.6 SUBSISTEMAS PRINCIPALES

El módulo PowerCore está integrado por subsistemas, los cuales en conjunto realizan las acciones específicas. Cada módulo tiene su función determinada y el funcionamiento de este depende del funcionamiento de cada uno de los subsistemas. En la figura 2.5 se muestra la disposición de los subsistemas en el módulo.

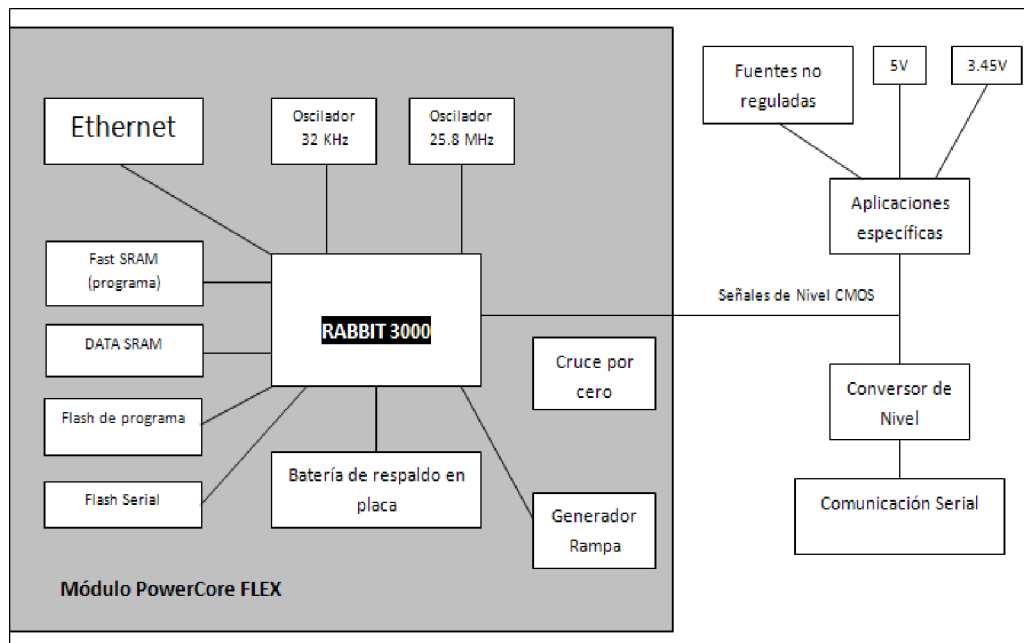


Figura 2.5 Disposición de Subsistemas del Módulo PowerCore Flex

2.2.7 DISPOSICIÓN DE PUERTOS DE ENTRADA Y SALIDA DIGITALES

La figura 2.6 ilustra la manera como se disponen las entradas y salidas digitales en el modulo PowerCore, la nomenclatura n.c simboliza que esos pines no tienen conexión [11].

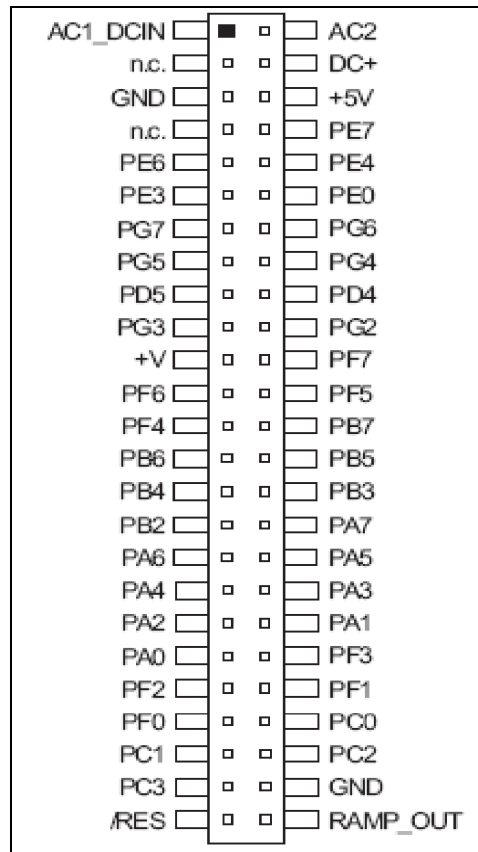


Figura 2.6 Disposición de Entradas y Salidas Digitales del Módulo PowerCore.

2.2.8 DISPOSICIÓN DE PUERTOS Y MEMORIA

La figura muestra el uso de los puertos del procesador Rabbit 3000 en el módulo PowerCore [7].

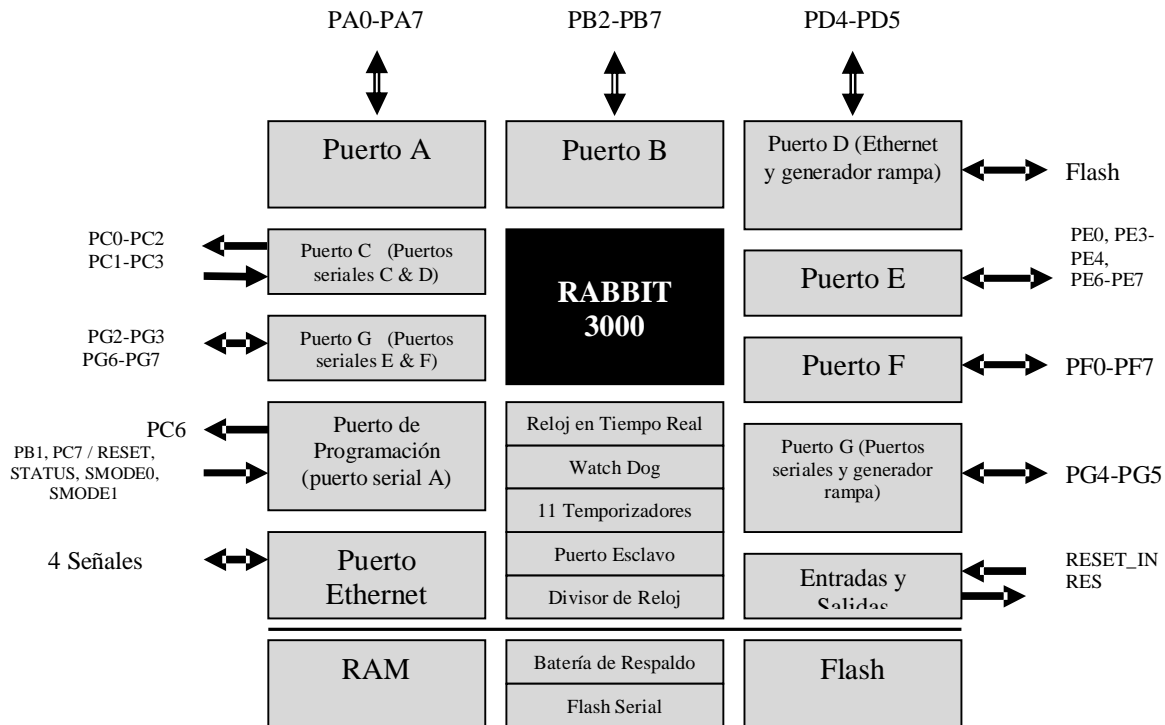


Figura 2.7 Disposición de Puertos y Memoria en el Módulo PowerCore.

2.2.9 DESCRIPCIÓN DE RECURSOS

2.2.9.1 BUSES INTERNOS Y EXTERNOS

Las líneas de dirección y de datos están ruteadas internamente en la tarjeta y son utilizadas por la memoria Flash, SRAM y los chips de Ethernet, estas corresponden a las A0-A19 (líneas de dirección) y D0-D7 (líneas de datos). Hay un bus separado completamente independiente de entradas y salidas, el cual es implementado mediante el uso de seis líneas de dirección y ocho de datos. Varios pulsos de corta duración pueden ser implementados para dato de reloj desde o hacia el bus. El bus de entrada/salida es una opción que puede ser habilitada para programas de usuario.

Las ocho líneas bidireccionales de entrada/salida comparten pines con el puerto paralelo A y las seis líneas de dirección comparten pines con parte del puerto paralelo B, aunque solo 64 direcciones de lectura o escritura son directamente disponibles, es

fácil expandir el espacio del registro del bus que es tan grande como se requiera añadiendo direcciones adicionales de bits, utilizando un registro cargado de 64 registros. Otro enfoque adicional es usar líneas adicionales separadas de pulsos electrónicos para crear más de 64 espacios de registros.

Las instrucciones de entrada y salida Rabbit son utilizadas para acceder a los registros creados en el bus de entrada y salida.

El puerto paralelo A, puede ser usado como un bus de datos externo de entrada y salida, para aislar entradas y salidas externas desde el bus de datos principal. Los pines en el puerto B usados como líneas de dirección de bus de entrada y salida pueden ser utilizados como líneas externas cuando el bus de entrada y salida no está habilitado. Los pines del puerto paralelo B, PB2-PB7, pueden ser utilizados también como un bus de direcciones auxiliar [6].

2.2.9.2 COMUNICACIÓN SERIAL

El módulo PowerCore Flex no posee ningún conector directamente en la placa. Sin embargo, una interfaz serial debe ser incorporada de la placa al módulo, ya que el módulo posee conectores RS-232 y RS-422. [8]

2.2.9.3 PUERTOS SERIALES

Existen 6 puertos seriales en la Rabbit 3800, los cuales son: A, B, C, D, E y F. Todos los 6 puertos seriales pueden operar en un modo asincrónico con una tasa en baudios correspondiente al sistema de reloj dividido para 8. Un puerto asincrónico puede manejar 7 u 8 bits de datos.

El puerto serial A es normalmente utilizado como puerto de programación, pero puede ser usado como asincrónico o como un puerto serial de reloj, una vez que el módulo PowerCore Flex ha sido programado y está operando en el modo Run.

El puerto serial B es utilizado para comunicar con la flash serial en el Módulo PowerCore y no está disponible para ninguna otra aplicación.

Los puertos seriales C y D pueden también operar en modo de reloj serial, de este modo el dato de entrada o salida tiene sincronismo.

Los puertos seriales E y F, pueden ser configurados como puertos seriales HDLC. El protocolo IrDA es también soportado por el formato SDLC para estos dos puertos. El puerto E puede ser configurado como SDLC.

Los tres puertos seriales que soportan comunicación serial de tiempo, los puertos A, C y D son apropiados para interfaces con dispositivos SPI [8].

2.2.9.4 PUERTO DE PROGRAMACIÓN

El Puerto A tiene características especiales que permiten el arranque del sistema después de ser reseteado. Es utilizado además para el desarrollo de software bajo Dynamic C.

El puerto de programación utiliza el puerto serial A del microprocesador para la comunicación y para operaciones de programación, tales como depuración, clonación y descarga, además para depuración remota sobre una conexión Ethernet, para lo cual se requiere de una placa auxiliar. El pin de status de la Rabbit 3800 también se encuentra en el puerto de programación, ya que esta es una salida que puede ser usada para enviar una señal digital general.

El puerto de programación transmite la información hacia o desde la PC mientras un programa está siendo depurado.

2.2.9.5 MEMORIA

SRAM

El módulo PowerCore trabaja a 51.6 MHz y necesitan 512K de SRAM para programación y ejecución.

Flash EPROM

El modulo PowerCore posee 512K de flash EPROM.

Flash Serial

El módulo PowerCore dispone de una memoria serial flash para almacenar datos y páginas web. Los programas de ejemplo ilustran el uso del flash serial [13].

2.2.9.6 SISTEMA DE RELOJ

El procesador Rabbit 3000 tiene una ventaja, posee oscilador principal utiliza un cristal externo con una frecuencia en el rango de 1.8 MHz a 26 MHz. El reloj del procesador se deriva de la salida del oscilador mediante un doblador de frecuencia, usando la frecuencia directamente, o por división de frecuencia para 2, 4, 6 u 8. El reloj del procesador puede también ser manejado por un oscilador en tiempo real de 32.768 KHz para operaciones de muy baja potencia, en cuyo caso puede ser desactivado mediante el control de software [5].

2.2.9.7 ENTRADA DE OSCILADOR DE 32.768 KHZ

La entrada de 32.768 KHz al oscilador está diseñada para aceptar un reloj de 32.768 KHz, el mismo que maneja la batería de respaldo. Además, es usado para manejar el watchdog timer y para generar el reloj en Baudios para el puerto serial A durante la secuencia de arranque [5].

2.2.9.8 GENERADOR RAMPA

El módulo PowerCore tiene un generador rampa incorporado en la tarjeta, el cual provee una precisa función continua “diente de sierra”. La calibración de esta función se realiza con un voltaje de referencia de 2.5V. La rampa de 400Hz tiene un crecimiento lineal de 0 a 3.1V en un tiempo aproximado de 1.9ms y de decrecimiento de 0.45ms.

El generador de rampa permite la medición de voltajes análogos usando comparadores LM339 y las capacidades de captura de pulso que ofrece el microprocesador Rabbit 3000 para convertir tiempo en voltaje [8].

2.3 PROTOCOLOS DE COMUNICACIÓN

Los protocolos de comunicaciones definen las normas que posibilitan que se establezca una comunicación entre varios equipos o dispositivos que forman parte de la red, de modo que para que dos nodos de la red se puedan comunicar entre si es necesario que ambos empleen la misma configuración de protocolos, es decir que hablen el mismo idioma.

Entre los protocolos propios de una red de área local podemos distinguir dos principales grupos. Por un lado están los protocolos de los niveles físicos y de enlace, niveles 1 y 2 del modelo OSI, que definen las funciones asociadas con el uso del medio de transmisión: envío de los datos a nivel de bits y trama, y el modo de acceso de los nodos al medio. Estos protocolos vienen unívocamente determinados por el tipo de red (Ethernet, Token Ring, etc.). El segundo grupo de protocolos se refiere a aquellos que realizan las funciones de los niveles de red y transporte, niveles 3 y 4 de OSI, es decir los que se encargan básicamente del encaminamiento de la información y garantizar una comunicación extremo a extremo libre de errores. Estos protocolos transmiten la información a través de la red en pequeños segmentos llamados paquetes. Si un ordenador quiere transmitir un fichero grande a otro, el fichero es dividido en paquetes en el origen y vueltos a ensamblar en el ordenador destino. Cada protocolo define su propio formato de los paquetes en el que se especifica el origen, destino, longitud y tipo del paquete, así como la información redundante para el control de errores. Los protocolos de los niveles 1 y 2 dependen del tipo de red, mientras que para los niveles 3 y 4 hay diferentes alternativas, siendo TCP/IP la configuración más extendida [14].

Los protocolos pueden estar implementados bien en hardware (tarjetas de red), software (drivers), o una combinación de ambos, existen propiedades típicas que estos utilizan los cuales se describen a continuación [15]:

- Detección de la conexión física sobre la que se realiza la conexión (cableada o sin cables)
- Pasos necesarios para comenzar a comunicarse (Handshaking)
- Negociación de las características de la conexión.
- La manera en la que se inicia y se termina un mensaje.

- Formato de los mensajes.
- Detección y corrección de errores
- Detección de la pérdida de conexión.
- Terminación de la sesión de conexión.
- Estrategias para asegurar la seguridad (autenticación, cifrado).

Para el desarrollo de este proyecto el módulo PowerCoreFlex al cual llegan las señales de los dispositivos sensores se comunica con la PC en la estación de cobro bajo el protocolo TCP/IP, con lo cual es posible el intercambio de datos entre la PC y el controlador Rabbit, permitiendo de esta manera reducir el tiempo de atención a cada vehículo al mínimo posible.

2.3.1 TCP/IP

TCP/IP son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet, un sistema de protocolos que hacen posibles servicios entre ordenadores que no pertenecen a la misma red, permitiendo el envío de paquetes de información a diferentes destinos, para lo cual requiere verificar y ordenar los paquetes en su llegada

El Protocolo de Control de Transmisión (TCP) permite a dos anfitriones establecer una conexión e intercambiar datos. El TCP garantiza la entrega de datos, es decir, que los datos no se pierdan durante la transmisión y también garantiza que los paquetes sean entregados en el mismo orden en el cual fueron enviados. La arquitectura que utiliza el protocolo TCP ve como iguales a todas las redes a conectarse, sin tomar en cuenta el tamaño de ellas, ya sean locales o de cobertura amplia. Define que todas las redes que intercambiarán información deben estar conectadas a una misma computadora o equipo de procesamiento; a tales computadoras se les denominan computas, pudiendo recibir otros nombres como enrutadores o puentes.

Para que exista comunicación en una red dos computadoras deben estar identificadas con precisión. Este identificador puede estar definido en niveles bajos (identificador físico) o en niveles altos (identificador lógico) dependiendo del protocolo utilizado. TCP/IP utiliza un identificador denominado dirección Internet o dirección IP, cuya longitud es de 32 bytes, estos se representan mediante cuatro octetos, escritos en

formato decimal, separados por puntos. La dirección IP identifica tanto a la red a la que pertenece una computadora como a ella misma dentro de dicha red [14].

2.3.1.1 CAPAS

La transmisión y recepción de datos dentro de la red se la realiza mediante niveles o capas que describen una pila de protocolos, aunque en la práctica no corresponde exactamente con el modelo en Internet. En una pila de protocolos, cada nivel soluciona una serie de problemas relacionados con la transmisión de datos, y proporciona un servicio bien definido a los niveles más altos. Los niveles superiores son los más cercanos al usuario y tratan con datos más abstractos, dejando a los niveles más bajos la labor de traducir los datos de forma que sean físicamente manipulables.

Los protocolos de cada capa tienen una interfaz bien definida. Una capa generalmente se comunica con la capa inmediata inferior, la inmediata superior, y la capa del mismo nivel en otros computadores de la red. Esta división de los protocolos ofrece abstracción en la comunicación. De este modo, se realiza el envío del software entre capas a través de la recepción de información almacenada en las cabeceras. Cada capa añade su cabecera en la parte delantera del mensaje. La cabecera es retirada por parte de su correspondiente capa en el lado receptor.

Normalmente, los tres niveles superiores del modelo OSI (Aplicación, Presentación y Sesión) son considerados simplemente como el nivel de aplicación en el conjunto TCP/IP. Como TCP/IP no tiene un nivel de sesión unificado sobre el que los niveles superiores se sostengan, estas funciones son típicamente desempeñadas (o ignoradas) por las aplicaciones de usuario. La diferencia más notable entre los modelos de TCP/IP y OSI es el nivel de Aplicación, en TCP/IP se integran algunos niveles del modelo OSI en su nivel de Aplicación. Una interpretación simplificada de la pila TCP/IP se muestra en la figura 2.8.

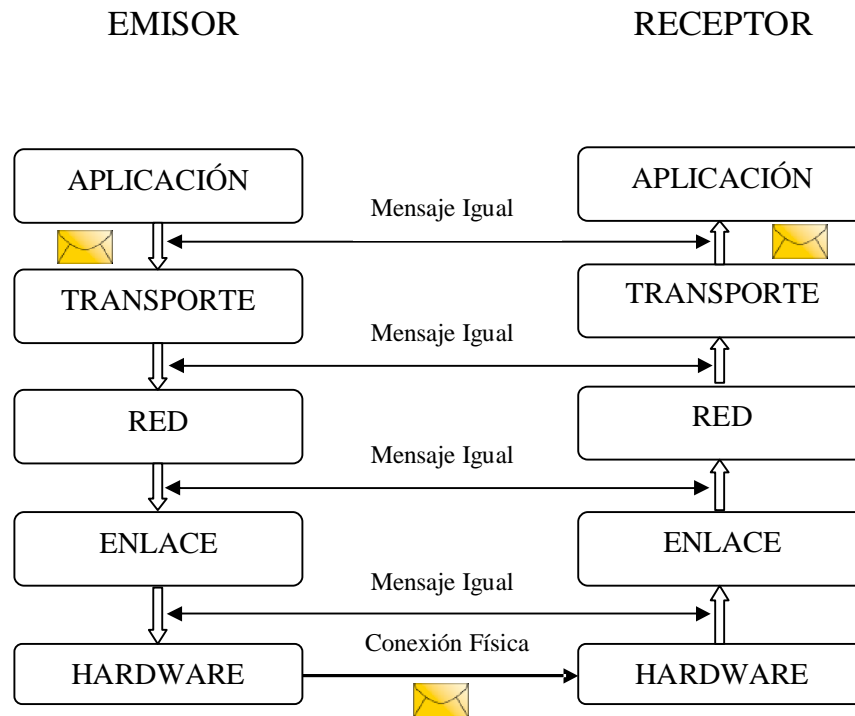


Figura 2.8 Modo de comunicación entre capas de red del modelo TCP/IP

Cada nivel de la jerarquía de protocolos OSI tiene una función específica y define un nivel de comunicaciones entre sistemas:

- **Nivel de Aplicación:** En este el sistema operativo de red y sus aplicaciones se hacen disponibles a los usuarios. Los procesos que acontecen en este nivel son aplicaciones específicas que pasan los datos al nivel de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar [16].
- **Nivel de Transporte:** Suministra el mayor nivel de control en el proceso que mueve actualmente datos de un equipo a otro. Los protocolos del nivel de transporte solucionan problemas como la fiabilidad y la seguridad de que los datos llegan en el orden correcto, también determinan a qué aplicación van destinados los datos.
- **Nivel de Red:** Define protocolos para abrir y mantener un camino entre equipos de la red. Se ocupa del modo en que se mueven los paquetes, es decir de la comunicación con el módulo PowerCoreFlex.
- **Nivel de Enlace de Datos:** Define las reglas para enviar y recibir información a través de la conexión física entre dos sistemas.

- **Nivel Físico:** Describe las características físicas de la comunicación, como las convenciones sobre la naturaleza del medio usado para la comunicación (comunicaciones por cable, fibra óptica o radio), y todo lo relativo a los detalles como los conectores, código de canales y modulación, potencias de señal, longitudes de onda, sincronización, temporización y distancias máximas.

2.3.2 ETHERNET

Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI. Se tomó como base para la redacción del estándar internacional IEEE 802.3. La figura 2.9 muestra que Ethernet opera en las dos capas inferiores del modelo OSI: el de la capa de enlace de datos y la capa física [17].

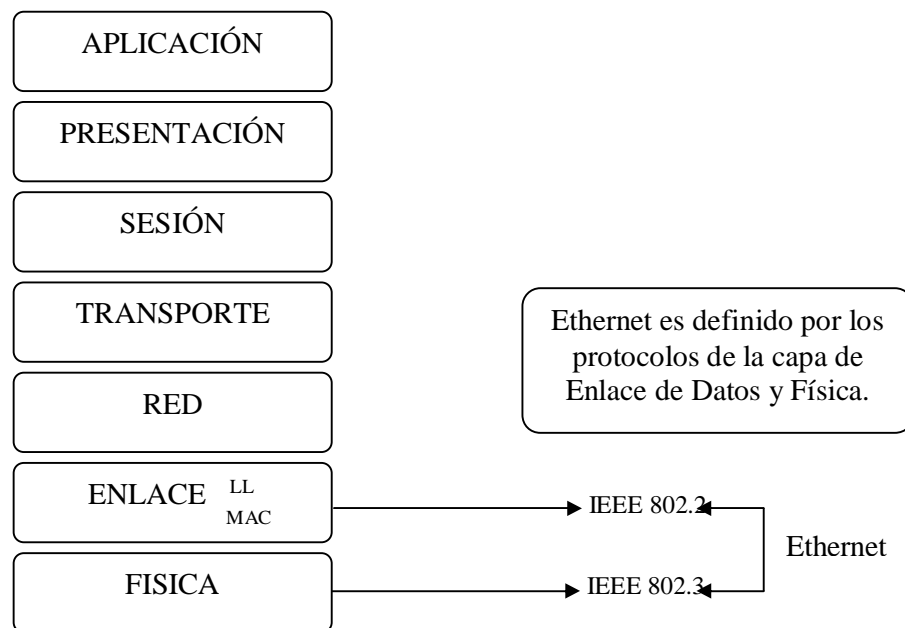


Figura 2.9 Capa Ethernet en el modelo OSI

Ethernet utiliza un medio de difusión de bus y se basa en el método de acceso conocido como CSMA/CD para regular el tráfico en la línea de comunicación principal.

La información en la red Ethernet se envía en tramas de longitud variable que contienen la información de control. La trama está formada por tres grupos llamados

Header, Packet y Trailer, integrados por un conjunto de bytes que realizan una función específica. El Header indica el inicio de la trama, la dirección, el tipo y la calidad de control; el Packet contiene los datos que van a ser transmitidos; y el Trailer se encarga de la detección de errores y del bit de parada [16]. La figura 2.9 muestra a cada uno de los grupos de la trama.

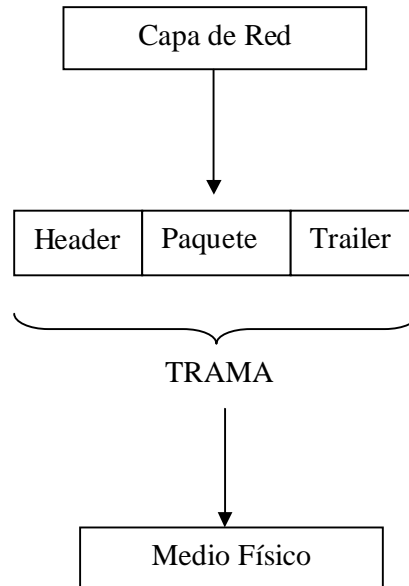


Figura 2.10 Trama de Ethernet

El estándar Ethernet original permite la transmisión en banda base a 10 Mbps; las tarjetas se denominan comúnmente 10BaseT, 10Base2..., según el tipo de cable de conexión [18].

2.3.2.1 FORMATO DE LA TRAMA DE ETHERNET

La estructura original de la trama Ethernet define un tamaño mínimo de 64 bytes y un máximo de 1518 bytes. Esto incluye todos los bytes de la dirección MAC de destino a través de la secuencia de verificación de trama (FCS). El preámbulo y el delimitador del inicio de la trama no están incluidos cuando se describe el tamaño de la trama [17]. La tabla 2.1 describe el formato de la trama Ethernet.

ETHERNET						
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46 a 1500 bytes	4 bytes
PREAMBULO	SOF	DESTINO	ORIGEN	TIPO	DATOS	FCS

Tabla 2.2 Formato de Trama Ethernet

Preámbulo

Campo de 7 bytes (56 bits) con una secuencia de bits usada para sincronizar y estabilizar el medio físico antes de iniciar la transmisión de datos. El patrón del preámbulo es:

10101010 10101010 10101010 10101010 10101010 10101010 10101010

Estos bits se transmiten en orden, de izquierda a derecha y en la codificación Manchester representan una forma de onda periódica.

SOF (Start Of Frame) Inicio de Trama

Campo de 1 byte (8 bits) con un patrón de 1s y 0s alternados y que termina con dos 1s consecutivos. El patrón del SOF es: 10101011. Indica que el siguiente bit será el bit más significativo del campo de dirección MAC de destino. Aunque se detecte una colisión durante la emisión del preámbulo o del SOF, el emisor debe continuar enviando todos los bits de ambos hasta el fin del SOF.

Dirección de destino

Campo de 6 bytes (48 bits) que especifica la dirección MAC hacia la que se envía la trama. Esta dirección de destino puede ser de una estación, de un grupo multicast o la dirección de broadcast de la red. Cada estación examina este campo para determinar si debe aceptar el paquete.

Dirección de origen

Campo de 6 bytes (48 bits) que especifica la dirección MAC desde la que se envía la trama. La estación que deba aceptar el paquete conoce por este campo la dirección de la estación origen con la cual intercambiará datos.

Tipo

Campo de 2 bytes (16 bits) que identifica el protocolo de red de alto nivel asociado con el paquete o, en su defecto, la longitud del campo de datos. La capa de enlace de datos interpreta este campo.

Datos

Campo de 46 a 1500 Bytes de longitud. Cada Byte contiene una secuencia arbitraria de valores. El campo de datos es la información recibida del nivel de red (la carga útil). Este campo, también incluye los H3 y H4 (cabeceras de los niveles 3 y 4), provenientes de niveles superiores. Si la longitud es inferior a 46 bytes se rellena con bytes al final.

FCS (Frame Check Sequence - Secuencia de Verificación de Trama)

Campo de 32 bits (4 bytes) que contiene un valor de verificación CRC (Control de redundancia cíclica). El emisor calcula el CRC de toda la trama, desde el campo destino al campo CRC suponiendo que vale 0. El receptor lo recalcula, si el valor calculado es 0 la trama es válida.

La trama Ethernet contiene la dirección física de la fuente y también la dirección física de destino, una vez transmitida la trama todos los dispositivos de red contienen por la siguiente oportunidad de transmitir una trama, la disputa es pareja, y para asegurar que el acceso al canal de comunicaciones sea equitativo, ningún dispositivo puede bloquear a otros dispositivos.

Control de Acceso al Medio

El acceso al canal de comunicaciones compartido es determinado por la subcapa MAC. El método utilizado por Ethernet es CSMA/CD, con Detección de Colisiones es el protocolo utilizado en redes Ethernet para asegurar que sólo un nodo de red se transmite en la red de cable en un momento dado. A través este proceso, Ethernet “escucha” en el medio físico antes de intentar transmitir, en caso de que esté ocupado este, espera un tiempo aleatorio y vuelve a intentarlo (Carrier Sense); además permite escuchar y transmitir a la vez (Multiple Access); en caso de que múltiples dispositivos transmitan al mismo tiempo, es posible la detección del error (Collision Detection).

2.3.3 ESTANDAR RJ-45

El RJ45 es una interfaz física comúnmente usada para conectar redes de cableado estructurado, (categorías 4, 5, 5e y 6). RJ es un acrónimo inglés de Registered Jack que a su vez es parte del Código Federal de Regulaciones de Estados Unidos. Posee ocho "pines" o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado.

Es utilizada comúnmente con estándares como TIA/EIA-568-B, que define la disposición de los pines; es empleado en cables de red Ethernet, donde suelen usarse 8 pines (4 pares) [19]. La figura 2.11 muestra la configuración de pines del conector RJ-45 para el puerto Ethernet.

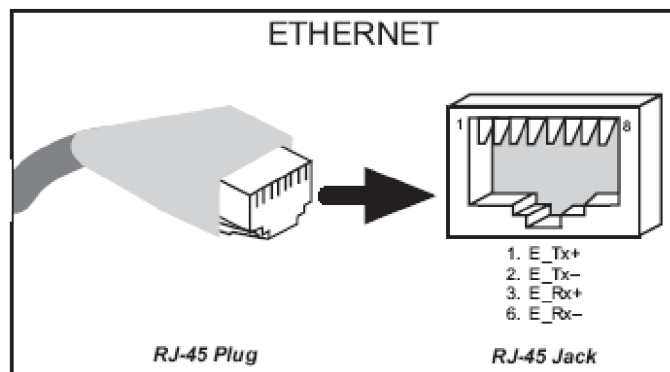


Figura 2.11 Configuración de pines del conector RJ-45 para el puerto Ethernet

Para el desarrollo de este proyecto esta interfaz física será utilizada para la conexión de la PC de control ubicada en la cabina del peaje con el dispositivo de control Rabbit, para esto se utilizara un cable cruzado que sirve para conectar dos dispositivos igualitarios, como 2 computadoras entre sí, en este caso la PC con el controlador Rabbit, para lo que se ordenan los colores de tal manera que no sea necesaria la presencia de un hub o switch. La distribución para un cable cruzado se muestra a continuación.

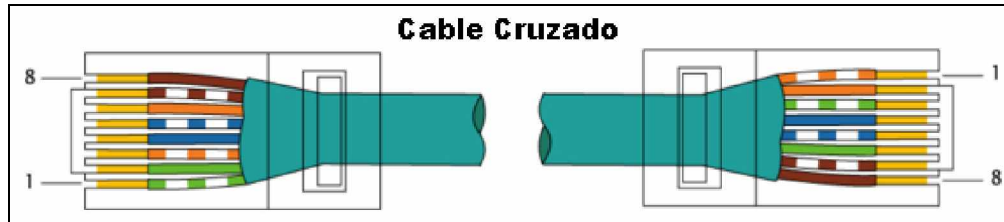


Figura 2.12 Distribución de un cable cruzado

2.4 COMUNICACIÓN SERIAL

La comunicación serial es un protocolo muy común para comunicación entre dispositivos que se incluye de manera estándar en cualquier computadora. La mayoría de estas incluyen dos puertos seriales, el concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. La comunicación serial es también un protocolo común utilizado por varios dispositivos para instrumentación; existen varios dispositivos compatibles que incluyen puertos seriales.

Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmitir, (3) Recibir. Debido a que la transmisión es asincrónica, es posible enviar datos por una línea mientras se reciben datos por otra. Existen otras líneas disponibles para realizar *handshaking*, o intercambio de pulsos de sincronización, pero no son requeridas. Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

El Módulo PowerCoreFlex brinda comunicación serial incorporada a través de sus puertos que se basan en los estándares RS-232, RS-422 Y RS-48. El estándar RS-232 es usado para la visualización de la tarifa a través del envío de datos del módulo PowerCoreFlex hacia el Display de visualización ubicado en la cabina de cobro.

2.4.1 TIPOS DE COMUNICACIONES SERIALES

2.4.1.1 SIMPLEX

En este caso el transmisor y el receptor están perfectamente definidos y la comunicación es unidireccional.

2.4.1.2 HALF DUPLEX

En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor y los datos se desplazan en ambos sentidos pero no simultáneamente.

2.4.1.3 FULL DUPLEX

Los datos se desplazan en ambos sentidos simultáneamente. Para ello ambos transmisores poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados.

2.4.2 RS-232

Corresponde al estándar ANSI/EIA-232, éste conector permite la transmisión de datos a través del puerto serial de forma asíncrona. Es utilizado para una gran variedad de propósitos por lo que se han ido desarrollando notables mejoras en las líneas de transmisión y en los cables para aumentar el desempeño de RS-232 en lo que respecta a la distancia y velocidad del estándar. Sin embargo está limitado a comunicaciones de punto a punto entre los dispositivos y el puerto serial de la computadora. El hardware de RS-232 se puede utilizar para comunicaciones seriales en distancias de hasta 15 metros y para velocidades de comunicación bajas, de no más de 20 [Kb/s] [20].

Se envían datos de 5, 6, 7, 8 o 9 bits. La velocidad se mide en baudios (bits/segundo) y sólo son necesarios tres cables, correspondientes a la transmisión y recepción y tierra (GND) [22]. Es imprescindible la comprensión de la función que cumple cada pin tanto de entrada como de salida de datos durante la comunicación, ya que existen dos tipos de conectores para este estándar los de 9 pines y los de 25 pines cuales permiten la transferencia de una mayor cantidad de datos.

Las señales con la que actúa el puerto son digitales (0 - 1) y la tensión a la que trabaja es de 12 Voltios, es decir 12 [V] corresponde a un "0" lógico, mientras que -12[V] indica un "1" lógico [21]. Las características de los pines y su configuración para cada uno de los tipos de conectores se muestran en la tabla 2.3

Nombre	Descripción	Conector 9 pines	Conector 25 pines
-	Chasis	1	1
TxD	Transmitir Datos Señal de salida	3	2
RxD	Recibir Datos Señal de entrada	2	3
RTS	Solicitud de envío Señal de salida.	7	4
CTS	Libre para envío Señal de entrada	8	5
DSR	Equipo de datos listo Señal de entrada	6	6
SG	Tierra Referencia para señales	5	7
DCD	Detección de portadora Señal de entrada	1	8
TxC	Reloj de transmisión	-	15
RxC	Reloj de recepción	-	17

DTR	Terminal de datos listo Señal de salida	4	20
RI	Indicador de llamada Señal de entrada	9	22
RTxC	Reloj de Recepción / Transmisión	-	24

Tabla 2.3. Configuración y descripción de conectores DB9 y DB25 (Estándar RS-232)

Antes de iniciar cualquier comunicación con el puerto RS232 se debe configurar el protocolo a utilizar. Siendo los parámetros a configurar los siguientes:

Velocidad de transmisión (baud rate): Indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds).

Bits de datos: Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. El número de bits que se envía depende en el tipo de información que se transfiere. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usar para referirse a todos los casos [21].

Bits de parada: Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits.

Paridad: Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible [21].

2.4.3 RS-422

Corresponde al estándar EIA RS-422-A, este conector serial utilizado usa señales eléctricas diferenciales, en comparación con señales referenciadas a tierra como en RS-232. La transmisión diferencial, que utiliza dos líneas para transmitir y recibir, tiene la ventaja que es más inmune al ruido y puede lograr mayores distancias que RS-232. La inmunidad al ruido y la distancia son dos puntos clave para ambientes y aplicaciones industriales [20].

2.4.4 RS-485

Corresponde al estándar EIA-485 es una mejora sobre RS-422 ya que incrementa el número de dispositivos que se pueden conectar (de 10 a 32) y define las características necesarias para asegurar los valores adecuados de voltaje cuando se tiene la carga máxima. Gracias a esta capacidad, es posible crear redes de dispositivos conectados a un solo puerto RS-485. Esta capacidad, y la gran inmunidad al ruido, hacen que este tipo de transmisión serial sea la elección de muchas aplicaciones industriales que necesitan dispositivos distribuidos en red conectados a una PC u otro controlador para la colección de datos, HMI, u otras operaciones. RS-485 es un conjunto que cubre RS-422, por lo que todos los dispositivos que se comunican usando RS-422 pueden ser controlados por RS-485. El hardware de RS-485 se puede utilizar en comunicaciones seriales de distancias de hasta 1000 metros de cable.

2.5 COMUNICACIÓN CON LA TARJETA ELECTRÓNICA RABBIT

La comunicación, constituye un aspecto imprescindible para coordinar, controlar y compartir datos. Es por ello que se requiere establecer una comunicación sólida entre cada uno de los dispositivos, garantizando de este modo el funcionamiento del sistema, confiabilidad en la información y seguridad en el tipo de datos adquiridos.

Es necesario considerar los modos de comunicación que ofrece el controlador Rabbit, tanto en hardware como en software, ya que a partir de ello es posible alcanzar las metas del proyecto, optimizando el tiempo de atención a cada vehículo al mínimo posible, esto se realiza a través de la comunicación Ethernet, la misma que sirve para la

comunicación entre el controlador y la PC en la estación de control. También se utiliza la comunicación serial del controlador Rabbit, tiene como fin enviar la tarifa al Display de visualización ubicado en la estación de control. Para establecer la comunicación, a través de la red Ethernet y serial es necesario desarrollar la programación respectiva mediante la plataforma de Dynamic C. La figura 2.13 indica el esquema gráfico de los tipos de comunicación.

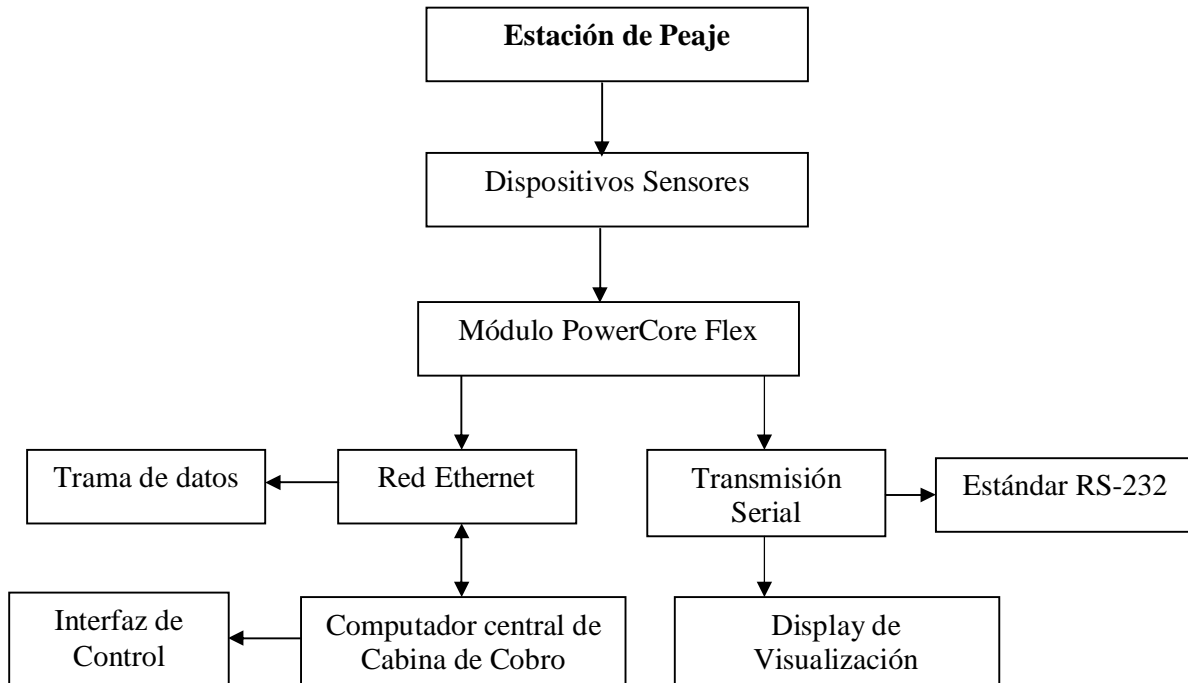


Figura 2.13 Esquema de Comunicación de la Red Ethernet y Comunicación Serial

2.5.1 RED ETHERNET

El sistema mediante el cual existe la comunicación entre el módulo PowerCoreFlex ubicado en la parte externa del carril y la máquina de operación en la caseta de peaje, está basado en la integración de una red de tipo local, usando puertos de comunicaciones punto a punto para Ethernet y cableado tipo par trenzado, de modo que es necesario utilizar un cable UTP categoría 5. En esta forma la posibilidad de pérdida de eventos es muy reducida.

Es importante mencionar que la transmisión de datos es realizada a través de un cable cruzado el cual permite el intercambio de información entre el ordenador principal y el módulo PowerCoreFlex de manera Half Duplex.

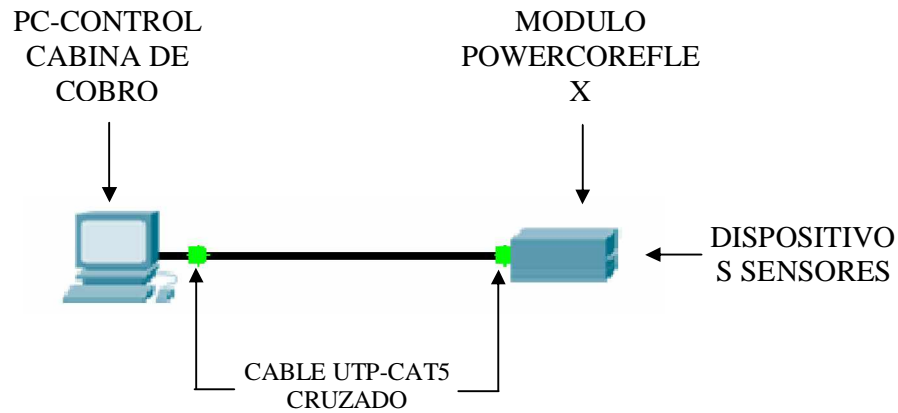


Figura 2.14 Red Ethernet del Peaje

El controlador Rabbit utiliza la versión Dynamic C 9.20 el cual posee sus propias librerías para realizar la programación de las distintas aplicaciones. A continuación se detalla las librerías utilizadas para la comunicación con el protocolo TCP/IP.

2.5.2 SOCKET TCP

Un socket TCP representa el estado de conexión entre un equipo local y un mando a distancia. Para propósitos prácticos un socket es una estructura en la memoria RAM que contiene toda la información necesaria de estado, requiere un buffer para transmitir y recibir la información. Los drivers TCP/IP están localizados en la carpeta LIB\TCPIP, cada socket debe estar asociado a tcp_Socket de 145 bytes. Los buffers de entrada y salida se encuentran en memoria extendida [23].

2.5.3 APERTURA DEL SOCKET

Existen dos maneras de abrir el socket TCP, de manera pasiva y activa. Para el desarrollo del presente proyecto se utilizara el socket de manera pasiva, en este modo, cuando se requiere establecer contacto con el dispositivo controlador, es necesario abrir

el socket con `tcp_listen()`. Esta forma es comúnmente utilizada para servidores Internet que escuchan en un puerto conocido, como el 80 para HTTP.

Es necesario suministrar `tcp_listen()` con un puntero de estructura `tcp_Socket` y la dirección IP viene dada por la tarjeta. Se debe configurar con 0 (cero), en caso de que se requiera realizar la conexión desde cualquier dirección IP o cualquier puerto.

Para manejar conexiones simultaneas múltiples, cada nueva conexión requerirá su propia estructura `tcp_Socket` y una llamada separada a `tcp_listen`, pero usando el mismo número de puerto local.

La llamada `tcp_listen()` retornará inmediatamente y se debe verificar la conexión entrante. Es posible el uso de la macro `sock_wait_established`, con la cual se llamará a `tcp_tick()` y bloqueará hasta que la conexión sea establecida o se puede probar manualmente el socket utilizando `sock_established` [24].

2.5.4 FUNCIONES TCP SOCKET

Existen diversas funciones que pueden ser aplicadas a `tcp_Socket`, entre las cuales se encuentran las de estado, entradas y salidas y Control, las cuales han sido consideradas durante el desarrollo del proyecto. Además de `tcp_open` y `tcp_listen()`, también está `sock_close()` que debe ser llamada cuando se espera finalizar una conexión. Una llamada a `sock_close()` no cierra inmediatamente la conexión, porque esta toma cierto tiempo en enviar la solicitud de finalización y recibir el reconocimiento. Si desea estar seguro que la conexión está completamente cerrada antes de continuar, es posible realizar el llamado de la función `tcp_tick()` con la dirección del socket. Cuando esta función retorna un 0 (cero), indica que el socket está completamente cerrado. Se debe tener en cuenta que si un dato sale para ser leído en el socket, este no puede ser cerrado completamente.

Para cancelar una conexión o volver a establecerla existe la función `sock_abort()`. Esta función reseteará al TCP y los paquetes enviados posteriormente serán ignorados [24].

2.5.5 COMUNICACIÓN SERIAL

El módulo PowerCoreFlex posee seis puertos seriales la comunicación. Las librerías que incluye Dynamic C ofrece un rango muy amplio de soporte. La librería RS232.LIB provee un conjunto de buffer circular basado en funciones seriales. La librería PACKET.LIB provee funciones seriales basadas en paquetes, en donde estos paquetes pueden ser delimitados por el noveno bit, por brechas de transmisión, o pueden ser configuradas de acuerdo a la aplicación. Las dos librerías proveen el bloqueo de funciones, la cual no retorna hasta que la transmisión o recepción ha finalizado, permitiendo que otras funciones se desempeñen entre llamadas [25].

La función voidserMode(int mode) permite desarrollar líneas de comunicación serial para el módulo PowerCore Flex. Para esto se llama a la función serXpen() por cualquiera de los puertos seriales. Cabe mencionar que estos puertos son el E y F. El modo se determina mediante el valor 0 o 1. Para el desarrollo de nuestra aplicación se va a utilizar el puerto E.

Modo	Puerto Serial	
	E	F
0	RS-232, 3 cables	RS-232, 3 cables
1	RS-232, 5 cables	RTS/CTS

Tabla 2.4. Configuración de Comunicación Serial en Dynamic C

2.6 SOFTWARE

Dynamic C es un sistema integrado de desarrollo para escribir software embebido. Está diseñado para el uso con controladores basados en el microprocesador Rabbit. Puede manejar aplicaciones de lenguaje C de aproximadamente 50000 C + estados o 1 MB.

Dynamic C integra en un programa las siguientes funciones de desarrollo:

- Edita
- Compila
- Enlaza
- Carga
- Depura

Dynamic C es una herramienta fácil de usar, con funciones completas, editor de texto, además se pueden ejecutar programas y depurar interactivamente en el código fuente o código máquina. Soporta lenguaje ensamblador conjuntamente posee menús desplegables lo que lo hace dinámico y fácil de usar.

Dynamic C viene con muchas funciones de las bibliotecas, todas en código fuente. Estas librerías soportan tiempo real de programación y proporcionan cadenas estándar y funciones matemáticas.

Dynamic C permite que el sistema desarrollado tenga el desempeño esperado, sin embargo, el operador responsable de la operación del peaje, no requiere manejar este software ya que el sistema implementado tiene un interfaz humano máquina la cual permite utilizar todas las funciones que ofrece el sistema brindando un control práctico del mismo.

CAPITULO III

DISEÑO DEL SISTEMA DE PRECLASIFICACIÓN VEHICULAR Y AUDITORIA

3.1 DISEÑO INTEGRAL DEL SISTEMA

El diseño integral del sistema de control constituye la esencia del proyecto, ya que representa el funcionamiento del sistema completo, esto conlleva a la integración de todos los dispositivos de hardware y software.

Para realizar el diseño es necesaria la consideración de características eléctricas de cada uno de los componentes, su disposición física y los elementos de protección. A continuación se describe de forma particular las características de cada uno de los subsistemas que lo componen además de su disposición en la estación de peaje.

3.2 DISPOSICIÓN DE LOS ELEMENTOS SENSORES

La función del conjunto de sensores es clasificar de forma automática a cualquier vehículo que cruce por la zona de los detectores. Esta clasificación automática tiene sus limitaciones, las cuales se refieren a la dificultad de clasificar vehículos de fabricación especial o que hayan sido modificados en sus dimensiones y características por sus propietarios o por fallas de los componentes del sistema.

Otra forma con la que se afecta la precisión de la clasificación automática, se da cuando el vehículo efectúa maniobras de entrada – salida, dentro de la zona de sensores, evasión de los mismos o circulación con exceso de velocidad.

Tomando en cuenta estas consideraciones a continuación se describe de forma particular cada uno de los dispositivos sensores que intervienen en el sistema de peaje, además de su ubicación en el mismo para que el equipo opere de forma correcta. El equipo de elementos sensores se encuentra conformado por:

Sensores de detección de ejes (Sensores de piso).

Este dispositivo es instalado regularmente en el piso o superficie del carril, son bandas de contacto seco que tienen internamente un muelle con dos láminas, que están encapsuladas en hule de alta resistencia.

Estas bandas detectan el número de ejes de un vehículo, por su arreglo en un ángulo agudo con respecto a la banqueta del carril puede distinguir si los ejes tienen rodada simple (eje ligero), o doble rodada (eje pesado). Hay 2 disposiciones frecuentemente utilizadas para detectar cantidad y tipo de ejes.

1. Dos sensores paralelos, dispuestos uno después del otro a una distancia determinada X , para verificar el número de ejes, un eje puede ser simple o múltiple (doble, triple o mayor), Además, los sensores pueden utilizarse para detectar la dirección del vehículo, de esta forma, si existe marcha inversa, se pueden restar los ejes [1].

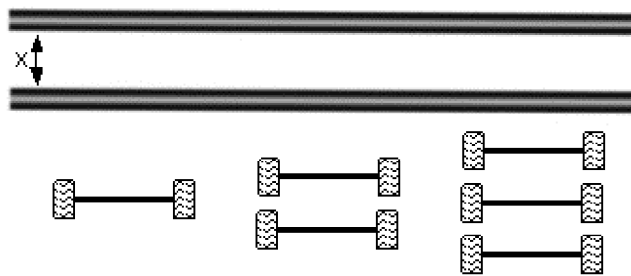


Figura 3.1. Disposición de dos Sensores Paralelos uno después de otro a una distancia X

2. Dos Sensores paralelos dispuestos uno después del otro a una distancia determinada (Y), de forma tal que un eje simple pisa los sensores uno a uno, y que uno dual pisa los dos sensores simultáneamente, estos sensores están inclinados a aproximadamente 53° (Z), medidos desde el eje longitudinal de la vía [1].

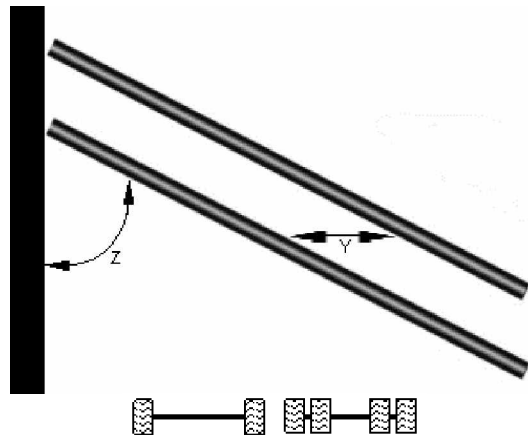


Figura 3.2. Disposición de dos Sensores inclinados.

Su disposición en la estación de peaje se la visualiza en la figura 3.3, cabe mencionar que el largo de los sensores se define de acuerdo al ancho de la vía.

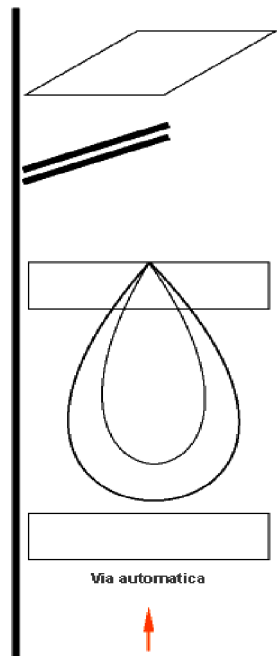


Figura 3.3. Disposición Típica de las Peanas en la Estación de Peaje.

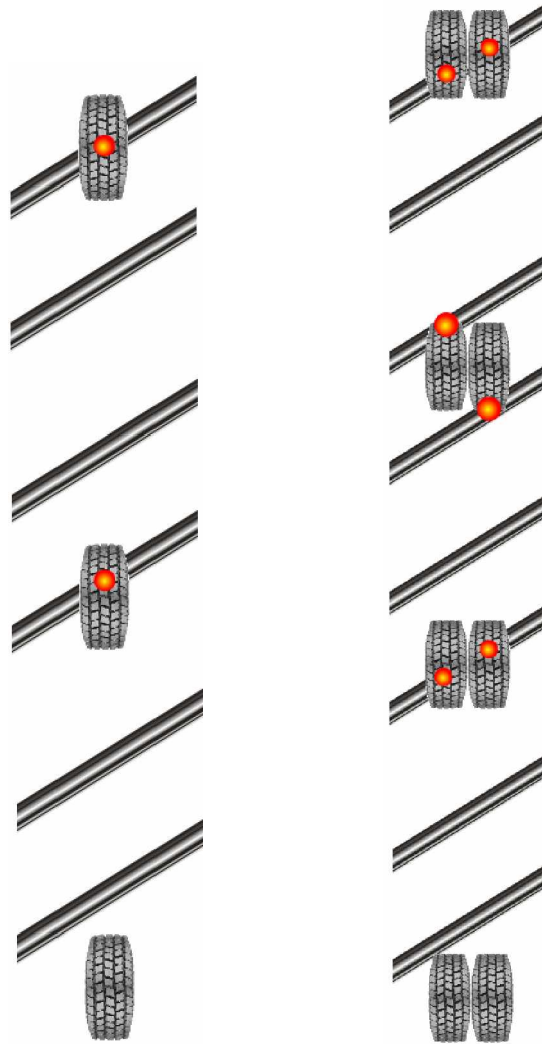


Figura 3.4. Discriminación del Número de Ejes a Través de la Peanas.

Sensor de presencia vehicular (Loop)

Consiste de un cable que forma lazos y que se encuentra ubicado por debajo de la superficie de rodamiento (pavimento), para el desarrollo de este proyecto este sensor se encuentra ubicado en la caseta de cobro bajo la barrera de acceso y/o salida, se utiliza para evitar que la barrera de acceso caiga sobre el vehículo cuando este aún no ha abandonado la caseta de cobro. Éste cable se encuentra interconectado a un controlador (PLC) situado en la caja interna de la barrera de acceso. El controlador detecta o mide la inductancia del lazo o circuito, esto indica que un vehículo ha sido detectado.

Barrera óptica

La barrera óptica es un dispositivo utilizado para diferenciar un vehículo de otro y para iniciar la cuenta del número de ejes de los vehículos. Es muy útil, sobre todo, cuando existe alto tráfico en el carril.

Cuando es interrumpido el paso de la luz (rayos) por un vehículo, el sensor envía una señal al módulo PowerCoreFlex de carril indicando el inicio de un vehículo y el principio de la cuenta del número de ejes, se encuentra ubicado al inicio del carril 1m antes de los sensores detectores de ejes. Esta señal de detección de vehículo es continua, hasta que el vehículo deja el carril de peaje, entonces los sensores indican fin de vehículo y se reinicia la cuenta del número de ejes para el siguiente vehículo.

Debido a que los elementos sensores actúan como un switch, se uso el mismo circuito para la entrada de estos al controlador Rabbit con excepción del Loop, se lo realizo a través del negador 74LS14 ya que este está alimentado con 5V y cuando una señal es activada su estado cambia de esta manera el controlador Rabbit lee la información de los sensores.

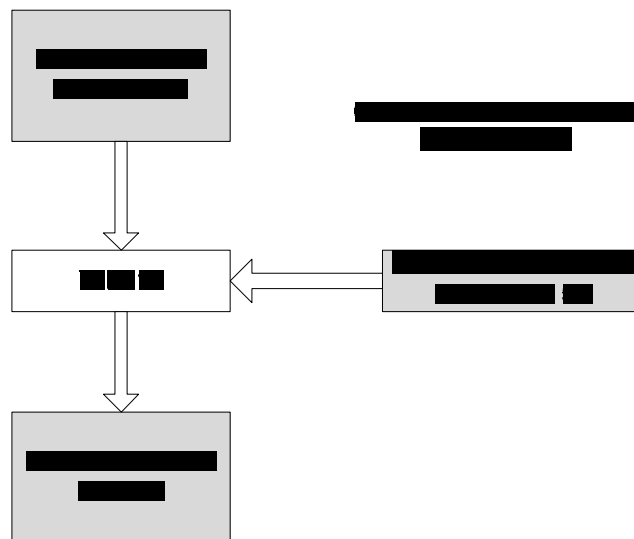


Figura 3.5. Diagrama de Disposición Electrónica de los Elementos Sensores

3.3 DISPOSICIÓN DE LA CABINA DE COBRO Y BARRERA DE ACCESO Y/O SALIDA

Cabina de Cobro

El equipo de cabina se compone de diversas herramientas que el peajista utilizará durante el turno de operación que le corresponda algunas de estas herramientas son: Monitor, teclado, impresora, rollos de papel, etc.

Aquí también se encuentra ubicada la PC de control, la cual realiza la comunicación con el módulo PowerCoreFlex, contiene todos los elementos de control y detección vehicular que requiere el sistema de preclasificación, además cuenta con la base de datos en la cual se registran todos los eventos que ocurren en el carril.

Debido a que este proyecto se enfoca en un prototipo de estación de peaje se sugiere que el gabinete este ubicado a la mitad del carril y este construido con lamina de acero inoxidable con tratamiento anticorrosivo que garantice una vida útil mínima de 7 años, en condiciones de ambiente salino y humedad relativa del 50%. Debe contar con un sistema de alarma para saber cuando existe una apertura por mantenimiento o en casos no autorizados; debe contar además con chapas de seguridad para restringir el acceso a personas no autorizadas.

Lo mencionado anteriormente garantizará condiciones de trabajo extremas y además facilitará las labores de limpieza, permitiendo al mismo tiempo el fácil acceso para el mantenimiento. La figura 3.6 muestra la parte interna de la cabina de control junto con sus herramientas. La disposición se muestra en el Anexo (1).

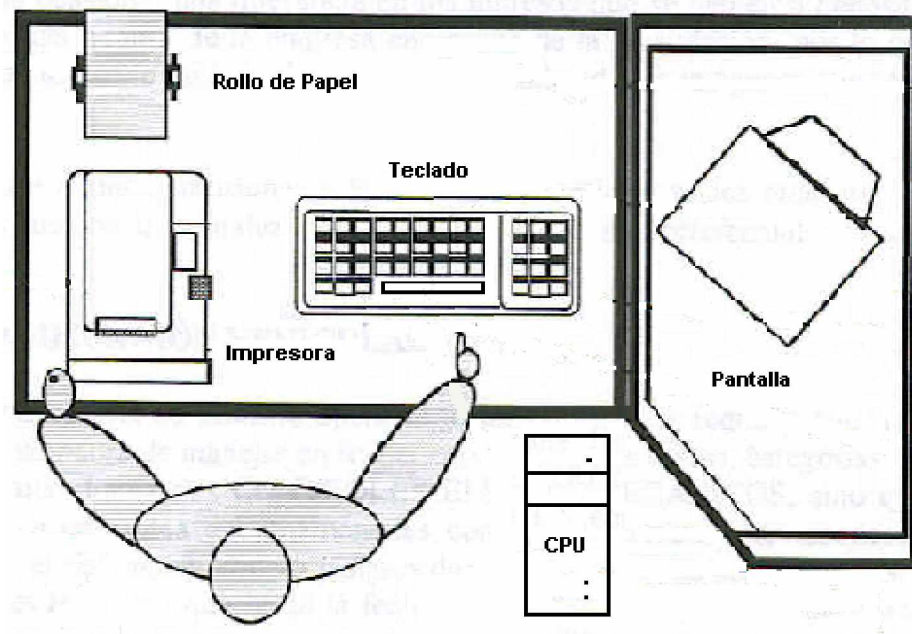


Figura 3.6. Parte Interna de la Cabina de Control

Barrera de acceso y/o salida

Es la encargada de controlar el acceso al carril, además detiene el tráfico hasta que el usuario cancele la tarifa a pagar, está compuesta por un mástil de 4m, con sus respectivos fines de carrera y el sensor de presencia vehicular, todo esto acoplado a un controlador PLC Siemens Logo el cual recibe la información del controlador PowerCoreFlex para abrir o continuar cerrado una vez confirmado el pago en la cabina de cobro. Esta barrera se encuentra al final del carril luego de la estación de cobro.

Está diseñada para trabajar 24 horas/días y 365 días/año. Sus características de fabricación la hacen ideal para trabajar indistintamente a la intemperie o en recintos cerrados.

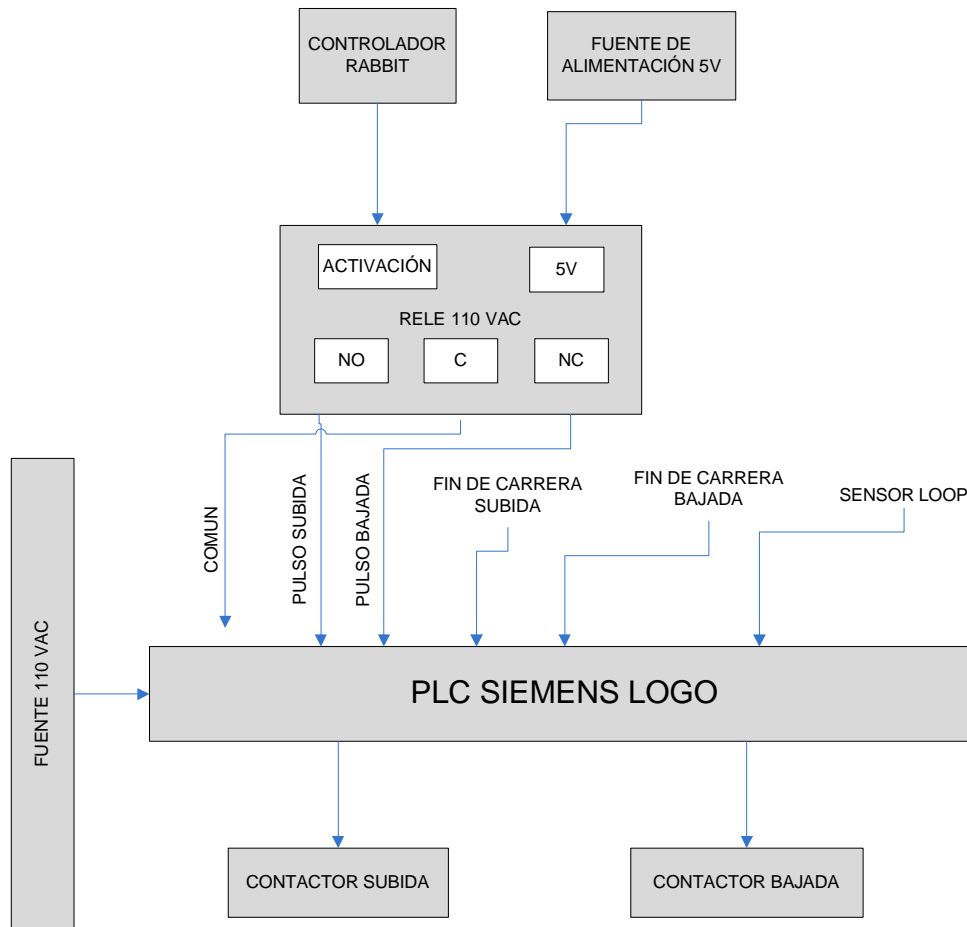


Figura 3.7. Diagrama de Disposición Electrónica de la Barrera de Acceso

3.4 DISPOSICIÓN DE SEÑALES LUMINOSAS Y VISUALIZADOR DE LA TARIFA

Las señales luminosas son indispensables en el manejo de sistemas de control de tráfico, es por esta razón que la estación de peaje posee todo un claro nivel de señalización debido a que la ubicación de los mismos desempeña un papel trascendental para la visibilidad de los usuarios. A continuación se detallan las señales luminosas.

Semáforo de techumbre

Dispositivo eléctrico que sirve para informar al usuario que carril está abierto o cerrado, ordenar y regular el tránsito de vehículos y peatones por un carril de cobro utilizando colores rojo y verde. Se encuentra en la parte superior del techo de la estación de cobro. Cabe mencionar que el semáforo de techumbre se activa de forma manual debido ya que no solamente indica si un carril está habilitado o no y no participa en el proceso de clasificación del vehículo.

Semáforo de Carril

Dispositivo eléctrico que sirve para ordenar y regular el tránsito de vehículos y peatones por un carril de cobro utilizando colores rojo y verde. Es utilizado para indicar a los vehículos que han cancelado su tarifa y que pueden abandonar la zona de cobro. Es activado por el módulo PowerCoreFlex a través de un relé una vez confirmado el pago en la caseta de cobro al igual que la barrera de acceso. El semáforo de carril se encuentra ubicado junto a la barrera de acceso al final del carril. La figura 3.3 muestra la disposición electrónica del semáforo de carril.

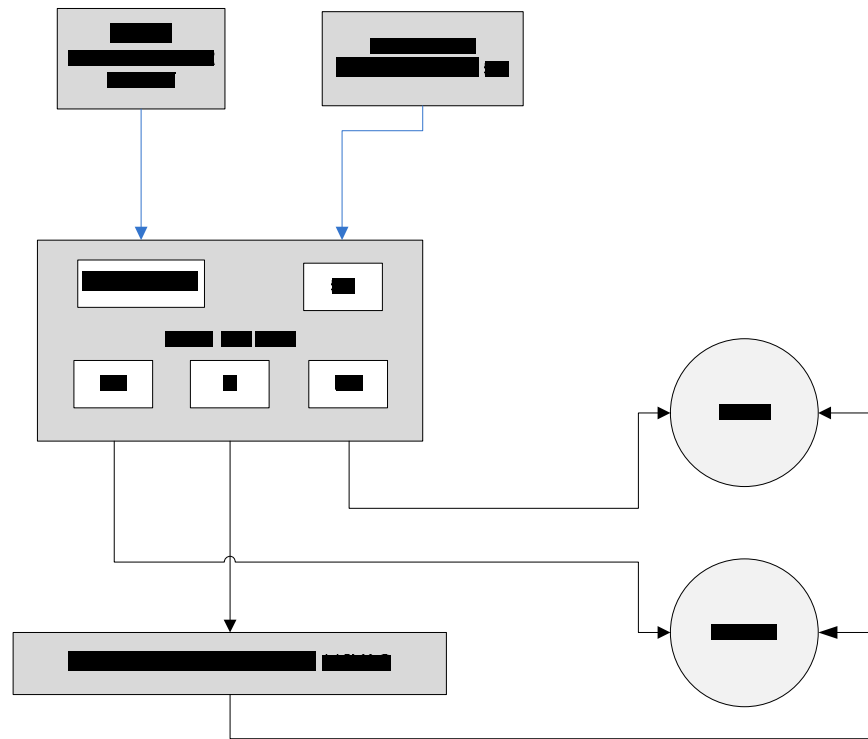


Figura 3.8. Diagrama de Disposición Electrónica del Semáforo de Carril

Display de Visualización

Permite desplegar la tarifa a cancelar a los usuarios de la autopista. Una vez hecha la clasificación el valor se envía a través del puerto serial del módulo PowerCoreFlex. Se ubica junto al semáforo de carril y ofrece una excelente visibilidad en cualquier hora del día y en circunstancias de escasa luminosidad. La figura 3.9 indica el modo de conexión y su disposición.

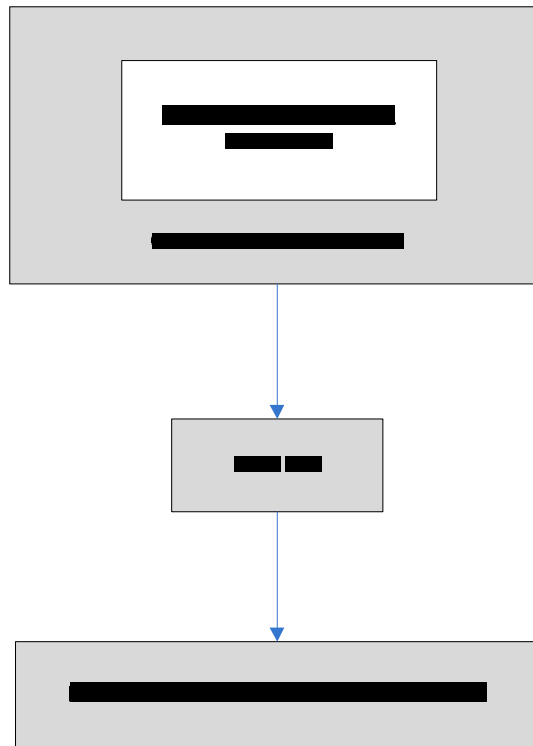


Figura 3.9. Diagrama de Disposición del Display

El anexo 1 muestra la disposición de todos los elementos de carril mencionados.

3.5 DISEÑO DEL CONTROLADOR

El control del sistema de clasificación automática de peaje es ejecutado por el controlador Rabbit con su módulo PowerCoreFlex debido a que posee los recursos necesarios, mismos que han sido considerados dentro del diseño general. La disposición y uso está determinado de acuerdo al número de señales a controlar.

En el diseño y avance del proyecto se tomo en cuenta el número de entradas y salidas de los dispositivos sensores y equipo que maneja en conjunto todo el sistema de clasificación vehicular, de esta manera se tomo los puertos A y B del controlador correspondientemente, además se especifica el tipo de comunicación para poder realizar

la clasificación vehicular mediante el puerto TCP/IP y el manejo del puerto serial para la visualización de la tarifa.

Adicionalmente se diseño una placa de control donde se integro el módulo PowerCoreFlex para el manejo adecuado de las señales. La placa de control integra las señales de entradas de los dispositivos sensores con sus respectivas protecciones y también integra las señales de salida para las funciones requeridas del carril de peaje.

3.5.1 PROGRAMACIÓN DEL CONTROLADOR

3.5.1.1 CONSIDERACIONES Y DIMENSIONAMIENTO

En la actualidad los sistemas de control embebido poseen los recursos necesarios incorporados en el mismo controlador, lo que facilita el manejo del dispositivo para el control de señales digitales dirigida a esta aplicación.

El módulo PowerCoreFlex posee líneas de entrada y salida configurables. El sistema prototipo de peaje requiere 3 líneas de entrada y 4 de salida, además se requiere inmunidad al ruido y un sistema robusto debido a que debe trabajar 24 horas y 365 días al año.

Se debe tomar en cuenta los sistemas de protección para el controlador y las condiciones ambientales de temperatura y humedad en la que va a trabajar para prever las condiciones de mantenimiento ubicación y ventilación. Esto es fundamental para el funcionamiento del prototipo. La tabla 3.1 describe la disposición de cada una de las entradas y salidas con sus correspondientes puertos:

TIPO	DESCRIPCIÓN	PUERTO
ENTRADA	BARRERA IR	PA0
ENTRADA	PEANA 1	PA1
ENTRADA	PEANA 2	PA2
SALIDA	AUXILIAR	PB5
SALIDA	SEMAFORO	PB6

ALIDA	BARRERA	PB7
SERIAL		PC2
SERIAL		PC3

Tabla 3.1 Descripción de Puertos en Controlador Rabbit

3.5.2 DYNAMIC C

Dynamic C es un sistema integrado de desarrollo para escribir software embebido, se fundamenta en la compilación de código en lenguaje C y grabación en la memoria de programa del controlador Rabbit con su módulo, posee sus propias librerías para el manejo respectivo de cada uno de los recursos del dispositivo.

La programación del módulo PowerCoreFlex se lo lleva a cabo a través de esta herramienta que utiliza la versión Dynamic C 9.20, y mediante la cual se desarrolla la lógica del sistema de clasificación automática. La comprensión de este ambiente de trabajo es fundamental para el desarrollo de este proyecto. La figura 3.2 muestra la pantalla principal de la interfaz de Dynamic C en donde se indican las funciones de los botones principales.

```

Dynamic C Version 9.20 - [C:\DOCUMENTOS AND SETTINGS\FERNANDO\MIS DOCUMENTOS\FERNANDO\TESIS\SOFTWARE\16-V-2008 V2.C]
File Edit Compile Run Inspect Options Window Help
#define DLYSEMAFORO 4//delay para activacion del semaforo
#define DIR_IP "192.168.100.100"//direccion IP del controlador
#define MASCARA "255.255.255.0"//mascara del controlador
#define PUERTO 13000//puerto de comunicacion
#define ACT 1
#define INACT 0

// __ PROTOTIPO DE FUNCIONES
void Ini_ES();//Inicializacion de entradas salidas
int Lec_Ent(int entrada);//Lectura de entradas
int Esc_Sal(int salida, int estado); //Escribe salidas
cofunc void Wr_Coni(char *buf, int tam);//Escribe en el puerto serial
void Ini_Serial();//Inicializa el puerto serial

// __ VARIABLES GLOBALES
unsigned char tipoAuto, pile[MAX_FILA], indiceFila;//almacena el tipo de auto que paso por el peaje.
unsigned char dolares, centavos;
CoData analisis, IR, ini_disp;//tarea para el analisis del proceso.

//TCP/IP

```

Figura 3.10. Pantalla Principal de Dynamic C



Este comando compila y ejecuta el programa, es posible realizar esta acción mediante el uso de la tecla F9.

Luego de realizar la compilación y ejecución se despliega una ventana llamada Stdio, la misma sirve como verificación, indica las salidas en caso de haber utilizado la instrucción *printf()*.



Compilar únicamente el programa, no lo ejecutarlo. Se puede hacer uso de esta acción mediante la tecla F5.



Permite detener la ejecución del programa, mientras se encuentre conectado a la tarjeta.



Sirve para realizar cambios en el código del programa



Permite moverse en el programa en ejecución a través de las instrucciones, sin ingresar a los bucles.



Sirve para mirar expresiones en el código del programa en ejecución.

3.5.3 DIAGRAMA DE BLOQUES

El controlador Rabbit realiza el control del sistema de preclasificación, se encarga de recibir las señales de los dispositivos sensores, procesarlas y enviar las señales de salidas a los actuadores correspondientes. Para comprender de mejor manera y de forma general el funcionamiento del sistema controlador se lo hace mediante bloques los que definen la organización de todo el proceso interno, sus entradas y sus salidas. La figura 3.11 muestra el diagrama de bloques del controlador Rabbit.

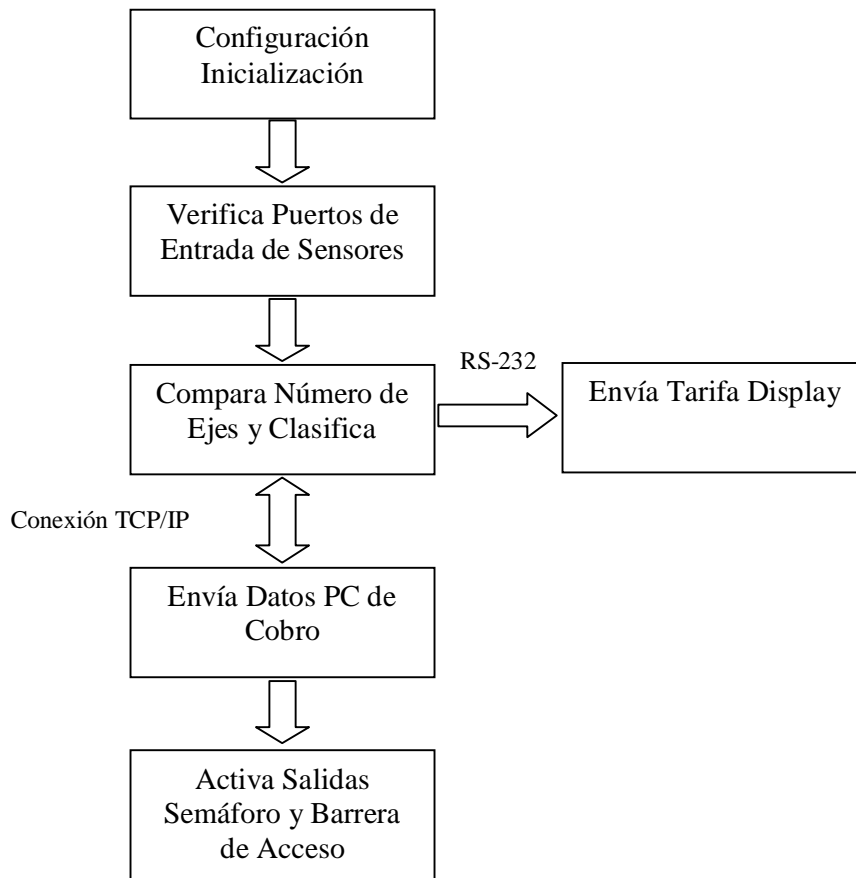
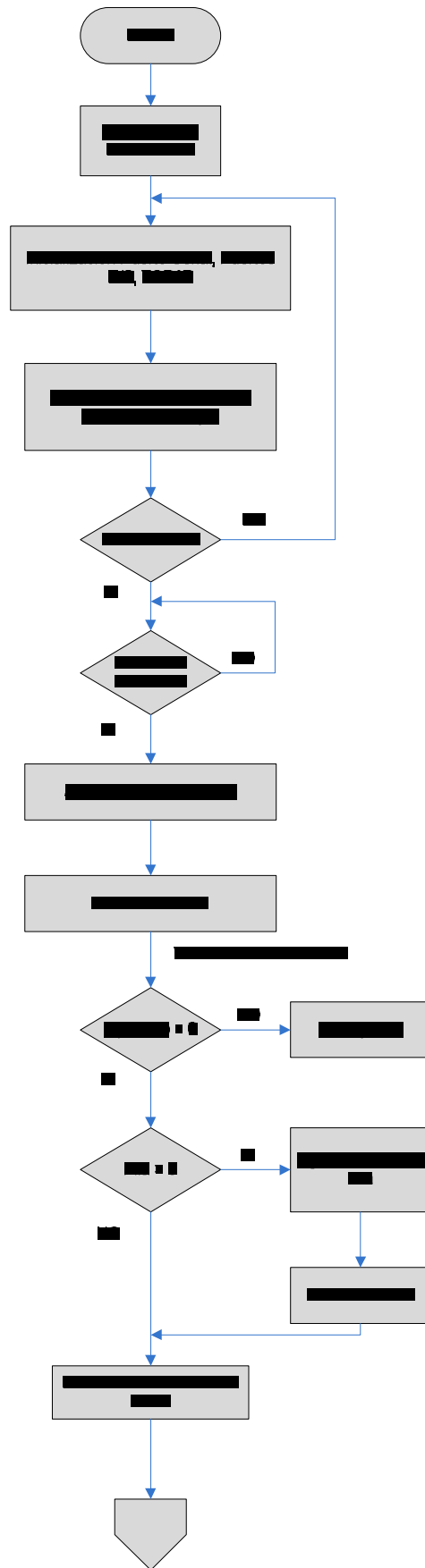


Figura 3.11. Diagrama de Bloques de Funcionamiento del Controlador

3.5.4 DIAGRAMA DE FLUJO

Una vez descrito el comportamiento del sistema de clasificación vehicular de manera general, se continúa con el desarrollo del software que es la parte esencial, es así que durante el progreso del mismo se acoplan varias funciones para la optimización del código, el desarrollo completo se encuentra en el código main(). La figura 3.4 muestra el diagrama de flujo general del sistema de clasificación.



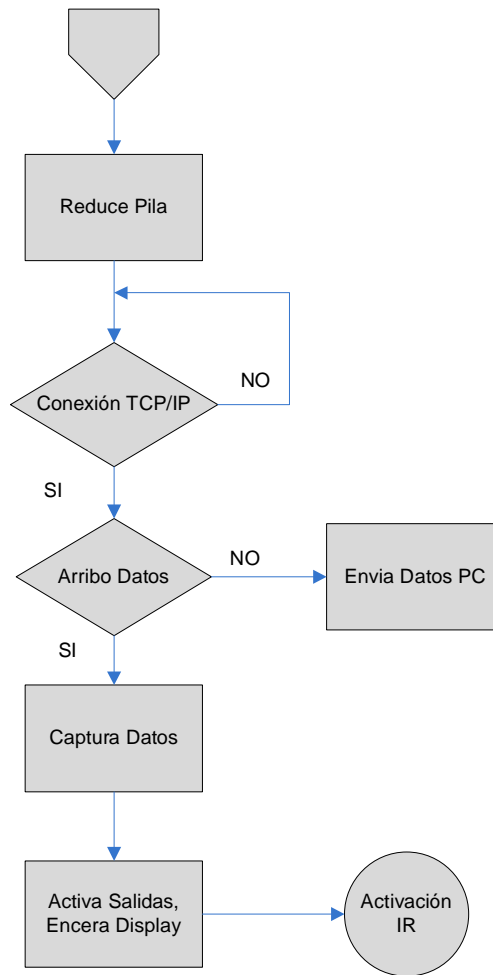


Figura 3.12. Diagrama de Flujo General del Sistema

3.5.5 PROGRAMACIÓN DE LA TARJETA

La programación de la tarjeta se la realiza mediante el uso de funciones, las que se utilizan para inicializar las entradas y salidas, lectura de entradas y escritura de salidas esto se lo realiza para que el encabezado del programa se reduzca y sea fácil para realizar cualquier modificación.

En la cabecera del código se define las constantes para facilitar la programación, se define las funciones a utilizar y las variables globales que actúan en el desarrollo del programa. Además se llama a las librerías y declaraciones de inicialización de los

puertos de entrada y salida y de cada recurso utilizado como son TCP/IP y transmisión serial. A continuación se muestra el encabezado y declaraciones del programa.

```
//PEAJE

#use "PowerCoreFLEX.lib"

//__CONSTANTES
#define COM_BAUD 115200 //Velocidad de comunicación del puerto serial
#define CINBUFSIZE 255 //Es el tamaño máximo de una trama RTU
#define COUTBUFSIZE 255
#define BARRERA_IR 0 //Entrada del puerto A donde se conecta la barrera
#define PEANA1 1 //Entrada del puerto A donde se conecta la peana1
#define PEANA2 2 //Entrada del puertoA donde se conecta la peana2
#define SEMAFORO 6 //Salida del puertoB donde se conecta el semáforo
#define BARRERA 7 //Salida del puertoB donde se conecta la barrera mecánica
#define MAX_PILA 10//nuero máximo de autos que se pueden mantener en memoria.
#define DLYBARRERA 1//delay para el pulso de la barrera
#define DLYSEMAFORO 4//delay para activación del semáforo
#define DIR_IP "192.168.100.100"//dirección IP del controlador
#define MASCARA "255.255.255.0"//mascara del controlador
#define PUERTO 13000//puerto de comunicacion
#define ACT 1
#define INACT 0

//__PROTOTIPO DEFUNCIONES
void Ini_ES();//Inicializacion de entradas salidas
int Lec_Ent(int entrada);//Lectura de entradas
int Esc_Sal(int salida, int estado); //Escribe salidas
cofunc void Wr_Com1(char *buf, int tam);//Escribe en el puerto serial
void Ini_Serial();//Inicializa el puerto serial

//__VARIAIBLES GLOBALES
unsigned char tipoAuto, pila[MAX_PILA], indicePila;//almacena el tipo de auto que paso por el
peaje.
unsigned char dolares, centavos;
CoData analisis, IR, ini_disp;//tarea para el analisis del proceso.

//__TCP/IP
#define TCPCONFIG 1//STATICO
#define SOCK_BUF_SIZE 512
#define MAX_SOCKETS 1
#define MAX_BUFSIZE_TCP 256

#use "dcrtcp.lib"
//Variables
tcp_Socket Socket; //Socket para comunicacion
//Funciones
//cofunc int Com_TCP(tcp_Socket *s, int puerto, void(*func)());

//__FUNCION PRINCIPAL
main()
{
```

```
unsigned char buf[MAX_BUFSIZE_TCP], aux, bufRS[25];
int length, i;
Ini_ES();
Ini_Serial();

//INICIALIZACION DE TCP
sock_init();
sock_mode(&Socket,TCP_MODE_ASCII);

//Encera Salidas
Esc_Sal(BARRERA,0);
Esc_Sal(SEMAFORO,0);

//inicializacion de variables
tipoAuto=0;//no hay ningun auto
indicePila=0;//no hay elementos en la pila

loopinit();
```

Configuración de Inicialización Controlador Rabbit.

A partir del encabezado, la secuencia de bucle infinito consta de dos partes principales que son la tarea de análisis del número de ejes y la conexión TCP/IP.

3.5.5.1 ANÁLISIS DEL NÚMERO DE EJES DE UN VEHÍCULO

Una vez que la función principal inicializa sus cabecera e ingresa al bucle infinito, se inicial la tarea de análisis del número de ejes de un auto, esta empieza cuando el sensor óptico (barrera IR) detecta la presencia de un vehículo en la zona de cobro. La barrera IR cambia de estado a ACTIVO y llama a la función análisis que es la encargada del proceso tal como se indica en la figura 3.5.

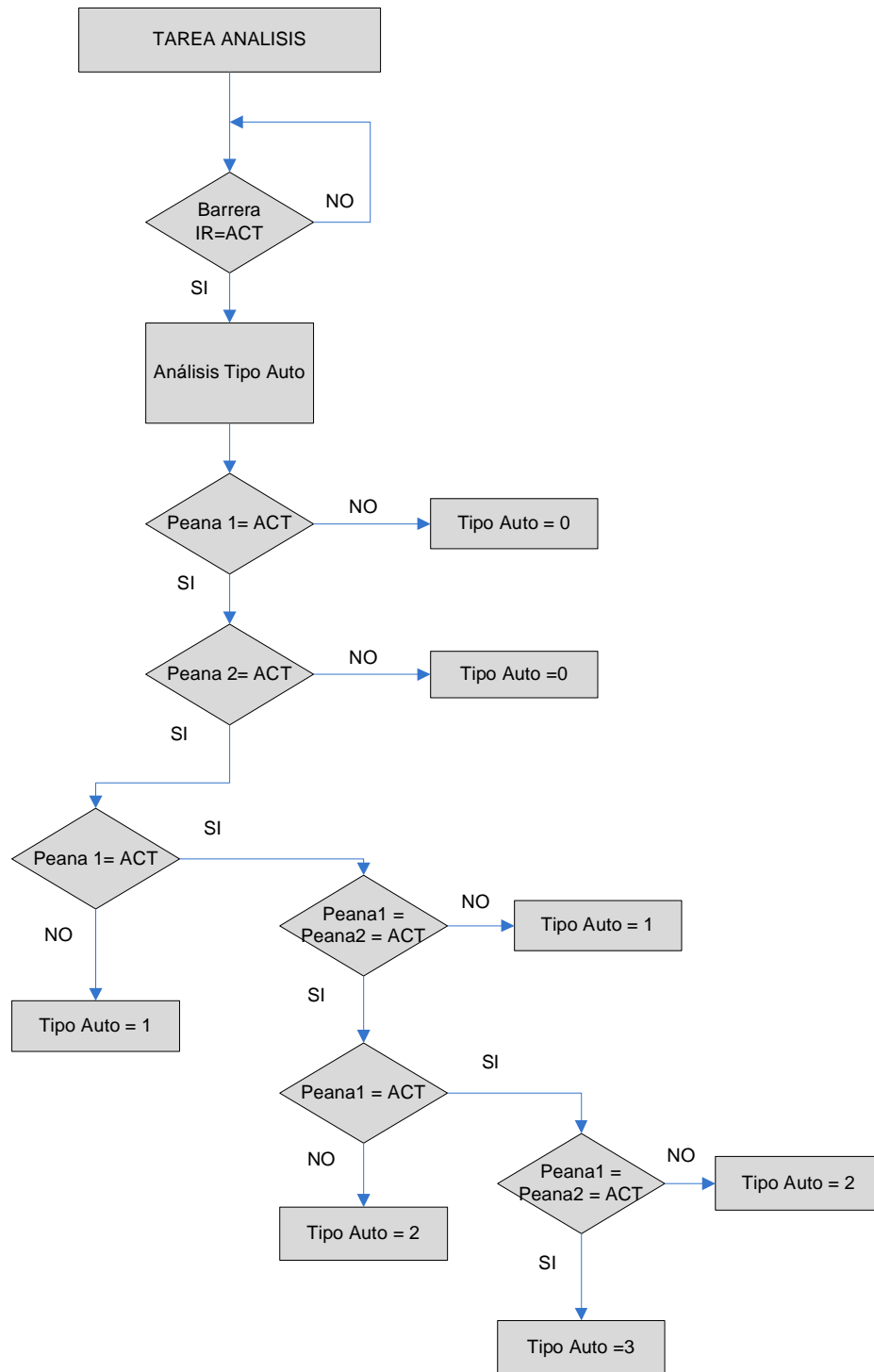


Figura 3.13. Diagrama de Flujo del Análisis del Número de Ejes de un Vehículo

Una parte de esta programación es descrita a continuación.

```

constate IR always_on
{
    printf("\nInicia Tarea IR");
    waitfor(Lec_Ent((int)BARRERA_IR)==ACT);//Espera a que se active la entrada (Entradas con
    pull up)
    waitfor(DelayMs(800)); //retaro para rebotes
    if(Lec_Ent((int)BARRERA_IR)==ACT) CoBegin(&analisis);//inicializa analisis del numero
    de ejes del auto
    else CoBegin(&IR);//reinicia tarea IR
    waitfor(Lec_Ent((int)BARRERA_IR)==INACT);//Espera a que haya pasado el auto
    CoPause(&analisis); //termina analisis
    if(tipoAuto>0) //verifica datos obtenidos
    {
        if(indicePila<MAX_PILA)
        {
            pila[indicePila]=tipoAuto; //ingresa dato a la pila
            indicePila++;//incrementa la pila
        }
    }
    printf("\nFinaliza tarea IR Tipo=%d",tipoAuto);
}

constate analisis init_on
{
    tipoAuto=0; //no hay auto

    printf("\nInicia Tarea Analisis");

    //Primer Eje
    waitfor(Lec_Ent((int)PEANA1)==ACT);//espera a que se active peana1
    printf("\nPrimer Eje");
    waitfor(DelayMs(800)); //retaro para rebotes
    waitfor(Lec_Ent((int)PEANA1)==INACT);//espera a que se desactive peana1

    tipoAuto=1;
    printf("\nTipo=%d",tipoAuto);

    //Segundo Eje
    waitfor(Lec_Ent((int)PEANA1)==ACT);//espera a que se active peana1
    printf("\nSegundo Eje");
    waitfor(DelayMs(800)); //retaro para rebotes
    while(Lec_Ent((int)PEANA1)==ACT)
    {
        if( (Lec_Ent((int)PEANA2)==ACT) )tipoAuto=2;
        yield;
    }
    //if( (Lec_Ent((int)PEANA1)==0) && (Lec_Ent((int)PEANA2)==0) )tipoAuto=2;
    waitfor(Lec_Ent((int)PEANA1)==INACT);//espera a que se desactive peana1
    printf("\nTipo=%d",tipoAuto);

    //Tercer Eje
    waitfor(Lec_Ent((int)PEANA1)==ACT);//espera a que se active peana1
    printf("\nTercero Eje");
    waitfor(DelayMs(800)); //retaro para rebotes
    while(Lec_Ent((int)PEANA1)==ACT)

```

```

{
    if( (Lec_Ent((int)PEANA2)==ACT) )tipoAuto=3;
    yield;
}
//if( (Lec_Ent((int)PEANA1)==0) && (Lec_Ent((int)PEANA2)==0) )tipoAuto=3;
waitfor(Lec_Ent((int)PEANA1)==INACT);//espera a que se desactive peana1
printf("\nTipo=%d",tipoAuto);
}

```

Análisis del Número de Ejes de un Vehículo

Una vez concluido el análisis los datos son enviados mediante comunicación TCP/IP a la PC de control para el cobro de la tarifa respectiva. A continuación se muestra las clasificaciones hechas por el dispositivo Rabbit. La tabla 3.2 muestra la clasificación del controlador Rabbit.

Tipo Auto = 0	No existe vehículo presente
Tipo Auto = 1	Vehículo Liviano
Tipo Auto = 2	Vehículo Pesado
Tipo Auto = 3	Vehículo Extra Pesado

Tabla 3.2. Clasificación del Tipo de Vehículos

3.5.5.2 CONEXIÓN TCP/IP

El sistema mediante el cual existe la comunicación entre el carril y la máquina de operación en la caseta de peaje, está basado en un protocolo de comunicaciones estándar tipo TCP/IP, es necesario que el dispositivo de control esté conectado a la red por lo que el sistema operativo del controlador está solicitando conexión con el programa máster constantemente para confirmar el arribo de datos en el buffer de entrada del puerto Ethernet, el cual se encarga de actualizar el tipo de vehículo que está pasando por el carril.

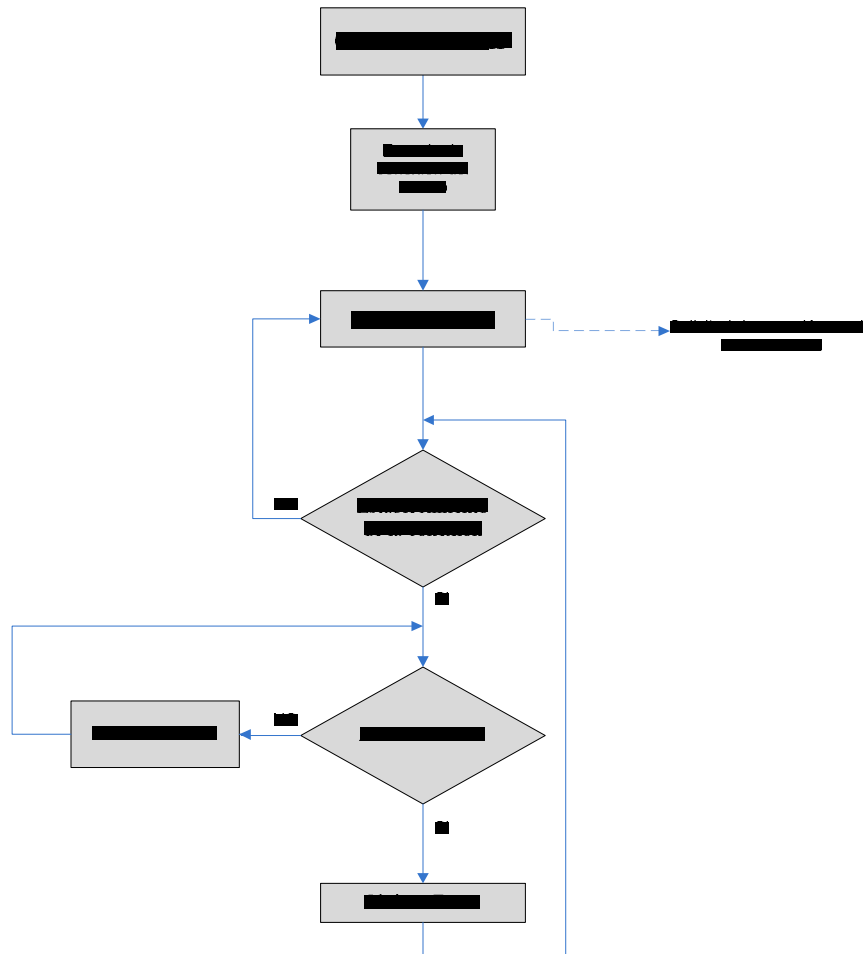


Figura 3.14. Diagrama de Flujo Conexión TCP/IP

La comunicación es de trascendental importancia para el funcionamiento del peaje. El Diagrama de flujo de la figura 3.9 muestra el proceso realizado para verificar el arribo y envío de datos.

```
//Comunicacion TCP/IP -----
costate
{
tcp_listen(&Socket,(int)PUERTO, 0, 0, NULL, 0);// escucha la llegada de una conexion en el
puerto
// espera por la coneccion
while((-1 == sock_bytesready(&Socket)) && (0 == sock_established(&Socket)))
{// suelta el control para que otras tareas se ejecuten
//printf("\nSocketEstablished=%d",sock_established(&Socket));
yield;
}
while(sock_established(&Socket))
```

```
{
// obtener datos
length = sock_fastread(&Socket, buf, MAX_BUFSIZE_TCP);
if(length > 0)
{ //si se recibio datos
printf("\nLeidos=%d\n",length);
for(i=0;i<length;i++) printf("%d ",buf[i]);
}
}
```

Programación de la Conexión TCP/IP

Toda la información del carril desde la tarjeta de control embebido hacia la PC de control es verificada a fin de que cada uno de los eventos llegue completo. En esta forma la posibilidad de pérdida de eventos es muy reducida.

3.6 DISEÑO DE LA INTERFAZ GRÁFICA

El programa considerado para la Interfaz gráfica es Visual Basic, se debe tomar en cuenta una serie de consideraciones con la finalidad de identificar cuáles son los alcances del programa en función a las necesidades debido a que es necesario tener un control visual del funcionamiento del sistema.

El HMI es el programa responsable de discriminar las distintas clasificaciones de vehículos e importe de peaje, es sencillamente una herramienta que lleva un control preciso del aforo marcado por el peajista y de manera paralela el registro del equipo de clasificación automática. Su función es concentrar la información de manera que esta se encuentre disponible en completo apego a los posibles reportes.

El software presenta una valiosa herramienta que servirá para identificar claramente las posibles discrepancias entre el equipo de piso y el peajista, ya que tiene la posibilidad de captar las imágenes de todas las transacciones, es decir, la clasificación de pago acorde a la clase de vehículo.

El software desarrollado brinda un conjunto de soluciones para conocer el aforo vehicular por turno, día, o periodos superiores de tiempo, el importe del recaudo por turno y las posibles discrepancias del ingreso en plata producto de mala clasificación ya sea del sistema o bien del peajista. También permitirá al peajista justificar los vehículos

marcados como pagados cuando estos no hayan cubierto el importe del peaje correspondiente.

3.6.1 CONSIDERACIONES

Para el desarrollo de la interfaz del proyecto han sido considerados los siguientes aspectos:

- Diseño de la base de datos
- Definición de las propiedades de los controles
- Generación del código asociado a los eventos que ocurran en el carril.
- Generación del código del programa.
- Creación de un interfaz de usuario.

3.6.2 BASE DE DATOS

La base de datos es la encargada de guardar la información administrativa y de operación del carril, se realiza mediante MySQL, ya que es un sistema que permite el desarrollo y tratamiento de datos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

3.6.2.1 DISEÑO DE BASE DE DATOS

Existen distintos modos de organizar la información y representar las relaciones entre los datos en una base de datos. Los Sistemas administradores convencionales usan bases de datos para hacer seguimiento de las entidades, atributos y relaciones.

En el desarrollo de este proyecto el almacenamiento de datos es indispensable para la administración del mismo, es así que se vuelve necesaria la creación de una base de datos que almacene las funciones principales de administración y operación del carril a fin de que se registren todos los eventos que ocurren en el mismo y en forma particular en los que existe reclasificación.

Con el fin de que la base de datos sea realmente funcional, debido a que debe almacenar grandes cantidades de registros y además sea de fácil acceso se uso el programa MySQL que es un software gratuito para bases de datos

MySQL es un software muy completo capaz de satisfacer las necesidades de almacenamiento del sistema de clasificación vehicular, puede realizar un enlace con visual Basic a través de un conector ODBC (Open Database Connectivity) es un estándar de acceso a Bases de Datos desarrollado por Microsoft, el objetivo de *ODBC* es hacer posible el acceder a cualquier dato desde cualquier aplicación.

La base de Datos posee cinco tablas básicas que almacenan los datos necesarios para la administración y operación del carril de peaje.

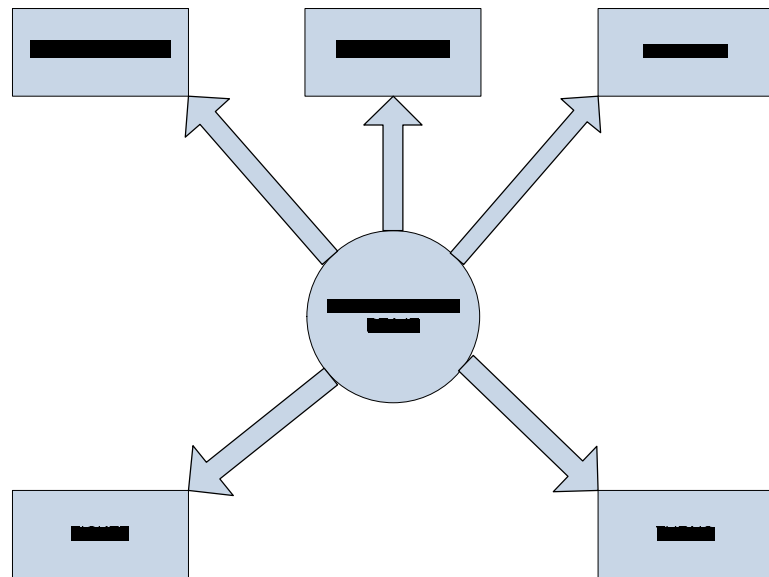


Figura 3.15 Diagrama de flujo de distribución de Base de Datos

Cada una de las tablas se creó a partir de las necesidades de almacenamiento del programa, debido a esto las tablas constan de los siguientes campos:

Tabla	Campo	Tipo	Descripción
Configuración	conf_cod	Int	Código de Facturación
	conf_iva	Varchar	Tarifa de Iva
	conf_auto	Float	Tarifa autos livianos
	conf_dos	Float	Tarifa autos de dos ejes
	conf_tres	Float	Tarifa autos de tres ejes
	conf_remolque	Float	Tarifa para autos con remolque
	conf_ruc	Varchar	Número de Ruc de la Empresa
	conf_ciudad	Varchar	Ciudad en la que opera la Empresa
	conf_denominación	Varchar	Denominación que consta en el Ruc
	conf_nom	Varchar	Nombre comercial
	conf_dir	Varchar	Dirección de la Empresa
	conf_dirfact	Varchar	Dirección que consta en la Factura
	conf_num	Varchar	Número de autorización del SRI
	conf_validez	Date	Fecha de validez de la autorización
conf_comp	Varchar	Denominación del Comprobante	
Parámetros	par_cod	Int	Configuración de Parámetros
	par_nom	Varchar	Nombre de Peaje
	par_ubi	Varchar	Ubicación de Peaje
	par_res	Varchar	Responsable del Peaje
	par_tra	Varchar	Tramo del Peaje
	par_logo	Varchar	Logotipo
Ticket	tic_cod	Double	Número de Ticket
	tic_val	Float	Valor del Ticket
	tic_fec	DateTime	Hora y fecha de emisión del Ticket
	tic_op	Varchar	Operador de turno
	tic_tipo	Varchar	Tipo de Vehículo
	tic_pago	Varchar	Tipo de Pago
	tic_pic	Varchar	Imagen del Vehículo
	tic_op_ced	Varchar	Número de identificación del

			Operador
Turno	tur_cod	Double	Código de Ticket
	Turno	Varchar	Tipo de Turno
	tur_tic_ini	Double	Ticket de Inicio de turno
	tur_tic_fin	Double	Ticket de fin de turno
	tur_fec_ini	DateTime	Fecha y hora de inicio de turno
	tur_fec_fin	DateTime	Fecha de cierre de turno
	tur_op	Varchar	Operador de Turno
	tur_val	Double	Efectivo generado en el turno
	tur_val_er	Double	Efectivo generado por error
	tur_bol	Double	Boletos generados en el turno
	tur_bol_er	Double	Boletos generados por error
Usuario	usu_cedula	Char	Cedula del personal
	usu_nombre	Varchar	Nombre del personal
	usu_apellido	Varchar	Apellido del personal
	usu_direccion	Varchar	Dirección del personal
	usu_telefono	Varchar	Teléfono
	usu_celular	Varchar	Celular
	usu_cargo	Varchar	Cargo
	usu_user	Varchar	Nombre de usuario
	usu_pass	Varchar	Contraseña

Tabla 3.3 Distribución de campos en las tablas de la base de datos

3.7 DISEÑO HMI

Para iniciar la operación del HMI se necesita estar dentro de Windows XP, ejecutar el software y automáticamente se pondrá en marcha el software de administración y operación del carril.

3.7.1 MENÚ PRINCIPAL

El menú principal localizado en la parte superior izquierda de la pantalla está compuesto por cuatro submenús los que se describen a continuación:

Peaje

- Parámetros
- Tarifas
- Personal de Peaje
- Entrada
- Cambiar Clave
- Salir

Procesos

- Iniciar Operación
- Cerrar turno

Reportes

- Reporte por turno
- Reporte por Fecha

Personal

- Operarios

Para ingresar a estos menús se puede utilizar el teclado empleando la tecla ALT + P para ingresar al sub menú de peaje, estando dentro del sub menú es posible cambiar de renglón o columna a través de las flechas. Para identificar la opción a la que se desea ingresar se resaltara dentro de una casilla de color.

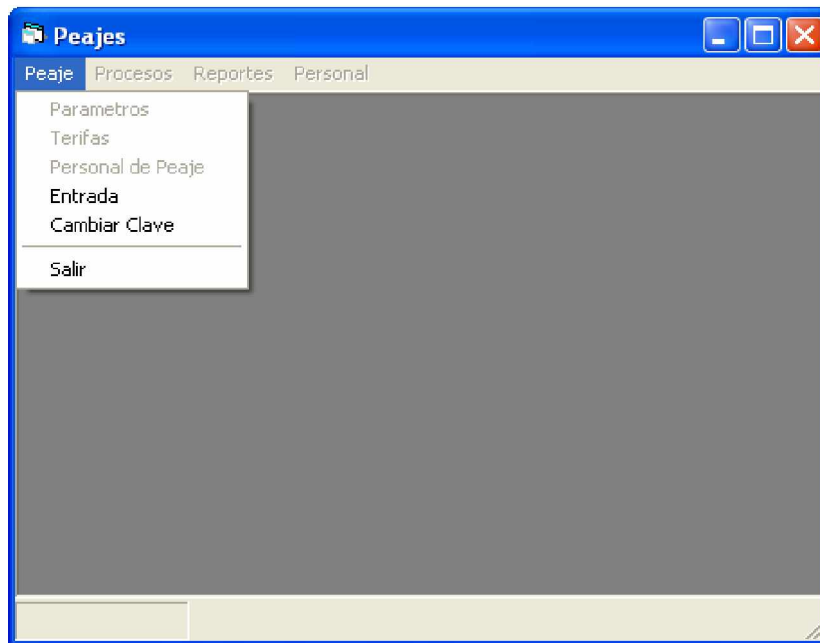


Figura 3.16 Menú Principal de HMI

a) PEAJE

El ingreso del usuario se realiza previo al manejo del programa como tal por medio de una pantalla auxiliar en la cual aparece un recuadro en el cuál se ingresara un nombre clave o Id, para posteriormente ingresar el password.



Figura 3.17 Pantalla de ingreso de usuario

- **Parámetros**

Al seleccionar la opción de parámetros se despliega una pantalla para la configuración de identificación de la caseta de peaje. En esta pantalla se configura el nombre de la caseta de peaje, la ubicación, el nombre del funcionario responsable de la caseta de peaje, el tramo concesionado, el logotipo de la empresa operadora y administradora y los horarios en que se dará inicio y fin a las operaciones de turnos de ocho horas, indicando la hora en que deben iniciarse y finalizar los turnos de operación.

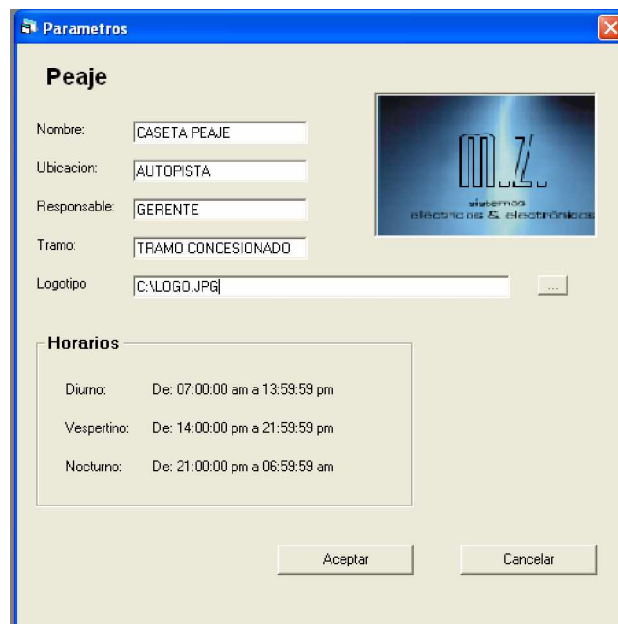


Figura 3.18 Pantalla de Configuración de Parámetros

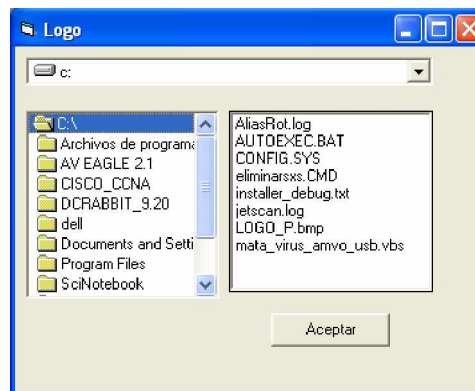


Figura 3.19 Pantalla de Selección del Logotipo de la Empresa Administradora

- **Tarifas**

En el sub menú tarifas se ingresa la configuración para la facturación del servicio de rentas internas SRI, además se integra la sub clasificación aplicable para las tarifas de los vehículos usuarios del peaje.



The screenshot shows a software window titled 'Tarifas' with a blue title bar. The main content area is titled 'Configuración' and is divided into two sections: 'Facturación' and 'Tarifas'.

Facturación Section:

- Numero de RUC: 1803665962001
- Ciudad: QUITO
- Denominación según consta en el RUC: MZ SISTEMAS
- Nombre Comercial: MZ SISTEMAS
- Dirección Casa Matriz: JUAN LEON MERA Y ROBLES
- Dirección emite factura: AUTOPISTA
- Nro. Autorización SRI: 0000001
- Validez/autorizac hasta: 2008-10-22
- Denom comprobante: 000000000000000000000001
- IVA: 12

Tarifas Section:

- Autos Livianos: 1.00
- Vehiculos Pesados 2 ejes : 2.00
- Vehiculos Pesados 3 ejes: 3.00
- Remolques: 1.50

Buttons for 'Guardar' and 'Cancelar' are located to the right of the 'Tarifas' section.

Figura 3.20 Pantalla de Configuración de Facturación y Tarifas

- **Personal de Peaje**

En el sub menú Personal de peaje se despliega una pantalla para el ingreso de nuevos usuarios, posee cajas de texto para el ingreso de los datos y botones llamados por la función que cumplen según el rango del empleado ya sea operador o administrador.



Creacion de Usuarios

Personal

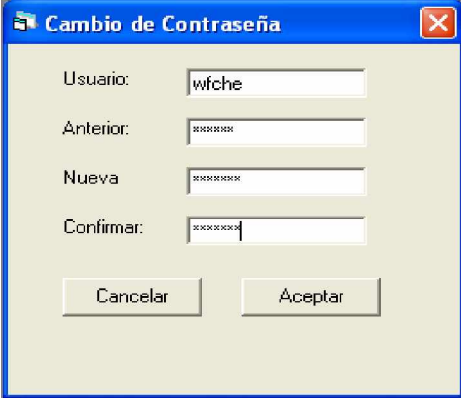
Cedula: 1803558418
Nombre: FERNANDO
Apellido: CHILUISA
Direccion: TEODORO WOLF N 15-56
Telefono: 022905542
Celular: 098367752
Cargo: Administrador
Usuario: wfche
Contraseña: xxxxxxxx
Confirme Contraseña: xxxxxxxx

Cancelar Crear Usuario

Figura 3.21 Pantalla de Creación de Nuevos Usuarios

- **Cambiar Clave**

Esta opción permite a los operarios y administradores del sistema cambiar su clave de ingreso al sistema, para esto es necesario indicar el nombre de usuario y la contraseña anterior, para posteriormente registrar una nueva clave de acceso con la debida confirmación.



Cambio de Contraseña

Usuario: wfche
Anterior: xxxxxxxx
Nueva: xxxxxxxx
Confirmar: xxxxxxxx

Cancelar Aceptar

Figura 3.22 Pantalla Para el Cambio de Contraseña

- **Salir**

Para finalizar esta columna de sub menú se encuentra la opción de salir la que estará restringida para operadores y disponible únicamente para personal técnico de mantenimiento.

b) PROCESOS

En el sub menú procesos se encuentran las pantallas de operación y cierre de turno de la caseta de peaje, las cuales se describen a continuación:

- **Iniciar Operación**

La pantalla principal del interfaz posee los tipos de vehículos para la visualización de la clasificación con el fin de verificar la clasificación efectuada por el equipo preclasificador, además de tener la fecha, hora, estado de conexión con el controlador Rabbit, nombre del operador, número de ticket o secuencia, si opera de forma manual o automática , además un botón para registrar el pago y la captación automática de imágenes y en caso de ser errónea la clasificación automática un botón de reclasificación para clasificar de forma manual el tipo de auto.

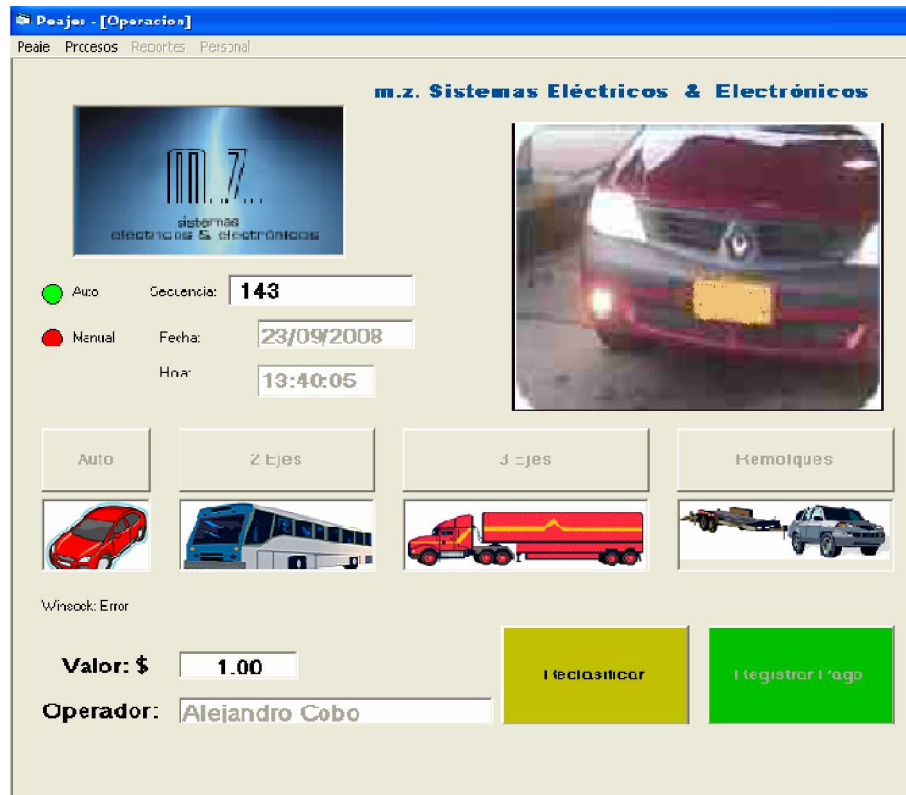
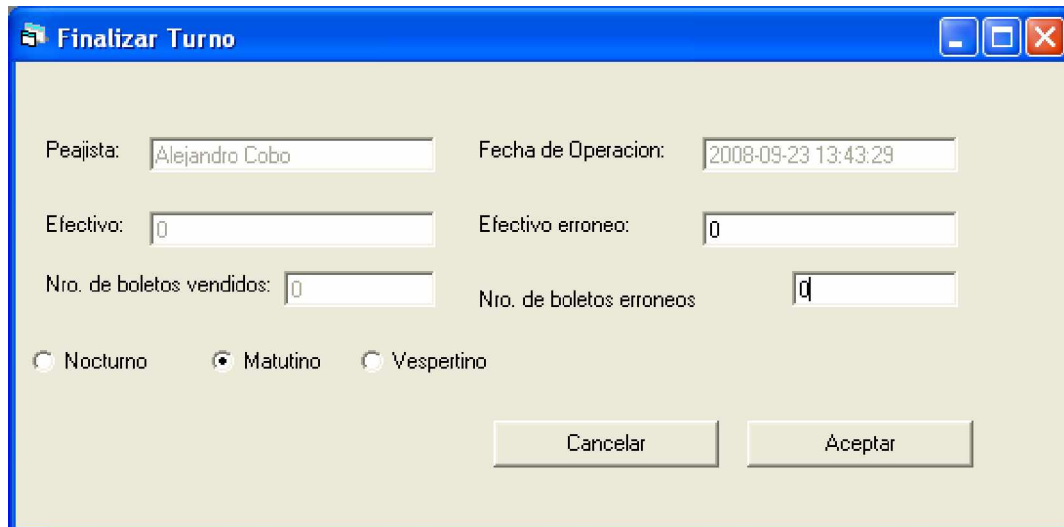


Figura 3.23 Pantalla de Operación Principal del HMI

- **Cerrar Turno**

Para el cierre de turno se toma como referencia los datos previamente almacenados del sistema de clasificación automática, una vez realizada esta operación se abre una pantalla de liquidación de turno en la cual se despliega el nombre del peajista, la fecha, la hora y el sistema genera el efectivo y el número de boletos vendidos, dejando para el peajista llenar los valores de efectivo erróneo en caso de haberlo y el número de boletos generados por error, además se debe escoger el turno en el cuál ha operado para poder realizar el cierre de caja.



Finalizar Turno

Peajista: Alejandro Cobo Fecha de Operacion: 2008-09-23 13:43:29

Efectivo: 0 Efectivo erroneo: 0

Nro. de boletos vendidos: 0 Nro. de boletos erroneos: 0

Nocturno Matutino Vespertino

Cancelar Aceptar

Figura 3.24 Pantalla de Liquidación de Turno

c) REPORTES

La pantalla de reporte es necesaria para obtener la información de todos los eventos que ocurren en el carril durante el turno de un operador, o según las fechas que se requiera, para ello se realiza una lectura de la base de datos, que se actualiza cada vez que se registra el paso de un vehículo por la estación de cobro.

- **Reportes por Turno**

Este tipo de reporte permite hacer una selección precisa de los registros deseados, en el cual se selecciona la fecha y el turno de operación. Una vez seleccionada la fecha de la que se requiere el reporte se procede a generarlo para su impresión.

	N° Ticket	Fecha y Hora	Operador	Clasificación	Valor	Tipo de pago	Imagen
1	69	01/06/2008 18:00:35	Fernando Canillo	Extra Pesado	3	Reclasificado	C:/imagenes/tcte20080601-18C005.jpg
2	70	01/06/2008 18:00:20	Fernando Canillo	Liviano	1	Reclasificado	C:/imagenes/tcte20080601-18C020.jpg
3	71	01/06/2008 18:00:30	Fernando Canillo	Liviano	1	Reclasificado	C:/imagenes/tcte20080601-18C030.jpg

Figura 3.25 Pantalla de Reportes Generados por Turno

- **Reportes por Fecha**

Este tipo de reporte nos provee un historial de eventos registrados en el carril, desde una fecha inicial hasta una fecha final, por lo que es posible indicar el periodo de tiempo ya sea de un día, una semana o más.

Peajes - [Reporte por Fechas]

Fecha de Inicio: 28/05/2008 Fecha de Fin: 04/06/2008

NºTicket	Fecha y Hora	Operador	Clasificación	Valor	Tipo de pago	Imagen
1	28/05/2008 1:22:10	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080527-01221
2	29/05/2008 1:22:35	Fernando Carrillo	Remolque	4	Reclasificado	C:\imagenes\foto20080527-01223
3	29/05/2008 2:24:48	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080527-27483
4	29/05/2008 2:24:57	Fernando Carrillo	Remolque	4	Reclasificado	C:\imagenes\foto20080527-27484
5	29/05/2008 2:24:04	Fernando Carrillo	Liviano	1	Reclasificado	C:\imagenes\foto20080527-27480
6	29/05/2008 2:25:17	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080527-27521
7	29/05/2008 2:25:24	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080527-27522
8	29/05/2008 2:25:40	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080527-27523
9	29/05/2008 2:25:30	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080527-27524
10	29/05/2008 2:25:36	Fernando Carrillo	Liviano	1	Reclasificado	C:\imagenes\foto20080527-27525
11	29/05/2008 2:25:40	Fernando Carrillo	Remolque	4	Reclasificado	C:\imagenes\foto20080527-27526
12	01/06/2008 11:38:28	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080601-11382
13	01/06/2008 11:38:39	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080601-11383
14	01/06/2008 11:38:46	Fernando Carrillo	Liviano	1	Reclasificado	C:\imagenes\foto20080601-11384
15	01/06/2008 11:39:02	Fernando Carrillo	Remolque	4	Reclasificado	C:\imagenes\foto20080601-11385
16	01/06/2008 11:43:00	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080601-11430
17	01/06/2008 11:43:00	Fernando Carrillo	Remolque	4	Reclasificado	C:\imagenes\foto20080601-11431
18	01/06/2008 11:43:09	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080601-11432
19	01/06/2008 11:43:35	Fernando Carrillo	Liviano	1	Reclasificado	C:\imagenes\foto20080601-11433
20	01/06/2008 11:43:43	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080601-11434
21	01/06/2008 11:43:50	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080601-11435
22	01/06/2008 11:43:55	Fernando Carrillo	Remolque	4	Reclasificado	C:\imagenes\foto20080601-11436
23	01/06/2008 11:43:59	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080601-11437
24	01/06/2008 11:53:08	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080601-11538
25	01/06/2008 11:53:15	Fernando Carrillo	Liviano	1	Reclasificado	C:\imagenes\foto20080601-11539
26	01/06/2008 11:53:20	Fernando Carrillo	Remolque	4	Reclasificado	C:\imagenes\foto20080601-11530
27	01/06/2008 11:53:25	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080601-11531
28	01/06/2008 11:53:34	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080601-11532
29	01/06/2008 11:53:41	Fernando Carrillo	Liviano	1	Reclasificado	C:\imagenes\foto20080601-11533
30	01/06/2008 12:12:53	Fernando Carrillo	Extra Pesado	3	Reclasificado	C:\imagenes\foto20080601-12124
31	01/06/2008 12:12:50	Fernando Carrillo	Pesado	2	Reclasificado	C:\imagenes\foto20080601-12125
32	01/06/2008 12:13:05	Fernando Carrillo	Liviano	1	Reclasificado	C:\imagenes\foto20080601-12126

Nº Tickets: 43
Total Efectivo: 104

Figura 3.26 Pantalla de Reportes Generados por Fecha

d) PERSONAL

Este sub menú permite al administrador de la estación de peaje llevar un control minucioso del personal que opera en el sistema con todos los datos generales del mismo, permite realizar modificaciones y crear nuevos registros del personal del peaje.

Peajes - [Personal del Peaje]

Usuarios

Cedula	Nombre	Apellido	Direccion	Telefono	Celular	Usuario	Password
1715597979	Fernando	Carrillo	Av Del Maestro	2398354	099108176	Administrador	luis
1803598418	Fernando	Chilaza	Teodoro Wallin 1568	022806542	096387752	Administrador	wlche
1803598841	Diego	Chilaza	Guillermo Adolfo Becker	032842268	092847857	Operador	diegoch

Actualizar Borrar Usuario Modificar Crear Cancelar

Figura 3.27 Pantalla Para Almacenamiento de Usuarios

3.7.2 DESARROLLO DEL SOFTWARE

La HMI diseñada se fundamenta en el intercambio de datos con el controlador ubicado en el carril, el cuál lee las entradas de los dispositivos sensores y de acuerdo a eso envía la clasificación de cada vehículo, además posee una base de datos en donde registra todos los eventos registrados por el equipo de carril y en los que el operador clasifica manualmente, conjuntamente la base de datos almacena todas las configuraciones administrativas de la empresa administradora de la estación de peaje.

Al iniciar el interfaz se realiza el ingreso de Usuario en un proceso de comparación con la base de Datos, y a la vez de discrimina las clases de usuarios, de administración y de operación.

Una vez que el operador ingresa al sistema se inicia la conexión con el controlador ubicado en el carril con sus respectivos dispositivos sensores realizando el envío y recepción de datos, con esto se procede a la operación normal del carril hasta finalizar el turno correspondiente.

Para fines de control y auditoría el administrador del sistema de carril tiene acceso al resto de funciones del software los que incluye configuración de parámetros, tarifas y la impresión de reportes por turno o por fecha, además de llevar un registro detallado del personal.

Cabe mencionar que dentro de la programación está implícito el dimensionamiento y estética de la presentación en general.

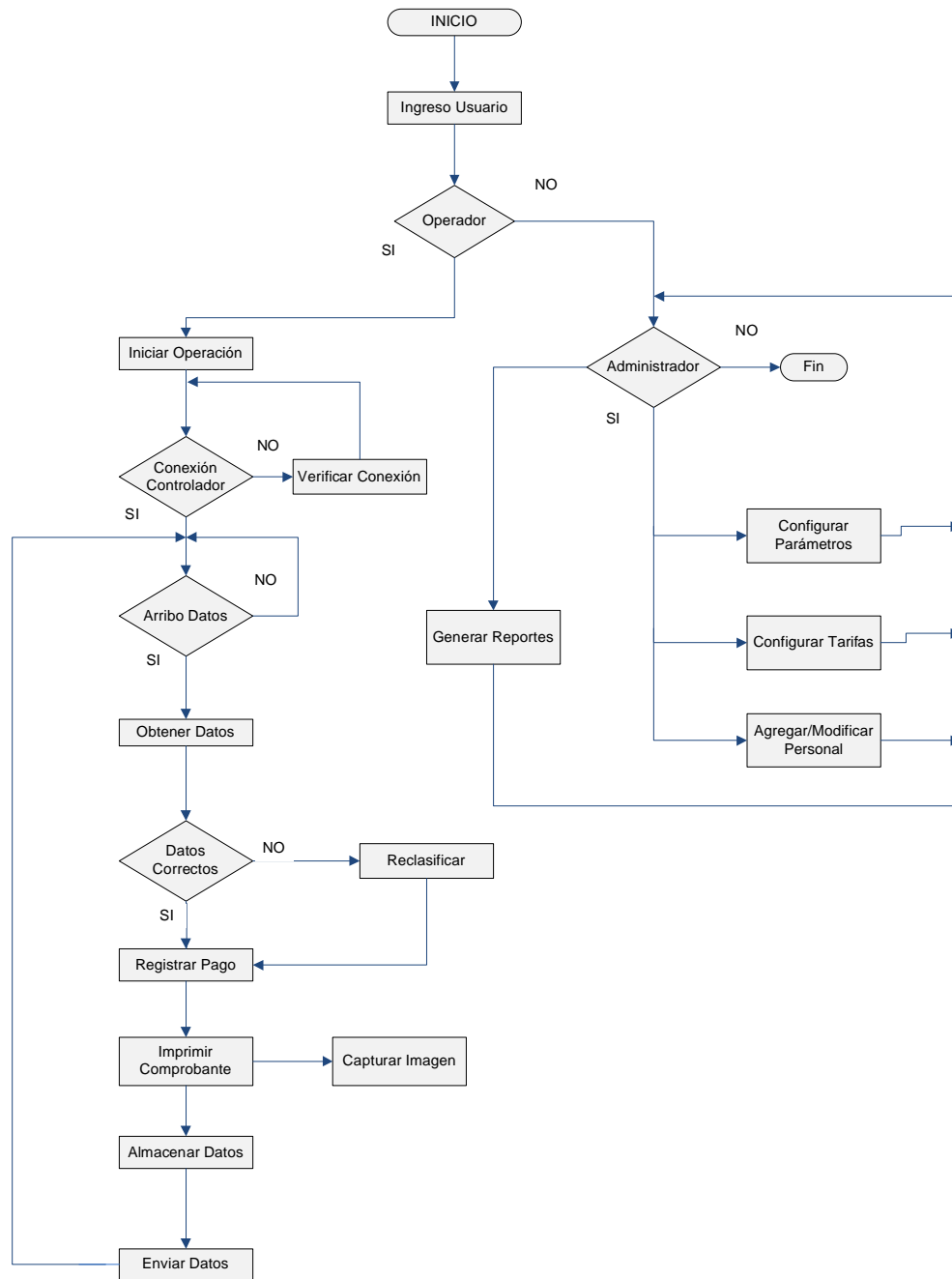


Figura 3.28 Diagrama de Flujo General del Funcionamiento de la Interfaz Gráfica

a) INGRESO DE USUARIOS

El proceso de ingreso de usuarios al sistema es fundamental para el funcionamiento del sistema, ya que el software está desarrollado para abrir las aplicaciones de manera discriminada si el usuario es operador o administrador del sistema, de este modo se procede a partir del ingreso de un nombre de usuario y una clave de acceso con la respectiva comparación con la base de datos.

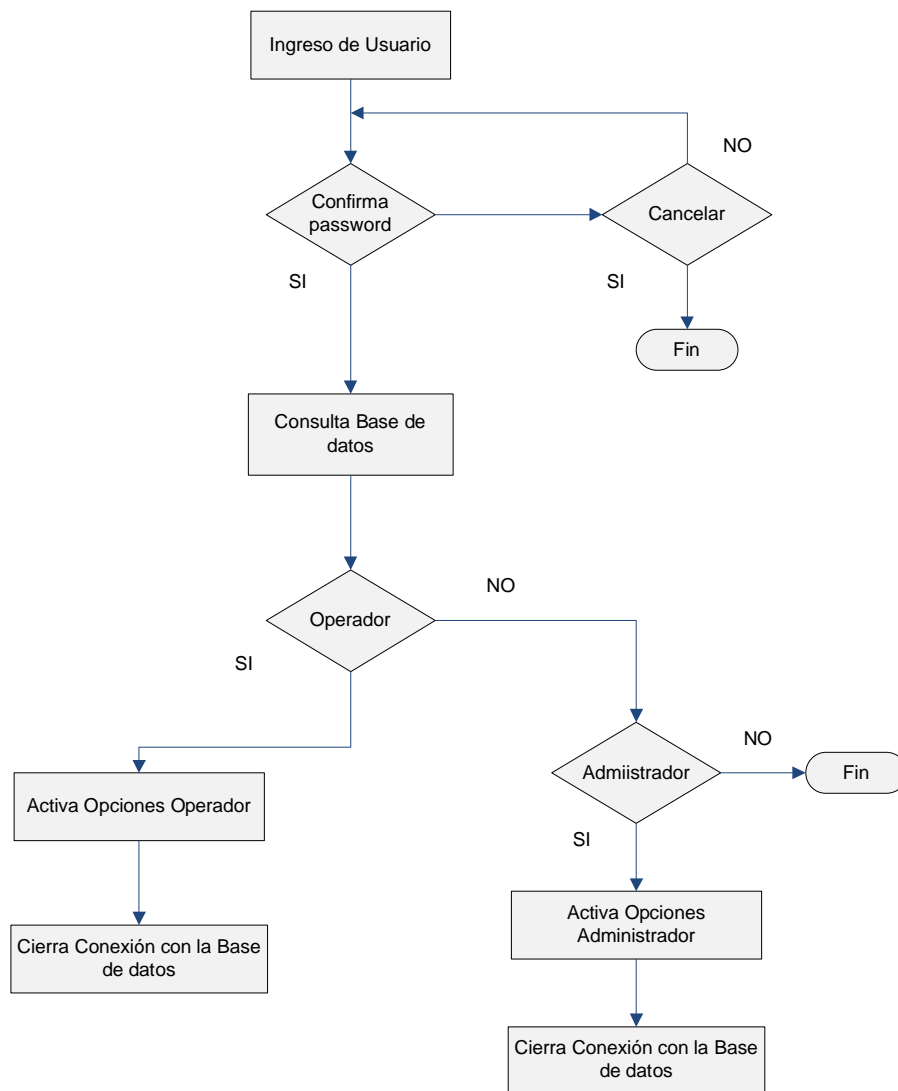


Figura 3.29 Diagrama de Flujo para Ingreso de Usuarios

En el caso de no existir coincidencia con los datos ingresados se despliega un mensaje de error y se retorna a la condición inicial.

b) OPERACIÓN

El proceso de operación de la estación de peaje permite poner en marcha el sistema completo para la clasificación automática de los vehículos usuarios y la correspondiente recaudación.

El proceso realizado es establecer la comunicación con el controlador, ya que el mismo está intentando conectarse periódicamente y a la vez se hace una actualización automática del registro de datos puesto que el controlador almacena en la pila los vehículos que se encuentran esperando para realizar el pago correspondiente.

La conexión con Visual Basic se realiza mediante una herramienta que permite hacer uso de funciones destinadas a comunicación y arribo de datos llamada Microsoft Winsock, esta herramienta es un manejador de red, que es necesario configurar en sus propiedades los campos como dirección IP local, dirección IP remota, numero de puerto local, numero de puerto remoto y el protocolo de comunicación a utilizar, ya sea UTP o TCP/IP. Para el desarrollo de esta aplicación se uso comunicación TCP/IP. . Una vez configurado se puede hacer uso tanto de funciones como comandos destinados a comunicación y arribo de datos por ejemplo:

```
.....  
.....  
WSockClient.RemoteHost = "192.168.100.100" ' Dirección IP del dispositivo  
(Rabbit)  
WSockClient.RemotePort = 13000 ' Puerto  
EstadoWinsock  
If WSockClient.State <> sckOpen Then  
    WSockClient.Close  
    WSockClient.Connect  
End If  
    EstadoWinsock  
.....  
.....
```

Programación de Configuración Winsock

Una vez configurados los parámetros de Winsock se procedió a elaborar funciones para la que permiten establecer las funciones de envío y arribo de datos, además de informar el estado de conexión con el controlador Rabbit.

```
Private Sub WSocketClient_DataArrival(ByVal bytesTotal As Long)
    Dim vtRecibido
    Dim DataRx() As Byte
    WSocketClient.GetData vtRecibido, vbArray + vbByte, bytesTotal 'Recibir Datos
    TCP

    DataRx = vtRecibido
    Select Case DataRx(0) ' Numero de Funcion
        ' Func 0: Tipo de Carro
        Case 0: ProcesarCarro (DataRx(1))
        ' Func 1: Tarifa
        'Case 1:
        ' Func 2: Abrir Valla
        'Case 2:

    End Select

End Sub

Public Function EstadoWinsock()

    Select Case WSocketClient.State

        Case 0: lblEstado.Caption = "Winsock: Cerrado"
        Case 1: lblEstado.Caption = "Winsock: Abierto"
        Case 2: lblEstado.Caption = "Winsock: Escuchando"
        Case 3: lblEstado.Caption = "Winsock: Conexión Pendiente"
        Case 4: lblEstado.Caption = "Winsock: Resolviendo Host"
        Case 5: lblEstado.Caption = "Winsock: Host Resuelto"
        Case 6: lblEstado.Caption = "Winsock: Conectando"
        Case 7: lblEstado.Caption = "Winsock: Conectado"
        Case 8: lblEstado.Caption = "Winsock: Equipo Cerrando Conexión"
        Case 9: lblEstado.Caption = "Winsock: Error"

    End Select

End Function
```

Para la captación automática de imágenes de los vehículos que ingresan a la cabina de cobro se utiliza una tarjeta de video llamada WinTV la cual permite la conexión con Visual Basic a través de un OCX desarrollado para este tipo de aplicaciones, a partir de esto el vídeo se mostrará automáticamente en directo, con esto se realiza la captación automática de imágenes.

El OCX puede funcionar independientemente, provee un control detallado sobre el vídeo y audio, además brinda acceso rápido a parámetros como la Fuente de vídeo, canal, volumen, contraste, color, etc.

Cabe mencionar que para la captación automática de imágenes es necesario tomar la hora y fecha actual para guardar la foto correspondiente, además de esto se debe direccionar una carpeta y configurar el formato y la resolución en la que se desea capturar la imagen. A continuación se muestra el código para la captación de imágenes.

```
.....  
.....  
fecha = Format(Date, "yyyy/mm/dd") & " " & Format(Time, "hh:mm:ss")  
nombre = "foto" & Format(Date, "yyyymmdd") & "-" & Format(Time,  
"hhmmss")  
    dirpic = "C:/imagenes/" & nombre & ".jpg"  
    video.SaveToDiskEx dirpic, hcwFF_JPEG_jpg, hcwFIF_16BPP,  
hcwCMP_JPEG, 60  
.....  
.....
```

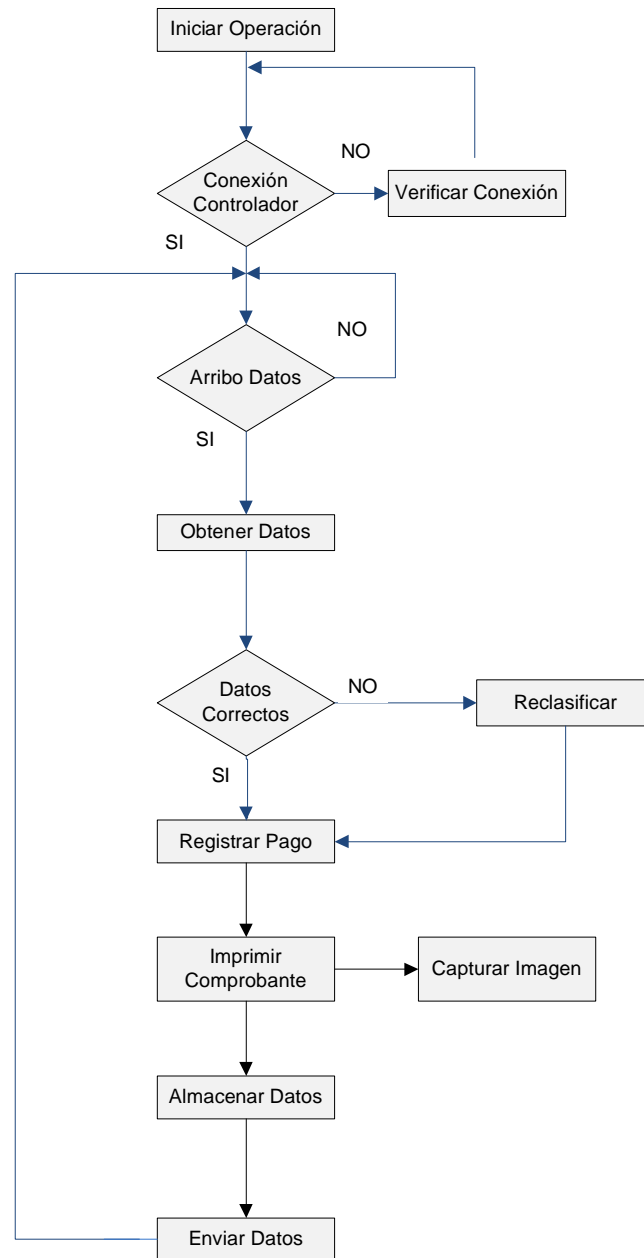


Figura 3.30 Diagrama de Flujo de Operación del Sistema

c) ENVÍO DE DATOS AL CONTROLADOR

Una vez confirmado el tipo de vehículo que transita por la cabina de cobro se procede al envío de datos al controlador para activar los pulsos de la barrera de acceso y el semáforo de carril, para esto es necesario el llamado de un comando que apunta al

winsock, WSocketClient.SendData vtEnviar, como se describió en las funciones anteriores.

Las tramas comunes que se transmiten en la red tienen la siguiente forma general:

FUNCIÓN	DATOS
1 Byte	n Bytes

Tabla 3.4 Trama General de Datos enviados al Controlador

Cabe mencionar que para el envío se utiliza transmisión de datos vía protocolo TCP/IP, bajo las características 100 Base T.

Las funciones y parámetros de las peticiones y respectivas respuestas que recibe el sistema se describen a continuación:

Tipo de Dato	Función		Parámetros	
	Valor	Descripción	Valor	Descripción
Pet.	0	Tipo de Carro	---	No tiene parámetros.
Res.			R1: 0-255	0: No hay carro 1: Livianos 2: Pesados 3: Extra Pesados
Pet.	1	Tarifa	P1: 0-255	Dólares
Res.			P2: 0-255	Centavos
Pet.	2	Abrir Barrera	---	No tiene parámetros.
Res.			---	No tiene parámetros.

Tabla 3.5 Funciones y Parámetros que Recibe el Sistema

d) ALMACENAMIENTO DE DATOS

El almacenamiento de datos se realiza a partir de las necesidades de almacenamiento del programa, dependiendo si el usuario es administrador u operador, debido a que la base de datos del programa está diseñada para guardar datos de configuración, tarifas o parámetros en caso de ser administrador o en cada proceso de recaudación una vez que se confirma el pago de cada vehículo y se captura la imagen respectiva.

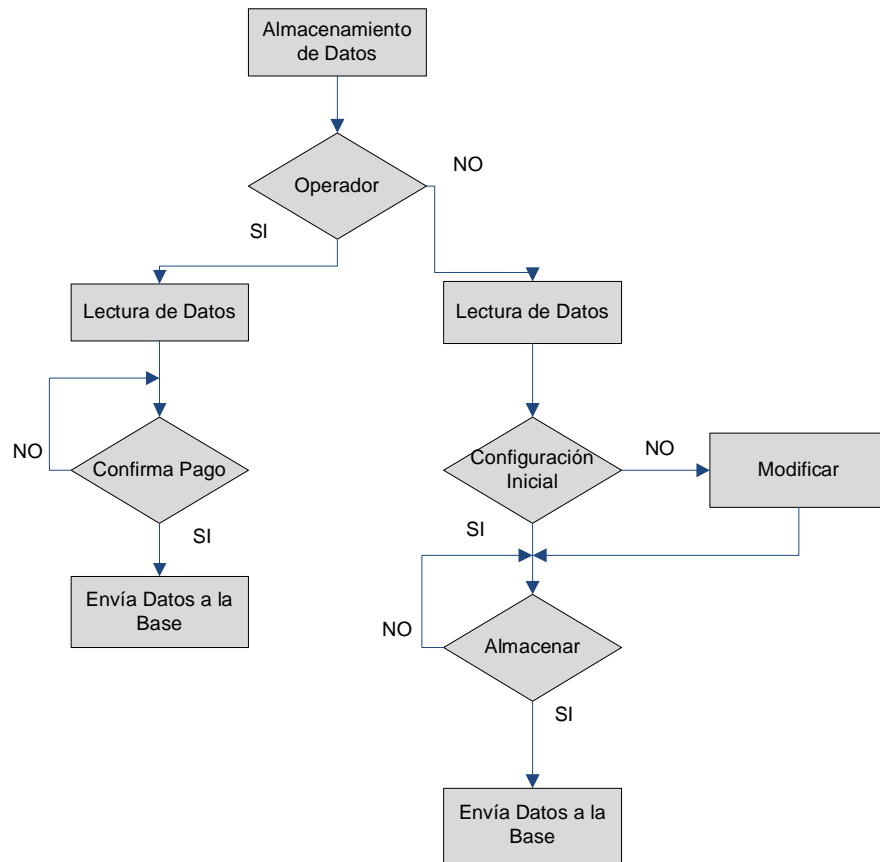


Figura 3.31 Diagrama de Flujo de Almacenamiento de Datos

CAPITULO IV

INSTALACIÓN DEL SISTEMA

4.1 DESCRIPCIÓN

Para la instalación y puesta en marcha del sistema de clasificación automática es preciso tomar en cuenta parámetros necesarios como: suministro de alimentación eléctrica, instalación de protecciones a tierra, para el correcto funcionamiento del mismo, realizar pruebas para comprobar la fiabilidad, conectividad y funcionamiento en general del mismo.

4.2 SISTEMA ELECTRICO

4.2.1 RED ELÉCTRICA Y FUENTE DE ALIMENTACIÓN

El funcionamiento del sistema requiere suministro de alimentación eléctrica de la de red a un nivel de tensión de 120 VAC y 60 Hz, además de esto se propone la conexión a una planta de emergencia, puesto que el sistema debe funcionar a tiempo completo y sin interrupciones. Tomando en cuenta este tipo de inconvenientes se ha previsto un sistema de respaldo de energía, para mantener en condiciones de funcionamiento el sistema de Administración y operación en modo manual por lo menos 15 min, después de un corte de suministro eléctrico.

Cabe mencionar que el controlador y elementos sensores requieren una alimentación diferente a la de la red pública, por lo que se ejecuta el acondicionamiento de esta señal, esto se efectúa mediante la rectificación y regulación a 5V y 12V dependiendo de los requerimientos.

4.2.2 SISTEMA DE PROTECCIÓN

DISPOSITIVOS DE PROTECCIÓN

El tablero de control donde se encuentra el módulo PowerCoreFlex y la tarjeta electrónica posee un interruptor termomagnético unipolar de 15A dispositivo que ayuda a la unidad principal a mantenerse a salvo de cortocircuitos y caídas de voltaje que se efectúen en el suministro eléctrico. De igual manera la fuente de voltaje que alimenta el controlador tiene un dispositivo de protección en función de la corriente nominal de carga, a partir de esto se dimensiona considerando un 25% de valor adicional, debido a que se pueden generar transitorios que afecten el funcionamiento de los componentes que conforman el sistema.

El valor de corriente máximo de carga es de 1 [A] por lo que el valor del fusible de protección es de 1.25 [A]. Sin embargo es necesario ajustar a un valor que se encuentre en el mercado, por lo que se consideró uno de 2 [A].

PUESTA A TIERRA

Un sistema de puesta a tierra consiste en la conexión de equipos eléctricos y electrónicos a tierra, para evitar que se produzcan daños en caso de una corriente transitoria peligrosa.

El objetivo de un sistema de puesta a tierra es:

- El de brindar seguridad a las personas.
- Proteger las instalaciones, equipos y bienes en general, al facilitar y garantizar la correcta operación de los dispositivos de protección.
- Establecer la permanencia, de un potencial de referencia, al estabilizar la tensión eléctrica a tierra, bajo condiciones normales de operación.

La importancia de realizar una conexión a tierra es trascendental dentro de toda instalación eléctrica, debido a que a través de esta es posible mantener altos niveles de seguridad para garantizar el óptimo desarrollo y comportamiento de los sistemas [1].

Se debe tomar en cuenta la forma en que el sistema es conectado a tierra, por el efecto directo en la magnitud de los voltajes que deben ser mantenidos en condiciones normales y bajo condiciones transitorias.

Es necesario considerar un parámetro muy relevante dentro del análisis y diseño de puestas a tierra, la resistividad del electrodo, pueden ser artificiales o naturales. La misma que a su vez depende de factores tales como la resistividad del contacto del electrodo con la tierra y la resistividad del suelo [1].

Se entiende por electrodos artificiales los establecidos con el exclusivo objeto de obtener la puesta a tierra, y por electrodos naturales las masas metálicas que puedan existir enterradas.

La resistividad del terreno se define como la resistencia que presenta 1 m³ de tierra, en esta influyen varios factores que pueden variarla, entre los más importantes se encuentran: Naturaleza del Terreno, Humedad, Temperatura, Salinidad, Estratigrafía, Compactación y las Variaciones estacionales [2]

De este modo, en la implementación realizada el potencial propio del circuito se referencia a tierra mediante la utilización de una barra de Cobre, el cual es altamente conductivo y ofrece baja resistencia óhmica, permitiendo la descarga de cualquier señal externa, ajena a las consideradas dentro del diseño y que pudieran afectar su correcto desempeño.

4.3 INTEGRACIÓN CONTROLADOR – ELEMENTOS DE DETECCIÓN

El funcionamiento integral del sistema de clasificación vehicular se rige al dispositivo principal que constituye el controlador Rabbit, el mismo que recibe las señales de entrada de los dispositivos sensores de piso y envía señales eléctricas de salida a través de la adaptación de una tarjeta electrónica a circuitos de disparo

conformados por relés hacia los distintos elementos que constituyen el equipo de preclasificación automática.

El modulo PowerCoreFlex se adapta en la tarjeta electrónica, la misma que está diseñada para adaptar las señales de entrada y salida enviadas al controlador.

Esta adaptación se la lleva a cabo a través del uso de transistores los cuales actúan como un switch para los relés de salida, con esto se evita el reingreso de corrientes parasitas o transitorias al controlador.

Para la adaptación de las señales de entrada se utiliza una compuerta negadora la cual cambia de estado al activarse cualquier sensor, con esto se asegura que la entrada al módulo sea un “1” lógico y de esta manera se protege al controlador.

Además de realizar la función de adaptar las señales la tarjeta electrónica cumple con la función de proteger el controlador, de manera que no puedan afectar el funcionamiento del mismo en caso de existir corrientes parásitas o mal funcionamiento de los sensores. Con esto se contribuye en la vida útil del controlador y el funcionamiento integral del equipo de clasificación automática.

4.4 PRUEBAS Y RESULTADOS

Para garantizar el desempeño óptimo del sistema de clasificación vehicular es estrictamente necesario realizar pruebas para comprobar la robustez, confiabilidad, conectividad y funcionamiento del sistema en general.

Con este enfoque se han ido desarrollando pruebas y ajustes a lo largo del desarrollo y avance de este proyecto con un cronograma establecido. A continuación se detalla las pruebas realizadas al sistema en la siguiente tabla.

Pruebas	Duración	Elementos	Resultado
Circuito de adaptación de señales	5 meses	Compuertas Negadoras, Fuentes de Alimentación, Transistores, Dispositivos Sensores	Ü
Lógica de Programación del Controlador	2 meses	Tarjeta Rabbit, Circuitos de Adaptación, Fuente de Alimentación, CPU de control.	Ü
Conexión de la Tarjeta Rabbit con Display	1 mes	Tarjeta Rabbit, Display de Visualización, Integrado Max 485	Ü
Conexión De la Tarjeta al Interfaz Gráfica	2 meses	Tarjeta Rabbit, CPU control, Cable Cruzado, Sensores.	Ü
Captación Automática de Imágenes	2 semanas	CPU de Control, Cable de Video, Fuente de Alimentación, Video Cámara.	Ü
Almacenamiento de Datos a la Base	1 mes	Tarjeta Rabbit, CPU Control, Cable Cruzado.	Ü
Integración del Controlador con Elementos de Detección	1 mes	Tarjeta Rabbit, CPU Control, Cable Cruzado, Fuente de Alimentación, Sensores de Piso, Cámara de Video, Semáforo, Display, Barrera de Acceso.	Ü

Tabla 4.1. Detalle de pruebas por períodos

4.5 PROCEDIMIENTO

4.5.1 CIRCUITO DE ADAPTACIÓN DE SEÑALES

Para el desarrollo del proyecto fue de vital importancia comprobar el funcionamiento del circuito de adaptación de las señales de entrada y salida de los sensores, ya que estas van a estar en funcionamiento todo el tiempo. Por esta razón se mantuvo en funcionamiento 24 horas por un periodo de 20 semanas.

4.5.2 LÓGICA DE PROGRAMACIÓN DEL CONTROLADOR

Desarrolladas las pruebas de adaptación de señales, se procedió a realizar su interconexión con el controlador, para estos se consideró la lógica de programación para la clasificación automática, los tiempos de retardo para rebotes y para activación de señales y el funcionamiento en conjunto de los diferentes sensores.

Además se configuró la conectividad en red. Esta verificación general del sistema se realizó durante ocho semanas.

4.5.3 CONEXIÓN DE LA TARJETA RABBITO CON EL DISPLAY

A través de la utilización del puerto serial del módulo PowerCoreFlex, se procedió a realizar la comprobación del envío y recepción de caracteres mediante la transmisión serial para la visualización de los distintos tipos de tarifa en el display. Estas pruebas se realizaron durante cuatro semanas.

4.5.4 CONEXIÓN DE LA TARJETA A LA INTERFAZ GRÁFICA

Antes de realizar la conexión con la interfaz gráfica fue necesaria la verificación del funcionamiento de la tarjeta en la red, para esto se configuró previamente su respectiva dirección IP y puerto de acceso, de este modo al hacer un ping de comprobación, se comprobó que la conexión se completó sin errores.

Una vez realizada esta comprobación se procedió a establecer la conexión del controlador con el interfaz gráfica se realizó la comunicación bidireccional de datos, mediante uso de herramientas que el software sobre el cual se programó, para lo cual se simulo el paso de un vehículo manipulando los sensores, de esta manera se comprobó el arribo de datos del tipo de vehículo y la activación de señales de salida una vez confirmado el pago. Su verificación se realizó durante ocho semanas.

4.5.5 CAPTACIÓN AUTOMÁTICA DE IMÁGENES

Comprobada la interconexión de la interfaz gráfica con el controlador se procedió con la captura de imágenes de los vehículos que ingresan a la cabina de cobro, para esto se utilizó una herramienta de software (OCX) sobre la cual se programó. Su verificación se realizó durante dos semanas.

4.5.6 ALMACENAMIENTO DE DATOS A LA BASE

Al establecer la conexión del controlador con la interfaz gráfica se realizó la comunicación bidireccional de datos. Una vez realizado este proceso era necesario guardar esos datos en una base para auditoría. Para esto se utilizó el software mysql el cual permite el enlace con Visual Basic para el almacenamiento de datos, además es de fácil manejo. Su verificación se realizó durante cuatro semanas.

4.5.7 INTEGRACIÓN DEL CONTROLADOR CON ELEMENTOS DE DETECCIÓN

Una vez verificados cada uno de los subsistemas, se procedió a realizar la interconexión y verificación integral, considerando la lógica de programación y el funcionamiento en conjunto tanto de hardware como de software.

Se hicieron pruebas de conectividad en red, discriminación del acceso de administración y operación, además se verificó el funcionamiento manual y automático del sistema unificado la activación de las señales de entrada y salida. Esta verificación general del sistema se realizó durante cuatro semanas.

CAPITULO V

ANÁLISIS ECONÓMICO

5.1 INTRODUCCIÓN

El diseño del Sistema de Preclasificado y Auditoría Para el Control Vehicular de una Estación de Peaje se realizo para incrementar la seguridad en la operación al no ser el peajista, quien determina la clase de vehículos que habrá de cobrar, eliminando con ello la mayor parte de inconvenientes que presentan los equipos de posclasificación.

Con el diseño de este sistema se elimina los inconvenientes del sistema posclasificador, ahora solo con la opción de que el peajista corrija en forma manual algunos errores de la clasificación automática y opere simplemente como un recaudador de efectivo, de tal manera que la evasión o falta de control en el recaudo es prácticamente inexistente o reducida solo a defectos de los procedimientos y controles posteriores. Gracias a esto se obtiene una herramienta de trabajo altamente eficaz, que permitirá hacer mucho más eficiente su operación, y el control aforo – ingreso.

Cabe mencionar que el diseño no solo se lo realizó en base al aspecto técnico, sino que, se llevó a cabo considerando la optimización de recursos económicos; por lo que fue necesario desarrollar un sistema que además de ser de bajo presupuesto garantice su control, funcionalidad y un bajo costo de mantenimiento, ofreciendo eficiente operación y control en la recaudación.

El análisis económico del sistema de clasificación vehicular se enfoca a los costos de Inversión Inicial debido a que es un modelo prototipo destinado a la comercialización.

5.2 INVERSIÓN INICIAL

Los dispositivos electrónicos, así como todos los elementos considerados dentro del diseño son diversos, cada uno con un objetivo en común, garantizar mediante su correcta utilización el funcionamiento general del sistema.

Las siguientes tablas muestran la inversión realizada, esta información es detallada en función de cada uno de los subsistemas que forman parte del proyecto en general.

<u>ELEMENTOS DE CARRIL</u>				
Ítem	Descripción	Cantidad	Precio Unitario	Precio Total
1	Barrera Óptica	1	74,07	74,07
2	Peanas	2	392,74	785,48
3	Display de Usuario	1	90	90
4	Barrera de Salida	1	1000	1000
5	Lazo Magnético	1	300	300
6	Semáforo	1	250	250
TOTAL:				US \$ 2499,55

Tabla 5.1. Inversión Económica de Elementos de Carril

<u>ELEMENTOS DE CABINA</u>				
Ítem	Descripción	Cantidad	Precio Unitario	Precio Total
1	PC con monitor y Teclado	1	700	700
2	Impresora de Reportes	1	80	80
3	Impresora para Tickets	1	100	100
4	UPS	1	60	60
TOTAL:				US \$940

Tabla 5.2. Inversión Económica de Elementos de Cabina

<u>TABLERO DE DISTRIBUCIÓN</u>				
Ítem	Descripción	Cantidad	Precio Unitario	Precio Total
1	Tarjeta Rabbit	1	269	269
2	Interruptor Termomagnético Unipolar 15 ^a	1	17	17
3	Regleta	1	7	7
4	Ventilador	1	5	5
5	Max 485	1	4,5	4,5
6	Resistencias	20	0,03	0,6
7	Transistores 2N3904	4	0,4	1,6
8	Diodos 1N4007	5	0,15	0,75
9	Relé de 110VAC de activación	3	3,6	10,8
10	Bornera de dos entradas	7	0,4	2,8
11	Bornera de tres entradas	5	0,6	3
12	Capacitor electrolítico 10uF, 50V	8	0,16	1,28
13	Capacitor electrolítico 1uF, 50V	6	0,16	0,96
14	Sujetadores	10	0,25	2,5
15	Riel Din	1	2,8	2,8
16	Puente de diodos KBL404	1	1,1	1,1
17	Transformador de 24V con tab central	1	6,5	6,5
18	Regulador LM 7805	2	0,7	1,4
19	Regulador LM 7812	1	0,7	0,7
20	Placa de Tarjeta Electrónica	172	0,2	34,4
21	Placa de Fuente 5 y 12 VDC	145	0,2	29
22	Borneras de alimentación AC	4	1	4
23	Gabinete de Control Metálico	1	70	70
24	Switch ON/OFF	1	0,54	0,54
25	Amarras Plástica	20	0,1	2
26	Header hebra	2	1,5	3
TOTAL:				US \$482,23

Tabla 5.3. Inversión Económica de Tablero de Distribución

<u>ELEMENTOS DE VIDEO</u>				
Ítem	Descripción	Cantidad	Precio Unitario	Precio Total
1	Cámara Autoiris con Visión Nocturna	1	690	690
2	Conectores de Video	2	1,5	3
3	Tarjeta de Video WinTV	1	80	80
TOTAL:				US \$773

Tabla 5.4. Inversión Económica de Elementos de Video

<u>CABLES</u>				
Ítem	Descripción	Cantidad	Precio Unitario	Precio Total
1	Cable UTP- CAT 5	20	0,6	12
2	Cable 2x14 AWG	80	0,7	56
3	Cable 2x12 AWG	15	0,75	11,25
4	Cable 2x22 AWG	20	0,4	8
5	Cable Coaxial	20	0,9	18
TOTAL:				US \$105,25

Tabla 5.5. Inversión Económica de Cables

<u>GASTOS ADICIONALES</u>				
Ítem	Descripción	Cantidad	Precio Unitario	Precio Total
1	Mano de Obra	1	400	400
2	Transporte	1	100	100
3	Dirección Técnica	1	600	600
TOTAL:				US \$1100

Tabla 5.6. Inversión Económica de Gastos Adicionales

<u>SISTEMA DE PRECLASIFICACIÓN AUTOMÁTICA</u>				
Ítem	Descripción	Cantidad	Precio Unitario	Precio Total
1	Elementos de Carril	1	2499,55	2499,55
2	Elementos de Cabina	1	940	940
3	Tablero de distribución	1	482,23	482,23
4	Elementos de Video	1	773	773
5	Cables	1	105,25	105,25
6	Gastos de Honorarios y Adicionales	1	1100	1100
TOTAL:				US \$5900,03

Tabla 5.7. Inversión Total del Proyecto

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 INTRODUCCIÓN

Actualmente sistemas de peaje tradicionales utilizan sistemas de posclasificación, estos métodos presentan muchos inconvenientes, porque existe evasión y falta de control sobre la recaudación, además está expuesto a fallas de tipo humano.

Es así, que se diseñó e implementó un sistema con el cual se ofrece una solución general, operativa y funcional ante el sistema posclasificador.

Además de tener un bajo costo inicial, tiene una garantía de funcionamiento a largo plazo; esto fue posible conseguir mediante la selección adecuada de dispositivos electrónicos que permitieron cumplir con las metas planteadas al inicio del proyecto.

6.2 CONCLUSIONES

El sistema permite diferenciar del número de ejes de los automotores de rodada sencilla o rodada doble, a través de sensores de contacto, ópticos e inductivos acoplados a un controlador.

En el caso del controlador se desarrollo la programación del módulo PowerCoreFlex a fin de que las señales sean captadas por la tarjeta para la clasificación de vehículos.

En el caso de que el controlador pierda conexión con el servidor, la lógica de la tarjeta está diseñada para que se vuelva a levantar la conexión con el usuario de la interfaz gráfica.

El controlador está en capacidad de almacenar un número determinado de vehículos dependiendo de la necesidad hasta restablecer la conexión con el servidor, o en casos de que exista tráfico en el carril.

Cuando se usa comunicación TCP/IP el módulo PowerCoreFlex, escucha la llegada de una conexión, en el puerto verifica si hay datos nuevos, en caso de haberlos los obtiene. A partir de esto suelta el control para que otras tareas se ejecuten. Es decir el controlador se mantiene verificando y actualizando constantemente los datos de llegada.

La tarjeta Rabbit puede ser programada únicamente mediante Dinamic C, ya que este posee las librerías necesarias para manejo de todas las características y recursos que poseen estas tarjetas.

El controlador Rabbit ofrece múltiples recursos para la transmisión de datos, es así que se utilizo dos tipos de configuración: serial para la transmisión de la tarifa al usuario y TCP/IP para la clasificación automática, manteniendo una alta eficiencia en la comunicación.

Se desarrollo un software aplicativo creando una herramienta de trabajo eficaz que permite entre otras posibilidades, la generación de reportes, la validación de usuarios y la interconexión con el sistema de grabación de video para la captación automática de imágenes de los vehículos en la estación de cobro. Además posee un módulo de administración para la configuración inicial del sistema y control del personal que opera en la estación.

A través del sistema auditor es posible aclarar todos los eventos que ocurren en el carril, en especial en los que el peajista corrige en forma manual algunos errores de la clasificación automática.

El sistema está respaldado con una base de datos que permite realizar consultas de reportes y establecer responsables del control aforo – ingreso.

6.3 RECOMENDACIONES

Dentro de la planeación del proyecto es imprescindible realizar la selección del controlador en función de la aplicación a desarrollar.

Para el desarrollo de la programación del controlador se recomienda realizar pruebas por módulos o subsistemas, y luego realizar la integración en un solo código y realizar una prueba total.

Antes de desarrollar la el software de aplicación se recomienda diseñar una base de datos para evitar inconvenientes en la programación.

Es estrictamente necesario considerar sistemas de protección eléctrica, para garantizar el buen desempeño de los elementos que constituyen el sistema y la vida útil de los mismos.

Además es ineludible considerar el desarrollo de sistemas de respaldo de energía eléctrica que permita su permanente funcionamiento, constituyendo una alternativa la utilización de paneles solares, los cuales requieren de una mínima inversión reduciendo el impacto en el medio ambiente.

Se recomienda dar capacitación al personal encargado del manejo del sistema, ya que conlleva una gran responsabilidad la administración vial

La instalación del presente proyecto requiere las obras civiles necesarias considerando las características viales y de crecimiento vehicular para la instalación de los elementos sensores, de manera que no puedan afectar o incidir en el sistema de clasificación planteado.

REFERENCIAS BIBLIOGRÁFICAS

CAPITULO I

- [1] CONTROLES ELECTROMECÁNICOS S.A. Sistemas Mexicanos de alta tecnología para detección, clasificación y auditoría en peajes, Junio de 1998.
- [2] DYETRON MANUAL DE SOLUCIONES PARA EL MANEJO INTEGRAL DE CONESIONES VIALES.
- [3] ENGINEERED PARKING SYSTEMS, INC. parking gate model P301, Operation & Instalación manual.
- [4] CAME CANCELLI AUTOMATICI Fococélulas documentación técnica 42 Rev. 5.0 119R42
- [5] CAME CANCELLI AUTOMATICI Controladores para campo magnético documentación técnica T20 Rev. 1.0 119RT20-E
- [6] www.mzsistemas.com
- [7] www.intecar.com.ar/spanish/Indexsp.htm
- [8] www.rabbit.com/products/PowerCore/

CAPITULO II

- [1] PowerCore Flex, Introduction, página 1, user's manual, 2004, www.rabbitsemiconductor.com.
- [2] PowerCore Flex, appendix A, Electrical and Mechanical Characteristics, página 94, user's manual, 2004, www.rabbitsemiconductor.com.
- [3] PowerCore Flex, Hardware Reference, página 27, user's manual, 2004, www.rabbitsemiconductor.com.
- [4] PowerCore Flex, Advantages, página 5, user's manual, 2004, www.rabbitsemiconductor.com.
- [5] Rabbit 3000 Microprocessor, Rabbit 3000 Design Features, página 12, user's manual, 2004, www.rabbitsemiconductor.com.
- [6] PowerCore Flex, Internal and External Buses, página 32, user's manual, 2004, www.rabbitsemiconductor.com.
- [7] PowerCore Flex, use of Rabbit 3000 Ports, página 29, user's manual, 2004, www.rabbitsemiconductor.com.

- [8] PowerCore Flex, Hardware Reference, páginas 34-38, user's manual, 2004, www.rabbitsemiconductor.com.
- [9] Prototyping Board, Appendix B, página 112, user's manual, 2004, www.rabbitsemiconductor.com.
- [10] Prototyping Board, Appendix B, páginas 110-111, user's manual, 2004, www.rabbitsemiconductor.com.
- [11] PowerCore Flex, PowerCore Digital Inputs and Outputs, página 28, user's manual, 2004. www.rabbitsemiconductor.com.
- [12] Prototyping Board, Appendix A, página 94, user's manual, 2004, www.rabbitsemiconductor.com.
- [13] PowerCore Flex, Memory, página 43, user's manual, 2004, www.rabbitsemiconductor.com.
- [14] <http://www.forest.ula.ve/~mana/cursos/redes/protocolos.html>
- [15] http://es.wikipedia.org/wiki/Protocolo_de_red#Introducci.C3.B3n
- [16] http://es.wikipedia.org/wiki/Familia_de_protocolos_de_Internet
- [17] <http://cisco.com/web/learning/netacad/index.html>, CCNA1, Capítulo 9 Ethernet Estándares e implementación
- [18] Microsoft ® Encarta ® 2007. © 1993--2006 Microsoft Corporation.
- [19] <http://es.wikipedia.org/wiki/RJ-45>
- [20] <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1#422>
- [21] <http://www.euskalnet.net/shizuka/rs232.htm>
- [22] http://es.wikipedia.org/wiki/Puerto_serial
- [23] Networking Solutions TCP/IP, Dynamic C TCP/IP User's Manual Vol. 1, TCP Socket Functions, página 34, www.rabbitsemiconductor.com.
- [24] Networking Solutions TCP/IP, An Introduction to TCP/IP, TCP Socket Interface, página 26, www.rabbitsemiconductor.com.
- [25] RabbitCore RCM3000 User's Manual, Software Reference, página 20, www.rabbitsemiconductor.com.

CAPITULO III

- [1] www.intecar.com.ar/spanish/Indexsp.htm

CAPITULO IV

[1]//www.lpi.tel.uva.es/~nacho/docencia/EMC/trabajos_02_03/Proteccion_contra_descargas_atmosfericas/12/12.htm

[2] <http://html.rincondelvago.com/sistemas-de-puesta-a-tierra.html>

ANEXOS

ANEXO A

DIAGRAMA GENERAL DE LA ESTACIÓN DE PEAJE

ANEXO A.1

DIAGRAMA GENERAL DE LA ESTACIÓN DE PEAJE VISTA SUPERIOR

ANEXO B

DIAGRAMA DE CONEXIÓN GENERAL “SISTEMA PEAJE”

ANEXO B.1

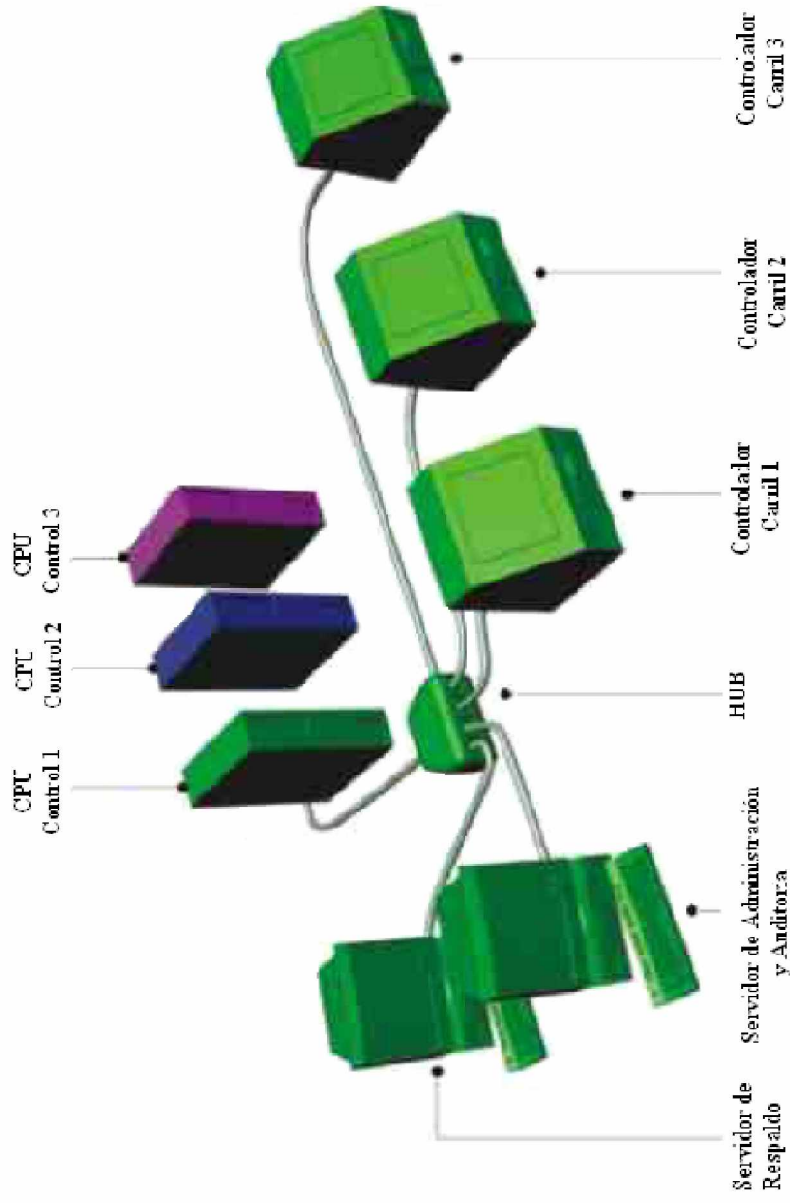
DIAGRAMA ELÉCTRICO UNIFILAR “SISTEMA PEAJE”

ANEXO B.2

CIRCUITO ELECTRÓNICO DE CONTROL “SISTEMA PEAJE”

ANEXO C

ESQUEMA DE PROYECCIÓN A TRES CARRILES



ESCUELA POLITÉCNICA DEL EJÉRCITO	
ESQUEMA DE PROYECCION A TRES CARRILES	
REALIZADO POR:	FECHA: 0CT/2008
<u>Wilson F. Chirio E.</u>	HOJA: 1 DE 1

ANEXO D

CÓDIGO DE PROGRAMA DEL CONTROLADOR TARJETA RABBIT

```

/*****
*           ESCUELA POLITÉCNICA DEL EJÉRCITO           *
*           INGENIERÍA ELECTRÓNICA EN AUTOMATIZACIÓN   *
*           TESIS DE GRADO                             *
*
*           SISTEMA DE PRECLASIFICACIÓN VEHICULAR     *
*
*           WILSON FERNANDO CHILUISA ESPIN           *
*
*****/

```

```
#use "PowerCoreFLEX.lib"
```

```
//__CONSTANTES_____
```

```

#define COM_BAUD 115200 //Velocidad de comunicacion del puerto serial
#define CINBUFSIZE 255 //Es el tamaño maximo de una trama RTU
#define COUTBUFSIZE 255
#define BARRERA_IR 0 //Entrda del puertoA donde se conecta la barrera
#define PEANA1 1 //Entrada del puertoA donde se conecta la peana1
#define PEANA2 2 //Entrada del puertoA donde se conecta la peana2
#define SEMAFORO 6 //Salida del puertoB donde se conecta el semaforo
#define BARRERA 7 //Salida del puertoB donde se conecta la barrera mecanica
#define MAX_PILA 10//nuero maximo de autos que se pueden mantener en memoria.
#define DLYBARRERA 1//delay para el pulso de la barrera
#define DLYSEMAFORO 4//delay para activacion del semaforo
#define DIR_IP "192.168.100.100"//direccion IP del controlador
#define MASCARA "255.255.255.0"//mascara del controlador
#define PUERTO 13000//puerto de comunicacion
#define ACT 1
#define INACT 0

```

```
//__PROTOTIPO DE FUNCIONES_____
```

```

void Ini_ES();//Inicializacion de entradas salidas
int Lec_Ent(int entrada);//Lectura de entradas
int Esc_Sal(int salida, int estado); //Escribe salidas
cofunc void Wr_Com1(char *buf, int tam); //Escribe en el puerto serial
void Ini_Serial();//Inicializa el puerto serial

```

```
//__VARIAIBLES GLOBALES_____
```

```

unsigned char tipoAuto, pila[MAX_PILA], indicePila;//almacena el tipo de auto que paso por el peaje.
unsigned char dolares, centavos;
CoData analisis, IR, ini_disp;//tarea para el analisis del proceso.

```

```
//TCP/IP_____
```

```

#define TCPCONFIG 1//STATICO
#define SOCK_BUF_SIZE 512
#define MAX_SOCKETS 1
#define MAX_BUFSIZE_TCP 256

```

```
#use "dcrtcp.lib"
```

```

//Variables
tcp_Socket Socket;//Socket para comunicacion

//Funciones
//cofunc int Com_TCP(tcp_Socket *s, int puerto, void(*func)());

//__FUNCION PRINCIPAL_____
main()
{
    unsigned char buf[MAX_BUFSIZE_TCP], aux, bufRS[25];
    int length, i;

    Ini_ES();
    Ini_Serial();

    //INICIALIZACION DE TCP

    sock_init();
    sock_mode(&Socket, TCP_MODE_ASCII);

    //Encera Salidas

    Esc_Sal(BARRERA,0);
    Esc_Sal(SEMAFORO,0);

    //inicializacion de variables
    tipoAuto=0;//no hay ningun auto
    indicePila=0;//no hay elementos en la pila

    loopinit();
    //----Bucle Infinito-----
    {
    while (1)

    loophead();

    costate tcp_ip init_on
    {
        ifdown(IF_ETH0);
        while (ifpending(IF_ETH0) != IF_DOWN)
        {
            tcp_tick(NULL);
            yield;
        }
        printf("\nInterfaz desactivada");

        ifconfig(IF_ETH0, IFS_IPADDR,atona(DIR_IP), IFS_NETMASK, atona(MASCARA), IFS_UP,
IFS_END);
    }
}

```

```

ifup(IF_ETH0);

while (ifpending(IF_ETH0) != IF_UP)
{
    tcp_tick(NULL);
    yield;
}
printf("\nInterfaz activada");
}

/*
costate
{

    Esc_Sal(BARRERA,1);
    Esc_Sal(SEMAFORO,1);
    waitfor(DelaySec(1));
    Esc_Sal(BARRERA,0);
    Esc_Sal(SEMAFORO,0);
    waitfor(DelaySec(1));

}

costate
{
    printf("\nBARRERA IR=%d",Lec_Ent((int)BARRERA_IR));
    printf("\nPEANA1=%d",Lec_Ent((int)PEANA1));
    printf("\nPEANA2=%d",Lec_Ent((int)PEANA2));
    waitfor(DelaySec(1));
} */

costate ini_disp init_on
{
    buf[0]=111;//direccion del display
    buf[1]=0;
    waitfordone{ Wr_Com1((char *)buf,2);}
    waitfor(DelaySec(1));
}
/*
costate
{
    buf[0]=1;
    buf[1]=2;
    buf[2]=3;
    buf[3]=4;
    waitfordone{ Wr_Com1((char *)buf,4);}
    waitfor(DelayMs(500));
    aux=serFread((char *)bufRS,4,100); // envia respuesta
    serFrdFlush();
    buf[0]=0;
    buf[1]=0;
    buf[2]=0;
    buf[3]=0;
    if(aux>0)
    printf("\n\nLeidos=%d : %d %d %d %d\n\n",aux,bufRS[0],bufRS[1],bufRS[2],bufRS[3]);
    waitfor(DelaySec(1));
}
*/

```

```

//LA barrera IR debe estar lo más cerca posible a las peanas

constate IR always_on
{
    printf("\nInicia Tarea IR");

    waitfor(Lec_Ent((int)BARRERA_IR)==ACT);//Espera a que se active la entrada (Entradas con pull
up)
    waitfor(DelayMs(800)); //retaro para rebotes
    if(Lec_Ent((int)BARRERA_IR)==ACT) CoBegin(& analisis);//inicializa analisis del numero de
ejes del auto
    else CoBegin(& IR);//reinicia tarea IR
    waitfor(Lec_Ent((int)BARRERA_IR)==INACT);//Espera a que haya pasado el auto
    CoPause(& analisis); //termina analisis
    if(tipoAuto>0) //verifica datos obtenidos
    {
        if(indicePila<MAX_PILA)
        {
            pila[indicePila]=tipoAuto; //ingresa dato a la pila
            indicePila++; //incrementa la pila
        }
    }
    printf("\nFinaliza tarea IR Tipo=%d", tipoAuto);
}

constate analisis init_on
{
    tipoAuto=0; //no hay auto

    printf("\nInicia Tarea Analisis");

    //Primer Eje
    waitfor(Lec_Ent((int)PEANA1)==ACT);//espera a que se active peana1
    printf("\nPrimer Eje");
    waitfor(DelayMs(800)); //retaro para rebotes
    waitfor(Lec_Ent((int)PEANA1)==INACT);//espera a que se desactive peana1

    tipoAuto=1;
    printf("\nTipo=%d", tipoAuto);

    //Segundo Eje
    waitfor(Lec_Ent((int)PEANA1)==ACT);//espera a que se active peana1
    printf("\nSegundo Eje");
    waitfor(DelayMs(800)); //retaro para rebotes
    while(Lec_Ent((int)PEANA1)==ACT)
    {
        if( (Lec_Ent((int)PEANA2)==ACT) )tipoAuto=2;
        yield;
    }
    //if( (Lec_Ent((int)PEANA1)==0) && (Lec_Ent((int)PEANA2)==0) )tipoAuto=2;
    waitfor(Lec_Ent((int)PEANA1)==INACT);//espera a que se desactive peana1
    printf("\nTipo=%d", tipoAuto);

    //Tercer Eje
    waitfor(Lec_Ent((int)PEANA1)==ACT);//espera a que se active peana1
    printf("\nTercero Eje");
    waitfor(DelayMs(800)); //retaro para rebotes
    while(Lec_Ent((int)PEANA1)==ACT)
    {
        if( (Lec_Ent((int)PEANA2)==ACT) )tipoAuto=3;
    }
}

```



```

    yield;
}
//if( (Lec_Ent((int)PEANA1)==0) && (Lec_Ent((int)PEANA2)==0) )tipoAuto=3;
waitfor(Lec_Ent((int)PEANA1)==INACT); //espera a que se desactive peanal
printf("\nTipo=%d",tipoAuto);
}

costate Salida//Pulso de activacion para la barrera de salida
{

    Esc_Sal(BARRERA,1);
    waitfor(DelaySec(DLYBARRERA));
    Esc_Sal(BARRERA,0);

}

costate Semaforo//Activacion del semaforo
{
    Esc_Sal(SEMAFORO,1);
    waitfor(DelaySec(DLYSEMAFORO));
    Esc_Sal(SEMAFORO,0);
}

costate //Pila TCP/IP-----
{

    tcp_tick(NULL);

    }//-----

costate //Comunicacion TCP/IP -----
{

    tcp_listen(&Socket,(int)PUERTO, 0, 0, NULL, 0); // escucha la llegada de una conexión en el
    puerto
    // espera por la coneccion
    while((-1 == sock_bytesready(&Socket)) && (0 == sock_established(&Socket)))
    { // suelta el control para que otras tareas se ejecuten
        //printf("\nSocketEstablished=%d",sock_established(&Socket));
        yield;
    }
    while(sock_established(&Socket))
    {

        // obtener datos
        length = sock_fastread(&Socket, buf, MAX_BUFSIZE_TCP);

        if(length > 0)
        { //si se recibio datos

            printf("\nLeidos=%d\n",length);
            for(i=0;i<length;i++) printf("%d ",buf[i]);

```

```

switch(buf[0])//Funcion
{
case 0: //Funcion 0 (devuelve el tipo de auto)

    buf[0]=0;//funcion
    if(indicePila>0)
    {
        buf[1]=pila[0];//tipo de auto

        for(i=0;i<indicePila;i++)
            pila[i]=pila[i+1];

        indicePila--;//reduce pila

    } else buf[1]=0;//no hay autos
    printf("\nResponde");
    sock_fastwrite(&Socket, buf, 2);//respuesta

    break;

case 1: //Funcion 1 (recibe tarifa)

        dolares=buf[1];
        centavos=buf[2];
        buf[0]=111;//direccion del display
        buf[1]=dolares;
        waitfordone{Wr_Com1((char *)buf,2);}
        //Envia por rs232 los datos

        // sock_fastwrite(&Socket, buf, length);//respuesta (eco)
        break;

case 2: //Funcion 2 (abre barreras)
    CoBegin(&Salida);//Activa salida
    CoBegin(&Semaforo);//Activa semaforo
    CoBegin(&ini_disp);//encera display
    break;

}

//printf("\nLeidos=%d\n",length);
//for(i=0;i<length;i++) printf("%d ",buf[i]);
//sock_fastwrite(&Socket, buf, length);//respuesta (eco)
}

yield; // suelta el control a otras tareas
}
sock_close(&Socket);

} //-----

} //while -----

}

```

```
// __DEFINICION DE FUNCIONES_____
```

```
/* START FUNCTION DESCRIPTION *****  
Ini_ES
```

SYNTAX: void Ini_ES()

DESCRIPTION: Inicializa todas las entradas salidas de la tarjeta
SIH. Debe ser llamada antes se utilizar cualquier
función de E/S.

```
END DESCRIPTION *****/
```

```
void Ini_ES()  
{  
  //SALIDAS=====  
  //WrPortI(PBDDR, &PBDDRShadow, 0xFF); //portB como Salidas  
  
  WrPortI(PBDR, &PBDRShadow, 0xFF);  
  WrPortI(PBDDR, &PBDDRShadow, 0xFF);  
  
  //ENTRADAS=====  
  
  //Entrada 1-8  
  WrPortI(SPCR, &SPCRShadow, 0x080); //PortA como entrada  
  
}
```

```
/* START FUNCTION DESCRIPTION *****  
Lec_Ent
```

SYNTAX: nodebug int Lec_Ent(int entrada)

DESCRIPTION: Devuelve el valor lógico, que se encuentra en una
entrada de la tarjeta.

PARAMETER: entrada-> Entero que indica la entrada a leer, los valores
que puede tomar son:

0-7: entradas PuertoA

RETURN VALUE: Valor lógico de la entrada leída (0-1). (-1) Si el parámetro
ingresado es incorrecto.

```
END DESCRIPTION *****/
```

```
int Lec_Ent(int entrada)  
{
```

```
  switch(entrada)  
  {  
    case 0:  
    case 1:  
    case 2:  
    case 3:  
    case 4:
```

```

case 5:
case 6:
case 7:
    return( BitRdPortI(PADR,entrada)); //devuelve un valor presente en la entrada
    break;

default: return(-1);//se ha ingresado mal parametro entrada

}

}

/* START FUNCTION DESCRIPTION *****
Esc_Sal

SYNTAX: nodebug int Esc_Sal(int salida, int estado)

DESCRIPTION: Escribe el estado lógico indicado par las salidas de la
tarjeta.

PARAMETER1: salida-> Entero que indica la entrada a leer, los valores
que puede tomar son:

                2-7: salidas del puertoB

PARAMETER2: estado-> Entero que indica el estado lógico de la salida. (0-1)

RETURN VALUE: (-1) Si el parametro ingresado es incorrecto.

END DESCRIPTION *****/

int Esc_Sal(int salida, int estado)
{

switch(salida)
{
case 2:
case 3:
case 4:
case 5:
case 6:
case 7:
    BitWrPortI(PBDR, &PBDRShadow,(int)(estado),(int)(salida)); //escribe la salida
    break;

default: return(-1);//se ha ingresado mal parametro salida

}

}

/* START FUNCTION DESCRIPTION *****

```

Ini_Serial

SYNTAX: void Ini_Serial()

DESCRIPTION: Inicializa los puertos RS232

END DESCRIPTION *****/

```
void Ini_Serial()
{
    // Open serial port E (COM1)
    serCopen(COM_BAUD);
    serCparity(PARAM_EPARITY);
    serC databits(PARAM_8BIT);
    serCflowcontrolOff();

    // Clear serial buffers (COM1)
    serCwrFlush();
    serCrdFlush();
}
```

/* START FUNCTION DESCRIPTION *****/

Wr_Com1

SYNTAX: cofunc void Wr_Com1(char *buf, int tam)

DESCRIPTION: Escribe en el puerto de comunicacion especifico, los datos de un buffer de un tamaño predefinido. La funcion detecta si el bus de datos esta ocupado con el fin de evitar colision y perdida de datos.

PARAMETER1: buf-> puntero a un buffer de datos a transmitir .

PARAMETER2: tam-> numero de datos del buffer a transmitir.

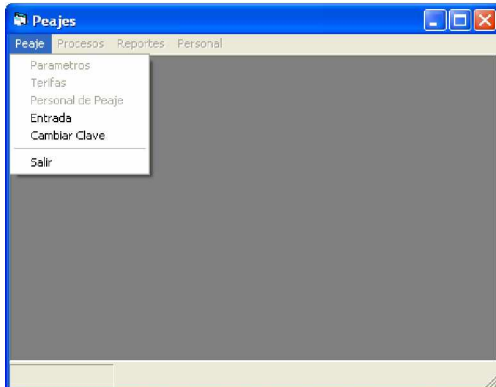
END DESCRIPTION *****/

```
cofunc void Wr_Com1(char *buf, int tam)
{
    serCwrite((char *)buf,tam); // envia respuesta
    while (serCwrFree() != COUTBUFSIZE)yield;
    // then, wait until the Tx data register and the Tx shift register
    // are both empty
    while (BitRdPortI(SCSR, 3) || BitRdPortI(SCSR, 2))yield;
    serCwrFlush(); //listo el buffer para la siguiente lectura
}
```

ANEXO E

CÓDIGO DE PROGRAMA DE LA INTERFAZ GRÁFICA

PANTALLA 1



```
Private Sub carriles_Click()
```

```
End Sub
```

```
Private Sub aforo_Click()
```

```
Form1.Show
```

```
End Sub
```

```
Private Sub cambia_Click()
```

```
Form6.Show
```

```
End Sub
```

```
Private Sub entrada_Click()
```

```
login.Show
```

```
End Sub
```

```
Private Sub inicio_Click()
```

```
menu.entrada.Enabled = False
```

```
Form9.Show
```

```
End Sub
```

```
Private Sub MDIForm_Load()
```

```
menu.procesos.Enabled = False
```

```
menu.reportes.Enabled = False
```

```
menu.para.Enabled = False
```

```
menu.tarifas.Enabled = False
```

```
menu.personal.Enabled = False
```

```
menu.salir.Enabled = False
```

```
End Sub
```

```
Private Sub para_Click()
```

```
Form2.Show
```

```
End Sub
```

```
Private Sub personal_Click()
```

```
Form5.Show
```

```
End Sub
```

```
Private Sub personal2_Click()
```

```
Form1.Show
```

```
End Sub
```

```

Private Sub pretirno_Click()
    menu.entrada.Enabled = True
    Unload Form9
    Form8.Show
    Form9.conexion.Close
    MDIForm_Load
End Sub

```

```

Private Sub salir_Click()
    End
End Sub

```

```

Private Sub tarifas_Click()
    Form4.Show
End Sub

```

```

Private Sub turno_Click()
    Form3.Show

```

```

End Sub

```

PANTALLA 2



```

Private Sub btn1_Click()
    Dim tam As Integer
    Dim user() As String, query As String
    Dim Pass() As String, cont As Integer
    Dim cargo() As String
    Dim consulta As New ADODB.Recordset
    Dim conexion As New ADODB.Connection
    Dim cnn As String
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    conexion.Open cnn
    query = "Select usu_user,usu_pass,usu_cargo from peaje.usuario"
    consulta.Open query, conexion
    If consulta.EOF = False Then
        Do Until consulta.EOF
            tam = tam + 1
            consulta.MoveNext
        Loop
        consulta.MoveFirst
        ReDim user(tam)

```



```

ReDim Pass(tam)
ReDim cargo(tam)
Do Until consulta.EOF
    user(cont) = consulta.Fields(0)
    Pass(cont) = consulta.Fields(1)
    cargo(cont) = consulta.Fields(2)
    cont = cont + 1
    consulta.MoveNext
Loop
End If
consulta.Close
conexion.Close 'se cierra la conexion con la base de datos
For i = 0 To tam
    If Trim(txt_user.Text) = user(i) And Trim(txt_pas.Text) = Pass(i) Then
        estado = "v"
        menu.Show
        nomglobal = user(i)
        If cargo(i) = "Operador" Then
            menu.para.Enabled = False
            menu.tarifas.Enabled = False
            menu.personal.Enabled = False
            menu.reportes.Enabled = False
            menu.procesos.Enabled = True
            menu.pretirno.Enabled = False
            menu.personal2.Enabled = False
            menu.personal.Enabled = False
        End If
        If cargo(i) = "Administrador" Then
            menu.para.Enabled = True
            menu.tarifas.Enabled = True
            menu.personal.Enabled = True
            menu.reportes.Enabled = True
            menu.procesos.Enabled = False
            menu.pretirno.Enabled = False
            menu.personal2.Enabled = True
            menu.salir.Enabled = True
        End If
        Unload Me
        i = tam + 1
    End If
Next
If estado <> "v" Then
    MsgBox "El nombre de Usuario o La contraseña son incorrectos. Intente de nuevo", vbOKOnly,
"Verifique El Usuario Y Contraseña"
    txt_user.Text = ""
    txt_pas.Text = ""

    End If
End Sub

Private Sub btn2_Click()
    Unload Me
End Sub

```

PANTALLA 3

Parametros

Peaje

Nombre: CASETA PEAJE

Ubicacion: AUTOPISTA

Responsable: GERENTE

Tramo: TRAMO CONCESIONADO

Logotipo: C:\LOGO.JPG

Horarios

Diurno:	De: 07:00:00 am a 13:59:59 pm
Vespertino:	De: 14:00:00 pm a 21:59:59 pm
Nocturno:	De: 21:00:00 pm a 06:59:59 am

Aceptar Cancelar

```
Private Sub Command1_Click()
```

```
Form10.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
Dim conexion As New ADODB.Connection
```

```
Dim cnn As String, query As String, fec As String, aux As Integer
```

```
cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
```

```
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
```

```
aux = Len(path1)
```

```
Do While (InStr(1, path1, "\") <> 0)
```

```
aux2 = InStr(1, path1, "\")
```

```
path1 = Mid(path1, 1, (aux2 - 1)) & "/" & Mid(path1, aux2 + 1, aux)
```

```
Loop
```

```
txtlogo.Text = path1
```

```
If txtnom <> "" And txtubi <> "" And txtres <> "" And txttra <> "" And txtlogo <> "" Then
```

```
conexion.Open cnn
```

```
conexion.Execute "update peaje.parametro set par_nom="" & txtnom.Text & ",par_ubi="" &
```

```
txtubi.Text & ",par_res="" & txtres.Text & ",par_tra="" & txttra.Text & ",par_logo="" & path1 & "
```

```
WHERE par_cod =1"
```

```
conexion.Close
```

```
MsgBox "Sus Datos han sido guardados con exito", vbOKOnly
```

```
Unload Me
```

```
Picture1.Picture = path2
```

```
Else
```

```
MsgBox "Ingrese correctamente los Datos", vbCritical, "Error"
```

```
End If
```

```
End Sub
```



```

",conf_remolque=" & CDb(txt4.Text) & ",conf_ruc=" & txtruc.Text & ",conf_ciudad=" & txtciu.Text
& ",conf_denominacion=" & txtlden.Text & ",conf_nom=" & txtnom.Text & ",conf_dir=" &
txtdir.Text & ",conf_dirfact=" & txtDirfact.Text & ",conf_num=" & txtnum.Text & ",conf_validez=" &
fecha.Value & ",conf_comp=" & txtcomp.Text & " WHERE conf_cod =1"
    fec = Format(fecha.Value, "yyyy-MM-dd")
    If txtruc <> "" And txtciu <> "" And txtlden <> "" And txtnom <> "" And txtdir <> "" And txtDirfact <>
"" And txtnum <> "" And fecha <> "" And txtcomp <> "" And txtiva <> "" And txtauto <> "" And txt2
<> "" And txt3 <> "" And txt4 <> "" Then
        conexion.Open cnn
        conexion.Execute "update peaje.configuracion set conf_iva=" & txtiva.Text & ",conf_auto=" &
Cdbl(txtauto.Text) & ",conf_dos=" & Cdbl(txt2.Text) & ",conf_tres=" & Cdbl(txt3.Text) &
",conf_remolque=" & Cdbl(txt4.Text) & ",conf_ruc=" & txtruc.Text & ",conf_ciudad=" & txtciu.Text
& ",conf_denominacion=" & txtlden.Text & ",conf_nom=" & txtnom.Text & ",conf_dir=" &
txtdir.Text & ",conf_dirfact=" & txtDirfact.Text & ",conf_num=" & txtnum.Text & ",conf_validez=" &
fecha.Value & ",conf_comp=" & txtcomp.Text & " WHERE conf_cod =1"
        MsgBox "Sus Datos han sido guardados con exito", vbOKOnly
        conexion.Close
        Unload Me
    Else
        MsgBox "Ingrese correctamente los Datos", vbCritical, "Error"
    End If

```

End Sub

```
Private Sub Command7_Click()
```

```
    Unload Me
```

```
End Sub
```

PANTALLA 6

```
Private Sub btnCancel_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Dim conexion As New ADODB.Connection
```

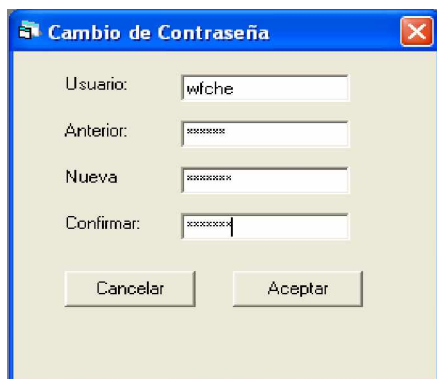
```
    Dim cnn As String, query As String
```

```

cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
conexion.Open cnn
If txtced <> "" And txtnom <> "" And txtape <> "" And txtdir <> "" And cmbcar <> "" And txtuser <>
"" And txtpass <> "" And txtpass = txtpass Then
    query = "insert into
peaje.usuario(usu_cedula,usu_nombre,usu_apellido,usu_direccion,usu_telefono,usu_celular,usu_cargo,us
u_user,usu_pass) values(" & txtced.Text & "," & txtnom.Text & "," & txtape.Text & "," & txtdir.Text
& "," & txttel.Text & "," & txtcel.Text & "," & cmbcar.Text & "," & txtuser.Text & "," &
txtpass.Text & ""
    conexion.Execute "insert into peaje.usuario(usu_cedula, usu_nombre, usu_apellido, usu_direccion,
usu_telefono, usu_celular, usu_cargo, usu_user, usu_pass) values(" & txtced.Text & "," & txtnom.Text
& "," & txtape.Text & "," & txtdir.Text & "," & txttel.Text & "," & txtcel.Text & "," & cmbcar.Text
& "," & txtuser.Text & "," & txtpass.Text & ""
    conexion.Close
Else
    MsgBox "Ingrese los datos correctamente", vbCritical
End If
Unload Me
End Sub

```

PANTALLA 7



```

Private Sub btn_exit_Click()
Unload Me
End Sub

```

```

Private Sub btn_ok_Click()
    Dim conexion As New ADODB.Connection
    Dim result As New ADODB.Recordset
    Dim cnn As String, query As String, cedula As String
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    If txt_usu.Text <> "" And txt_pass.Text <> "" And txt_n.Text <> "" And txt_conf.Text <> "" Then
        conexion.Open cnn
        query = "Select usu_cedula, usu_user, usu_pass from usuario Where usu_user =" & txt_usu.Text &
""
        result.Open query, conexion
        If result.Fields(0) = "" Then
            MsgBox "El usuario no existe", vbOKOnly
        Else
            If txt_pass.Text = result.Fields(2) Then
                If txt_n.Text <> txt_conf Then
                    MsgBox "La contraseña nueva no coincide", vbOKOnly
                Else
                    cedula = result.Fields(0)
                End If
            End If
        End If
    End If

```

```

cedula & ""
        query = "update peaje.usuario set usu_pass="" & txt_n.Text & "" WHERE usu_cedula ="" &
conexion.Execute query
conexion.Close
MsgBox "El cambio se realizo exitosamente", vbOKOnly
txt_usu.Text = ""
txt_pass.Text = ""
txt_n.Text = ""
txt_conf.Text = ""
    End If
Else
    MsgBox "El usuario o contraseña no coincide", vbOKOnly
End If
End If
Else
    MsgBox "Los datos se encuentran vacios", vbOKOnly
End If

End Sub

```

PANTALLA 8



```

Dim consulta As New ADODB.Recordset
Dim con_usu As New ADODB.Recordset
Dim conexion As New ADODB.Connection
Dim cnn As String, query As String, rec As Integer, auxn As Integer
Dim valor As Double, auto As String, Tipo As String, empresa As String, ruc As String

```

```

Private Sub Form_Load()
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    Dim consulta3 As New ADODB.Recordset, fecha As String
    txt_fecha.Text = Date
    auxn = 0
    txt_hora.Text = Time
    WsockClient.RemoteHost = "192.168.100.100" ' Dirección IP del dispositivo (Rabbit)

```

```

WSockClient.RemotePort = 13000 ' Puerto
EstadoWinsock
If WSockClient.State <> sckOpen Then
    WSockClient.Close
    WSockClient.Connect
End If
EstadoWinsock

conexion.Open cnn
fecha = Format(Date, "yyyy/mm/dd") & " " & Format(Time, "hh:mm:ss")
query = "SELECT tic_cod FROM ticket order by tic_cod DESC"
consulta3.Open query, conexion
txt_sec.Text = Str(Int(consulta3.Fields(0)) + 1)
consulta3.Close
conexion.Execute "Insert into turno (tur_tic_ini,tur_fec_ini,tur_op)values(" & Int(txt_sec.Text) & ","
& fecha & "," & nomglobal & """)
query = "Select conf_iva,conf_auto,conf_dos,conf_tres,conf_remolque,conf_ruc,conf_nom from
peaje.configuracion where conf_cod = 1"
consulta.Open query, conexion
query = "Select usu_nombre, usu_apellido, usu_cedula from peaje.usuario where usu_user = " &
nomglobal & """
con_usu.Open query, conexion
txt_op.Text = con_usu.Fields(0) & " " & con_usu.Fields(1)
codglobal = con_usu.Fields(2)
con_usu.Close
ruc = consulta.Fields(5)
empresa = consulta.Fields(6)
menu.pretirno.Enabled = True
End Sub
Private Sub btn_auto_Click()
txt_val.Text = consulta.Fields(1) & ".00"
btn_auto.BackColor = 255
btn_2.BackColor = -2147483633
btn_3.BackColor = -2147483633
btn_rem.BackColor = -2147483633
auto = "Liviano"
btn_pago.Enabled = True
If auxn = 1 Then
    ProcesarCarro 1
    auxn = 0
End If
End Sub
Private Sub bot1()
txt_val.Text = consulta.Fields(1) & ".00"
btn_auto.BackColor = 255
btn_2.BackColor = -2147483633
btn_3.BackColor = -2147483633
btn_rem.BackColor = -2147483633
auto = "Liviano"
btn_pago.Enabled = True

End Sub
Private Sub btn_2_Click()
txt_val.Text = consulta.Fields(2) & ".00"
btn_auto.BackColor = -2147483633
btn_2.BackColor = 255
btn_3.BackColor = -2147483633
btn_rem.BackColor = -2147483633
auto = "Pesado"
btn_pago.Enabled = True

```

```

    If auxn = 1 Then
        ProcesarCarro 2
        auxn = 0
    End If
End Sub
Private Sub bot2()
    txt_val.Text = consulta.Fields(2) & ".00"
    btn_auto.BackColor = -2147483633
    btn_2.BackColor = 255
    btn_3.BackColor = -2147483633
    btn_rem.BackColor = -2147483633
    auto = "Pesado"
    btn_pago.Enabled = True

End Sub

Private Sub btn_3_Click()
    txt_val.Text = consulta.Fields(3) & ".00"
    btn_auto.BackColor = -2147483633
    btn_2.BackColor = -2147483633
    btn_3.BackColor = 255
    btn_rem.BackColor = -2147483633
    auto = "Extra Pesado"
    btn_pago.Enabled = True
    If auxn = 1 Then
        ProcesarCarro 3
        auxn = 0
    End If
End Sub
Private Sub bot3()
    txt_val.Text = consulta.Fields(3) & ".00"
    btn_auto.BackColor = -2147483633
    btn_2.BackColor = -2147483633
    btn_3.BackColor = 255
    btn_rem.BackColor = -2147483633
    auto = "Extra Pesado"
    btn_pago.Enabled = True

End Sub
Private Sub btn_rem_Click()
    txt_val.Text = consulta.Fields(4) & ".00"
    btn_auto.BackColor = -2147483633
    btn_2.BackColor = -2147483633
    btn_3.BackColor = -2147483633
    btn_rem.BackColor = 255
    auto = "Remolque"
    btn_pago.Enabled = True
    If auxn = 1 Then
        ProcesarCarro 4
        auxn = 0
    End If
End Sub
Private Sub bot4()
    txt_val.Text = consulta.Fields(4) & ".00"
    btn_auto.BackColor = -2147483633
    btn_2.BackColor = -2147483633
    btn_3.BackColor = -2147483633
    btn_rem.BackColor = 255
    auto = "Remolque"
    btn_pago.Enabled = True

```



```

End Sub
Private Sub btn_pago_Click()
    Dim nombre As String, fecha As String, factura As Double
    Dim consulta2 As New ADODB.Recordset
    btn_auto.Enabled = False
    btn_2.Enabled = False
    btn_3.Enabled = False
    btn_rem.Enabled = False

    v1.Visible = True
    r1.Visible = False
    v2.Visible = False
    r2.Visible = True
    fecha = Format(Date, "yyyy/mm/dd") & " " & Format(Time, "hh:mm:ss")
    nombre = "foto" & Format(Date, "yyyymmdd") & "-" & Format(Time, "hhmmss")
    dirpic = "C:/imagenes/" & nombre & ".jpg"
    video.SaveToDiskEx dirpic, hcwFF_JPEG_jpg, hcwFIF_16BPP, hcwCMP_JPEG, 60
    query = "Insert into ticket(tic_val,tic_fec,tic_op,tic_tipo,tic_pago,tic_pic,tic_op_ced)values(" &
    txt_val.Text & "," & fecha & "," & txt_op.Text & "," & auto & "," & Tipo & "," & dirpic & "," &
    codglobal & ")"
    conexion.Execute query
    query = "SELECT tic_cod FROM ticket order by tic_cod DESC"
    consulta2.Open query, conexion
    factura = consulta2.Fields(0)
    txt_sec = factura + 1
    'Codigo Impresion
    Printer.ScaleMode = vbCharacters 'establezco caracteres para controlar la impresion
    Printer.TrackDefault = True 'siempre apunta a la impresora predeter
    'el punto y coma (;) hace que la linea se imprima en el mismo renglon

    'para el logo y datos fijos
    i = 1
    For i = 1 To 3
        Printer.Print " "
    Next
    Printer.Font.Name = "Arial"
    Printer.Font.Size = 12
    Printer.FontBold = True
    Printer.Print Tab(5); empresa
    Printer.Print Tab(5); "RUC: " & ruc
    Printer.Print Tab(5); "N° Fact: " & factura
    Printer.Print Tab(5); "Consumidor Final"
    Printer.Print Tab(5); "FECHA: "; UCase(Format(Now, "dd/mmm/yyyy"))
    Printer.Print Tab(5); "HORA: " & UCase(Format(Now, "hh:mm am/pm"))
    Printer.Print Tab(5); "Forma de Pago: EFECTIVO"
    Printer.Print Tab(5); "Categoria: " & auto
    Printer.Print Tab(5); "Tarifa: ";
    Printer.Print Tab(20); "$ " & txt_val.Text
    Printer.Print Tab(5); "IVA:";
    Printer.Print Tab(20); "$ 0.00"
    Printer.Print Tab(5); "Total:";
    Printer.Print Tab(20); "$ " & txt_val.Text
    Printer.EndDoc
    rec = 0

    EstadoWinsock
    If WSocketClient.State = sckConnected Then
        WSocketClient.SendData 2 ' Funcion 2
    End If

```

```

txt_val.Text = "00"
btn_auto.BackColor = -2147483633
btn_2.BackColor = -2147483633
btn_3.BackColor = -2147483633
btn_rem.BackColor = -2147483633
auto = ""
btn_pago.Enabled = False
Tipo = ""
End Sub

Private Sub WSocketClient_Connect()
    EstadoWinsock
End Sub

Private Sub btn_re_Click()
    btn_pago.Enabled = True
    btn_auto.Enabled = True
    btn_2.Enabled = True
    btn_3.Enabled = True
    btn_rem.Enabled = True
    v1.Visible = False
    r1.Visible = True
    v2.Visible = True
    r2.Visible = False
    Tipo = "Reclasificado"
    auxn = 1
    WSocketClient.SendData 0 ' Funcion 0
End Sub

Private Sub Form_Unload(Cancel As Integer)
    conexion.Close
    WSocketClient.Close
    EstadoWinsock
End Sub

Private Sub Timer1_Timer()
    txt_hora.Text = Time
    If rec = 0 Then
        EstadoWinsock
        If WSocketClient.State = sckConnected Then
            WSocketClient.SendData 0 ' Funcion 0
        End If
    End If
End Sub

Private Sub WSocketClient_DataArrival(ByVal bytesTotal As Long)
    Dim vtRecibido
    Dim DataRx() As Byte

    WSocketClient.GetData vtRecibido, vbArray + vbByte, bytesTotal 'Recibir Datos TCP
    DataRx = vtRecibido

    Select Case DataRx(0) ' Numero de Funcion
        ' Func 0: Tipo de Carro
        Case 0: ProcesarCarro (DataRx(1))

        ' Func 1: Tarifa
        Case 1:

        ' Func 2: Abrir Puerta
        Case 2:
    End Select
End Sub

```

End Select

End Sub

Public Function EstadoWinsock()

Select Case WSocket.State

Case 0: lblEstado.Caption = "Winsock: Cerrado"

Case 1: lblEstado.Caption = "Winsock: Abierto"

Case 2: lblEstado.Caption = "Winsock: Escuchando"

Case 3: lblEstado.Caption = "Winsock: Conexión Pendiente"

Case 4: lblEstado.Caption = "Winsock: Resolviendo Host"

Case 5: lblEstado.Caption = "Winsock: Host Resuelto"

Case 6: lblEstado.Caption = "Winsock: Conectando"

Case 7: lblEstado.Caption = "Winsock: Conectado"

Case 8: lblEstado.Caption = "Winsock: Equipo Cerrando Conexión"

Case 9: lblEstado.Caption = "Winsock: Error"

End Select

End Function

Public Function ProcesarCarro(ByVal Tipo As Byte)

Dim Dolares As Byte

Dim Centavos As Byte

Select Case Tipo

Case 0: lblTipoCarro.Caption = "No hay carro presente"
rec = 0

Case 1:

' Asigna Tarifa para <<Livianos>>

Dolares = CInt(consulta.Fields(1)) ' entre 0 y 255

' Centavos = 10 ' entre 0 y 99

bot1

rec = 1

Case 2:

' Asigna Tarifa para <<Pesados>>

Dolares = CInt(consulta.Fields(2)) ' entre 0 y 255

' Centavos = 20 ' entre 0 y 99

bot2

rec = 2

Case 3:

' Asigna Tarifa para <<Extra Pesados>>

Dolares = CInt(consulta.Fields(3)) ' entre 0 y 255

' Centavos = 30 ' entre 0 y 99

bot3

rec = 3

Case 4:

' Asigna Tarifa para <<Extra Pesados>>

Dolares = CInt(consulta.Fields(4)) ' entre 0 y 255

' Centavos = 30 ' entre 0 y 99

bot4

rec = 4

End Select

If Tipo > 0 Then ' Si existe un carro presente, envia la Tarifa

Dim bDataTx(2) As Byte

Dim vtEnviar As Variant

bDataTx(0) = 1 ' Funcion 1: Tarifa

```

bDataTx(1) = Dolares
bDataTx(2) = Centavos
vtEnviar = bDataTx
WSockClient.SendData vtEnviar
End If
End Function

```

PANTALLA 9

```

Dim res As Integer
Dim ticket As Double
Private Sub Command1_Click()
    Dim conexion As New ADODB.Connection, result As New ADODB.Recordset
    Dim cnn As String, query As String, aux As Integer, turno As String
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    conexion.Open cnn
    query = "Select tur_cod from turno order by tur_cod DESC"
    result.Open query, conexion
    aux = result.Fields(0)
    If Option1.Value = True Then
        turno = "Nocturno"
    End If
    If Option2.Value = True Then
        turno = "Matutino"
    End If
    If Option3.Value = True Then
        turno = "Vespertino"
    End If
    If turno <> "" Then
        res = MsgBox("Una vez confirmados los datos no podran ser modificados. ¿Esta seguro que los
datos son correctos?", vbYesNo)
        If res = 6 Then ' presiona Si
            query = "update peaje.turno set tur_tic_fin = " & ticket & ", tur_fec_fin = " & txt_fin.Text & ",
turno = " & turno & ", tur_val = " & CDb1(txtef.Text) & ",tur_bol = " & CDb1(txttic.Text) & ",tur_val_er
= " & CDb1(txtferr.Text) & ",tur_bol_er = " & CDb1(txt_error.Text) & " Where tur_cod = " & aux
            conexion.Execute "update peaje.turno set tur_tic_fin = " & ticket & ", tur_fec_fin = " &
txt_fin.Text & ", turno = " & turno & ", tur_val = " & CDb1(txtef.Text) & ",tur_bol = " &
CDbl(txttic.Text) & ",tur_val_er = " & CDb1(txtferr.Text) & ",tur_bol_er = " & CDb1(txt_error.Text) &
" Where tur_cod = " & aux
        End If
    Else
        MsgBox "Seleccione el turno", vbCritical, "Error"
    End If
    conexion.Close
    Unload Me

```

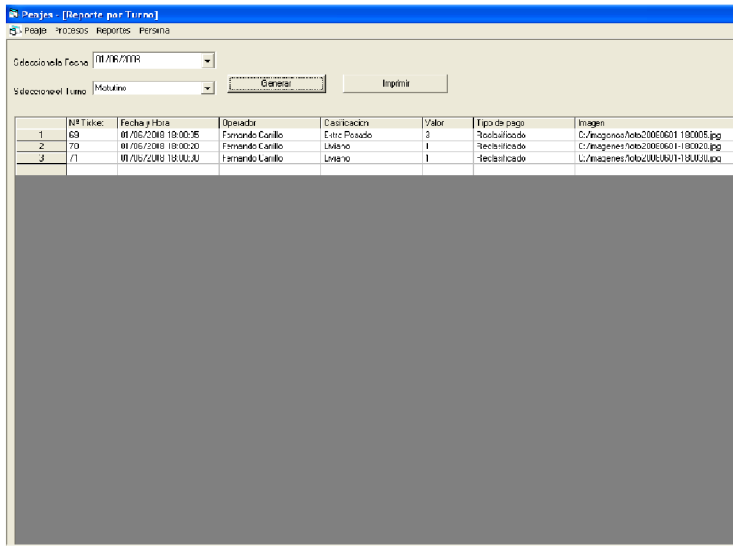
End Sub

```
Private Sub Command2_Click()  
Form9.Show  
menu.procesos.Enabled = True  
End Sub
```

```
Private Sub Form_Load()  
    Dim conexion As New ADODB.Connection  
    Dim cnn As String, aux As Double, val As Double, tic As Double  
    Dim consulta As New ADODB.Recordset, consulta2 As New ADODB.Recordset  
    val = 0  
    tic = 0  
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended  
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"  
    conexion.Open cnn  
    query = "Select usu_nombre,usu_apellido from peaje.usuario where usu_user = " & nomglobal & ""  
    consulta.Open query, conexion  
    txt_nom.Text = consulta.Fields(0) & " " & consulta.Fields(1)  
    txt_fin.Text = Format(Date, "yyyy-MM-dd") & " " & Time  
    consulta.Close  
    query = "Select tur_tic_ini from turno order by tur_tic_ini DESC"  
    consulta2.Open query, conexion  
    aux = consulta2.Fields(0)  
    query = "Select tic_cod, tic_val from peaje.ticket where tic_cod >= " & aux  
    consulta.Open query, conexion  
    Do Until (consulta.EOF)  
        val = val + consulta.Fields(1)  
        tic = tic + 1  
        ticket = consulta.Fields(0)  
        consulta.MoveNext  
    Loop  
    txtef.Text = val  
    txteferr.Text = val  
    txttic.Text = tic  
    txt_error.Text = tic  
    consulta2.Close  
    consulta.Close
```

End Sub

PANTALLA 10



Private Sub btncon_Click()

Dim date1 As String, date2 As String

Dim conexion As New ADODB.Connection, result As New ADODB.Recordset

Dim cnn As String, query As String, fec As String, aux As Integer, i As Integer, tic1 As Double, tic2 As Double

i = 1

cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"

conexion.Open cnn

date1 = Format(fecha.Value, "yyyy/mm/dd") & " 00:00:00"

turno = Turnos.Text

query = "Select turno, tur_tic_ini, tur_tic_fin, tur_fec_ini, tur_val from turno where tur_fec_ini >= " & date1 & " and turno = " & turno & """

result.Open query, conexion

If result.EOF = False Then

tic1 = result.Fields(1)

tic2 = result.Fields(2)

txtefe.Text = result.Fields(4)

End If

result.Close

query = "Select tic_cod, tic_fec, tic_op, tic_tipo, tic_val, tic_pago, tic_pic from ticket Where tic_cod >= " & tic1 & " and tic_cod <= " & tic2

result.Open query, conexion

If result.EOF = True Then

No existen Resultados

Else

Do Until (result.EOF)

flex.TextMatrix(i, 0) = i

flex.TextMatrix(i, 1) = result.Fields(0)

flex.TextMatrix(i, 2) = result.Fields(1)

flex.TextMatrix(i, 3) = result.Fields(2)

flex.TextMatrix(i, 4) = result.Fields(3)

flex.TextMatrix(i, 5) = result.Fields(4)

flex.TextMatrix(i, 6) = result.Fields(5)

flex.TextMatrix(i, 7) = result.Fields(6)

i = i + 1

```

        flex.Rows = i + 1
        result.MoveNext
    Loop
End If
result.Close
conexion.Close
txttic.Text = flex.Rows - 2
For i = 0 To flex.Cols - 1
    flex.TextMatrix(flex.Rows - 1, i) = ""
Next
End Sub
Private Sub Command1_Click()

    Dim conexion As New ADODB.Connection, result As New ADODB.Recordset
    Dim cnn As String, query As String, nom As String
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    conexion.Open cnn
    query = "Select usu_nombre, usu_apellido from usuario where usu_user = " & nomglobal & ""
    result.Open query, conexion
    nom = result.Fields(0) & " " & result.Fields(1)

    Printer.ScaleMode = vbCharacters 'establezco caracteres para controlar la impresion
    Printer.TrackDefault = True 'siempre apunta a la impresora predeter
    'el punto y coma (;) hace que la linea se imprima en el mismo renglon
    'para el logo y datos fijos
    i = 1
    For i = 1 To 3
        Printer.Print " "
    Next
    Printer.Font.Name = "Arial"
    Printer.Font.Size = 16
    Printer.FontBold = True
    Printer.Print Tab(31); "Estacion de Peaje"
    Printer.Font.Size = 14
    Printer.Print Tab(37); "Hoja de Reporte"
    Printer.Font.Size = 10
    Printer.Print " "
    Printer.Print " "
    Printer.Print Tab(5); "Nombre: ";
    Printer.Print Tab(20); nom
    Printer.Print Tab(5); "Fecha inicio: ";
    Printer.Print Tab(20); fecha.Value
    Printer.Print Tab(5); "Turno: ";
    Printer.Print Tab(20); Turnos.Text
    Printer.Print " "
    Printer.FontBold = False
    For i = 0 To flex.Rows - 1
        If i = 1 Then
            Printer.Print " "
        End If
        Printer.Print Tab(2); flex.TextMatrix(i, 0); 'Enumeracion
        Printer.Print Tab(15); flex.TextMatrix(i, 1); 'Ticket
        Printer.Print Tab(35); flex.TextMatrix(i, 2); 'Fecha y Hora
        Printer.Print Tab(60); flex.TextMatrix(i, 3); 'Operador
        Printer.Print Tab(82); flex.TextMatrix(i, 4); 'Clasificacion
        Printer.Print Tab(99); flex.TextMatrix(i, 5); 'Valor
        Printer.Print Tab(107); flex.TextMatrix(i, 6) 'Tipo de Pago
        Printer.Print Tab(84); flex.TextMatrix(i, 7)
    Next

```

```

Printer.FontBold = True
Printer.Print " "
Printer.Print " "
Printer.Print Tab(90); "Total Tickets:";
Printer.Print Tab(110); txttic.Text
Printer.Print " "
Printer.Print Tab(90); "Total Efectivo:";
Printer.Print Tab(110); "$ " & txtefe.Text

```

```
Printer.EndDoc
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
flex.ColWidth(0) = 1000
```

```
flex.ColWidth(1) = 1000
```

```
flex.ColWidth(2) = 2000
```

```
flex.ColWidth(3) = 2000
```

```
flex.ColWidth(4) = 2000
```

```
flex.ColWidth(5) = 1000
```

```
flex.ColWidth(6) = 2000
```

```
flex.ColWidth(7) = 4000
```

```
flex.ColAlignment(0) = 3
```

```
flex.ColAlignment(1) = 0
```

```
flex.ColAlignment(2) = 0
```

```
flex.ColAlignment(5) = 0
```

```
flex.TextMatrix(0, 1) = "N° Ticket"
```

```
flex.TextMatrix(0, 2) = "Fecha y Hora"
```

```
flex.TextMatrix(0, 3) = "Operador"
```

```
flex.TextMatrix(0, 4) = "Clasificacion"
```

```
flex.TextMatrix(0, 5) = "Valor"
```

```
flex.TextMatrix(0, 6) = "Tipo de pago"
```

```
flex.TextMatrix(0, 7) = "Imagen"
```

```
End Sub
```

PANTALLA 11

Peajes - (Resorte por Fecha)

Inicio: 28/05/2008 Fecha Fin: 01/05/2008

N° Ticket	Fecha y Hora	Operador	Clasificación	Valor	Tipo de pago	Imagen	
1	20	30/05/2008 11:22:10	Fernando Carrillo	Peasido	2	Reclaficada	C:\imagenes\Auto20080527-09221
2	20	28/05/2008 11:22:35	Fernando Carrillo	Remolque	4	Reclaficada	C:\imagenes\Auto20080527-01212
3	01	24/05/2008 22:48:48	Fernando Carrillo	Pura Pasero	3	Reclaficada	C:\imagenes\Auto20080525-2248
4	32	23/05/2008 22:48:57	Fernando Carrillo	Remolque	4	Reclaficada	C:\imagenes\Auto20080525-2248
5	25	28/05/2008 22:49:04	Fernando Carrillo	Liviano	1	Reclaficada	C:\imagenes\Auto20080525-2248
6	24	20/05/2008 22:52:17	Fernando Carrillo	Liviano	2	Reclaficada	C:\imagenes\Auto20080525-22521
7	25	28/05/2008 22:52:24	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080525-2252
8	27	24/05/2008 22:52:40	Fernando Carrillo	Peasido	2	Reclaficada	C:\imagenes\Auto20080525-2252
9	30	28/05/2008 22:58:30	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080525-2258
10	30	28/05/2008 22:58:38	Fernando Carrillo	Liviano	1	Reclaficada	C:\imagenes\Auto20080525-2258
11	20	23/05/2008 22:59:43	Fernando Carrillo	Remolque	4	Reclaficada	C:\imagenes\Auto20080525-2259
12	40	07/05/2008 11:29:28	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080521-1129
13	41	07/05/2008 11:28:38	Fernando Carrillo	Peasido	2	Reclaficada	C:\imagenes\Auto20080521-1128
14	42	07/05/2008 11:28:48	Fernando Carrillo	Liviano	1	Reclaficada	C:\imagenes\Auto20080521-1128
15	43	07/05/2008 11:29:56	Fernando Carrillo	Remolque	4	Reclaficada	C:\imagenes\Auto20080521-1129
16	44	07/05/2008 11:29:02	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080521-1129
17	45	07/05/2008 11:29:09	Fernando Carrillo	Remolque	4	Reclaficada	C:\imagenes\Auto20080521-1129
18	46	07/05/2008 11:30:01	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080521-1130
19	47	07/05/2008 11:43:35	Fernando Carrillo	Liviano	1	Reclaficada	C:\imagenes\Auto20080521-1143
20	48	07/05/2008 11:43:43	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080521-1143
21	49	07/05/2008 11:43:53	Fernando Carrillo	Peasido	2	Reclaficada	C:\imagenes\Auto20080521-1143
22	50	07/05/2008 11:43:55	Fernando Carrillo	Remolque	4	Reclaficada	C:\imagenes\Auto20080521-1143
23	51	07/05/2008 11:43:58	Fernando Carrillo	Peasido	2	Reclaficada	C:\imagenes\Auto20080521-1143
24	52	07/05/2008 11:53:09	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080521-1153
25	53	07/05/2008 11:53:15	Fernando Carrillo	Liviano	1	Reclaficada	C:\imagenes\Auto20080521-1153
26	54	07/05/2008 11:53:20	Fernando Carrillo	Remolque	4	Reclaficada	C:\imagenes\Auto20080521-1153
27	55	07/05/2008 11:53:25	Fernando Carrillo	Peasido	2	Reclaficada	C:\imagenes\Auto20080521-1153
28	56	07/05/2008 11:53:34	Fernando Carrillo	Pura Pasero	3	Reclaficada	C:\imagenes\Auto20080521-1153
29	57	07/05/2008 11:53:41	Fernando Carrillo	Liviano	1	Reclaficada	C:\imagenes\Auto20080521-1153
30	58	07/05/2008 12:12:53	Fernando Carrillo	Estra Pasero	3	Reclaficada	C:\imagenes\Auto20080521-1212
31	59	07/05/2008 12:12:59	Fernando Carrillo	Peasido	2	Reclaficada	C:\imagenes\Auto20080521-1212
32	60	07/05/2008 12:13:05	Fernando Carrillo	Liviano	1	Reclaficada	C:\imagenes\Auto20080521-1212

N° Tickets: 43
Total Efectivo: 104


```

Private Sub btncon_Click()
    Dim date1 As String, date2 As String
    Dim conexion As New ADODB.Connection, result As New ADODB.Recordset
    Dim cnn As String, query As String, fec As String, aux As Integer, i As Integer
    Dim total As Double
    i = 1
    total = 0
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    conexion.Open cnn

    date1 = Format(fecha1.Value, "yyyy/mm/dd") & " 00:00:00"
    date2 = Format(fecha2.Value, "yyyy/mm/dd") & " 23:59:59"
    query = "Select tic_cod, tic_fec, tic_op, tic_tipo, tic_val, tic_pago, tic_pic from ticket Where tic_fec >=
" & date1 & " and tic_fec <= " & date2 & ""
    result.Open query, conexion
    Do Until (result.EOF)
        flex.TextMatrix(i, 0) = i
        flex.TextMatrix(i, 1) = result.Fields(0)
        flex.TextMatrix(i, 2) = result.Fields(1)
        flex.TextMatrix(i, 3) = result.Fields(2)
        flex.TextMatrix(i, 4) = result.Fields(3)
        flex.TextMatrix(i, 5) = result.Fields(4)
        flex.TextMatrix(i, 6) = result.Fields(5)
        flex.TextMatrix(i, 7) = result.Fields(6)
        total = total + CDbl(result.Fields(4))
        i = i + 1
        flex.Rows = i + 1
        result.MoveNext
    Loop
    result.Close
    conexion.Close
    txtefe.Text = total
    txttic.Text = flex.Rows - 2
    For i = 0 To flex.Cols - 1
        flex.TextMatrix(flex.Rows - 1, i) = ""
    Next
End Sub

```

```

Private Sub Command1_Click()
    Dim conexion As New ADODB.Connection, result As New ADODB.Recordset
    Dim cnn As String, query As String, nom As String
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    conexion.Open cnn
    query = "Select usu_nombre, usu_apellido from usuario where usu_user = " & nomglobal & ""
    result.Open query, conexion
    nom = result.Fields(0) & " " & result.Fields(1)

    Printer.ScaleMode = vbCharacters 'establezco caracteres para controlar la impresion
    Printer.TrackDefault = True 'siempre apunta a la impresora predeter
    'el punto y coma (;) hace que la linea se imprima en el mismo renglon
    'para el logo y datos fijos
    i = 1
    For i = 1 To 3
        Printer.Print " "
    Next
    Printer.Font.Name = "Arial"

```

```

Printer.Font.Size = 16
Printer.FontBold = True
Printer.Print Tab(31); "Estacion de Peaje"
Printer.Font.Size = 14
Printer.Print Tab(37); "Hoja de Reporte"
Printer.Font.Size = 10
Printer.Print " "
Printer.Print " "
Printer.Print Tab(5); "Nombre: ";
Printer.Print Tab(20); nom
Printer.Print Tab(5); "Fecha inicio: ";
Printer.Print Tab(20); fecha1.Value
Printer.Print Tab(5); "Fecha fin: ";
Printer.Print Tab(20); fecha2.Value
Printer.Print " "
Printer.FontBold = False
For i = 0 To flex.Rows - 1
  If i = 1 Then
    Printer.Print " "
  End If
  Printer.Print Tab(2); flex.TextMatrix(i, 0); Enumeracion
  Printer.Print Tab(15); flex.TextMatrix(i, 1); Ticket
  Printer.Print Tab(35); flex.TextMatrix(i, 2); Fecha y Hora
  Printer.Print Tab(60); flex.TextMatrix(i, 3); Operador
  Printer.Print Tab(82); flex.TextMatrix(i, 4); Clasificacion
  Printer.Print Tab(99); flex.TextMatrix(i, 5); Valor
  Printer.Print Tab(107); flex.TextMatrix(i, 6) Tipo de Pago
  Printer.Print Tab(84); flex.TextMatrix(i, 7)
Next
Printer.FontBold = True
Printer.Print " "
Printer.Print " "
Printer.Print Tab(90); "Total Tickets:";
Printer.Print Tab(110); txttic.Text
Printer.Print " "
Printer.Print Tab(90); "Total Efectivo:";
Printer.Print Tab(110); "$ " & txtefe.Text

Printer.EndDoc

```

End Sub

```

Private Sub Command2_Click()
Unload Me
End Sub

```

```

Private Sub Form_Load()
flex.ColWidth(0) = 1000
flex.ColWidth(1) = 1000
flex.ColWidth(2) = 2000
flex.ColWidth(3) = 2000
flex.ColWidth(4) = 2000
flex.ColWidth(5) = 1000
flex.ColWidth(6) = 2000
flex.ColWidth(7) = 4000
flex.ColAlignment(0) = 3
flex.ColAlignment(1) = 0
flex.ColAlignment(2) = 0
flex.ColAlignment(5) = 0

```

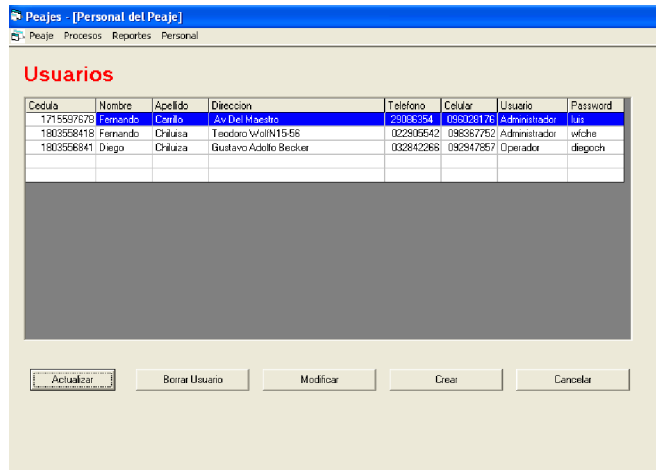
```

flex.TextMatrix(0, 1) = "N° Ticket"
flex.TextMatrix(0, 2) = "Fecha y Hora"
flex.TextMatrix(0, 3) = "Operador"
flex.TextMatrix(0, 4) = "Clasificacion"
flex.TextMatrix(0, 5) = "Valor"
flex.TextMatrix(0, 6) = "Tipo de pago"
flex.TextMatrix(0, 7) = "Imagen"

```

End Sub

PANTALLA 12



```

Dim i As Integer, j As Integer, ced As String
Private Sub Command1_Click()
    Dim query As String
    Dim conexion As New ADODB.Connection
    Dim cnn As String
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    ced = flex1.TextMatrix(flex1.Row, 0)
    If ced <> "" Then
        conexion.Open cnn
        Text1.Text = ced
        query = "Delete from usuario Where usu_cedula = " & ced & ""
        conexion.Execute query
        Command5_Click
    End If
End Sub

Private Sub Command2_Click()
    codusu = flex1.TextMatrix(flex1.Row, 0)
    Form7.Show
End Sub

Private Sub Command3_Click()
    Unload Me
End Sub

Private Sub Command4_Click()
    Form5.Show
End Sub

Private Sub Command5_Click()

```

```
flex1.Rows = 2
For i = 0 To 7
    flex1.TextMatrix(1, i) = ""
Next
Form_Load
End Sub
```

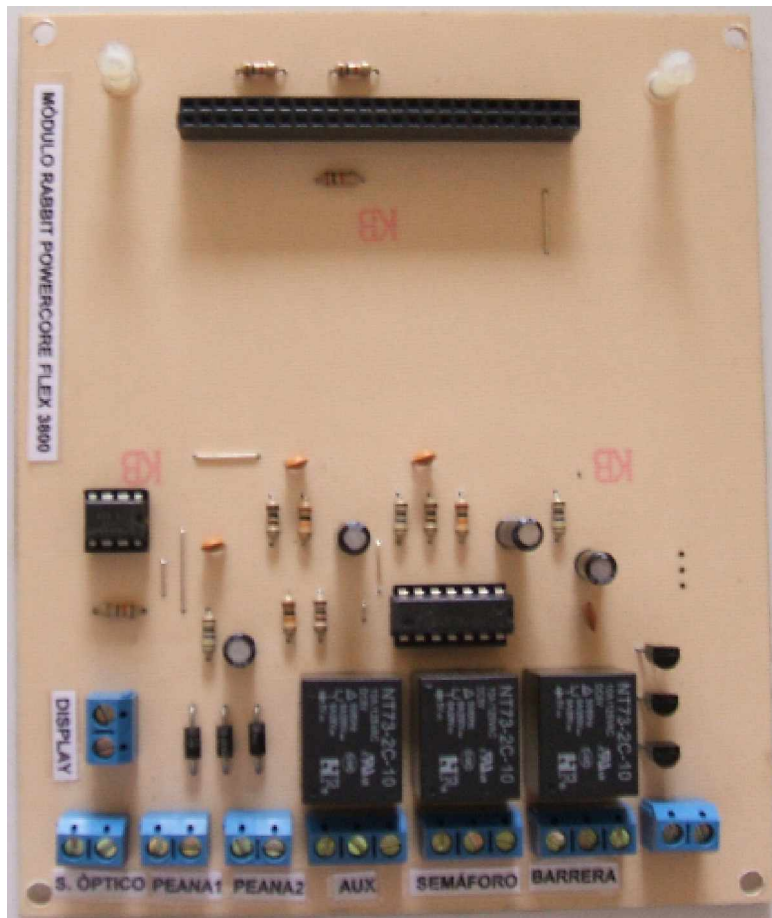
```
Private Sub Form_Load()
    Dim consulta As New ADODB.Recordset
    Dim conexion As New ADODB.Connection
    Dim cnn As String
    cnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties=DATABASE=peaje;DSN=peaje;OPTION=0;PORT=0;SERVER=localhost;UID=root"
    flex1.TextMatrix(0, 0) = "Cedula"
    flex1.ColWidth(0) = 1200
    flex1.TextMatrix(0, 1) = "Nombre"
    flex1.TextMatrix(0, 2) = "Apellido"
    flex1.TextMatrix(0, 3) = "Direccion"
    flex1.ColWidth(3) = 3000
    flex1.TextMatrix(0, 4) = "Telefono"
    flex1.TextMatrix(0, 5) = "Celular"
    flex1.TextMatrix(0, 6) = "Usuario"
    flex1.ColWidth(6) = 1200
    flex1.TextMatrix(0, 7) = "Password"
    conexion.Open cnn
    consulta.Open "Select * from peaje.usuario;", conexion
    i = 1
    flex1.Rows = flex1.Rows + 1
    Do Until consulta.EOF
        flex1.TextMatrix(i, 0) = consulta.Fields(0)
        flex1.TextMatrix(i, 1) = consulta.Fields(1)
        flex1.TextMatrix(i, 2) = consulta.Fields(2)
        flex1.TextMatrix(i, 3) = consulta.Fields(3)
        flex1.TextMatrix(i, 4) = consulta.Fields(4)
        flex1.TextMatrix(i, 5) = consulta.Fields(5)
        flex1.TextMatrix(i, 6) = consulta.Fields(6)
        flex1.TextMatrix(i, 7) = consulta.Fields(7)
        consulta.MoveNext
        i = i + 1
        flex1.Rows = flex1.Rows + 1
    Loop
    conexion.Close

End Sub
```

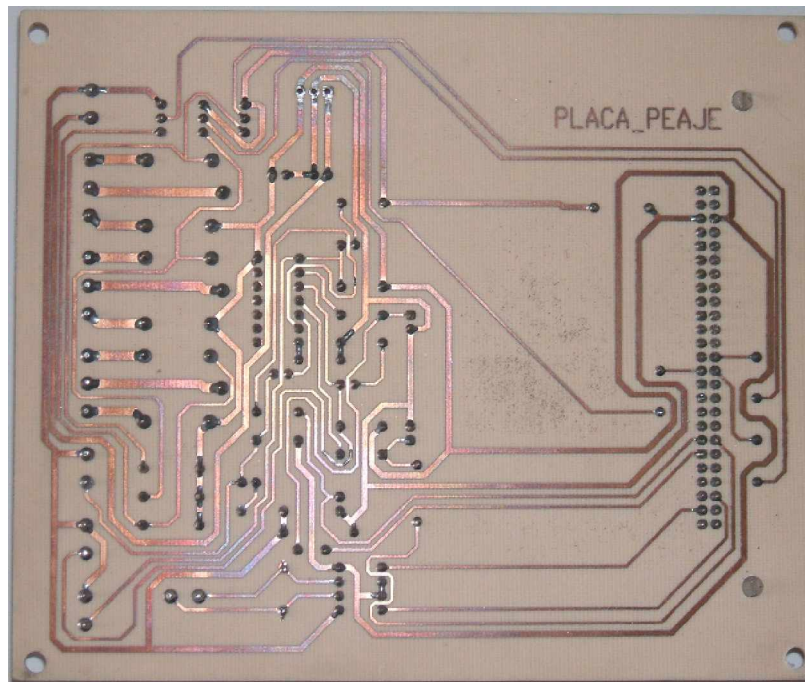
ANEXO F
FOTOGRAFÍAS

1.- Circuito de Tarjeta Electrónica de Control Sistema Peaje

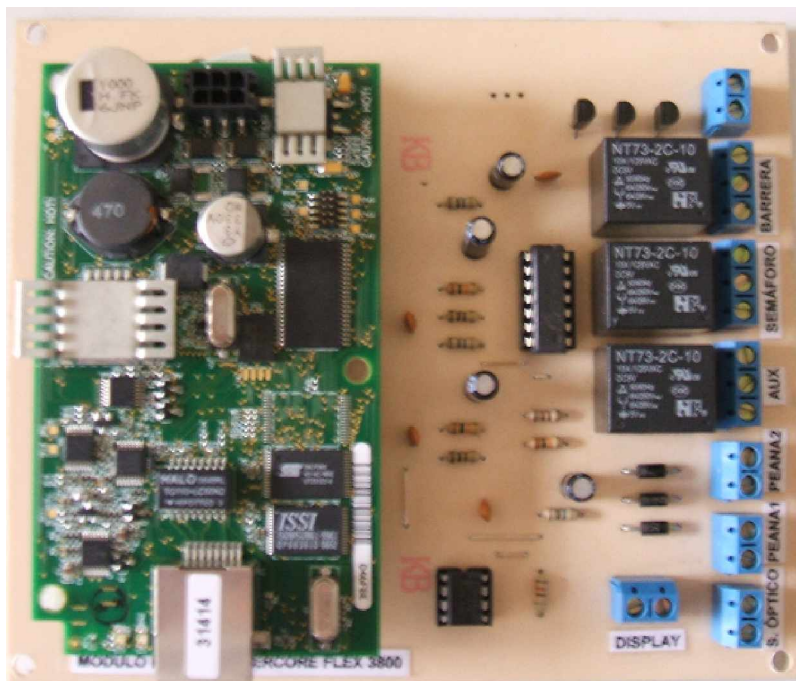
1.1 Anverso



1.2 Reverso

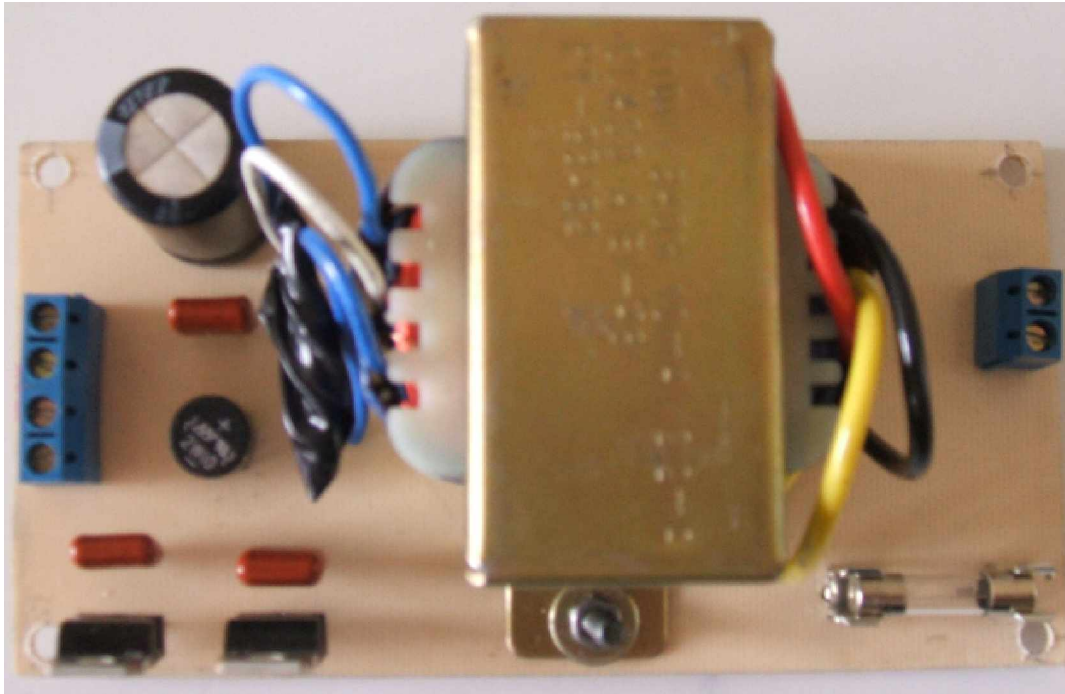


1.3 Tarjeta de Control con Módulo PowerCoreFlex

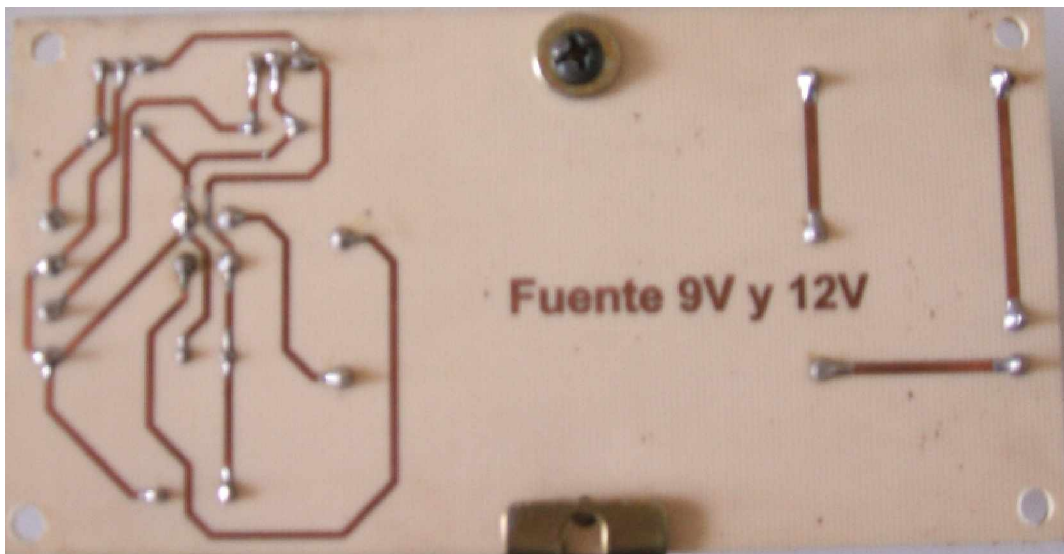


2.- Fuente de 5 y 9V DC

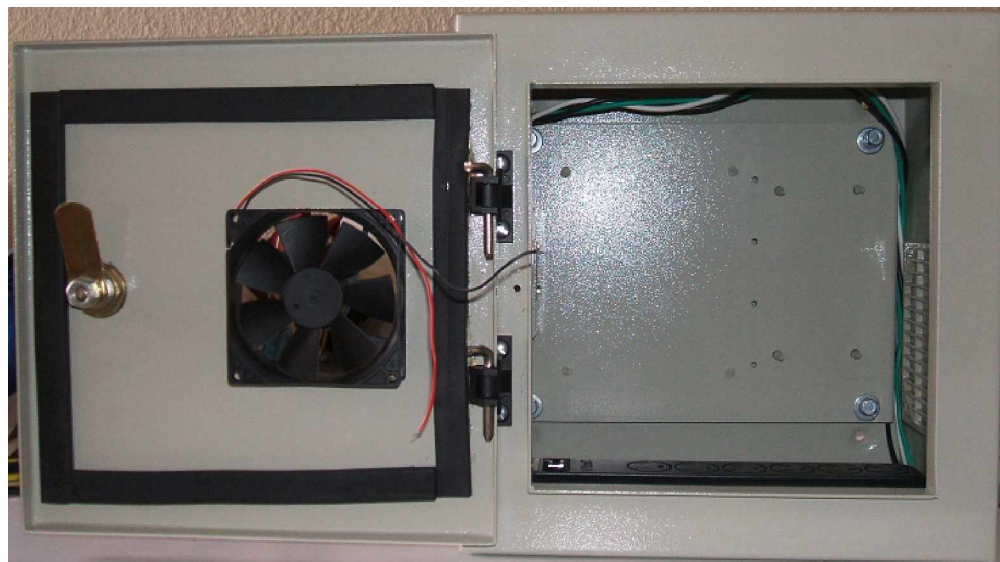
2.1 Anverso



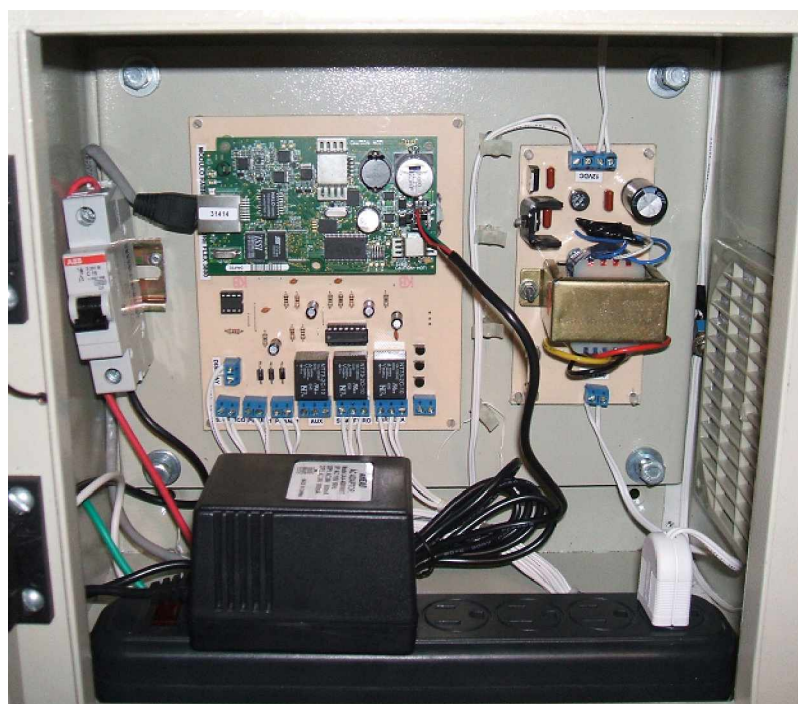
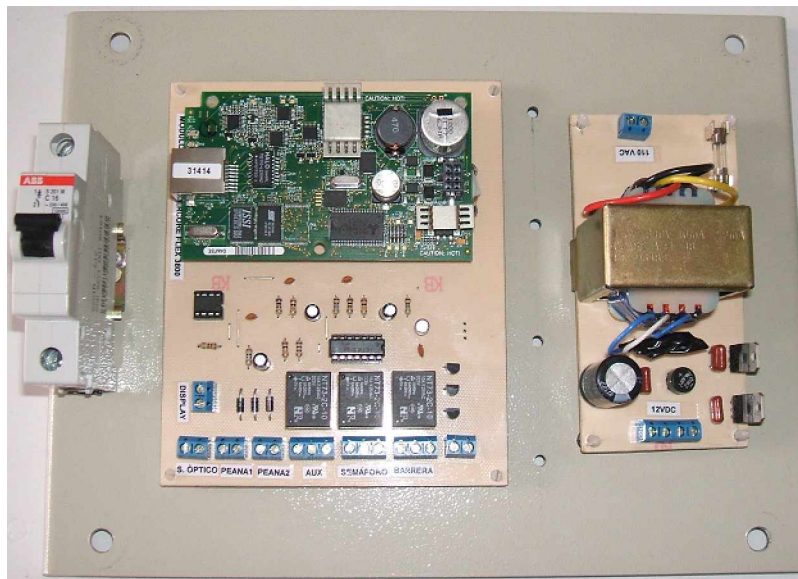
2.2 Reverso



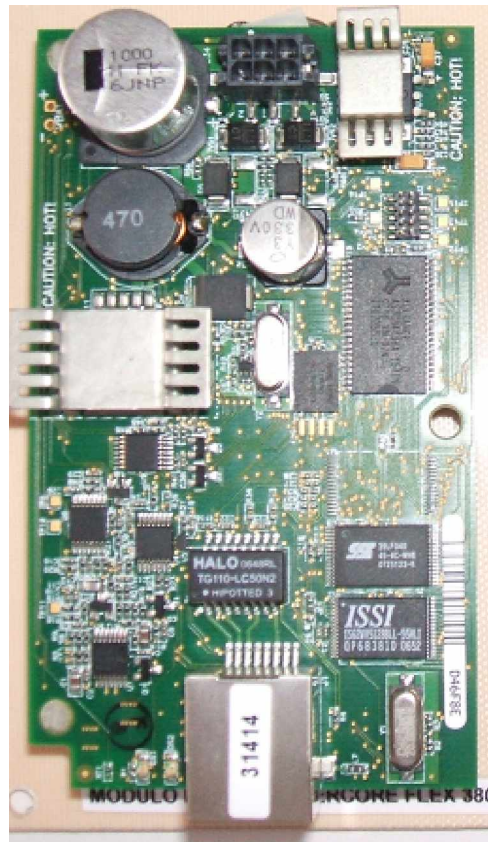
3.- Armario de Control



4.- Tablero de Control



4.- Módulo PowerCoreFlex



ÍNDICE DE FIGURAS

CAPITULO I

Figura 1.1 Fococélulas.....	5
Figura 1.2 Módulo PowerCore Flex 3800 colocada sobre la placa.....	6
Figura 1.3 Funcionamiento del Sensor de Contacto.....	8
Figura 1.4 Banda Detectora de Ejes.....	8
Figura 1.5 Detector de Masa Metálica (loop).....	10
Figura 1.6 Barrera de Acceso.....	11
Figura 1.7 Display del Usuario.....	13
Figura 1.8 Semáforo de Carril.....	13

CAPITULO II

Figura 2.1. Módulo PowerCore Flex 3800.....	20
Figura 2.2. Vista superior del módulo PowerCore.....	23
Figura 2.3. Vista frontal del módulo PowerCore.....	24
Figura 2.4. Vista lateral del módulo PowerCore.....	24
Figura 2.5 Disposición de Subsistemas del Módulo PowerCore Flex.....	24
Figura 2.6 Disposición de Entradas y Salidas Digitales del Módulo PowerCore.....	25
Figura 2.7 Disposición de Puertos y Memoria en el Módulo PowerCore.....	26
Figura 2.8 Modo de comunicación entre capas de red del modelo TCP/IP.....	33
Figura 2.9 Capa Ethernet en el modelo OSI.....	34
Figura 2.10 Trama de Ethernet.....	35
Figura 2.11 Configuración de pines del conector RJ-45 para el puerto Ethernet.....	38
Figura 2.12 Distribución de un cable cruzado.....	39
Figura 2.13 Esquema de Comunicación de la Red Ethernet y Comunicación Serial.....	44
Figura 2.14 Red Ethernet del Peaje.....	45

CAPITULO III

Figura 3.1. Disposición de dos Sensores Paralelos a una distancia X.....	50
Figura 3.2. Disposición de dos Sensores inclinados.....	51
Figura 3.3. Disposición Típica de las Peanas en la Estación de Peaje.....	51
Figura 3.4. Discriminación del Número de Ejes a Través de la Peanas.....	52

Figura 3.5. Diagrama de Disposición Electrónica de los Elementos Sensores	53
Figura 3.6. Parte Interna de la Cabina de Control	55
Figura 3.7. Diagrama de Disposición Electrónica de la Barrera de Acceso	56
Figura 3.8. Diagrama de Disposición Electrónica del Semáforo de Carril.....	58
Figura 3.9. Diagrama de Disposición del Display	59
Figura 3.10. Pantalla Principal de Dynamic C	61
Figura 3.11. Diagrama de Bloques de Funcionamiento del Controlador.....	63
Figura 3.12. Diagrama de Flujo General del Sistema.....	65
Figura 3.13. Diagrama de Flujo del Análisis del Número de Ejes de un Vehículo.....	68
Figura 3.15 Diagrama de flujo de distribución de Base de Datos	74
Figura 3.17 Pantalla de ingreso de usuario.....	78
Figura 3.18 Pantalla de Configuración de Parámetros.....	79
Figura 3.19 Pantalla de Selección del Logotipo de la Empresa Administradora	79
Figura 3.20 Pantalla de Configuración de Facturación y Tarifas	80
Figura 3.21 Pantalla de Creación de Nuevos Usuarios.....	81
Figura 3.22 Pantalla Para el Cambio de Contraseña.....	81
Figura 3.23 Pantalla de Operación Principal del HMI.....	83
Figura 3.24 Pantalla de Liquidación de Turno	84
Figura 3.25 Pantalla de Reportes Generados por Turno	85
Figura 3.26 Pantalla de Reportes Generados por Fecha	86
Figura 3.27 Pantalla Para Almacenamiento de Usuarios	86
Figura 3.28 Diagrama de Flujo General del Funcionamiento de la Interfaz Gráfica	88
Figura 3.29 Diagrama de Flujo para Ingreso de Usuarios	89
Figura 3.30 Diagrama de Flujo de Operación del Sistema	93
Figura 3.31 Diagrama de Flujo de Almacenamiento de Datos	95

ÍNDICE DE TABLAS

CAPITULO II

Tabla 2.1. Configuración Estándar.	22
Tabla 2.2 Formato de Trama Ethernet	36
Tabla 2.3. Configuración de conectores DB9 y DB25 (Estándar RS-232).....	42
Tabla 2.4. Configuración de Comunicación Serial en Dynamic C.....	47

CAPITULO III

Tabla 3.1 Descripción de Puertos en Controlador Rabbit.....	61
Tabla 3.2. Clasificación del Tipo de Vehículos.....	70
Tabla 3.3 Distribución de campos en las tablas de la base de datos.....	76
Tabla 3.4 Trama General de Datos enviados al Controlador	94
Tabla 3.5 Funciones y Parámetros que Recibe el Sistema	94

CAPITULO IV

Tabla 4.1. Detalle de pruebas por períodos	100
--	-----

CAPITULO V

Tabla 5.1. Inversión Económica de Elementos de Carril.....	104
Tabla 5.2. Inversión Económica de Elementos de Cabina.....	104
Tabla 5.3. Inversión Económica de Tablero de Distribución.....	105
Tabla 5.4. Inversión Económica de Elementos de Video	106
Tabla 5.5. Inversión Económica de Cables	106
Tabla 5.6. Inversión Económica de Gastos Adicionales.....	106
Tabla 5.7. Inversión Total del Proyecto	107

GLOSARIO

AFORO Número de vehículos que cruzan una Plaza de Cobro o en uno de sus carriles, en un turno o en un día.

CARRIL Vía de acceso, a la Plaza de Cobro donde se clasifica y registra el peaje a los vehículos por el uso del camino o puente

CLASIFICACIÓN DE VEHÍCULOS Subsistema del Sistema de Peaje que clasifica automáticamente los vehículos por el número y tipo de ejes

DISPLAY Dispositivo electrónico que permite desplegar mensajes variables luminosos, escritos o imágenes a los usuarios de la autopista. Pueden ser vistos a cualquier hora del día y en circunstancias de escasa visibilidad.

EJE Conjunto de ruedas en ambos contados de un vehículo, medidos por una barra transversal ubicada en la parte inferior del mismo.

LIQUIDACIÓN Es el resultado final de la revisión de las operaciones realizadas por el Cajero Receptor en cada turno-carril.

PRELIQUIDACIÓN Es la entrega sistematizada del efectivo recaudado y boletos cancelados que hace un Cajero Receptor, al término de su jornada laboral o su turno de trabajo, del carril que le fue asignado.

SENSOR DE PRESENCIA VEHICULAR. (LOOP) Consiste de un cable que forma lazos y que se encuentra ubicado por debajo de la superficie de rodamiento (pavimento). Éste cable se encuentra interconectado a un controlador situado al lado de la autopista, el cual está conformado básicamente por un amplificador de señales una fuente de poder y otros elementos

TURNO El turno es la jornada de 8 horas en que se divide un día de trabajo en las Plazas de Cobro

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) Acceso Múltiple con Escucha de Portadora y Detección de Colisiones, es una técnica usada en redes Ethernet para mejorar sus prestaciones.

DYNAMIC C Compilador, editor, cargador y depurador específico de los microprocesadores Rabbit.

ETHERNET Es el nombre de una tecnología de redes de computadoras de área local (LANs) basada en tramas de datos. Ethernet define las características de cableado y señalización de nivel físico y los formatos de trama del nivel de enlace de datos del modelo OSI.

FCS (Frame Check Sequence) Secuencia de Verificación de Trama

LLC (Logical Link Control) Control de Enlace Lógico, está en la parte superior de la subcapa de enlace dentro del modelo OSI

MAC (Media Access Control) Dirección de control de acceso al medio. Es un identificador hexadecimal de 48 bits que corresponde de forma única a una tarjeta o interfaz de red.

OSI (OSI, Open System Interconnection) modelo de referencia de Interconexión de Sistemas Abiertos. Sirve de marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

PLC (Programmable Logic Controller) Controlador Lógico Programable.

RED Es un conjunto de nodos y enlaces que proporcionan conexiones entre dos o más puntos definidos para facilitar la telecomunicación entre ellos.

RJ-45 interfaz física comúnmente usada para conectar redes de cableado estructurado, (categorías 4, 5, 5e y 6). RJ es un acrónimo inglés de Registered Jack que a su vez es parte del Código Federal de Regulaciones de Estados Unidos.

RS-232 Estándar para la conexión serial de señales de datos binarias entre un DTE (Equipo terminal de datos) y un DCE (Equipo de terminación del circuito de datos)

RS-422 Estándar aprobado por la EIA para conectar dispositivos en forma serial.

SOCKET Un socket queda definido por una dirección IP, un protocolo y un número de puerto

SOF (Start Of Frame) Conjunto de bits, indica el Inicio de Trama.

TCP/IP Protocolo de Control de Transmisión / Protocolo Internet de capa de Transporte que asegura la entrega satisfactoria de extremo a extremo de paquetes de datos sin error, como lo define el IETF.

USB (Universal Serial Bus) Es un puerto que sirve para conectar periféricos a un computador.

UTP Definido en el estándar EIA/TIA 568, soporta velocidades de transmisión de 10 Mbps en Ethernet 10Base-T, y 4 Mbps. Este cable tiene cuatro pares y su impedancia es de 100 Ω .

ÍNDICE DE DATA SHEETS

Tarjeta Rabbit con módulo PowerCoreFlex 3800	1
Diodo 1N4001	2
Transistor 2N3904	3
Inversor DM74LS14	4
Regulador LM78XX	5
MAX 485.....	6
Relé NT73-2C-10 (10/125VAC – 5VDC)	7

DATA SHEETS

1

**TARJETA RABBIT CON MÓDULO
POWERCORE FLEX**

2

Diodo 1N4001

3

Transistor 2N3904

4

Inversor DM74LS14

5

Regulador LM78XX

6

MAX 485

7

Relé NT73-2C-10

FECHA DE ENTREGA:

**COORDINADOR DE LA CARRERA DE
INGENIERÍA ELECTRÓNICA**

**WILSON FERNANDO
CHILUISA ESPÍN**