

ESCUELA POLITÉCNICA DEL EJÉRCITO

FACULTAD DE INGENIERÍA ELECTRÓNICA

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO EN
INGENIERÍA ELECTRÓNICA –ESPECIALIDAD
TELECOMUNICACIONES**

**“DISEÑO DE SISTEMAS PARA CONTROL ELECTRÓNICO DE
PEAJES”**

FRANCISCO JAVIER HOYOS MEJÍA

SANGOLQUÍ – MAYO 2005

CERTIFICACIÓN

Certificamos que el presente Proyecto de Grado fue realizado en su totalidad por el señor Francisco Javier Hoyos Mejía, bajo nuestra dirección.

Ing. MSc. Fabián Sáenz
DIRECTOR

Ing. Víctor Proaño
CODIRECTOR

AGRADECIMIENTO

A mi Dios, porque ha iluminado cada uno de mis pasos y siempre me ha acompañado fortaleciendo mi espíritu en los momentos difíciles de mi vida.

A mis padres, por su amor, comprensión, dedicación y cuidado, esto es fruto de su esfuerzo y trabajo.

A mi hermanos, quienes siempre me han acompañado con sus juegos y sonrisas que me han servido para alimentar de momentos alegres mi vida.

A la Escuela Politécnica del Ejército, en cuyas aulas adquirí los conocimientos que hoy me permiten ser un profesional.

A mis profesores, que con capacidad y responsabilidad supieron transmitir sus conocimientos. En especial a mi Director y Codirector de tesis, Ing. Fabián Sáenz e Ing. Víctor Proaño respectivamente.

A mis amigos y compañeros, que supieron brindarme su amistad, compartiendo y llenando de momentos gratos todos los instantes de este caminar.

Finalmente a todas las personas que depositaron su confianza en mi, de todo corazón
GRACIAS.

Paco

DEDICATORIA

Dedico este trabajo a Dios, ese ser supremo quien con ese infinito amor me ha dado la oportunidad de aprender y llenar mi vida experiencias.

Dedico todo este trabajo, a mis padres los seres a los que debo mi vida por ser fuente de amor, cariño y respaldo.

Dedico también a todas las personas que me han acompañado y colaborado con el desarrollo y culminación de este proyecto, porque con sus conocimientos técnicos y no tan técnicos siempre me alentaron con su voz de amor y afecto, para todos ellos este trabajo.

PRÓLOGO

El presente proyecto de grado titulado “**DISEÑO DE SISTEMAS PARA CONTROL ELECTRÓNICO DE PEAJES**” se propone con la intención de mostrar el funcionamiento de un prototipo que esta dirigido a controlar electrónicamente la recaudación de peajes. El cual tiene como objetivo principal mejorar el servicio de recaudación, haciéndolo totalmente transparente para el usuario, consiguiendo así que las vías concesionadas, y principalmente los puntos de recaudación de peaje dejen de ser sitios de estrangulamiento vehicular.

Este prototipo esta compuesto de tres subsistemas :

- ❑ Transmisión de datos
- ❑ Adquisición de datos, y
- ❑ Centro de control y administración.

El subsistema de transmisión de datos esta compuesto por un transmisor que está colocado dentro del vehículo, el mismo que envía un código vía RF, que es detectado a su vez en el receptor de RF, el cual trabaja conjuntamente con el subsistema de adquisición de datos, quien ordena y controla dos acciones: alarma o descuento, la primera es ejecutada si el usuario no se a registrado previamente como cliente y la segunda cuando uno de los clientes del sistema ha hecho uso del mismo descontando al efecto de una cuenta bancaria el valor del servicio mensualmente.

De esta manera se optimiza la vía, algo que le es de interés al estado puesto que en muchos casos, los gobierno seccionales no cuenta con los recursos económicos para expandir vías, sin embargo, de esta manera se resolvería el problema a la brevedad posible, consiguiendo también, resolver otros problemas que tienen que ver con la salud de usuarios y operarios del sistema, dado que se disminuiría el grado de

contaminación tanto de gases como de ruido algo que mejorará el bienestar y desempeño de las personas.

Se recomienda utilizar este prototipo como modelo de funcionamiento, mas no de implementación.

ÍNDICE

Contenido	Pág
PRÓLOGO	
CAPITULO 1.....	1
INTRODUCCIÓN.....	1
1.1 Antecedentes.....	1
1.2 Servicios de Pago Electrónico de Peajes.....	2
CAPITULO 2.....	5
ANÁLISIS DE TECNOLOGÍAS.....	5
2.1 SISTEMAS INTELIGENTES DE TRANSPORTE ITS.....	5
2.1.1 Definición.....	5
2.1.2 Objetivos.....	6
2.1.3 Tecnologías Involucradas.....	6
2.1.4 Arquitectura del Sistema.....	7
2.2 COMPONENTES DEL SISTEMA DE PEAJE.....	8
2.2.1 Transmisión de los datos.....	9
2.2.2 Adquisición de datos.....	10
2.2.3 Centro de control.....	10
2.3 TRANSMISIÓN, RECEPCIÓN Y COMUNICACIÓN MEDIANTE PICs.....	11
2.3.1 rfPIC12f675 Transmisor UHF ASK/FSK.....	11
2.3.1.1 Información General.....	11
2.3.1.2 Transmisor UHF ASK/FSK.....	12
2.3.1.3 Modulación ASK.....	13
2.3.2 rfRXD0420 Receptor UHF ASK / FSK / F M.....	14
2.3.2.1 Características Generales.....	14
2.3.2.2 Visión Global del rfRXD0420.....	16
2.3.2.3 Modulación por desplazamiento en Amplitud (ASK).....	18
2.3.2.4 Detector de filtrado RSSI.....	18
2.3.2.5 Período de la Señal.....	19
2.3.3 Descripción del PIC16F870.....	19

2.3.3.1	Características Generales	19
2.3.4	Descripción del PIC16F627	20
2.3.4.1	Características Generales	20
2.3.5	Interfase Serial RS-232	21
2.3.5.1	Circuito integrado MAX232	23
2.4	SOFTWARE DEL SISTEMA	24
2.4.1	Interfaz Hombre Máquina HMI	24
2.4.2	Bases de Datos	24
2.4.2.1	Características de una Base de Datos	25
2.4.2.2	Organizaciones de Datos	25
2.4.2.3	Bases de Datos Relacionales	26
2.4.2.4	Conceptos Básicos del Modelo Relacional	26
2.4.2.5	Relaciones	28
CAPITULO 3	31
DISEÑO DEL SISTEMA	31
3.1	GENERALIDADES	31
3.2	COMPONENTES DEL SISTEMA	32
3.2.1	Transmisión de Datos	32
3.2.1.1	Diseño de Hardware del Subsistema de Transmisión de Datos	32
3.2.1.2	Diseño del Software del Subsistema de Transmisión de Datos	39
3.2.2	Adquisición de Datos	47
3.2.2.1	Diseño del Hardware del Subsistema de Adquisición de Datos	47
3.2.2.2	Diseño del Software del subsistema de Adquisición de datos	52
3.2.3	Centro de Control	57
3.2.3.1	Diseño del software de monitoreo (MONITOR)	58
3.2.3.2	Diseño del software de administración (SIADPE)	60
CAPITULO 4	69
INTEGRACIÓN Y PRUEBAS	69
4.1	INTEGRACIÓN	69
4.1.1	Suministro de Energía	69
4.1.2	Sensores	70
4.1.3	Transmisor Receptor de RF	71
4.1.4	Tarjeta de Adquisición de Datos	71
4.1.5	Alarma	72

4.1.6 PC.....	72
4.2 PRUEBAS.....	73
4.2.1 Subsistema de Transmisión de Datos.....	73
4.2.2 Subsistema de Adquisición de Datos	75
4.2.3 Subsistema Centro de Control.....	75
CAPÍTULO 5.....	77
ANÁLISIS ECONÓMICO	77
5.1 LISTADO DE MATERIALES	77
5.1.1 Materiales Eléctricos y Electrónicos	77
5.1.2 Materiales para Maqueta.....	78
5.1.3 Software.....	79
5.2 ANÁLISIS DE COSTOS	79
5.2.1 Costos de Materiales Eléctricos y Electrónicos.....	79
5.2.2 Costos de Materiales para Maqueta	80
5.2.3 Costos de Diseño Software	80
CAPÍTULO 6.....	81
CONCLUSIONES Y RECOMENDACIONES.....	81
6.1 CONCLUSIONES	81
6.2 RECOMENDACIONES	83

REFERENCIAS BIBLIOGRÁFICAS

ANEXO 1

- A. DIAGRAMA ELECTRÓNICO Y PCB DE PLACA DE SUMINISTRO DE ENERGÍA

ANEXO 2

- B. DIAGRAMA ELECTRÓNICO Y PCB DE PLACA DE ADQUISICIÓN DE DATOS

ANEXO 3

- C. CÓDIGOS FUENTE DEL SOFTWARE DE TRANSMISIÓN Y ADQUISICIÓN DE DATOS

ANEXO 4

D. CÓDIGOS FUENTE DEL SOFTWARE DE ADMINISTRACIÓN Y MONITOREO

ANEXO 5

E. COMUNICACIÓN SERIAL EN VISUAL BASIC 6.0

ANEXO 6

F. MANUAL DEL USUARIO PARA EL PROGRAMA DE ADMINISTRACIÓN (SIADPE)

ANEXO 7

G. MANUAL DEL USUARIO PARA EL PROGRAMA DE MONITOREO (MONITOR)

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO

CAPÍTULO 1

INTRODUCCIÓN

1.1 Antecedentes

Aunque son muchos los aspectos de la sociedad que han mejorado gracias a la introducción de tecnologías avanzadas, hemos seguido contentándonos con un sistema de transporte cuya tecnología de control primaria es la señal de tráfico, prácticamente inmutable desde su invención en 1923.

Cuando se aborda la necesidad de mejorar el sistema de transporte, es necesario plantear algunas cuestiones: los puntos de estrangulamiento siguen constituyendo un problema importante en las principales rutas nacionales, en tanto que la congestión urbana e interurbana exige una adecuación inmediata. Por otra parte, resulta evidente que la emisión de gases nocivos constituye un peligro real para el futuro de la humanidad; en las zonas urbanas siguen incrementándose los problemas respiratorios, mientras que para el mundo en que vivimos, desplazarse consiste en una necesidad cotidiana.

La carretera es la vía de transporte que más costos le genera al Estado desde el punto de vista de la seguridad; puesto que causa aproximadamente 2000 víctimas mortales al año en el Ecuador.

Tradicionalmente en nuestro país ha existido despreocupación por parte de las autoridades en la gestión de tránsito, específicamente en la vialidad urbana de las grandes y pequeñas ciudades siendo ésta una forma de incrementar los problemas de congestión vehicular. Principalmente, dicha desatención se ha fijado en la mala programación de semáforos y en el inadecuado manejo de peajes, los cuales generan un caos vehicular que va complicando cada vez más el desplazamiento y la salud de las personas.

Es por esto que se insiste en el estudio de los procesos que, analizando simultáneamente las características de los medios y recursos puestos en juego para un fin determinado, reflejen su productibilidad o rentabilidad, al mismo tiempo que permitan disponer de una información rápida y oportuna para la toma de decisiones.

En el campo de la gestión, los sistemas inteligentes han tenido su principal aplicación y desarrollo, pues en la actualidad se habla comúnmente de edificios inteligentes, vehículos inteligentes, tarjetas inteligentes, etc. Los sistemas inteligentes pueden definirse como aquellos que, optimizando los procesos y recursos para la obtención de un producto deseado, facilitan una permanente información, en tiempo real, para su seguimiento, evaluación y control, permitiendo introducir variaciones durante el curso de su gestión.

Estos son una combinación de la informática y las telecomunicaciones, o sea la telemática; con ello se multiplica la capacidad humana en las operaciones y se evitan, en gran medida, los fallos de su intervención, se comprende así, el fundamental peso que en estos procesos tiene el desarrollo y aplicación de las innovaciones tecnológicas.

1.2 Servicios de Pago Electrónico de Peajes

Se trata de transacciones electrónicas instantáneas de tipo económico, que minimizan la detención de los vehículos, e incluso suprimen ésta para efectuar el pago, evitando así importantes pérdidas de tiempo en los peajes de carretera.

Uno de los sistemas más extendido es el de las tarjetas de crédito, que operan al insertarlas en un lector al abordar el peaje.

Existe la tarjeta inteligente de proximidad, que incorpora un microchip en lugar de la banda magnética, bastando deslizarla sobre un lector para accionar las barreras, es un sistema muy utilizado en centros públicos como, estacionamientos, parques, museos, etc.

El sistema más avanzado es el de peaje automático canalizado, cuya base principal es un elemento sensor denominado *tag*, situado en el parabrisas del vehículo como receptor. El proceso se inicia a 20 metros del peaje a través de un detector colocado al efecto, un segundo sensor, ubicado en el pavimento, confirma la presencia del vehículo, desencadenando en la antena del puesto de pago una señal de radio hacia el receptor *tag*, el que responde proporcionando los datos del vehículo y conductor propietario, cargando en su cuenta el importe correspondiente. Todo este proceso se realiza sin detención del vehículo y a velocidades que pueden ser elevadas.

Finalmente se pretende desarrollar un sistema muy avanzado de peaje, cuyo valor no es fijo sino variable, en función del uso real de la carretera. La evolución de estas tasas se encarece o abarata, en tramos congestionados o libres, tratando de actuar como disuasores del uso del vehículo privado, para encaminar el tráfico hacia el transporte público colectivo.

Estos cobros por uso de infraestructuras requieren un complicado sistema de peaje continuo de pago cuando el conductor entra en la zona afectada, con un equipo muy sofisticado y caro que, por el momento, se encuentra en el campo de la experimentación.

Después de analizar varios factores e indicadores que ubican a estos problemas como graves para el desarrollo de la vida humana dentro de las zonas urbanas y suburbanas de nuestro país, pongo a consideración el estudio y la elaboración de este prototipo de control electrónico para peajes, que persigue el objetivo de dar solución a los grandes problemas de congestión en los puntos de recaudación de peajes, tomando en cuenta una serie de

estudios y experiencias en países del primer mundo, para de esta manera poder proyectar las posibles soluciones a estos problemas, que son producto de una comercialización desmedida y una previsión obsoleta y desconsiderada de las autoridades competentes.

En adelante se propone un estudio y desarrollo de un prototipo para control electrónico de peajes, en el cual se darán detalles de cada una de sus partes, referidos al gran conglomerado de los Sistemas Inteligentes de Transportes ITS.

CAPÍTULO 2

ANÁLISIS DE TECNOLOGÍAS

2.1 SISTEMAS INTELIGENTES DE TRANSPORTE ITS

2.1.1 Definición

Los Sistemas Inteligentes de Transporte son sistemas que aplican tecnología de información y control para apoyar sus operaciones. Con esta amplia definición, ITS es un abanico que abarca una extensa gama de sistemas de transporte, algunos de los cuales han sido implementados hace años y otros recién se están aplicando o están aún en investigación y desarrollo para futuras aplicaciones.

De este modo, una definición más comúnmente aceptada de ITS es *"la aplicación de tecnologías computacionales, de control y de comunicaciones para ayudar a los conductores y operadores a tomar decisiones inteligentes mientras conducen vehículos inteligentes o controlan el tránsito en caminos o vías férreas inteligentes"*. Aunque no es muy rigurosa, esta definición deja en claro la noción de que ITS mantiene al operador y conductor humano al centro. Por lo tanto, ITS no es de ningún modo sinónimo de automatización, aún cuando la conducción automática es una opción para el futuro bajo el mundo ITS.

2.1.2 Objetivos

Uno de los principales objetivos del ITS es la optimización del sistema de transporte, facilitando a los operadores de transportes públicos y de carreteras información más precisa y actualizada sobre la situación de sus redes, con lo que les resulta más fácil informar a los viajeros y ofrecer soluciones alternativas.

Ejemplos concretos:

- Sistemas de control de las señales de tráfico en las ciudades que se ajustan automáticamente para optimizar el flujo del tráfico y conceder preferencia al transporte público.
- Sistemas de gestión de autopistas que facilitan información a los automovilistas, recomendándoles por ejemplo, la velocidad óptima, contribuyendo así a incrementar la capacidad y el flujo y reducir el número de accidentes debidos a la congestión.
- Cobro electrónico de peajes, para hacer posibles transacciones automáticas, cómodas y fiables y mejorar el flujo del tráfico en las zonas de peaje.
- Sistemas de pago electrónico que permitan a una persona abonar con una sola tarjeta inteligente los peajes, las tarifas del aparcamiento y el autobús.

2.1.3 Tecnologías Involucradas

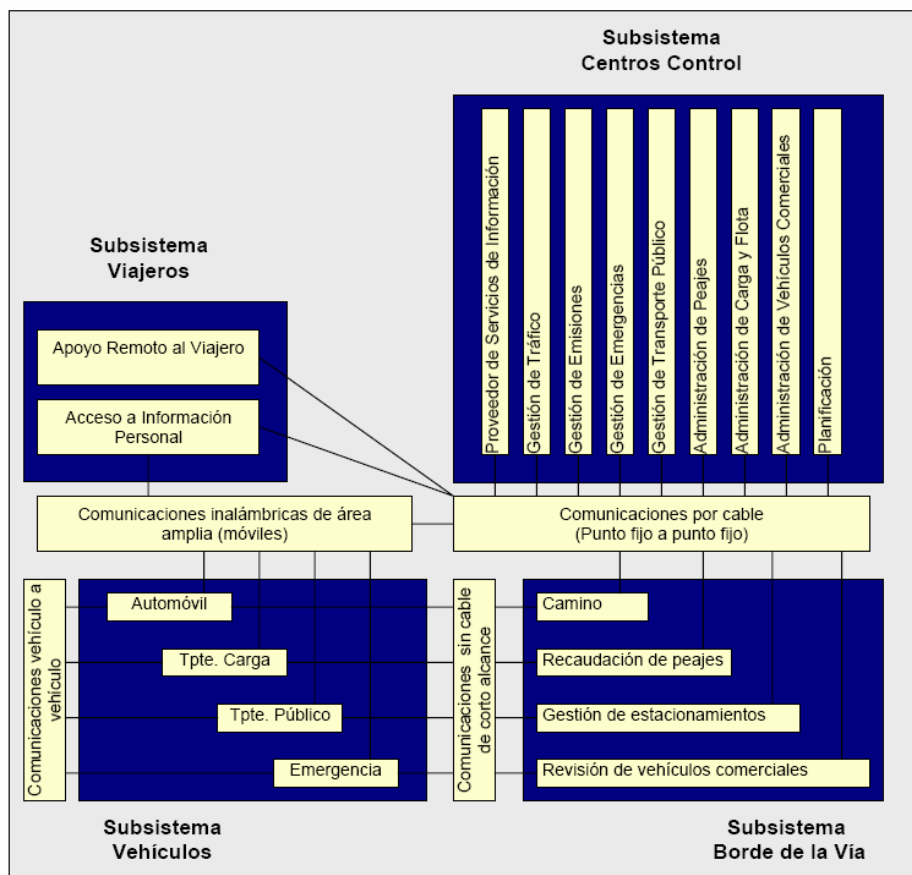
La adaptación de estas tecnologías al transporte requiere el conocimiento de múltiples campos de la ingeniería: civil, electrónica, mecánica, industrial y sus disciplinas relacionadas por ejemplo ingeniería de tránsito, dinámica de vehículos, ciencias de la computación, investigación de operaciones y recursos humanos. La conjunción de estas tecnologías para desarrollar funciones ITS está basada en la ingeniería de sistemas.

Además, dichas tecnologías se construyen una sobre la otra y producen sinergias. Por ejemplo, la misma información para recaudación electrónica de peajes puede ser usada para gestión de tránsito

2.1.4 Arquitectura del Sistema

La arquitectura del sistema es un marco para asegurar la interoperabilidad e integración sinérgica entre todas las funciones del ITS, independiente de las tecnologías específicas a ser desarrolladas. La interoperabilidad puede ser ejemplificada por la capacidad de un vehículo que usa un solo juego de antena y unidad dentro del vehículo para recibir a un conjunto de servicios sin importar donde el vehículo está operando. En cuanto a la integración sinérgica, en general hay cuatro fuentes en ITS:

La Figura 2.1 muestra la arquitectura física nacional de ITS desarrollada para EE.UU.



Fuente: Chen et al, 1999

Figura. 2.1. Diagrama de la arquitectura física de ITS en EE.UU.

- ❑ El uso común de tecnologías del ITS, como aquellas para la recaudación de peajes y de información del automóvil mencionadas previamente.
- ❑ Bases de datos compartidas, como el uso común de datos entre tráfico y los centros de gestión de tránsito.
- ❑ El intercambio y coordinación entre unidades de control y auxilio vial, como son la patrulla de la carretera y los servicios de emergencia.

2.2 COMPONENTES DEL SISTEMA DE PEAJE

Cualquier sistema que usa tecnologías de información y control puede ser descompuesto en las subfunciones de recolección, proceso y difusión de la información relevante y de decisión, es así que analizaremos este diseño al hacer la descomposición del sistema total, en sus subfunciones.

Estas subfunciones ayudan a mantener un control al tráfico de vehículos y personas involucradas, además pueden usar una amplia variedad de tecnologías. Las cuales analizaremos y estudiaremos para la elaboración de un posterior diseño. Dichas subfunciones pueden ser resumidas en:

- ❑ Transmisión de los datos
- ❑ Adquisición de los datos
- ❑ Centro de control



Figura. 2.2. Descripción gráfica de los componentes del sistema

2.2.1 Transmisión de los datos

La transmisión de los datos es una subfunción clave en los servicios. Existen dos categorías de medios técnicos para transmitir información: los terminales fijos, y los terminales móviles. Normalmente, se clasifican también como transmisión alámbrica e inalámbrica.

La Transmisión de Datos, además es un sistema, formado de varios bloques como son: medio de transmisión, transmisor y receptor.

A continuación, se enumeran los principales medios de transmisión: Par de hilos, par trenzado (UTP, STP), cable coaxial, fibra óptica (monomodo, multimodo), satélite, radioenlace, **transmisión inalámbrica**.

Enlaces inalámbricos

Servicio que consiste en ofrecer, al cliente acceso, mediante un enlace inalámbrico por medio de antenas, que permiten utilizar un ancho de banda. Todos estos enlaces trabajan por medio de radio frecuencia .

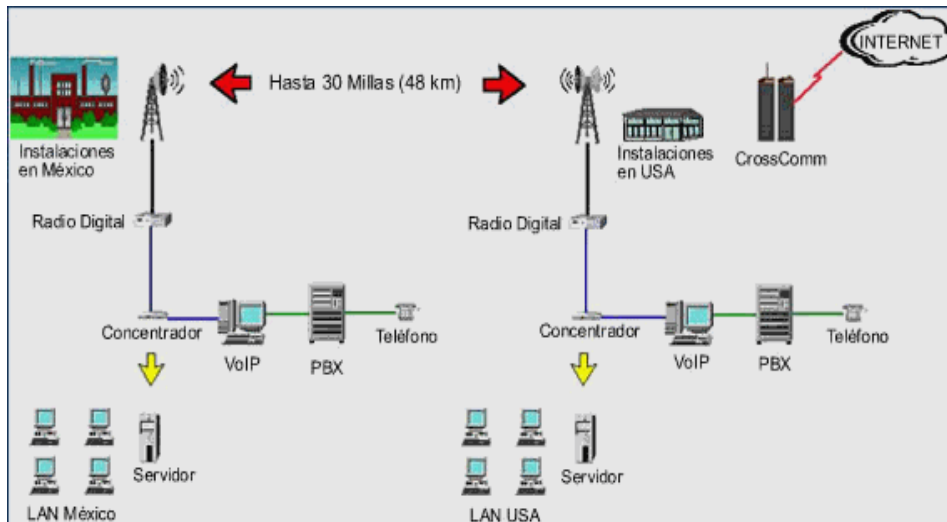


Figura. 2.3. Ejemplo de Enlace Inalámbricos

2.2.2 Adquisición de datos

En esta clasificación se encuentran todas las funciones que tienen que ver con la recolección de la información. El principal requisito, para esta subfunción, es que la información recolectada debe ser, precisa y disponible en el momento en que se necesita.

Los principales dispositivos utilizados ampliamente en esta subfunción son: **PICs (Microcontroladores)**, PLCs (Controladores Lógicos Programables), DAQs (Tarjetas de Adquisición de Datos)

2.2.3 Centro de control

Dado que, información de distinta naturaleza se está recolectando simultáneamente en diversos lugares, se necesita de un centro de control que consolide y procese todos los datos. En el centro de control es donde los operadores reciben los datos y apoyados por herramientas de software y hardware, toman decisiones sobre el sistema.

Los principales elementos que constituyen un centro de control son: **Interfaz Hombre Máquina (HMI), bases de datos.**

2.3 SISTEMA DE TRANSMISIÓN, RECEPCIÓN Y COMUNICACIÓN MEDIANTE PICs

2.3.1 rfPIC12f675 Transmisor UHF ASK/FSK

2.3.1.1 Información General

Los elementos microcontroladores rfPIC integran la potencia de los PICmicro de Microchip con las capacidades de comunicación wireless, para aplicaciones de RF con baja potencia. Los elementos ofrecen un pequeño encapsulado y componentes externos de menor tamaño, a fin de aprovechar espacio dentro de una aplicación.

El rfPIC12f675 es un transmisor ASK, de bajo costo y alto desempeño de radio UHF que se usa como el PIC rf12f675K para 315 MHz y el rf12f675F para 433.92 MHz.

Características técnicas:

- ❑ Velocidad de operación: oscilador interno de 4MHz, calibrado a $\pm 1\%$
- ❑ Capacidad de interrupción
- ❑ 8 niveles de pila
- ❑ modo de direccionamiento directo e indirecto

Características Generales:

- ❑ Memoria:
 - Memoria FLASH de programa 1024 x 8 bytes
 - Memoria EEPROM de datos 128 x 8 bytes
 - Memoria SRAM de datos 64 x 8 bytes
 - 100 000 escrituras en FLASH
 - 1 000 000 de escrituras en EEPROM
 - Retención de Datos FLASH / EEPROM
- ❑ Protección de código programable

- ❑ 6 pines de entrada salida, weak pull – ups y detección de cambio en el pin
- ❑ Comparador análogo de 16 niveles de referencia interna
- ❑ Conversor análogo digital 10 bits, 4 canales
- ❑ TIMER0 con 8 bits timer/contador con 8 bits de prescaler
- ❑ TIMER1 timer /contador con 16 bits con tres bits de prescaler
- ❑ 5 μ s de reacción del sleep con 3V de alimentación

Características Secundarias.

- ❑ Consumo bajo de energía
- ❑ 14 mA transmitiendo +6dBm a 434 MHz
- ❑ 4 mA transmitiendo -15dBm a 434 MHz
- ❑ 500 μ A, 4.0 MHz. INTOSC (Oscilador interno)
- ❑ Rango de voltaje para operación 2.0 a 5.5V
- ❑ Rango de temperatura industrial

2.3.1.2 Transmisor UHF ASK/FSK

- ❑ Tasa de transmisión de datos ASK 0-40 Kbps
- ❑ Tasa de transmisión de datos FSK 0-40 Kbps
- ❑ Potencia de salida +10 dBm a -12 dBm en 4 pasos
- ❑ Consumo de potencia ajustable para la transmisión

El rPIC12F675 viene como paquete SSOP de pines, como se muestra en la figura 2.4

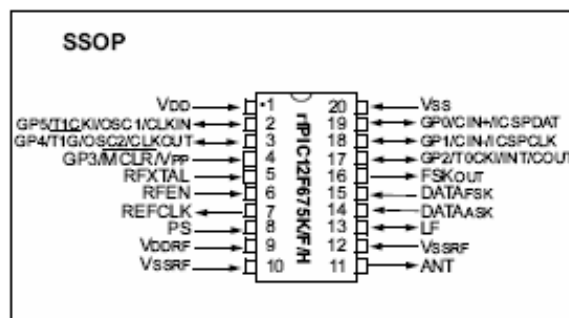


Figura. 2.4. Diagrama de pines del rPIC12F675

Este transmisor consta de un cristal oscilador, Lazo cerrado de fase (PLL), un amplificador de potencia (PA) con salida de colector abierto y modo de control lógico.

Hay 3 variaciones de este dispositivo para optimizar su funcionamiento para las bandas de frecuencia más comúnmente usadas. (véase la Tabla 2.1).

Dispositivo	Frecuencia	Modulación
rfPIC12F675K	290 - 350 MHz	ASK/FSK
rfPIC12F675F	390 - 450 MHz	ASK/FSK
rfPIC12F675H	850 - 930 MHz	ASK/FSK

Tabla. 2. 1. Banda de frecuencias de los rfPICs

El oscilador Colpitts genera la frecuencia de referencia establecida por el cristal adjunto. El oscilador controlado por voltaje (VCO) convierte el voltaje en el pin LF a frecuencia. Esta frecuencia es dividida para 32 y comparada con la referencia del cristal. La señal de salida del VCO es también amplificada por el PA.

Las dos señales de control del microcontrolador son conectadas externamente para una máxima flexibilidad de diseño. El rfPIC12F675 es capaz de transmitir en las modulaciones ASK o FSK.

2.3.1.3 Modulación ASK

En la modulación ASK, el dato es transmitido variando la potencia de salida. El pin $DATA_{ASK}$ habilita el PA, haciendo de esa forma que la señal de salida RF se active y desactive. Un simple receptor usando un filtro sintonizador y un diodo detector de picos puede capturar un dato. Un receptor más avanzado tal como el superheterodino rfRXD0420 puede incrementar el rango y reducir la susceptibilidad a la interferencia.

En modo ASK el pin $DATA_{FSK}$ y FSK_{OUT} no son usados y deberían ambos ser unidos a tierra. Un ejemplo de un típico circuito ASK es mostrado en la figura 2.5. El capacitor C1 puede ser reemplazado por un pequeño para simplificar el transmisor si el receptor tiene un

gran ancho de banda. Para un receptor de banda angosta el capacitor C1 puede necesitar ser reemplazado por un trimmer cap para sintonizar el transmisor a la frecuencia exacta.

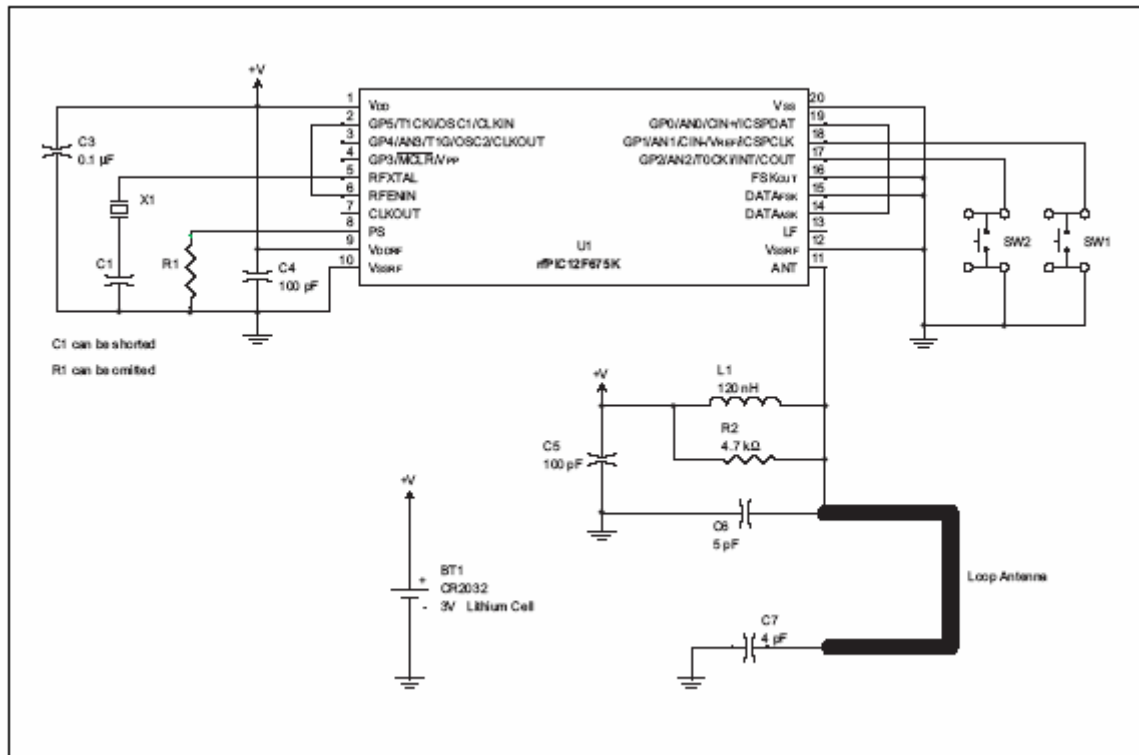


Figura. 2.5. Esquema del transmisor ASK típico

2.3.2 rfRXD0420 Receptor UHF ASK / FSK / F M

2.3.2.1 Características Generales

- ❑ Arquitectura del receptor de conversión superheterodino simple de bajo costo.
- ❑ Compatible con los transmisores RF de las series rPIC y rHCS.
- ❑ Fácil interfaz para el microcontrolador (MCU) y los decodificadores KEELOQ.
- ❑ La etapa del voltaje de oscilador controlado (VCO) está bajo seguro para el cristal de cuarzo de referencia:
 - Ancho de banda del receptor angosta.
 - Máximos rangos e inmune a las interferencias.
- ❑ Control de ganancia del LNA (Amplificador de bajo ruido) para mejorar el rango dinámico.

- ❑ Ancho de banda de la frecuencia intermedia seleccionable vía filtro cerámico externo.
- ❑ Indicador de la potencia de la señal recibida (RSSI) para la indicación de la potencia de la señal en FSK y FM y para demodulación ASK.
- ❑ Demodulador detector de cuadratura FSK / FM.

Receptor UHF ASK / FSK

- ❑ La frecuencia simple del receptor es establecido por la frecuencia del cristal.
- ❑ El rango de frecuencia recibida es:

DISPOSITIVO	RANGO DE FRECUENCIA
rfRXD0420	300 MHz – 400 MHz
rfRXD0920	800 MHz – 930 MHz

Tabla. 2.2. Banda de frecuencias de los receptores

- ❑ Máxima tasa de datos:

ASK: 80 Kbps NRZ

FSK: 40 Kbps NRZ

- ❑ Rango de la frecuencia intermedia: 455 KHz a 1.4 MHz.
- ❑ Rango del indicador de potencia de la señal recibida (RSSI): 70 dB.
- ❑ Rango de la división de frecuencia: ± 5 KHz a ± 120 KHz.
- ❑ Máxima modulación en frecuencia FM: 15 KHz.

Tecnología BI-CMOS

- ❑ Amplio rango de voltaje de operación.
- ❑ Bajo consumo de corriente en modo Standby o modo Activo

- rfRXD0420

- 8.2 mA (típico, en modo de alta ganancia LNA).
- < 100 nA en standby.

- rfRXD0920

- 9.2 mA (típico, en modo de alta ganancia LNA).
- < 100 nA en standby.

- Amplio rango de temperatura: -40 °C a + 85 °C.

2.3.2.2 Visión Global del rfRXD0420

rfRXD0420/0920 es un receptor de bajo costo, con recepción de corta distancia, compacto por lo que requiere un mínimo de componentes externos para un sistema de recepción completa. El rfRXD0420 cubre la frecuencia recibida en el rango de 300 MHz a 450 MHz, el rfRXD0920 cubre las frecuencias de 800 MHz a 930 MHz. El rfRXD0420 y rfRXD0920 comparten una arquitectura común. Estos están configurados para trabajar en la modulación ASK y FSK o modulación FM. Los rfRXD0420 y rfRXD0920 son compatibles con los rfPIC y la serie de rfHCS de los transmisores de RF.

- Alta estabilidad en la frecuencia sobre las variaciones de temperatura y variaciones de voltaje de alimentación.
- Baja señal falsa de emisión.
- Alto manejo de señales largas con control de ganancia LNA para mejorar el rango dinámico.
- Selección externa del ancho de banda de la frecuencia intermedia. El rango de la frecuencia intermedia está entre 455 KHz y 21.4 MHz.
- Modulación ASK o FSK para la recepción de datos digitales.
- Modulación FM para la recepción de señales análogas.
- Modulación FSK/FM empleando un detector de cuadratura.
- Indicador de potencia de señal recibida (RSSI) para detección ASK.
- Amplio rango de voltaje de alimentación.
- Bajo consumo de corriente en modo activo.

- Baja corriente en modo standby.

El rfRXD0420/0920 consiste de:

- Amplificador de bajo ruido (LNA) – ganancia seleccionable.
- Mezclador de baja conversión de la señal RF a la frecuencia intermedia seguido de un preamplificador de frecuencia intermedia.
- Integrado totalmente el sintetizador de frecuencia del Phase Locked Loop (PLL) para generar la señal de oscilador local (LO).
- El sintetizador de frecuencia consiste en:
 - Cristal oscilador
 - Detector de frecuencia de fase.
 - Controlador de voltaje del oscilador (VCO) de alta frecuencia.
 - Divisor de ancho de banda fijo.
 - rfRXD0420 = se divide por 16
 - rfRXD0920 = se divide por 32.
- Amplificador limitante de IF que es el que amplifica y limita la señal de IF y para la generación del indicador de la potencia de la señal recibida (RSSI).
- La parte del demodulador consiste (DEMODO) en un detector de fase (MIXER2) y un amplificador creando un detector de cuadratura, conocido también como detector de coincidencia de fase para demodular la señal de IF en las aplicaciones de modulación de FSK y FM.
- Amplificador operacional (OPA) el que puede ser configurado como un comparador para decisión de datos de modulación ASK o FSK o como un filtro para modulación FM.

La configuración de sus pines se presenta en la figura 2.6.

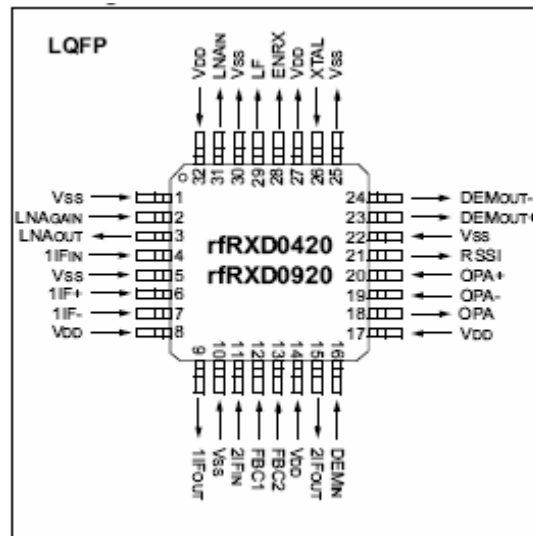


Figura. 2.6. Distribución de pines del rfRXD0420

2.3.2.3 Modulación por desplazamiento en Amplitud (ASK)

El amplificador limitador de IF con RSSI (Indicador de la potencia de la señal recibida) es usado como un detector ASK. La señal RSSI es filtrada y luego comparada a un voltaje de referencia para determinar si la señal RF entrante es un uno o cero lógico. El voltaje de referencia puede ser configurado como un nivel de voltaje dinámico determinado por la intensidad de señal de RF entrante o por un nivel predeterminado del mezclador.

2.3.2.4 Detector de filtrado RSSI

La señal RSSI es filtrada para remover las altas frecuencias y el ruido del pulso ayuda a tomar una decisión de proceso al comparar e incrementar la sensibilidad del receptor. La señal del filtro pasa bajos RSSI es de un filtro RC creado por impedancia de 36 ohmios de la salida RSSI y un capacitor C1. Configurando un tiempo constante ($RC = \tau$) del filtro RC depende el período de la señal y la decisión de cuando la señal se genera.

2.3.2.5 Período de la Señal

Una optimización de la sensibilidad del receptor con razonable distorsión del pulso ocurre cuando el filtro RC tiene un tiempo constante de entre 1 y 2 veces el período de la señal. Si la constante de tiempo del filtro RC es colocada demasiado corta, hay poco ruido beneficiando el filtrado. Sin embargo, si la constante de tiempo del filtro RC es demasiado larga, el pulso del dato se alargará causando interferencia intersimbólica.

2.3.3 Descripción del PIC16F870

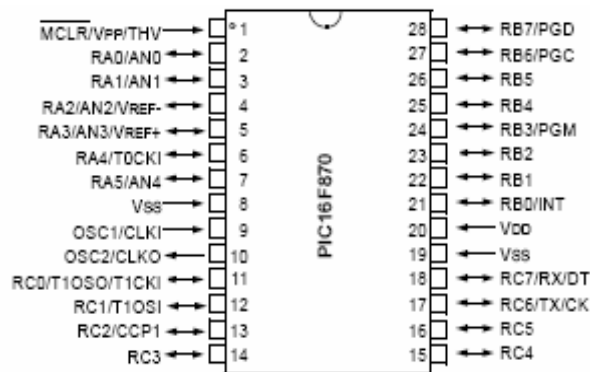


Figura. 2.7. Descripción de pines del PIC 16F870

2.3.3.1 Características Generales

- ❑ Procesador de arquitectura RISC avanzada.
- ❑ Juego de 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción, menos las de salto que tardan dos.
- ❑ Frecuencia de hasta 20 MHz. con 200ns de ciclo de reloj
- ❑ Memoria FLASH de hasta 2K x 14 bits
- ❑ Memoria de Datos RAM de hasta 128 x 8 bytes.
- ❑ Memoria de Datos EEPROM de hasta 64 x 8 bytes.
- ❑ Encapsulados compatibles con los PIC16CXXX de 28 y 40 pines.
- ❑ Hasta 11 fuentes de interrupción internas y externas.
- ❑ 8 niveles de Pila.

- ❑ Modos de direccionamiento directo e indirecto
- ❑ Perro guardián (WDT).
- ❑ Código de protección Programable.
- ❑ Modo SLEEP de bajo consumo.
- ❑ Programación serie en circuito con dos patitas.
- ❑ Voltaje de alimentación comprendido entre 2.0 y 5.5 V.
- ❑ Bajo consumo (menos de 1.6 mA a 5V y 4 MHz).

Dispositivos Periféricos:

- ❑ Timer 0: temporizador-contador de 8 bits con prescaler.
- ❑ Timer 1: temporizador-contador de 16 bits con prescaler .
- ❑ Un módulo de Captura-Comparación-PWM.
 Captura máxima de 16 bit con resolución de 12.5 ns
 Comparación máxima de 16 bit con resolución de 200 ns
- ❑ Conversor A/D de 10 bits.
- ❑ USART.

2.3.4 Descripción del PIC16F627

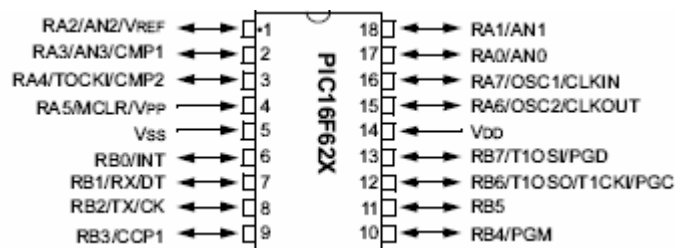


Figura. 2.8. Descripción de pines del PIC 16F627

2.3.4.2 Características Generales

- ❑ Juego de 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción, menos las de salto que tardan dos.
- ❑ Frecuencia de hasta 20 MHz. con 200ns de ciclo de reloj

- ❑ Memoria FLASH de hasta 1K x 14 bits
- ❑ Memoria de Datos RAM de hasta 224 x 8 bytes.
- ❑ Memoria de Datos EEPROM de hasta 128 x 8 bytes.
- ❑ 8 niveles de Pila.
- ❑ Modos de direccionamiento directo e indirecto
- ❑ Perro guardián (WDT).
- ❑ Código de protección Programable.
- ❑ Modo SLEEP de bajo consumo.
- ❑ Programación serie en circuito con dos patitas.
- ❑ Voltaje de alimentación comprendido entre 2.0 y 5.5 V.
- ❑ Bajo consumo (menos de 1.6 mA a 5V y 4 MHz).

Dispositivos Periféricos:

- ❑ 16 pines de entrada/salida con control individual
- ❑ Módulo de comparación análoga:
 - 2 comparadores análogos
 - las salidas del comparador son accesibles externamente
- ❑ Timer 0: temporizador-contador de 8 bits con prescaler.
- ❑ Timer 1: temporizador-contador de 16 bits con prescaler .
- ❑ Timer 2: temporizador-contador de 8 bits con prescaler .
- ❑ Un módulo de Captura-Comparación-PWM.
 - Captura máxima de 16 bit con resolución de 12.5 ns
 - Comparación máxima de 16 bit con resolución de 200 ns
- ❑ Conversor A/D con resolución de 10 bits.
- ❑ USART.

2.3.5 Interfase Serial RS-232

La PC utiliza la norma RS-232, que es un estándar mundial que rige los parámetros de uno de los modos de comunicación serial. Por medio de este protocolo se estandarizan las velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia,

los niveles de voltajes utilizados, el tipo de cable permitido, las distancias entre equipos, los conectores, etc.

Este estándar especifica que los niveles de salida son -5 a -15 voltios para el 1 lógico y +5 a +15 voltios para el 0 lógico, mientras que los niveles de entrada son -3 a -15 voltios para un 1 lógico y +3 a +15 voltios para un 0 lógico.

En una transmisión serial se forma un “tren” de bits; uno tras de otro viajan del lugar de emisión al receptor utilizando una sola vía, en este caso será un conductor eléctrico bus Serial, como en caso de los trenes con una sola vía si se desea transmitir en el sentido contrario, se debe esperar que la vía este libre.

Los datos serie se encuentran encapsulados en tramas de la forma:

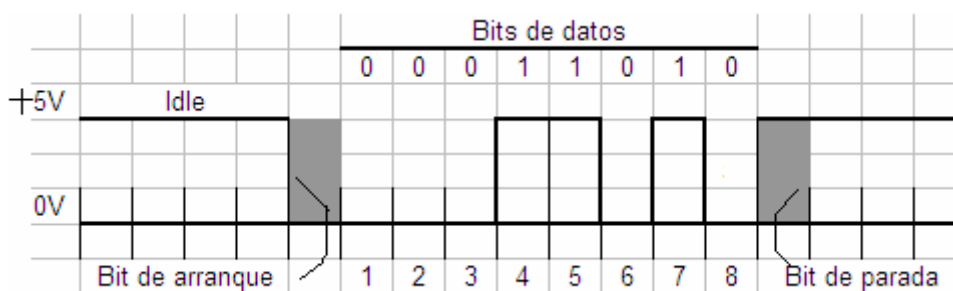


Figura. 2.9. Estructura de un carácter que se transmite serialmente.

Primero se envía un bit de Inicio, a continuación los bits de Datos (primero el bit de mayor peso) y finalmente los bits de Parada. El número de bits de datos y de bits de parada es uno de los parámetros configurables, así como el criterio de paridad par o impar para la detección de errores.

Normalmente, cuando no se realiza ninguna transferencia de datos, la línea del transmisor es pasiva (idle) y permanece en un estado alto. Para empezar a transmitir datos, el transmisor coloca esta línea en bajo durante el tiempo de un bit.

Tanto el transmisor como el receptor deben tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y de bits de parada.

2.3.5.1 Circuito integrado MAX232

Descripción: Este circuito integrado soluciona los problemas de niveles de voltaje cuando se requiere enviar señales digitales sobre una línea RS-232.

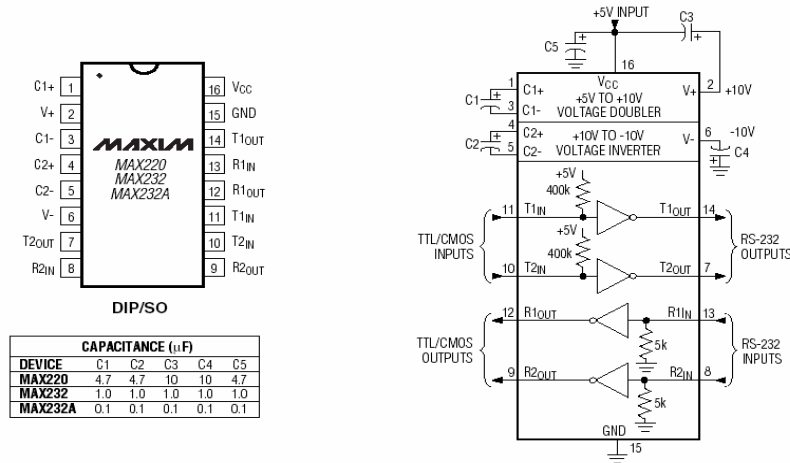


Figura. 2.10. Diagrama de pines y estructura interna del MAX232.

Funcionamiento: El MAX232 convierte los niveles RS-232 (cerca de +10 y -10 V) a voltaje TTL (0 a +5 V) y viceversa, sin requerir nada más que una fuente de +5 V. Un elevador de voltaje interno convierte el voltaje de +5 V al de doble polaridad de ± 12 V. Lleva internamente 2 conversores de nivel de TTL a RS232 y otros 2 de RS232 a TTL con lo que en total se puede manejar 4 señales del puerto serie de la PC, por lo general las más usadas son; TX, RX, RTS, CTS, estas dos últimas son las usadas para el protocolo handshaking pero no es imprescindible su uso. Para que el MAX232 funcione correctamente se debe poner unos condensadores externos de 10 uF para generar el voltaje RS-232 internamente, todo esto se puede ver en la figura 2.11, en la que solo se han cableado las líneas TX y RX que son las más usadas para casi cualquier aplicación.

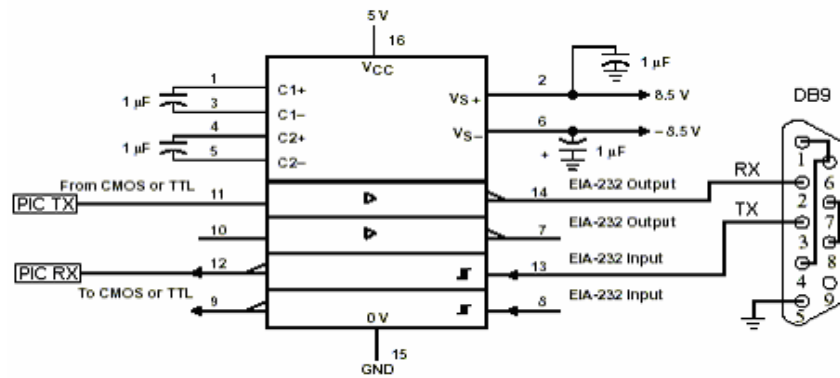


Figura. 2.11. Diagrama de conexión de pines del MAX232 con PIC

2.4 SOFTWARE DEL SISTEMA

2.4.1 Interfaz Hombre Máquina HMI

Un HMI o Interfaz Hombre Máquina no es sino un software que es utilizado para monitorear y controlar a un sistema o planta haciendo uso de distintas plataformas computacionales las cuales, sirven para crear una interfaz gráfica y de entorno amigable con el usuario u operario. Siendo el mismo capaz de manejarlo sin necesidad de ser un experto ni de haber desarrollado su código fuente.

2.4.2 Bases de Datos

Las bases de datos se pueden definir como una colección de datos interrelacionados entre sí, almacenados en conjunto *sin redundancias perjudiciales e innecesarias*. Su finalidad es servir a una aplicación o más, de la mejor manera posible, los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados.

Generalmente se habla de la independencia como un gran atributo de las bases de datos. La independencia se refiere a que los programas de aplicación y los datos que de

ellos se sirven son mutuamente independientes. Esto significa que el modificar los programas no afecta los datos o su organización.

2.4.2.1 Características de una Base de Datos

Las bases de datos tienen muchas formas de organización pero deben tener determinadas características:

- Versatilidad para la representación de relaciones
- Desempeño
- Costo mínimo
- Redundancia mínima
- Capacidad de búsqueda
- Integridad
- Seguridad de acceso a los datos y protección de la información
- Interfaz con el pasado
- Interfaz con el futuro
- Modificable en el tiempo
- Migración de datos
- Simplicidad

2.4.2.2 Organizaciones de Datos

Una base de datos es una colección de información. Y su característica principal es que los datos los organiza de acuerdo a modelos de datos.

Se considera la definición de modelo de datos como "un conjunto de conceptos y de reglas de composición de esos conceptos que permite describir unos datos", es decir, proveen un marco teórico que sustenta una metodología bien definida para el almacenamiento y recuperación de la información.

En la práctica existen tres modelos principales de datos: el modelo jerárquico, el modelo red, y el modelo relacional. Este último está relacionado con las bases de datos relacionales y el enfoque Entidad-Relación, al cual está enfocado este estudio.

2.4.2.3 Bases de Datos Relacionales

El modelo relacional tiene asociada una teoría que no puede ser separada del modelo: la teoría de la normalización de las relaciones. Esta teoría tiene por objeto eliminar los comportamientos anormales de las relaciones durante las actualizaciones. También permite eliminar los datos redundantes y facilita la comprensión de las relaciones semánticas entre los datos.

2.4.2.4 Conceptos Básicos del Modelo Relacional

Una entidad es un objeto que existe y puede distinguirse de otros objetos. Una entidad puede ser concreta, por ejemplo una persona o un libro, o abstracta, como un día festivo o un concepto. Una entidad identifica en forma unívoca al objeto que representa en el universo, por ejemplo Cuenta Corriente 143-2343-34-21.

Un conjunto de entidades es un grupo de entidades del mismo tipo. Por ejemplo, el conjunto de todas las personas que tienen una cuenta en el banco puede definirse como el conjunto de entidades *CuentaCorrentistas* del banco. Así mismo, el conjunto de entidades *Cuenta* podría representar el conjunto de todas las cuentas en un banco determinado. Las entidades pueden pertenecer a más de un conjunto, por ejemplo, la entidad *Persona* puede pertenecer a los conjuntos *Empleado*, *Cliente*, *Asociado*, etc.

Las entidades están representadas y tienen sentido por sus atributos. Los posibles atributos del conjunto de entidades *CuentaCorrentista* son *Nombre*, *Ruc*, *Calle*, *Número*, *Ciudad*. Los valores permitidos para un atributo constituyen su dominio. El dominio de el atributo *Nombre* podría ser el conjunto de todas las cadenas de caracteres de una determinada longitud.

Fuera del marco teórico, un conjunto de entidades es lo que llamamos Tabla. Cada Columna en la tabla corresponde a un atributo, suele recibir el nombre de Campos. Cada Fila en la tabla corresponde a los datos de una entidad del mundo real, se le llama habitualmente Registro. La tabla 2.3 ilustra esto:

Número cuenta	Ruc cliente	Saldo	Giros	Depósitos

Tabla. 2..3. Ejemplo de tabla para base de datos

Cuando se analiza un problema informático una de las primeras tareas es reconocer el universo de información. Si se pudiera almacenar y recuperar mágicamente toda la información que constituye el universo tendríamos resuelto nuestro problema. Sin embargo, es necesario dar una estructuración adecuada a la información para evitar los problemas de redundancia y duplicidad. Esta estructuración significa identificar y definir las entidades que constituyen el universo. Para ello se aplica una descomposición al universo de modo que:

- Las entidades sean conceptualmente puras, es decir, sólo contengan información sobre el conjunto de entidades del mundo real que representa. Por ejemplo, en una tabla que contenga información sobre la cuenta corriente bancaria no deberá contener información sobre la previsión de salud de las personas. Para manejar esta información se deben llevar dos tablas: Cuenta_corriente y Previsión_Salud.
- Los datos almacenados en una tabla deben ser "atómicos", es decir, la intersección de una columna con una fila no podrá contener más de un valor. Por ejemplo:
Si una tabla contiene un campo definido como edades de cargas familiares se podría definir para que contuviera una cadena de caracteres con esos valores: 5; 11;

18 lo que no sería correcto. En tal caso sería necesario definir una columna para carga o una tabla que permita almacenarlas.

2.4.2.5 Relaciones

- **Asociación entre Entidades (relación)** Es la asociación entre varias entidades. Por ejemplo, es posible definir una relación que asocia al CuentaHabiente "Harris" con la cuenta 323. Esto especifica que Harris es un CuentaHabiente con la cuenta bancaria 323.

- **Conjunto de Asociaciones (relaciones)** Un conjunto de relaciones es un grupo de relaciones del mismo tipo. Formalmente es una relación matemática de N mayor o igual que 2 (posiblemente idénticos) conjuntos de entidades. Por ejemplo, se puede definir la relación CtHabCta para notar la asociación entre los cuentahabientes y las cuentas bancarias. Una relación también puede tener atributos, por ejemplo, fecha podría ser un atributo del conjunto de relaciones CtHabCta; esto especifica la última vez que el CuentaHabiente tuvo acceso a su cuenta.

- **Limitantes** Una limitante importante es la de las cardinalidades de mapeo entre las distintas entidades, es decir, el N° de entidades con las que puede asociarse otra entidad a través de una asociación. Para un conjunto binario de relaciones se tiene lo siguiente:

Relación	Asociación	Descripción
Una a Una	(1 : 1)	Función inyectiva de A en B
Una a Muchas	(1 : n)	Una entidad A está relacionada con varias entidades de B.
Muchas a Una	(n : 1)	Muchas entidades de A está solo relaciona con una entidad de B.
Muchas a Muchas	(n : n)	Varias entidades de A están relacionadas con varias entidades de B.

Tabla. 2.4. Asociación de Limitantes

- **Llaves Primarias** Como ya se ha mencionado anteriormente las llaves permiten identificar una entidad o un grupo. La llave está constituida por atributos (puede ser 1 o un grupo). Una entidad fuerte posee una llave primaria. Una llave primaria me permite identificar una entidad y la llave secundaria me permite identificar un grupo de entidades.

- **Llaves Foráneas** Corresponden a un atributo de una tabla relacionada que es llave primaria en una tabla principal, es decir, es el atributo de una entidad, que es llave primaria de otra.

- **Integridad referencial** Conjunto de reglas que se siguen para preservar las relaciones establecidas entre las tablas cuando se introducen o se agregan registros. Si se exige integridad referencial el SABDR (Sistema Administrador de Bases de Datos Relacionales) impedirá agregar registros a una tabla relacionada para la cual no hay un registro principal; también impedirá cambiar valores en una tabla principal cuando el resultado de la modificación implica dejar registros huérfanos en la tabla relacionada, así como eliminar registros cuando existen registros coincidentes en una tabla relacionada.

-
- **La Aproximación por Descomposición** La aproximación por descomposición para concebir esquemas relacionales parte de una relación compuesta de todos los atributos, llamada la relación universal para descomponer después esta relación en subrelaciones que no padecen las anomalías anteriormente descritas. El proceso es un proceso de depuración sucesiva que debe lograr aislar unas entidades y unas asociaciones del mundo real. Debe estar realizado por un algoritmo a partir de una buena comprensión semántica de los datos.

En resumen, puede decirse que, en términos tradicionales, una relación se asemeja a un *archivo*, una tupla a un *registro* y un atributo a un *campo*. Sin embargo, estas correspondencias son a lo sumo aproximadas. Dicho de otra manera, las relaciones pueden considerarse archivos "altamente disciplinados".

Esta disciplina se traduce en una simplificación considerable de las estructuras de datos que el usuario debe manejar, y, por tanto, en una simplificación correspondiente de los operadores necesarios para manipularlas.

CAPÍTULO 3

DISEÑO DEL SISTEMA

3.1 GENERALIDADES

El sistema funciona una vez registrado el cliente en la base de datos del sistema, este registro se lo efectúa utilizando el software de administración (SIADPE), la placa del vehículo es utilizada como el código del usuario, la misma que es grabada en el módulo transmisor, quien transmitirá este código una vez que el cliente acceda al área de cobertura del peaje, una vez que él es detectado por un sensor ubicado a poca distancia del centro de control, este sensor activa un tiempo en el cual espera haber recibido el código del usuario para poder debitar de la cuenta bancaria del mismo, caso contrario, el centro de control ordena la activación de una alarma para que las autoridades competentes actúen. A continuación se detalla el diagrama funcional (figura. 3.1) y la descripción de las acciones de cada bloque que constituyen el sistema de control electrónico de peaje.

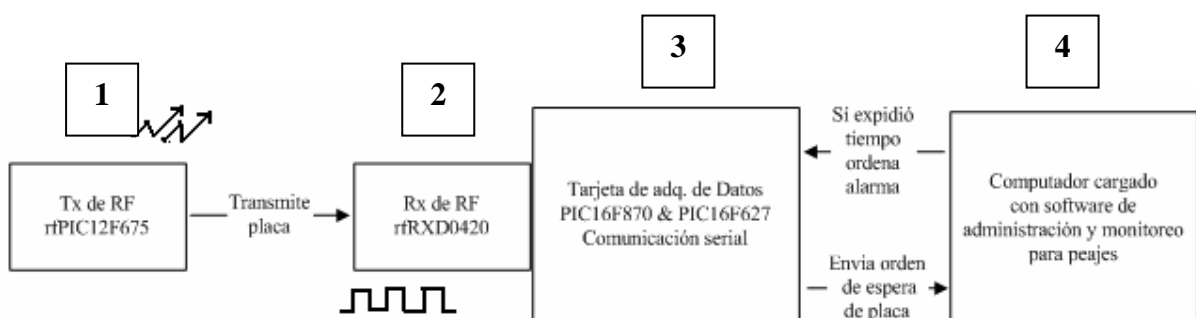


Figura. 3.1. Diagrama funcional del sistema.

- 1.- Envía código (placa de vehículo) grabado en módulo transmisor cada 150 mseg.
- 2.- Recibe información modulada en ASK y transmitida en la frecuencia de 433.92 MHz.
- 3.- Discrimina información, recibiendo solamente código grabado en el módulo transmisor. Una vez activado el sensor el cual detecta la presencia del usuario en el sistema emite una señal al software de monitoreo (figura. 3.1.(4).)
- 4.- La misma que al ser detectada, inicia un tiempo de espera para que llegue el código (placa vehículo). Luego de este tiempo toma dos decisiones : descuenta si llegó el código del vehículo, u ordena la activación de la alarma del sistema enviando una señal nuevamente a la tarjeta de adquisición de datos (figura. 3.1.(3)), si no llegó el código del vehículo.

3.2 COMPONENTES DEL SISTEMA

Para referirnos al diseño del sistema, es necesario utilizar todos los conceptos antes mencionados, en lo que respecta a cada uno de los subsistemas, a continuación se muestra el diagrama de bloques del sistema total (figura. 3.2)



Figura. 3.2. Diagrama de Bloques de Sistema

3.2.1 Transmisión de Datos

3.2.1.1 Diseño de Hardware del Subsistema de Transmisión de Datos

Este subsistema está compuesto de dos bloques, como se muestra en el siguiente diagrama de bloques (figura 3.3):

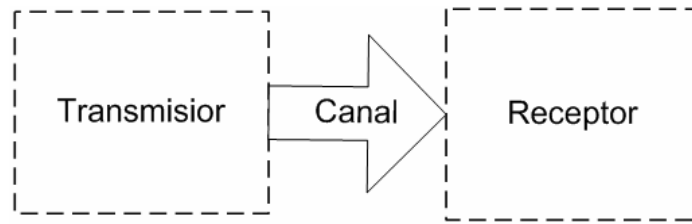


Figura. 3.3. Diagrama de Bloques del subsistema de Transmisión de Datos

El subsistema de Transmisión de Datos realiza la comunicación del vehículo el cual contiene el transmisor de RF, con la tarjeta de Adquisición de Datos la cual tiene dentro el receptor de RF. La función específica de cada una de estas partes es:

Transmisor de RF: Es el que transforma el código del usuario al formato RS232, para luego ser transmitido utilizando modulación ASK.

Canal RF: Este es el camino por el cual viajan los datos en forma de ondas electromagnéticas que oscilan a una frecuencia de 433.92 MHz, es decir dentro del rango UHF del espectro radioeléctrico.

Receptor de RF: Es el que demodula la señal, para a su salida obtener nuevamente el código del usuario en formato RS232.

Para el desarrollo de este subsistema, hemos decidido, utilizar como base un sistema de transmisión de datos inalámbrica, desarrollado por la empresa Microchip, denominado rfPIC™ Development Kit (figura. 3.4), se tomó la decisión de llevar a cabo este subsistema con la argumentación, que según el ITS, el sistema mas óptimo, es el que evita que el vehículo se detenga para desarrollar cualquier tipo de transacción. Utilizando estas recomendaciones y observando las características de esta tecnología, se decidió aplicarla. Puesto que también su costo de desarrollo fue módico, tomando en consideración que su comunicación es simplex, es decir la comunicación se desarrolla en una dirección es así que el módulo de transmisión se puede comunicar con el módulo de recepción pero no en sentido opuesto.

A continuación detallaremos las partes de este Kit de desarrollo, el cual nos permitió ejecutar de una forma fácil la transmisión de datos del sistema.

rfPIC™ Development Kit



Figura. 3.4. rfPIC™ Development Kit

Este kit de desarrollo contiene:

Una tarjeta de desarrollo PICKIT™ 1 Flash Starter Kit

Un módulo de transmisión de RF de 315 MHz.

Un módulo de transmisión de RF de 433.92 MHz.

Un módulo de recepción de RF de 315 MHz.

Un módulo de recepción de RF de 433.92 MHz.

Software del compilador.

PICKIT™ 1 Flash Starter Kit

El PICKIT™ 1 Flash Starter Kit figura 3.5 es un programador introductorio, con el cual es fácil empezar programando PICs de Microchip de la familia de 8/14-pines. Además fue diseñado para la programación de los módulos de transmisión de RF que contienen PICs de RF (rfPIC12F675K/F).

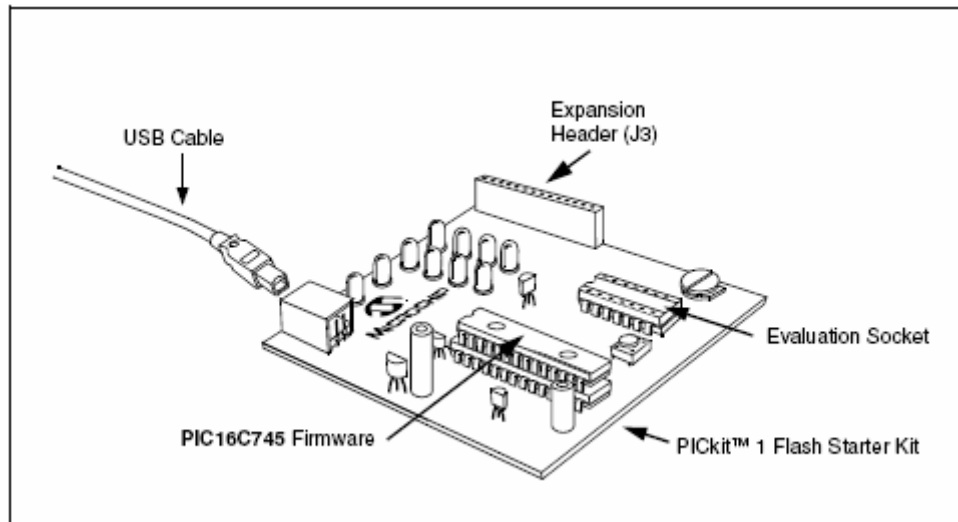


Figura. 3.5. PICkit 1 Flash Starter Kit

Módulo de Transmisión de RF

El rfPIC12F675F es un transmisor de radio ASK de rango corto UHF de alto desempeño diseñado usando PICs de la compañía Microchip como rfPIC12F675K para 315MHz y rfPIC12F675F para 433.92MHz.

El rfPIC12F675 figura 3.6 es un módulo transmisor autónomo que puede ser usado de distintas maneras con el rfPIC Development Kit 1

El módulo transmisor contiene:

2 pulsantes conectados a GP3 y GP4 del microcontrolador

2 potenciómetros conectados a GP0 y GP1

Habilitador de RF (RF_{ENIN}) conectado a GP5

Dato ASK ($DATA_{ASK}$) conectado a GP2

Soket opcional de 8 pines (U2) para emulación en circuito o insertando un DIP de 8 pines.

PIN	DESCRIPCION
1	GP5
2	GP4
3	GP3
4, 5, 6	No conexión
7	GP0
8	GP1
9	GP2
10, 11, 12	No conexión
13	Alimentación: 2.0 - 5.5 VDC
14	Tierra

Tabla. 3.1 Pines del módulo rPIC12F675

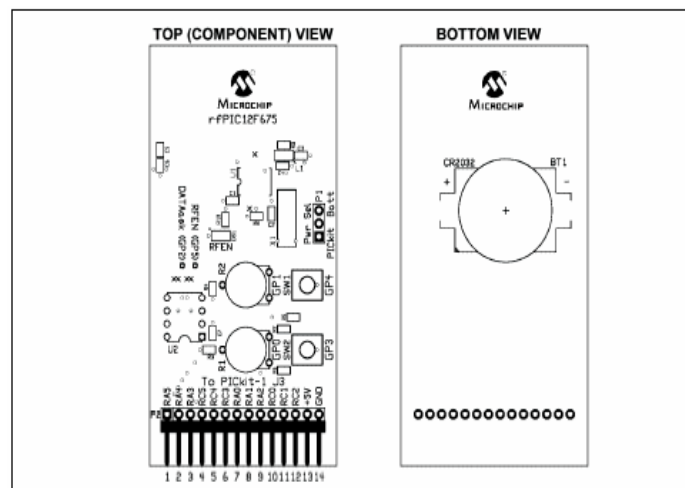


Figura. 3.6. Módulo Transmisor

Requerimientos de Alimentación

El jumper Pwr Sel P1 selecciona dos fuentes de alimentación para el rPIC12F675:

Colocando el jumper en la posición PICKit (pines 1 y 2) permite al módulo transmisor ser alimentado desde el conector P2 pin 13. Cuando el módulo del transmisor es conectado en la expansión J3, el módulo transmisor es alimentado desde el PICKit™ Starter Kit, y además puede ser programado desde él.

Colocando el jumper en la posición Batt (pines 2 y 3) permite al módulo transmisor ser alimentado desde una batería de Litio. Cuando el módulo del transmisor es alimentado por la batería puede ser usado para operaciones portátiles.

El rfPIC12f675 puede ser programado por el PICKIT™ 1 Flash Starter Kit, seguidamente se detallarán los pasos para su programación:

Paso 1: Quitar el PIC16F676 o el PIC12F675 del soket de evaluación del PICKIT™ 1 Flash Starter Kit figura 3.7.

Paso 2: Conecte el PICKIT™ 1 Flash Starter Kit al puerto USB del PC figura 3.7.

Paso 3: Conecte el módulo del transmisor en la expansión J3 del PICKIT™ 1 Flash Starter Kit figura 3.7.

Paso 4: El interno rfPIC12F675 en el dispositivo RF ahora viene a ser el objeto de programación. El módulo de transmisión puede ser removido para su operación autónoma.

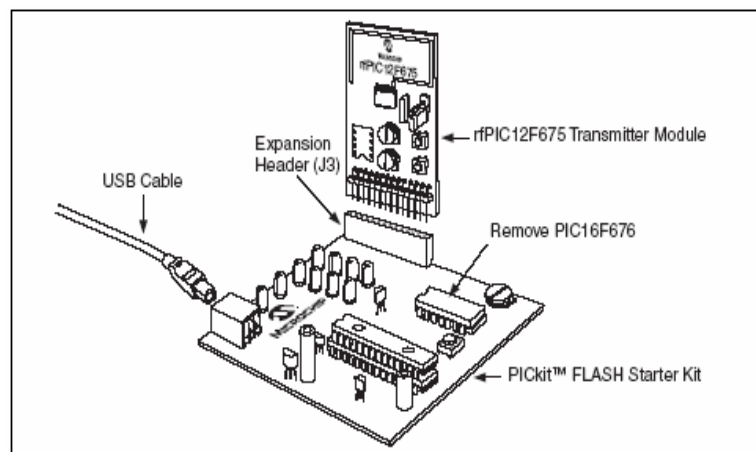


Figura. 3.7 Descripción de conexión del rfPIC12F675 sobre el PICKIT™

Paso 5: Abrir el software del grabador seleccionando Inicio>Programas>PICKit (tm) 1 Flash Starter Kit> PICKit (tm) 1 Flash Starter Kit . Aparecerá la interface de programación como se muestra en la figura 3.8

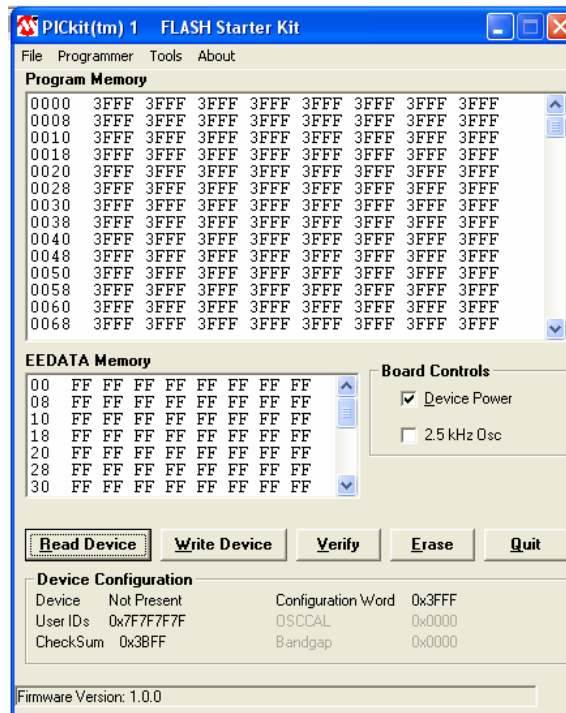


Figura. 3.8 Interface del Software del Grabador

Paso 6: Importar el programa compilado (archivo .HEX), seleccione *File>Import HEX*, como se muestra en la figura 3.9. Busque el archivo hex y haga clic en Open .

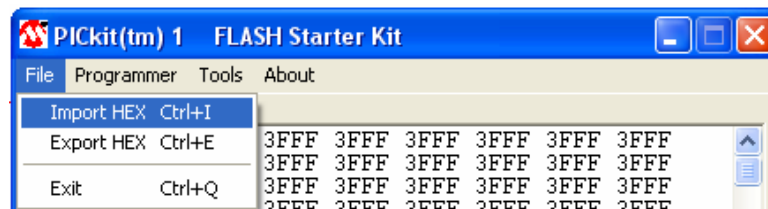


Figura. 3.9. Importar archivo .HEX

Paso 7: Para Grabar el dispositivo haga clic en Write Device

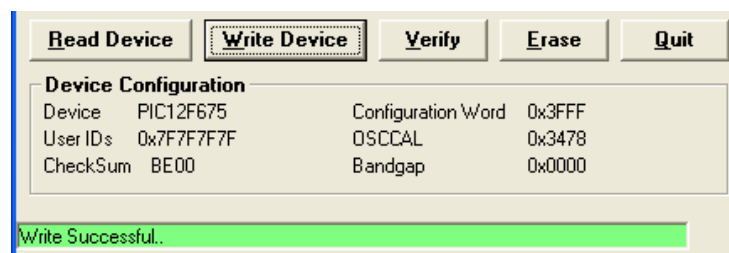


Figura. 3.10. Grabando el Dispositivo

Módulo de Recepción de RF

El módulo receptor rfRXD0420, es receptor de radio ASK de rango corto UHF de alto desempeño diseñado usando PICs de la compañía Microchip como rfRXD0420.

PIN	DESCRIPCION
1-10	No conexión
11	Entrada de dato recibido
12	No conexión
13	Alimentación: 2.5 - 5.5 VDC
14	Tierra
ANT	Conexión antena

Tabla. 3.2 Pines del módulo rfRXD0420

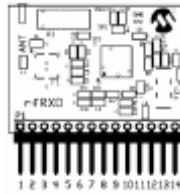


Figura. 3.11. Módulo Receptor

Las especificaciones de diseño del módulo receptor son:

Un solo canal, con mezclador de frecuencia en 315 MHz y 433.92 MHz.

Modulación ASK.

Tasa de señal de 4800 baudios.

3.2.1.2 Diseño del Software del Subsistema de Transmisión de Datos

Como mencionamos anteriormente, el módulo de transmisión es programable, gracias a esta característica se puede ajustar toda su circuitería a nuestra aplicación solo debemos observar como está construido el módulo de transmisión puesto que ésta ya contiene elementos y debemos configurar todos sus registros de tal forma que no afecte su diseño interno como se muestra en la figura. 3.12.

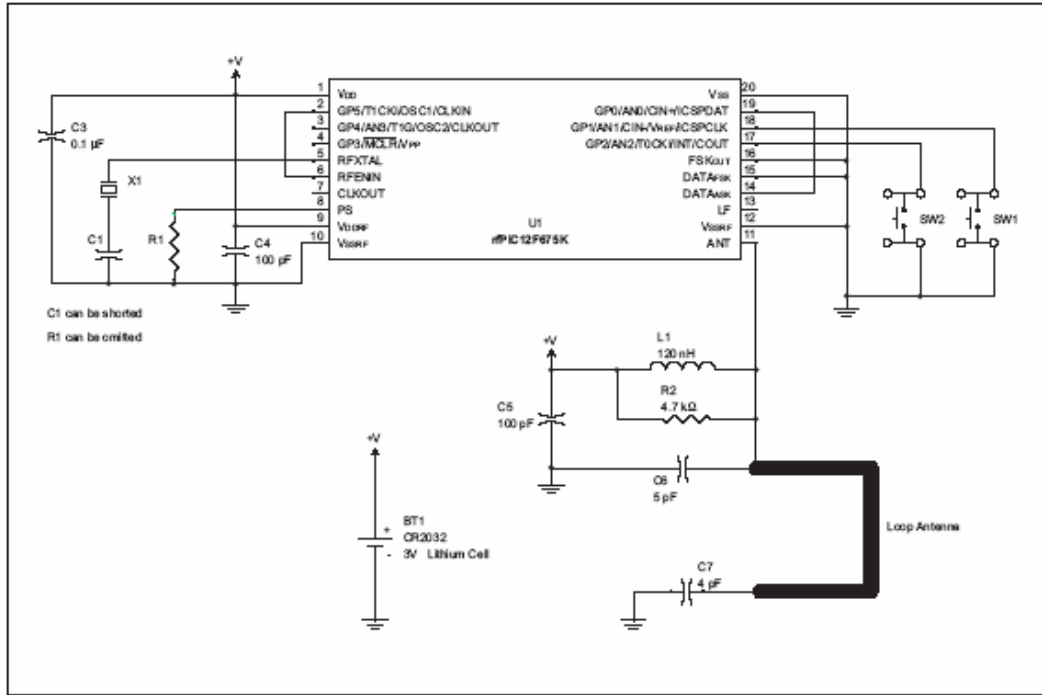


Figura. 3.12. Esquema del transmisor ASK típico

Diagrama de Flujo del Programa del Módulo Transmisor

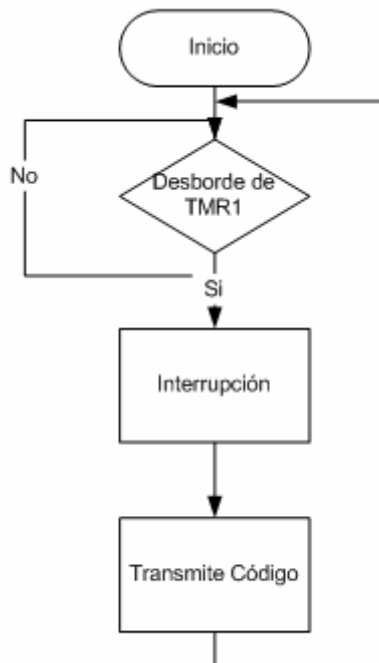


Figura. 3.13. Diagrama de Flujo del Programa del Módulo transmisor

Como se indica en el diagrama de flujo del programa del módulo transmisor, éste emitirá el código del usuario en cada desborde del TMR1 que está configurado como se muestra a continuación

Configuración de los Registros del módulo de Transmisión

Registro OPTION (81h)

1	0	0	1	1	0	0	0
(GPPU)'	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7						bit 0	

bit 7 (GPPU)': Bit habilitador de pull-up en GPIO.

1 = GPIO pull-ups están deshabilitados.

0 = GPIO pull-ups están habilitados por valores de puerto match individuales.

bit 6 INTEDG: Bit selector de interrupción de borde.

1 = Interrumpe con el flanco ascendente en el pin GP2/INT.

0 = Interrumpe con el flanco descendente en el pin GP2/INT.

bit 5 T0CS: Bit de selección de reloj TMR0 de origen .

1 = Transición en el pin GP2/T0CK1

0 = Instrucción interna de ciclo de reloj (CLKOUT).

bit 4 T0SE: Bit de selección de origen de desborde del TMR0.

1 = Incrementa la transición de alto a bajo en el pin GP2/T0CK1.

0 = Incrementa la transición de bajo a alto en el pin GP2/T0CK1.

bit 3 PSA: Bit de asignación del escalador.

1 = El escalador es asignado al WDT.

0 = El escalador es asignado para el módulo TMR0.

bit 2-0 PS2-PS0: Bits de selección de la razón del escalador.

Valor Bit	TMR0 Rate	WDT Rate
000	1:2	1:1

Registro INTCON (0B_h o 8B_h)

1	1	0	0	0	0	0	0	
GIE	PEIE	TOIE	INTE	GPIE	TOIF	INTF	GPIF	
bit 7								bit 0

- bit 7 GIE: Bit habilitador de la interrupción global.
1 = Habilita todas las interrupciones.
0 = Deshabilita todas las interrupciones.
- bit 6 PEIE: Bit habilitador de interrupciones periféricas.
1 = Habilita todas las interrupciones periféricas.
0 = Deshabilita todas las interrupciones periféricas.
- bit 5 TOIE: Bit habilitador de interrupción por desborde del TMR0.
1 = Habilita las interrupciones del TMR0.
0 = Deshabilita las interrupciones del TMR0.
- bit 4 INTE: Bit habilitador de interrupción externa del GP2/INT.
1 = Habilita la interrupción externa del GP2/INT.
0 = Deshabilita la interrupción externa del GP2/INT.
- bit 3 GPIE: Bit habilitador de interrupción por cambio de puerto⁽¹⁾.
1 = Habilita la interrupción por cambio de puerto en el GPIO.
0 = Deshabilita la interrupción por cambio de puerto en el GPIO.
- bit 2 TOIF: Bit bandera por interrupción de desborde del TMR0⁽²⁾.
1 = El registro TMR0 tiene desborde (debe ser limpiado por software).

0 = El registro TMR0 no tiene desborde.

bit 1 INTF: Bit Bandera de Interrupción externa en GP2/INT.

1 = Ocurrió una interrupción externa (se pone a cero por software).

0 = No ha ocurrido una interrupción externa en GP2/INT.

bit 0 GPIF: Bit bandera de interrupción por cambio de puerto.

1 = Cuando en uno de los más pequeños entre los pines GP5-GP0 cambian de estado (se pone a cero por software).

0 = Ninguno de los pines GP5 a GP0 han cambiado de estado.

Registro PIE - PERIPHERAL INTERRUPT ENABLE REGISTER 1 (8C_h)

0	0	0	0	0	0	0	1
EEIE	ADIE	-	-	CMIE	-	-	TMR1IE
bit 7							bit 0

bit 7 EEIE: Bit habilitador de interrupción de escritura completa EE.

1 = Habilita la interrupción de escritura completada EE (EEPROM).

0 = Deshabilita la interrupción de escritura completada EE.

bit 6 ADIE: Bit habilitador de la interrupción del convertor análogo-digital A/D.

1 = Habilita la interrupción del convertor A/D.

0 = Deshabilita la interrupción del convertor A/D.

bit 5-4 No implementado. Leer como “0” (cero).

bit 3 CMIE: Bit habilitador de la interrupción del comparador.

1 = Habilita la interrupción del comparador.

0 = Deshabilita la interrupción del comparador.

bit 2-1 No implementado. Leer como “0” (cero).

bit 0 TMR1IE: Bit habilitador de interrupción de desborde del TMR1.

1 = Habilita la interrupción de desborde del TMR1.

0 = Deshabilita la interrupción de desborde del TMR1.

Registro PIR1- PERIPHERAL INTERRUPT REGISTER 1 (0Ch)

0	0	0	0	0	0	0	0
EEIF	ADIF	-	-	CMIF	-	-	TMR1IF
bit 7							bit 0

bit 7 EEIF: Bit bandera de interrupción en la operación de escritura EEPROM.
 1 = Completada la operación de escritura (Se pone a cero por software).
 0 = La operación de escritura no ha sido completada o no ha comenzado.

bit 6 ADIF: Bit bandera de interrupción del conversor A/D.
 1 = La conversión A/D está completa (Se debe poner a cero por software).
 0 = La conversión A/D no está completa.

bit 5-4 No implementado. Se lee como ceros.

bit 3 CMIF: Bit bandera de interrupción del comparador.
 1 = La entrada del comparador ha cambiado (Se pone a cero por software).
 0 = La entrada del comparador no ha cambiado.

bit 2-1 No implementado. Se lee como ceros.

bit 0 TMR1IF: Bit bandera de interrupción de desborde del TMR1.
 1 = Hay desborde en el registro TMR1 (Poner a cero por software).
 0 = No hay desborde en el registro TMR1.

Registro ANSEL - ANALOG SELECT REGISTER - (9Fh)

0	0	0	0	0	0	1	1
-	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

bit 7: No implementado. Se lee como “0”.

bit 6-4 ADCS: Bit selector de reloj del conversor A/D.

000 = $F_{OSC/2}$.

001 = $F_{OSC/8}$

010 = $F_{OSC/32}$

x11 = F_{RC} (reloj obtenido del oscilador interno dedicado máx. 500 KHz)

100 = $F_{OSC/4}$

101 = $F_{OSC/16}$

110 = $F_{OSC/64}$

bit 3-0 ANS3-ANS0: Bit selector análogo.

Entre funciones análogas o digitales en los pines AN respectivamente.

1 = Entrada análoga. El pin es asignado como entrada análoga ⁽¹⁾.

0 = I/O digital. El pin es asignado para puerto o función especial.

Nota (1): Poniendo al pin como entrada análoga automáticamente deshabilita la entrada digital, weak pull-ups y la interrupción por cambio. El correspondiente bit TRISIO debe estar en “1” para que permita el control externo del voltaje en el pin.

Registro TRISIO (85_h)

0	0	0	1	1	0	1	1
-	-	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0
bit 7							bit 0

bit 7-6 No implementado. Se lee como “0”

bit 5-0 TRISIO: Bit de control de tri-estado de I/O de propósito general.

1 = Pin GPIO es configurado como una entrada. (Tres estados).

0 = Pin GPIO es configurado como una salida.

Nota: GP2 Y GP5, como salidas por RF_{ENIN} y Salida de $DATO_{ASK}$

Registro TICON - TIMER1 CONTROL REGISTER - (10h)

0	0	0	0	0	1	0	1
-	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	(T1SYNC)'	TMR1CS	TMR1ON
bit 7							bit 0

- bit 7 No implementado. Se lee como "0".
- bit 6 TMR1GE: Bit habilitador de la entrada al Timer1.
 Sí TMR1ON = 0:
 Este bit es ignorado.
 Sí TMR1ON = 1:
 1 = Timer1 está prendido si el pin (T1G)' está en bajo.
 0 = Timer1 está prendido.
- bit 5-4 T1CKPS1 – T1CKPS0: Bits de selección de la preescaler de la entrada del reloj del Timer1.
 11 = El valor de la preescaler es 1:8.
 10 = El valor de la prescaler es 1:4.
 01 = El valor de la preescaler es 1:2.
 00 = El valor de la preescaler es 1:1.
- bit 3 T1OSCEN: Bit habilitador de control del oscilador LP.
 Sí INTOSC sin el oscilador CLKOUT está activo:
 1 = El oscilador LP es habilitado para el reloj del Timer1.
 0 = El oscilador LP está apagado.
 Caso contrario:
 Este bit es ignorado.
- bit 2 (T1SYNC)': Bit de control de sincronización de la entrada del reloj externo del Timer1.
 TMR1CS = 1:
 1 = Entrada del reloj externo no es sincronizada.
 0 = Entrada del reloj externo es sincronizada.

TMR1CS = 0:

Este bit es ignorado. El Timer1 usa el reloj interno.

bit 1 TMR1CS: Bit de selección de reloj de origen del Timer1.

1 = Reloj externo del pin T1OSO/T1CKI (en la subida de borde).

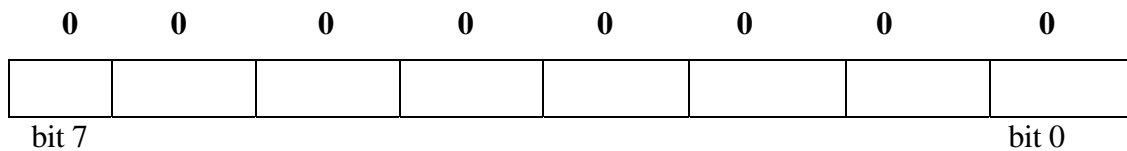
0 = Reloj interno ($F_{osc}/4$).

bit 0 TMR1ON: Bit para encender el Timer1.

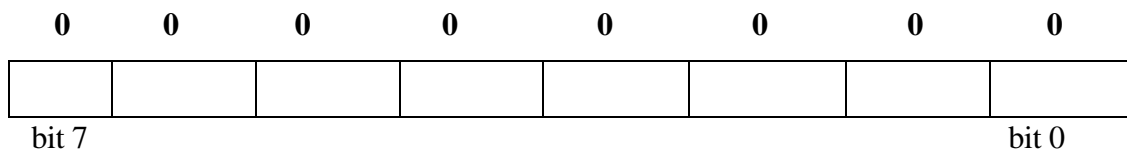
1 = Habilita el Timer1.

0 = Detiene el Timer1.

Registro TMR1H - TIMER1 HIGH REGISTER - (0Fh)



Registro TMR1L - TIMER1 LOW REGISTER - (0Eh)



3.2.2 Adquisición de Datos

3.2.2.1 Diseño del Hardware del Subsistema de Adquisición de Datos

Este subsistema está compuesto de cuatro bloques, como se muestra en el siguiente diagrama:

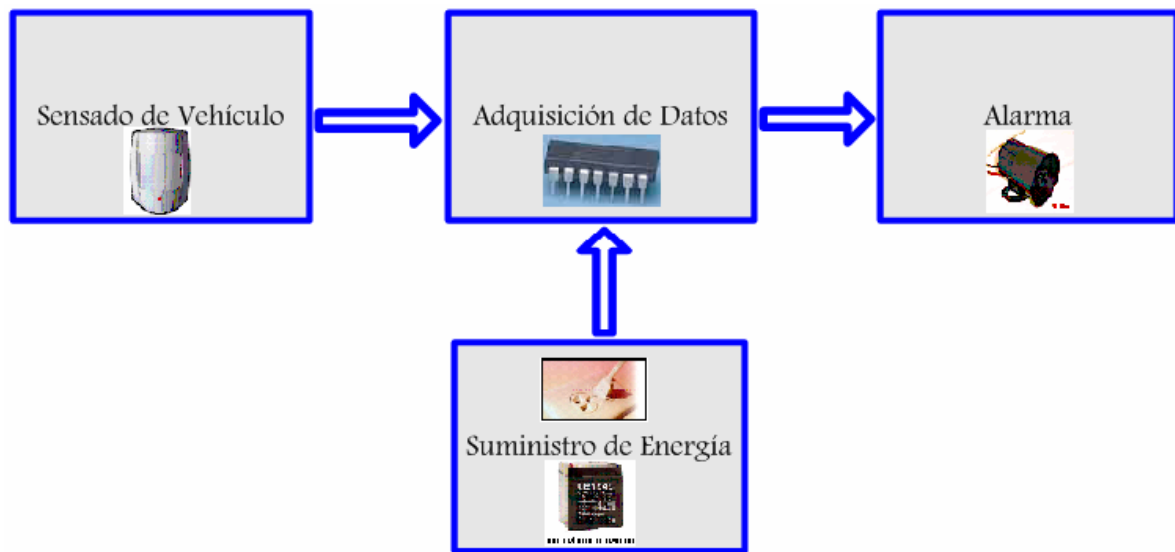


Figura. 3.14. Diagrama de Bloques del subsistema de Adquisición de Datos

Sensado de Vehículo: En este bloque se realiza la detección del vehículo el mismo emite una señal al bloque de Adquisición de Datos para luego ser evaluado como cliente o no cliente.

Suministro de Energía: Es el que alimenta a todo el bloque de Adquisición de Datos es decir al bloque de Sensado de Vehículo y Alarma. Además tiene un sistema de emergencia ante fallos en la red pública, haciendo que el mismo no se vea afectado ante cualquier falla en la red por más de 24 horas. Haciéndolo más robusto y confiable.

Alarma: Esta es una acción que el sistema ejecuta cuando ha verificado que el vehículo que ha ingresado al sistema no es cliente del mismo.

Adquisición de Datos: Este es el cerebro central del sistema el cual recibe y procesa las señales que entran hacia a él, tomando además acciones ante eventos de alarma.

Sensado de Vehículo

Es un detector que consta de un transmisor y un receptor tipo reflex entre los que se establece un haz (no visible) de infrarrojos, la interrupción de este haz provoca la activación de un tiempo en el cual se espera que se emita el código del usuario.

El haz de infrarrojos no es totalmente lineal, sino que tiene una cierta dispersión, por lo que es conveniente instalar barreras compuestas de transmisor-receptor uno a uno. La distancia máxima alcanzada es de 15 m.

Suministro de Energía

La alimentación es tomada de la red pública (110 VAC), por el circuito mostrado en la figura 3.15, y acondicionada para entregar 5 VDC para la etapa de control y 12 VDC para la etapa de potencia (manejo de la sirena), además en caso de fallas cuenta con una batería de respaldo para la alimentación del sistema por más de 24 horas.

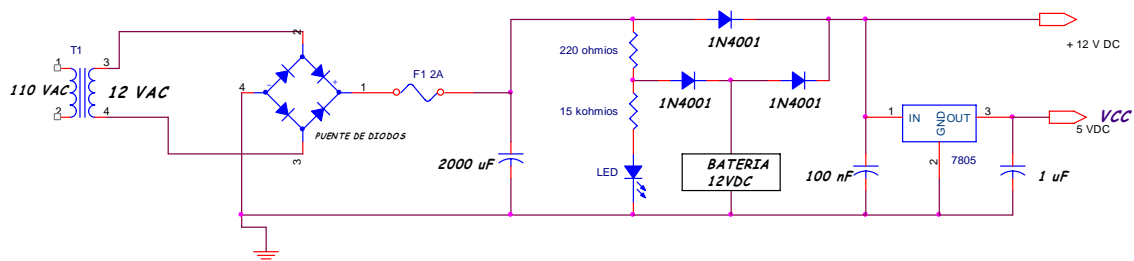


Figura 3.15. Circuito de alimentación del bloque de Adquisición de Datos.

Alarma

Para la identificación audible de una alarma se utiliza una sirena que recibe una señal de activación del bloque de Adquisición de Datos, específicamente del microcontrolador.

Para controlar la sirena con un pin del microcontrolador se lo hace a través de un optoacoplador y un transistor Darlington, para lo que se necesita identificar las características eléctricas de estos componentes, las mismas que se muestran en las tablas 3.3, 3.4 y 3.5.

PIC 16F627			SIRENA		
V_{OH}	3.8	V	V_L	12	V
V_{OL}	0.6				
I_{OH}	3	MA	I_L	1.7	A
I_{OL}	8.5				

Tabla 3.3. Características eléctricas del PIC 16F627 y de la SIRENA.

Los dispositivos elegidos son el optoacoplador 4N25 y el TBJ-NPN TIP 122. Las características principales de estos componentes se presentan en las tablas 3.4. y 3.5.

OPTO-ACOPLADOR (4N25)		
$P_{D\ max}$	250	MW
$V_{CE\ max}$	30	V
$V_{CE\ saturación}$	0.5	
$I_{C\ max}$	150	MA
$V_{D\ max}$	1.5	V
$I_{D\ min}$	5	Ma
β_{MIN}	20	%

Tabla 3.4. Características eléctricas del opto-acoplador 4N25

TBJ DARLINGTON-NPN TIP122		
$P_{D \max}$	2	W
$V_{BE \text{ on}}$	2.5	V
$V_{CE \max}$	100	
$V_{CE \text{ saturación}}$	2	
$I_{C \max}$	5	A
β_{MIN}	1000	-

Tabla 3.5. Características eléctricas del TBJ DARLINGTON-NPN TIP122.

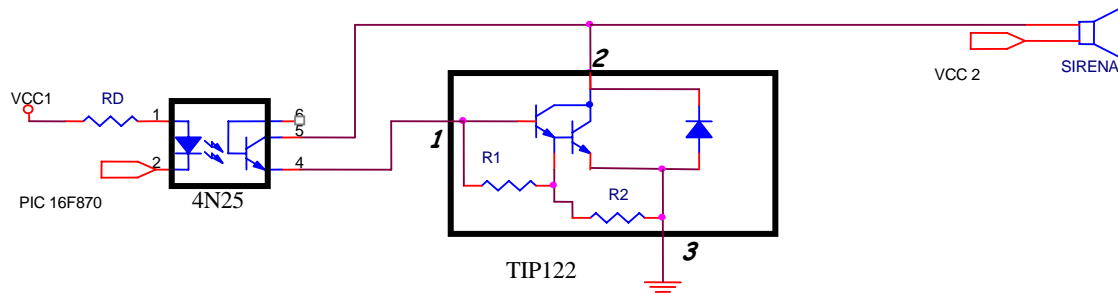


Figura 3.16. Circuito de control de la sirena.

Adquisición de Datos

Esta etapa constituye el cerebro del Sistema de Control de Peaje, cuenta con los PICs 16F870 y 16F627 programados, para recibir y enviar señales.

La configuración de los pines del PIC 16F870 y del PIC 16F627 se muestra la figura 3.17, y 3.18 respectivamente. El circuito completo del Sistema de Control de Peaje, se muestra en el Anexo (Esquemático del Circuito de Adquisición de Datos).

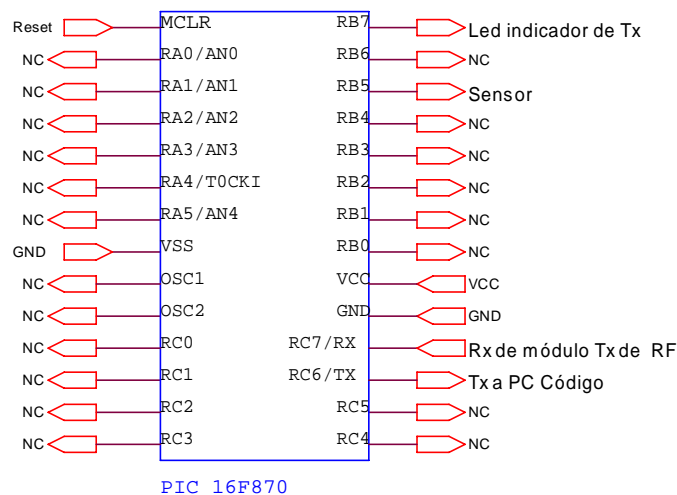


Figura 3.17. Esquema de la configuración de los pines del PIC 16F870

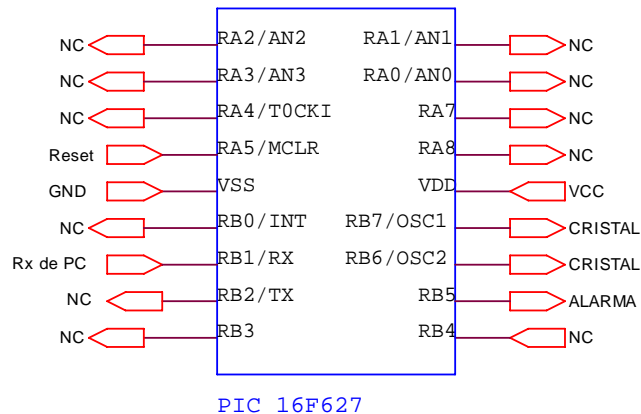


Figura 3.18. Esquema de la configuración de los pines del PIC 16F627

3.2.2.2 Diseño del Software del subsistema de Adquisición de datos

Para programar el microcontrolador y desarrollar la aplicación en el lenguaje de programación Basic, es necesario tener un editor de textos, el compilador PicBasic y al menos un sistema de desarrollo en donde el programa pueda ser verificado. La mejor solución, es usar algún editor especializado en la escritura de código de programas como Microcode Studio.

En la figura 3.19 se muestra la interface gráfica de Microcode Studio en el cual se escribe el código del programa, éste a su vez puede corregir errores de sintaxis, ordena visualmente las subrutinas. El Microcode queda enlazado con el PicBasic y el MPLAB IDE,

de manera que una vez que se termina el programa, se compila y éste genera el archivo *.HEX, los programas se guardan en formato Picbasic *.BAS.

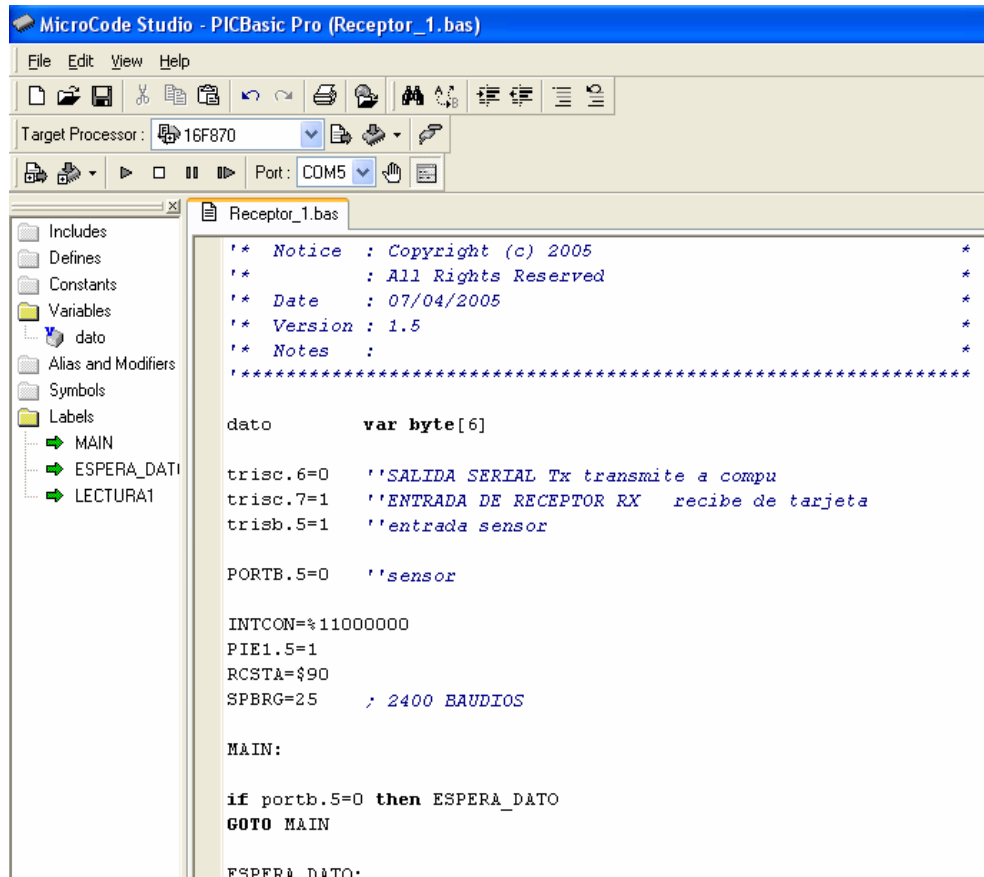


Figura 3.19. Interfaz gráfica del Microcode Studio.

El software utilizado para programar el microcontrolador es el MPLAB IDE, el mismo que necesita hardware extra (PICSTAR Plus de la Microchip), y el archivo con extensión .HEX, para poder grabar en el dispositivo.

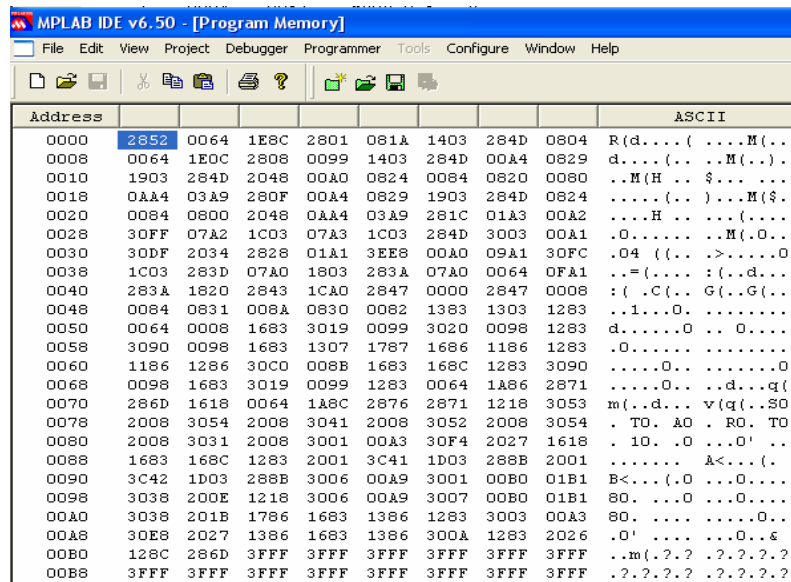


Figura 3.20. Interfaz gráfica del MPLAB IDE V.6.50.

Diagramas de Flujo de los programas para la Adquisición de Datos

Como se mencionó anteriormente utilizamos dos PICs de la Microchip 16F870 y el 16F627 los cuales obedecen a los siguientes diagramas de flujo (figuras 3.21, 3.22)

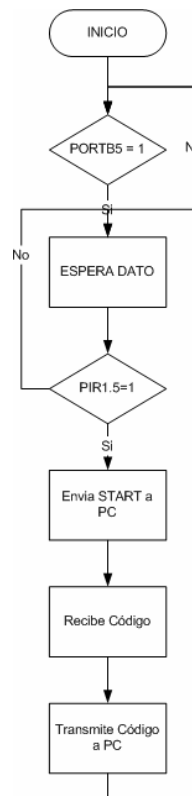


Figura. 3.21. Diagrama de flujo del PIC 16f870

El PIC 16F870 espera recibir la señal del sensor en el pin PORTB.5 el mismo que hace que este vaya a un lazo en el que espera que se active la bandera por interrupción en la recepción USART del PIC luego de la cual transmite hacia el PC una cadena de caracteres (“START1”) la que indica al PC un retardo de tiempo máximo que debe esperar para que se le envíe el código del usuario luego del cual se ejecutan dos acciones.

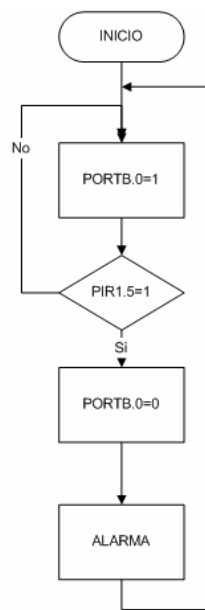


Figura. 3.22. Diagrama de flujo del PIC 16f627

Si se envió el código del usuario este descuenta automáticamente de la base de datos; caso contrario este activará una alarma a través del PIC 16F627 cuando el PC verifica que expidió el tiempo de espera y no llegó el código del usuario al sistema. El PIC 16F627 recibe un carácter (“R”) luego, que el mismo es detectado por interrupción en la recepción del USART se activará una alarma por un tiempo programado.

Reporte Serial

El HMI de la estación central de monitoreo, siempre está verificando la llegada de datos al puerto serial desde el Sistema, los mismos que son enviados al ocurrir los siguientes eventos:

- Descuento
- Alarma

Estructura del Programa Receptor1.bas

El programa Receptor1.bas es el que ejecuta el PIC 16F870 del Sistema, y está estructurado de la siguiente manera:

- ❑ Asignación de variables utilizadas en el programa.
- ❑ Configuración de pines del PIC 16F870.
- ❑ Inicialización de variables.
- ❑ Rutinas de programación basadas en los algoritmos explicados anteriormente en este capítulo.

Estructura del Programa Receptor2.asm

El programa Receptor2.asm es el que ejecuta el PIC 16F627 del Sistema, y está estructurado de la siguiente manera:

- ❑ Configuración e inicialización de registros para el PIC 16F627.
- ❑ Rutinas de programación basadas en los algoritmos explicados anteriormente en este capítulo.

Instrucciones para la Comunicación Serial

La recepción de los datos que llegan desde el subsistema de transmisión de datos hacia el subsistema de adquisición de datos se realiza utilizando el comando hserin, el cual necesita que se indique la variable donde se almacenará el dato a recibir. Figura 3.23.

Ejemplo:

HSERIN[str dato\6]

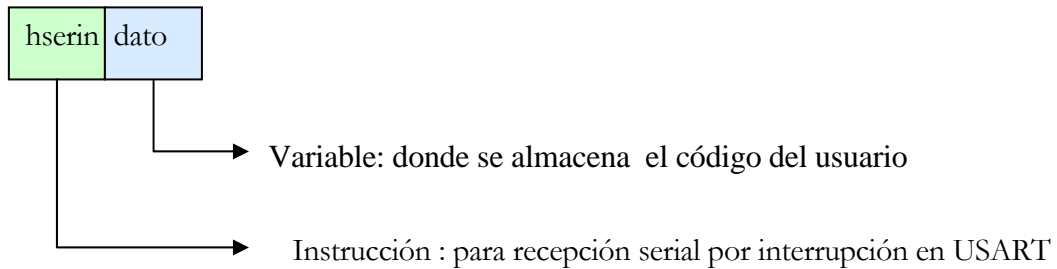


Figura 3.23. Instrucción de recepción serial para la programación del PIC.

La transmisión serial de la trama se lo hace por medio de la instrucción hserout, la que necesita que se especifique el dato o cadena de caracteres a transmitir Figura 3.24

Ejemplo:

HSEROUT ["START1"]

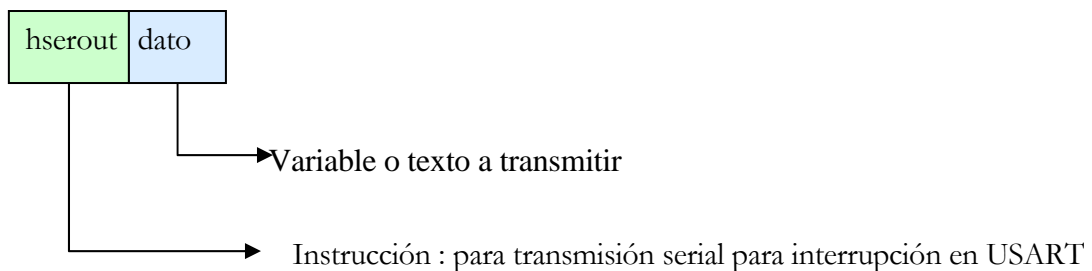


Figura 3.24. Instrucción de transmisión serial para la programación del PIC.

3.2.3 Centro de Control

El subsistema, Centro de Control es una parte medular puesto que permite la comunicación de los anteriores subsistemas con el operario, es decir, es el HMI del conjunto del bloques antes estudiados.

El software del Centro de Control se desarrolló en Microsoft Visual Basic 6.0 y está compuesto por dos partes una de las cuales se encarga de monitorear el sistema (MONITOR) y la otra de la administración del sistema (SIADPE), los cuales están enlazados a través de una base de datos desarrollada en Microsoft Access llamada Peaje.

MONITOR este programa se encarga de monitorear al sistema gracias a toda la información que recibe de él. Esta información es interpretada, por el software el mismo que actúa sobre la eventualidades para luego registrar cada uno de sus eventos en la base de datos del sistema.

SIADPE es el programa base que se encarga de las funciones administrativas de la Estación Central, almacenando, recuperando y actualizando información en la base de datos.

3.2.3.1 Diseño del software de monitoreo (MONITOR)

Este programa establece una comunicación serial, con el subsistema de Adquisición de datos el cual constantemente envía información al mismo, luego este ejecuta acciones como por ejemplo la acción *alarma* en caso de no ser cliente y *descuento* en caso de ser cliente del sistema, a su vez actualiza la base de datos cada que ocurre uno de estos eventos.

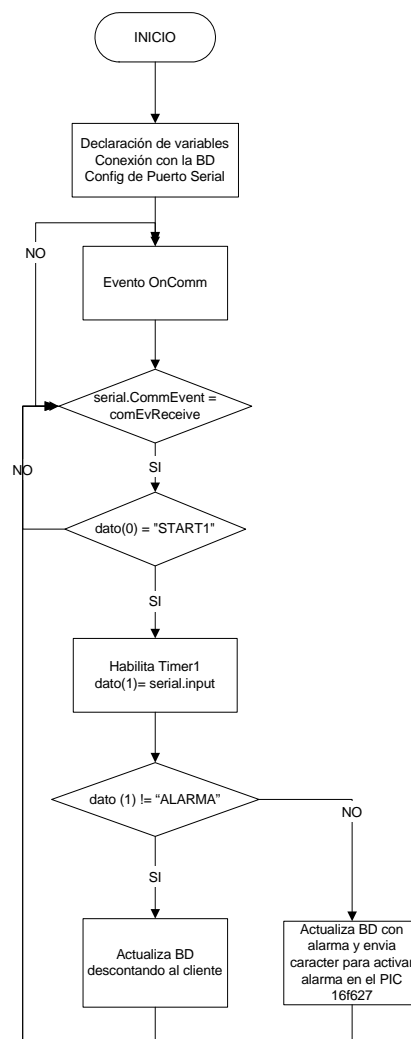


Figura. 3.25. Diagrama de flujo del Software MONITOR

Este software está diseñado para ser cargado una vez que se enciende el equipo donde está instalado, es fundamental esta característica puesto que se sabe que un sistema de peaje funciona las 24 horas del día. A continuación se muestra el diagrama de flujo del software MONITOR.

El programa MONITOR, constantemente verifica la llegada de datos al buffer de entrada, si hay un evento almacena los datos en las variables desde las cuales nos permiten interactuar con la base de datos para consultas y actualizaciones de la misma junto con la hora y fecha del sistema, luego muestra en la pantalla un reporte visual, como se puede observar en la figura 3.26

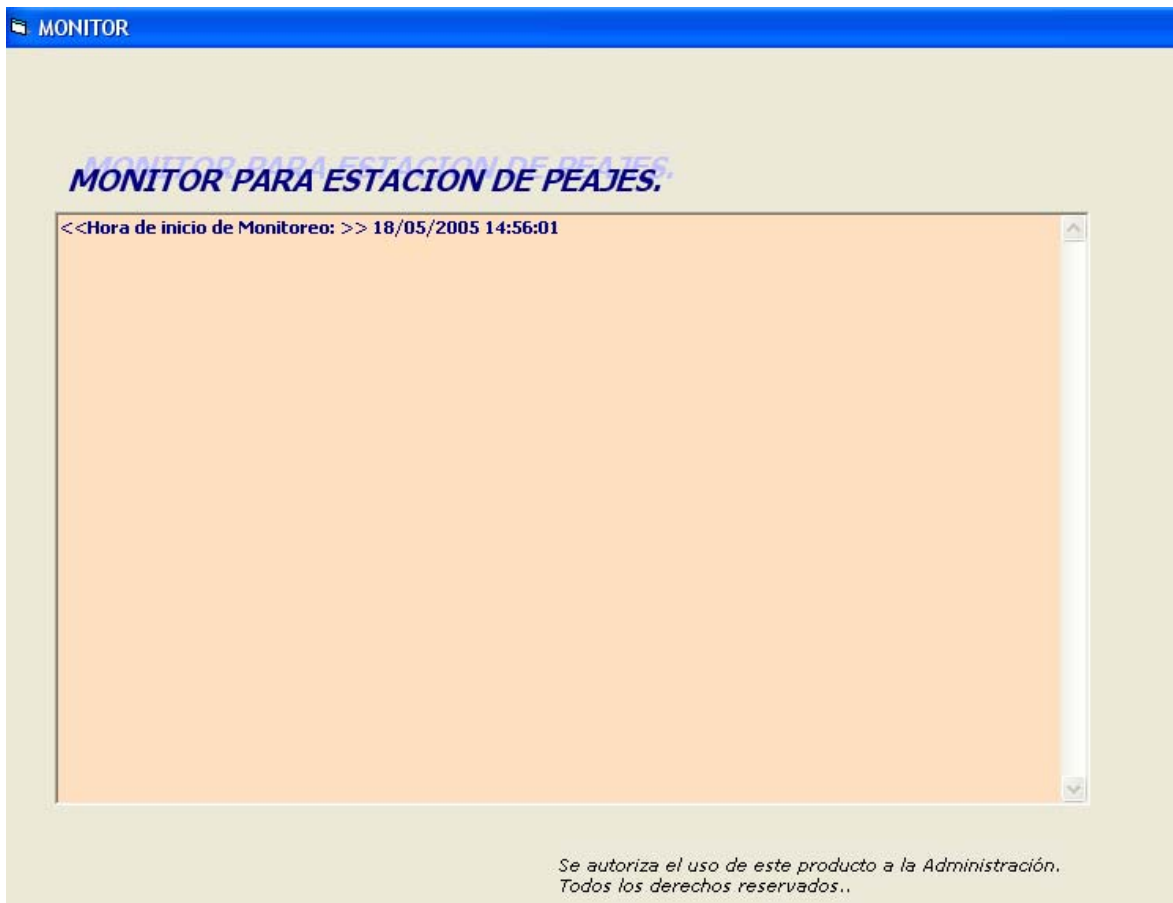


Figura. 3.26. Interface del software de monitoreo

3.2.3.2 Diseño del software de administración para la estación central (SIADPE)

Como se explicó anteriormente el software del Centro de Control consta de dos programas enlazados ADMINISTRACIÓN y MONITOREO; ahora, haremos referencia al software de ADMINISTRACIÓN, el cual actualiza y recupera información de la base de datos (Peaje).

Diseño de la bases de datos (Peaje)

Es sencillo diseñar una base de datos, pero estructurarla de la mejor manera, es la clave para iniciar bien la implementación de un sistema.

Es de destacar la importancia de un buen diseño. Un diseño apresurado o simplemente bosquejado puede mostrarse inservible o muy mejorable cuando la aplicación ya está parcialmente codificada, o el administrador de la base de datos ya tiene organizados el mantenimiento y el control de acceso a los datos.

Las fases del diseño de una base de datos son las siguientes:

1. Descripción en lenguaje natural.
2. Diagrama Entidad-Relación (E-R). También conocido como "diagrama de Chen". Estos diagramas modelizan el problema mediante entidades asociadas por relaciones. Adoptan la forma de gráficos donde los datos se relacionan mediante flechas. El diagrama E-R no depende del modelo de datos.
3. Elección del modelo de datos (usualmente el relacional)
4. Conversión del diagrama E-R al modelo relacional (tablas)
5. Normalización (eliminar diversos defectos de diseño).
6. Optimización (según criterios de almacenamiento interno, como el espacio en disco, redundancia de la información y el tiempo medio de acceso).

Modelo Entidad-Relación.- El modelo entidad-relación aporta una herramienta de modelado para representar las entidades, relaciones y atributos, mediante éstos, el esquema conceptual abstracto puede ser mostrado gráficamente y mantener una independencia conceptual con respecto a la implementación propiamente dicha.

En realidad, podemos hacer que los diagramas sean un reflejo fiel de las relaciones, interrelaciones y atributos del modelo relacional de datos o podemos englobar diversas relaciones en una sola entidad o conjunto de propiedades.

El modelo entidad-relación ofrece la posibilidad de enlazar dos o más tablas existentes en una base de datos. De hecho, ésta es la característica que dota de mayor potencia al modelo.

Entidad .- Una entidad puede ser un objeto físico (una persona, un auto, una casa o un empleado) o un objeto conceptual (una compañía, un puesto de trabajo o un curso universitario). En nuestro proyecto podemos definir siete entidades: Cliente, Vehículo, Banco, Cuenta, Estación, Provincia, Evento.

Atributos.- Cada entidad tiene propiedades específicas, llamadas atributos, que la describen. Por ejemplo, la entidad "Usuarios" posee los atributos Código de Usuario, Cédula, Nombre, Dirección, Teléfono de la Casa, Teléfono Móvil, E-mail. Una entidad particular tiene un valor para cada uno de sus atributos.

Relación.- Una relación se puede definir como una asociación entre entidades. Por ejemplo, la entidad "Cliente" puede estar relacionada con la entidad "Evento" por medio de la relación "descuento". Una relación también puede tener atributos.

Herramientas de Base de Datos

Microsoft Access permite administrar toda la información desde un único archivo de base de datos. Dentro del archivo, puede dividir los datos en contenedores de almacenamiento independientes denominados tablas; puede ver, agregar y actualizar datos de la tabla por medio de formularios en pantalla; puede asimismo buscar y recuperar sólo los datos que desee por medio de consultas; y puede analizar o imprimir datos con un diseño específico por medio de informes.

Para almacenar los datos, se crea una tabla para cada tipo de información de la que hace el seguimiento. Las tablas organizan datos en columnas (denominadas campos) y filas

(denominadas registros). Un campo común relaciona dos tablas de manera que Microsoft Access puede combinar los datos de las dos tablas para ver, modificar o imprimir. La utilización de una tabla diferente para cada tema significa que se almacenan los datos sólo una vez, lo cual hace aumentar la eficacia de la base de datos, y reduce errores de entrada de datos.

Para este proyecto se ha creado diez tablas: Cliente, Vehículo, Banco, Cuenta, Estación, Provincia, Evento, Tipo evento, Tipo vehículo y Tipo cuenta, como se muestra en la figura 3.27

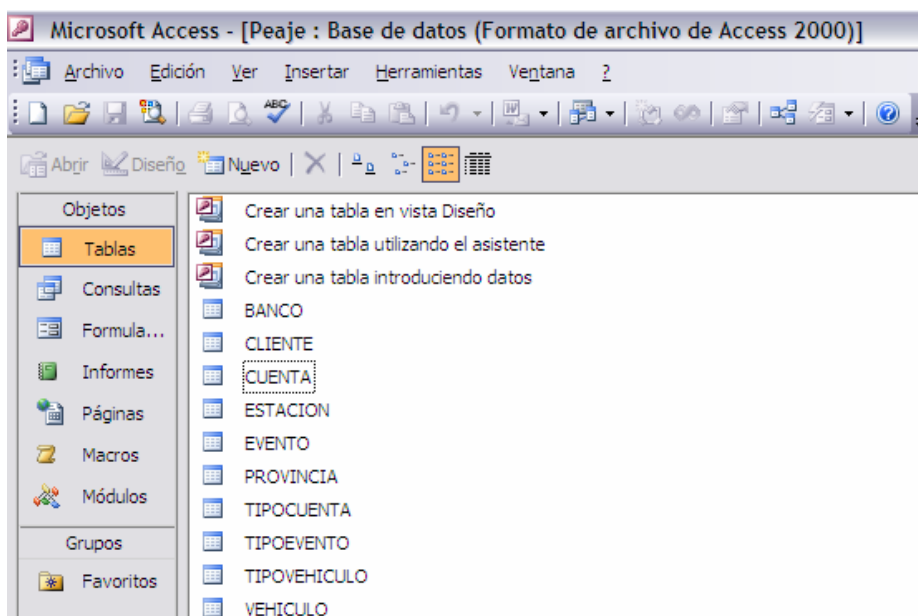


Figura 3.27. Tablas creadas para cada tipo de información.

La configuración del tipo de datos de un campo define el tipo de valores que puede escribir en el mismo. Por ejemplo, no puede escribir texto en un campo autonumérico. Si se elige un tipo de datos adecuado, puede asegurarse de que los datos se escriben de la forma correcta para ordenar, realizar cálculos y otras operaciones.

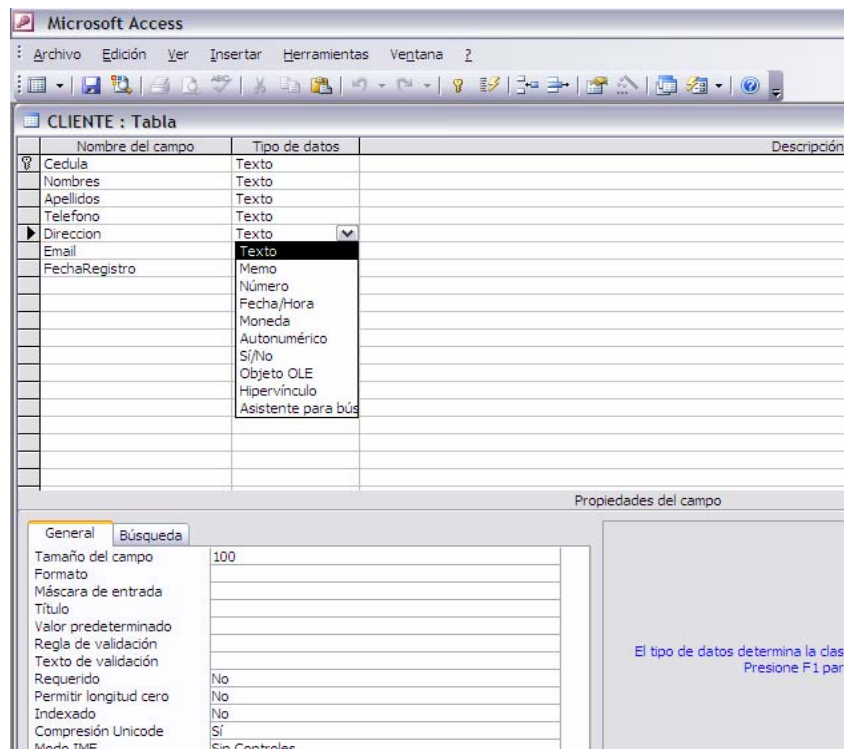


Figura 3.28. Configuración del tipo de datos.

Se utiliza una etiqueta única, denominada clave principal, para identificar los registros de la tabla. De la misma manera que la cédula de identidad de una persona identifica a la misma, la clave principal identifica un registro de manera única.

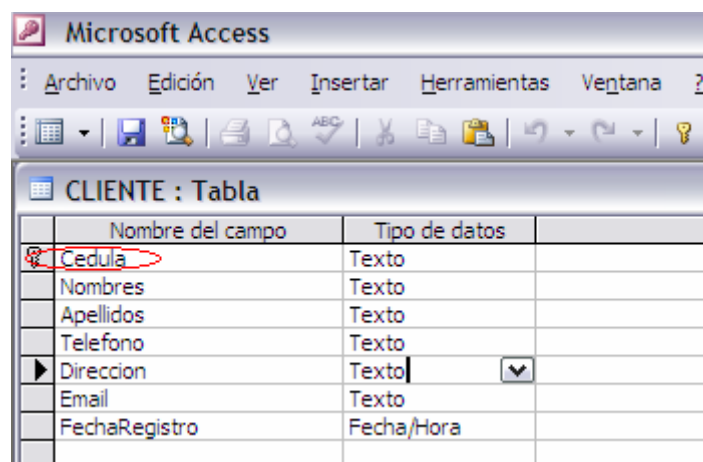


Figura 3.29. Clave principal.

Para reunir los datos de varias tablas en una consulta, formulario, informe o página de acceso a datos, se define relaciones entre las tablas.

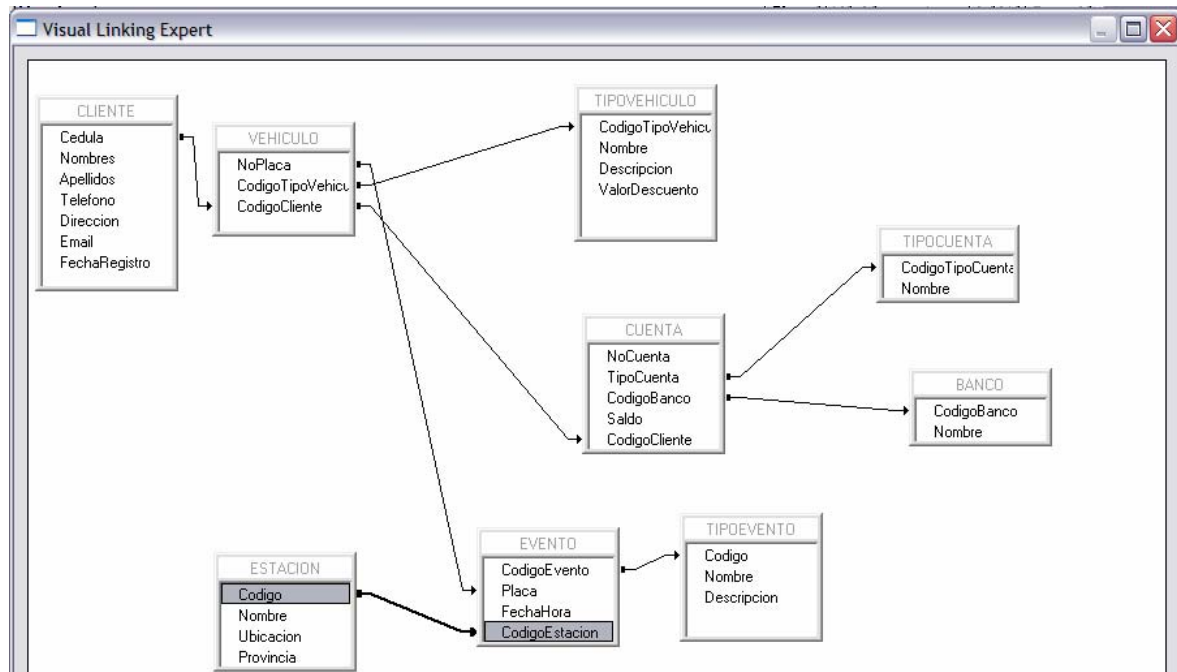


Figura 3.30. Relaciones entre tablas para reportes.

Para buscar y recuperar tan sólo los datos que cumplen las condiciones especificadas, incluyendo datos de varias tablas, puede crear una consulta, la misma que permite también actualizar o eliminar varios registros al mismo tiempo, y realizar cálculos predefinidos o personalizados en los datos.

Reporte de Clientes

REPORTE DE CLIENTES

Fecha de emisión: 22/04/2005

Cedula	Nombres	Apellidos	Telefono	Direccion	Fecha de Registro	No de Cuenta	Fecha de Servicio	Placa
1803394871	Mario	Erazo		Av. Mariscal Sucre	17/03/2005	1234567890	04/04/2005	SRA.148
							04/04/2005	SRA.148
							04/04/2005	SRA.148
							04/04/2005	SRA.148
							04/04/2005	SRA.148
							04/04/2005	SRA.148
							04/04/2005	SRA.148
							04/04/2005	SRA.148
Total de uso del servicio:								7

Figura 3.31. Presentación de contenido de reportes.

Controles, Conexión y Acceso a Bases de Datos en Visual Basic 6.0

Control

ADO (ActiveX Data Objects - Objetos de datos ActiveX). Tecnología multilenguaje para el acceso a datos, cuyo modelo de objetos incorpora objetos de conexión de datos, comandos de datos, objetos Recordset y colecciones que los incluyan. El modelo de objetos ADO proporciona un conjunto de objetos, propiedades y métodos fáciles de usar para la creación de secuencias de comandos que proporcionen acceso a la información de bases de datos. Más adelante se indican algunas de las sentencias que se utilizaron para el desarrollo del software SIADPE para el manejo de base de datos en Visual Basic 6.0

Métodos del Control ADO.CONNECTION

Command: Representa una sentencia SQL.

Connect: Establece o devuelve el nombre de la base de datos desde la que se ejecutan las operaciones de consulta y actualización.

Connection: Permite el intercambio de los datos. Especifica un origen de datos existente con el que se establece la conexión. Recupera datos de una o más tablas. Un objeto Connection representa una conexión abierta con un origen de datos.

Error: Representa un error en una conexión. Nombre de archivo no válido.

Execute: Ejecuta una consulta de acciones o una instrucción SQL.

Recordset (Objeto de manipulación de la Base de Datos): Permite la exploración de los datos y la manipulación de los mismos. Es un objeto de acceso a datos que contiene una colección de registros tomados, bien de una tabla, bien de un conjunto de tablas (a través de una consulta). Al utilizar objetos de acceso a datos, casi toda la interacción con los datos se produce a través de objetos Recordset. Todos los objetos Recordset están formados por registros (filas) y campos (columnas).

Forma de conexión a Bases de datos

OLE DB: Es una estrategia más extendida y eficiente para el acceso a datos que **ODBC** (Open Database Connectivity = Conectividad Abierta con Bases de Datos), ya que permite el acceso a más tipos de datos y está basada en el Modelo de objetos de Componente de Microsoft (Component Object Model-COM). Arquitectura de bases de datos que proporciona integración universal de datos a través de la red corporativa, desde los grandes sistemas a las estaciones de trabajo, sin importar el tipo de datos.

Lenguaje de manipulación de base de datos

SQL: Para consulta y selección de registros se utiliza el lenguaje estándar para bases de datos relacionales SQL (Structured Query Language = Lenguaje de Consulta Estructurado), el mismo que contiene un conjunto de herramientas que optimizan las consultas.

Instrucciones del lenguaje

Select: Especifica los campos, constantes y expresiones que se mostrarán en el resultado de la consulta.

Update: Guarda el contenido del búffer de copia en un objeto Recordset de tipo hoja de respuestas dinámica o tabla especificada. Es decir, ingresa en la Base de Datos el contenido del registro que estaba en la memoria.

Where: Indica a Visual Basic que incluya únicamente ciertos registros en el resultado de la consulta. Where es necesario para recuperar datos de varias tablas.

Ejemplo 1:

Contenido del módulo del proyecto. Declaración de variables globales, conexión y apertura de la base de datos.

Dim strigconexion As String

```
strigconexion = "DRIVER=Driver do Microsoft Access (*.mdb);UID = admin;FIL=MS  
Access;DefaultDir=C:\Peaje final;DBQ=C:\Peaje final\Peaje.mdb"
```

```
Set conecbd = New ADODB.Connection
```

```
conecbd.Open strigconexion
```

Ejemplo 2:

Ingreso de información a la base de datos.

```
sql = "select * from cliente"
```

```
rs.Open sql, conecbd, adOpenStatic, adLockOptimistic
```

With rs

```
.AddNew
```

```
!cedula = cedula
```

```
!nombres = nombres
```

```
!apellidos = apellidos
```

```
!telefono = telefono
```

```
!direccion = direccion
```

```
!email = email
```

```
!fecharegistro = fecharegistro
```

```
.Update
```

End With

Interface Gráfica para Administración del Sistema

En la figura 3.32, se puede apreciar la estructura que tienen las diferentes pantallas de la interface gráfica.

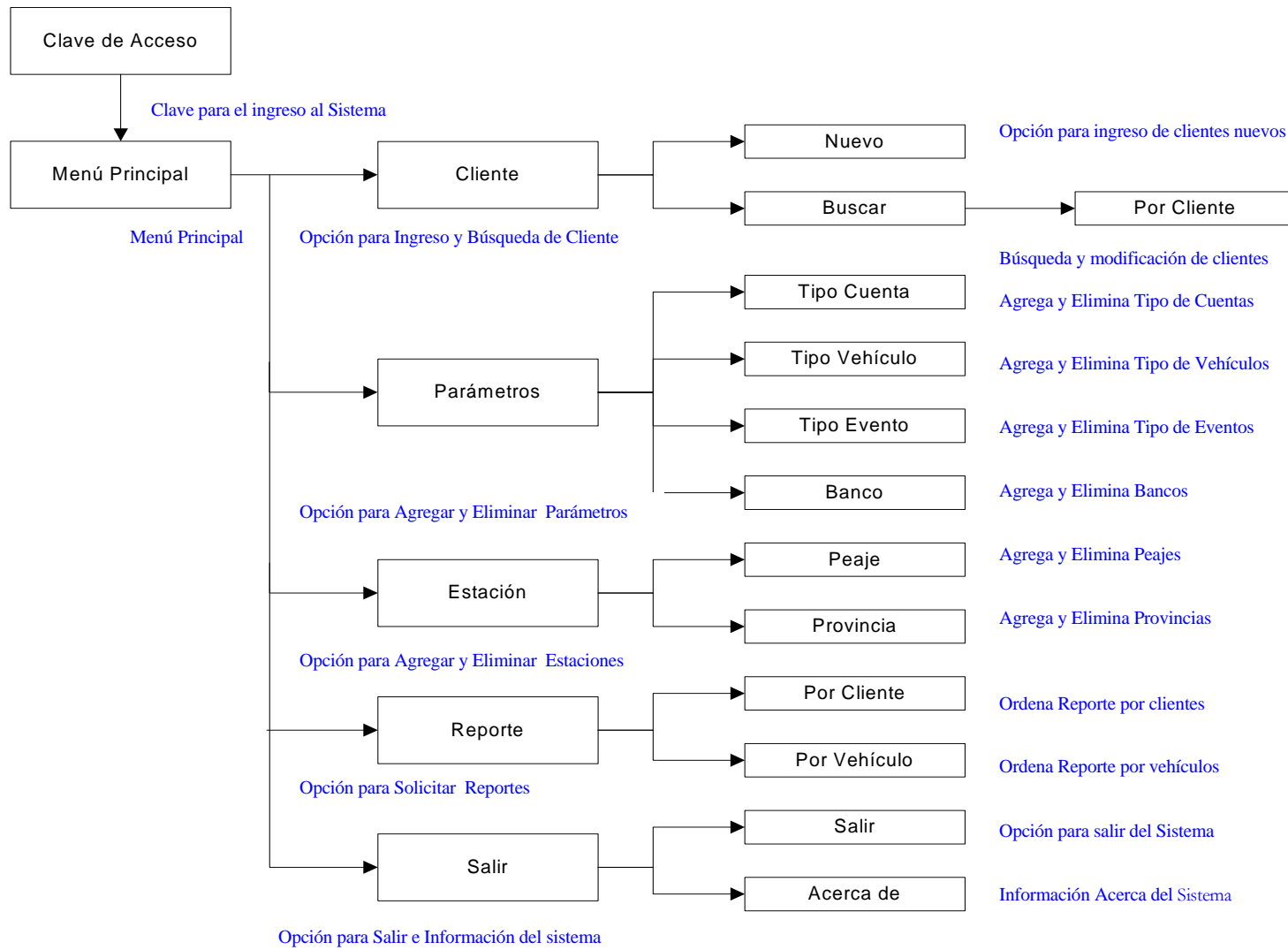


Figura. 3.32. Diagrama de bloques de la interfase grafica para la administración del sistema

CAPITULO 4

INTERGRACIÓN Y PRUEBAS

4.1 INTEGRACIÓN

4.1.1 Suministro de Energía

Es la fuente de energía del sistema, que se alimenta con un voltaje de entrada (110Vac) y entrega dos voltajes a su salida (12Vdc, 5Vdc). Además cuenta con un sistema de respaldo de energía a través de una batería recargable de 12V. 5 Amp.Hr.



Figura. 4.1. Suministro de Energía

4.1.2 Sensores

El sistema funciona con dos tipos de sensores cuya característica se describen a continuación:

- ❑ Marca: CAME sensor de fototransistores (transmisor / receptor)
- ❑ Vin: 12/24 Vdc
- ❑ Salida tipo relé
- ❑ Lógica negativa
- ❑ Distancia de alcance máxima: 15mts.
- ❑ Figura 4.2



Figura. 4.2. Sensor transmisor receptor

Para la maqueta se utilizó el siguiente sensor:

- ❑ Marca: OMRON sensor de infrarrojos (transmisor/reflector)
- ❑ Vin: 10 a 30 Vdc.
- ❑ Salida tipo transistor
- ❑ Logica positiva
- ❑ Distancia de alcance máxima: 0.30 mts
- ❑ Figura 4.3



Figura. 4.3. Sensor de infrarrojos tipo reflex

4.1.3 Transmisor Receptor de RF

Módulos transmisor receptor de RF circuitería diseñada para trabajar en la frecuencia de 433.92 MHz. con una modulación ASK a una distancia máxima de alcance entre módulos receptor transmisor de 50 mts.



Figura. 4.4. Módulos transmisor receptor de RF

4.1.4 Tarjeta de Adquisición de Datos

Utiliza el estándar RS232 para la comunicación con la PC, se alimenta con dos voltajes 5Vdc y 12Vdc puesto que desde ella se alimentan tanto el módulo de recepción, los sensores y la sirena .

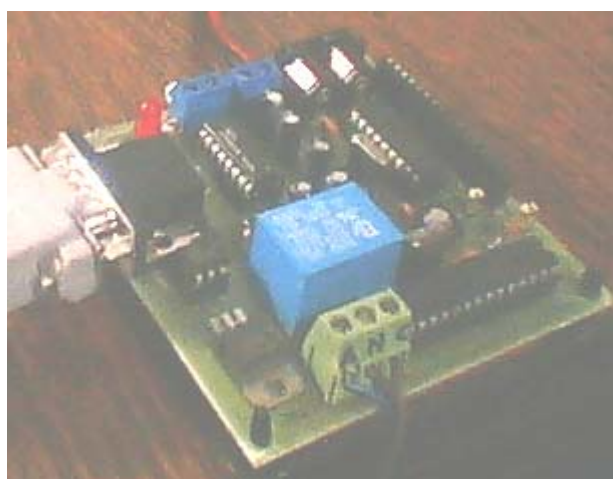


Figura. 4.5. Tarjeta de adquisición de los datos

4.1.5 Alarma

Se utilizó una sirena de 12Vdc para anunciar las alarmas que ocurren en el sistema.



Figura. 4.6. Sirena 12 Vdc

4.1.6 PC

Desde aquí se puede monitorear y administrar el sistema gracias a dos programas elaborados en Visual Basic. 6.0

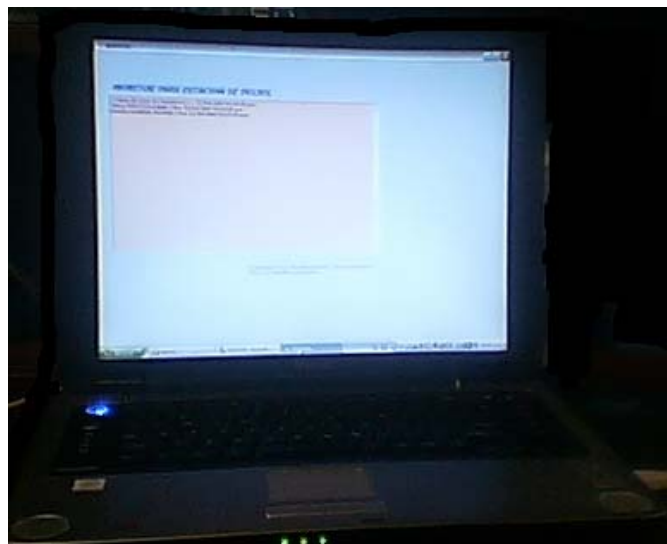


Figura. 4.7. PC del sistema

En conclusión, todos estos elementos unidos conforman el sistema electrónico para peajes el mismo que fue sometido a ciertas pruebas para probar su operabilidad, y funcionamiento.

4.2 PRUEBAS

4.2.1 Subsistema de Transmisión de Datos

Formato para el envío de datos

El formato de la trama de datos es el mostrado en la figura 4.1. Este formato de datos es igual al del estándar RS232, el mismo que está compuesto por un bit de inicio (BI), 8 bits de datos (DATOS) y un bit de parada (BP).

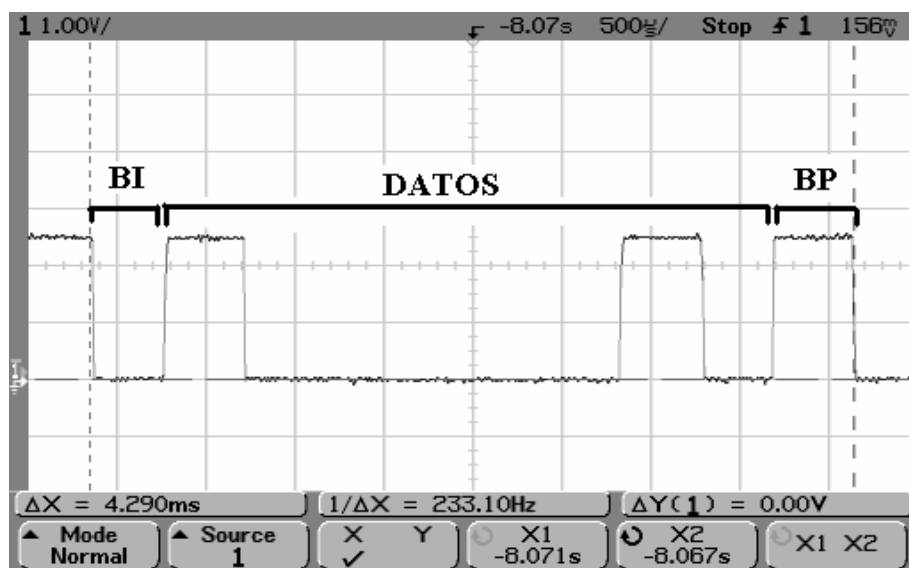


Figura. 4.8. Formato del envío de datos

La transmisión de los datos se realiza a una tasa de transmisión de 2.400 baudios, que es igual a 416 μseg para el tiempo de bit. Esta es una de las pruebas que se realizaron para constatar si los datos grabados en el módulo de transmisión se emitían a esa velocidad. Para lo cual conectamos la salida del módulo de recepción a la sonda del osciloscopio.

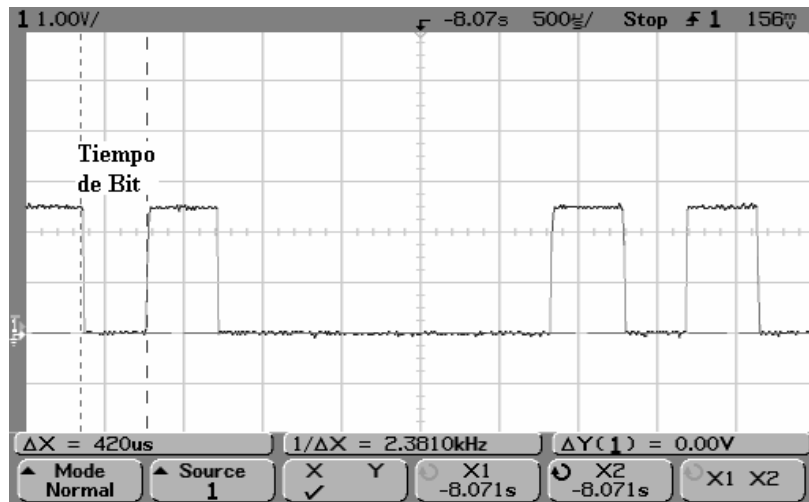


Figura. 4.9. Tiempo de bit de los datos

Después de observar la señal a la salida del módulo de recepción con la ayuda del osciloscopio se obtuvo esta señal (figura 4.9) la que nos indica que el tiempo de bit es de 420 μ seg y su frecuencia es de 2.381 Hz que por estar modulado en banda base da como resultado una tasa de transmisión de 2.381 baudios. Todos estos resultados dan un error de 0.8%, el cual es un porcentaje bajo que no afecta la decisión de bit en el receptor.

Luego de esta prueba se verificó que el código grabado en el módulo de transmisión se emitía correctamente, para lo cual grabamos un código en el transmisor (PRA148), el mismo que indica la placa del vehículo.

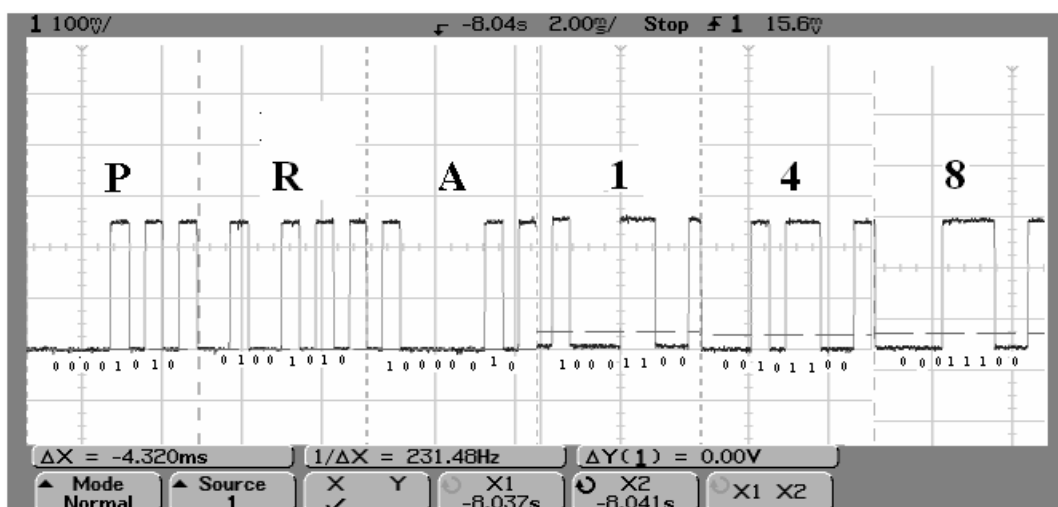


Figura. 4.10. Transmisión de la placa de vehículo

4.2.2 Subsistema de Adquisición de Datos

En este subsistema se realizaron pruebas de respuesta para una posterior calibración de los tiempos de programación tanto con el módulo de transmisión como con la alarma. Hay que tomar en cuenta que el sistema en conjunto trabaja a velocidad considerable puesto que los vehículos que acceden al servicio circulan dentro de él a una velocidad recomendada de 40 Km/Hr. El diseño de la maqueta muestra claramente la velocidad, señalización e infraestructura recomendada para el funcionamiento óptimo del sistema.

Para esto se activó a distintos tiempos la señal del sensor, la misma que propicia todo el proceso de identificación, reconocimiento y actuación ante los distintos eventos. Así se obtuvo resultados tales como:

El sistema atiende a distintos usuarios con intervalos mínimos de 0.60 seg en caso de ser clientes del sistema y 2 seg en caso de no ser clientes del sistema, este retardo de aproximadamente el triple del intervalo se produce por el tiempo de activación de la alarma. Estos intervalos se los puede conseguir con la ubicación de un rompe velocidad al ingreso del área de cobertura del sistema con lo cual se asegura estos intervalos entre usuarios.

4.2.3 Subsistema Centro de Control

Para la realización de pruebas en este subsistema fue necesario integrar todo el sistema electrónico de peaje, para lo cual se colocaron los sensores a 50 mts de la tarjeta de adquisición de datos la misma que fue colocada a 5 mts del centro de monitoreo. Una vez integrado todo el sistema se hicieron las pruebas con dos vehículos, con las siguientes alternativas:

- ❑ Vehículo 1: cliente, vehículo 2: no cliente
- ❑ Vehículo 1: no cliente, vehículo 2: no cliente
- ❑ Vehículo 1: cliente, vehículo 2: cliente

Los resultados y respuestas obtenidos del sistema, fueron los esperados, garantizando de esta manera el funcionamiento únicamente para un carril.

CAPÍTULO 5

ANÁLISIS ECONÓMICO

5.1 Listado de Materiales

Se enlistarán a continuación todos los materiales utilizados para la elaboración de este proyecto:

5.1.1 Materiales Electrónicos y Eléctricos

- 1 - Batería recargable de 12V 5Amp.Hr
- 1 - Bornera de 3 pines
- 5 - Bornera de 2 pines
- 1 - Caja para proyectos pequeña
- 1 - Caja para proyectos mediana
- 1 - Caja galvanizada 20*20*10
- 1 - Condensador polarizado de 2200 μ F / 50V
- 5 - Condensador polarizado de 10 μ F / 50V
- 1 - Condensador polarizado de 1 μ F / 50V
- 1 - Condensador de 100nF
- 4 - Condensador de 33pF
- 2 - Conector DB9 hembra
- 1 - Conector DB9 macho
- 1 - Conector de 16 pines
- 2 - Cristal de 4MHz.
- 3 - Diodos rectificador 1N4001
- 1 - Fusible de 2A.
- 1 - Juego de Placas con circuito impreso para proyecto

- 1 - Kit de desarrollo para PIC's de RF
- 3 - Led's (2 rojos 1 verde)
- 1 - MAX232
- 1 - Microcontrolador PIC 16F870
- 1 - Microcontrolador PIC 16F627
- 2 - Mts de cable de parlante 2*18
- 2 - Mts de cable sucre 2*18
- 2 - Mts cable multipar 2 pares
- 1 - Optoacoplador 4N25
- 4 - Plug macho para alimentación
- 4 - Plug hembra para alimentación
- 1 - Portafusible para placa
- 8 - Postes para Placa
- 1 - Puente de diodos 1N4001
- 1 - Pulsante
- 1 - Relé 12 VDC
- 1 - Regulador de voltaje LM7805
- 1 - Resistencia de 15K Ω
- 3 - Resistencias de 1K Ω
- 3 - Resistencias de 220 Ω
- 1 - Sensor barrera de infrarrojos
- 1 - Sirena 12 VDC
- 2 - Switch ON-OFF
- 1 - Transformador de voltaje 110V/12V. 3A.
- 1 - Transistor Darlington TIP122
- 1 - Zócalo 28 pines
- 1 - Zócalo 18 pines
- 1 - Zócalo 16 pines
- 1 - Zócalo 6 pines

5.1.2 Materiales para maqueta

- 8 - Planchas de madera de balsa

- 1/4- Galón de Estuco para interior
- 3 - Pega Super Bonder La Brujita
- 3 - Spray de colores
- 1 - Lámina de madera de balsa
- 3 - Acetato A4

5.1.3 Software

- 1- Caja Visual Studio 6.0 (incluye: manual, CD's y licencia para un computador)

5.2 Análisis de Costos

5.2.1 Costos de Materiales Electrónicos y Eléctricos

CANTIDAD	DESCRIPCIÓN	PRECIO UNIT.	PRECIO TOTAL
1	Batería recargable de 12V. 5Amp.Hr	24.67	24.67
1	Bornera de 3 pines	1.00	1.00
5	Bornera de 2 pines	0.60	3.00
1	Caja para proyecto pequeña	2.50	2.50
1	Caja para proyecto mediana	4.80	4.80
1	Caja galvanizada de 20*20*10	8.50	8.50
1	Condensador polarizado de 2200 μ F / 50V	1.00	1.00
5	Condensador polarizado de 10 μ F / 50V	0.08	0.40
1	Condensador polarizado de 1 μ F / 50V	0.08	0.08
1	Condensador de 100nF	0.08	0.08
4	Condensador de 33pF	0.10	0.40
2	Conector DB9 hembra	0.50	1.00
1	Conector DB9 macho	0.80	0.80
1	Conector de 16 pines	1.00	1.00
2	Cristal de 4MHz.	0.65	1.30
3	Diodos rectificador 1N4005	0.05	0.15
1	Fusible de vidrio 2A. 20mm	0.10	0.10
1	Juego de placa con circuito impreso de proyecto	41.66	41.66
1	Kit de desarrollo para PIC's de RF	200.00	200.00
3	Led's (2 rojos 1 verde)	0.15	0.45
1	MAX232	1.75	1.75
1	Microcontrolador PIC 16F870	6.00	6.00
1	Microcontrolador PIC 16F627	6.00	6.00
1	Optoacoplador 4N25	0.50	0.50
4	Plug hembra para alimentación	0.10	0.40
4	Plug macho para alimentación	0.20	0.80
1	Portafusible para placa	0.15	0.15
1	Puente de diodos 1N4001	0.50	0.50
2	Pulsantes	0.40	0.80
1	Regulador de voltaje LM7805	0.68	0.68
1	Relé 12VDC	1.50	1.50

1	Resistencia de 15K Ω	0.04	0.04
3	Resistencia de 1K Ω	0.04	0.12
3	Resistencia de 220 Ω	0.04	0.12
1	Sensor barrera de infrarrojos	42.00	42.00
1	Sirena 12 VDC	5.00	5.00
2	Switch ON-OFF	0.80	1.60
1	Transformador de voltaje 110V/12V. 3A.	3.50	3.50
1	Transistor Darlington TIP122	0.70	0.70
1	Zócalo 28 pines	0.20	0.20
1	Zócalo 18 pines	0.15	0.15
1	Zócalo 16 pines	0.10	0.10
1	Zócalo 6 pines	0.10	0.10
		SUBTOTAL	365.60
		12% IVA	43.87
		TOTAL	409.47

Tabla. 5.1. Costos de Material Electrónico y Eléctrico

5.2.2 Costos de Materiales para Maqueta

CANTIDAD	DESCRIPCIÓN	PRECIO UNIT.	PRECIO TOTAL
7	Planchas de madera de balsa	2.50	17.50
1	Lámina de madera de balsa	0.20	0.20
4	Tira de madera de balsa	0.50	2.00
1/4	Galón de Estuco para interior	1.86	1.86
3	Pega Super Bonder La Brujita	2.98	8.94
3	Spray de colores	1.94	5.82
3	Acetato A4	0.25	0.75
1	Tablero de MDF	7.00	7.00
	Varios	5.00	5.00
		SUBTOTAL	49.07
		12% IVA	5.88
		TOTAL	54.96

Tabla. 5.2. Costos de material de maqueta

5.2.3 Costo de Diseño y Software

CANTIDAD	DESCRIPCIÓN	PRECIO UNIT.	PRECIO TOTAL
1	Caja Visual Studio 6.0	1,723	1,723
	Desarrollo de Software para Monitoreo y Administración	500	500
	Diseño de prototipo electrónico y maqueta	500	500
		SUBTOTAL	2,723
		12% IVA	326.76
		TOTAL	3,049.76

Tabla. 5.3. Costos de Diseño y Software

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Se utilizó los rPIC's como medio de comunicación unidireccional, entre el automóvil y el centro de control, puesto que es una herramienta de fácil manejo por ser programable como un microcontrolador en el cual se puede personalizar toda la aplicación, además su costo es relativamente bajo.
- La programación del módulo transmisor se lo efectuó en lenguaje Assembler, puesto que se obtuvo un mejor control en lo que respecta a los tiempos de bit de los datos que serán transmitidos, y la optimización de memoria del microcontrolador.
- Se efectuó la programación de los chips que reciben los datos del módulo de recepción con la herramienta Microcode por su fácil lenguaje y sus funciones de comunicación que simplifican el trabajo del programador.
- El desarrollo del HMI, se efectuó en Visual Basic 6.0, por ser una herramienta de programación muy versátil la misma que por su ambiente grafico, opciones de comunicación y fácil manejo, es hoy en día una de las herramientas más utilizadas para el desarrollo de software en muchas aplicaciones.
- La presencia de la tecnología ha resuelto, algunos procesos que para el ser humano son difíciles y repetitivos, debemos pensar entonces en como resolver los problemas de tránsito vehicular de una forma tecnológicamente fácil y no costosa, haciendo que el status de vida crezca con lo que se conseguirá

proporcionalmente un crecimiento en el rendimiento de cada uno de los ciudadanos. Este proyecto se muestra con las intenciones de lograr mejorar la vida de la personas.

- El proyecto presentado muestra en forma de prototipo, una posible solución para el problema de congestión vehicular en los peajes. En el mismo se indica la forma de funcionamiento y la arquitectura del diseño, la misma que puede irse expandiendo con la idea de cubrir la mayor parte de servicios asociados, utilizando toda o parte de la arquitectura del diseño original.
- Este estudio ha conseguido, como conclusión la necesidad de concentrar esfuerzos en el desarrollo tecnológico de los medios que involucran a transportes y flotas puesto que se ha visto un gran descuido y poca infraestructura en lo que concierne a las rutas y caminos.
- La telemática como resultado de la electrónica avanzada a logrado mejorar en muchos aspectos los procesos y servicios en los que a diario están involucrados los seres humanos. Por todas estas características que brinda esta rama de la tecnología se pensó en utilizarla como parte fundamental del diseño del prototipo para automatizar todo el proceso de recaudación del peaje.
- En la actualidad existen estudios realizados por los países de primer mundo en los cuales se detallan importantes aspectos sobre carreteras y rutas totalmente automatizadas estos estudios son los denominados ITS, los que ha tomado como referencia este proyecto tratando así de encuadrarnos en las recomendaciones sin dejar de lado la realidad y las distancias de infraestructura y tecnología, que de estos nos separan.
- Una parte muy importante en el diseño del sistema es la creación de tablas en la base de datos, las cuales deben estar bien diseñadas para que el software pueda tener opción al mejoramiento y ampliación de parámetros de los cuales depende que el software pueda ser optimizado manejando además otros servicios asociados a él.

- Es importante el dimensionamiento del código con el cual se pueda identificar fácilmente al cliente, para nuestro caso el código es la placa del vehículo dentro de la cual se encuentra mucha información adyacente del cliente puesto que con ella se hace relación a varias bases de datos con lo cual se consigue un enlace de la información. Adicionalmente hablando en bits tenemos 2^{32} bits que significa $4 \cdot 10^9$ posibilidades de código, dado el primer carácter de la placa indica la provincia y la segunda el año de matriculación, los cuales prácticamente son fijos, el uno para provincias y el otro por años.
- Finalmente se ha concluido con este trabajo con resultados y objetivos planteados en el perfil de este proyecto, los cuales mostrarían el funcionamiento de cada una de las partes que conforma al sistema total de recaudación del peaje. Podemos además concluir que este prototipo sugiere parámetros recomendados para el funcionamiento de cada una de sus partes, así como: distancias, velocidades, infraestructura vial, etc.

6.2 RECOMENDACIONES

- Las autoridades deben tomar decisiones urgentes, para poder resolver los problemas de tránsito generados en los puntos de recaudación de peaje, como son la contaminación y la gran congestión vehicular lo que ocasiona grandes pérdidas económicas para el país ya que es comprobado que el ser humano no se desarrolla bien en sus actividades después de haber soportado niveles de estrés durante las primeras horas del día.
- La educación como parte fundamental en la formación de los seres humanos debe ser enfocada también a una educación vial, puesto que hoy en día manejar un vehículo ya no es un lujo si no una necesidad, la que con lleva gran responsabilidad por parte de los conductores, pues estudios realizados indican que los accidentes de tránsito es una de las principales causas de mortalidad, con cifras alarmantes de más de 1500, muertos al año en nuestro país.
- Para conseguir los mejores resultados dentro de lo que abarca la magnitud de este proyecto, es necesario que las organizaciones gubernamentales trabajen

conjuntamente, logrando leyes y acuerdos interinstitucionales dentro del margen legal a los que se deben regir cada una de las personas e instituciones involucradas.

- Este prototipo, se ha desarrollado utilizando PIC's de radio frecuencia, con los cuales se consiguieron los objetivos planteados en el proyecto, los que querían indicar el funcionamiento de un sistema con control electrónico para peaje, de la que ingresen al sistema, de una forma óptima y segura.

- El proyecto presentado, utiliza algunos tipos de tecnología, los cuales por falta de presupuesto no han sido los mejores para el desarrollo del mismo. Se puede incrementar el misma forma se debe advertir que el diseño no puede ser utilizado para la implementación real puesto que es necesario aumentar la capacidad de comunicación de usuarios con la central, para lo cual es necesario la adquisición de equipo sofisticado el mismo que deberá atender como mínimo a 20 usuarios potencial de este prototipo si puesto en marcha en uno de los peajes de nuestro país y con herramientas de mejor funcionalidad, se logra que en vez de utilizar una sirena como alarma que advierta a las autoridades del peaje, del ingreso de un usuario que no está registrado dentro de su base de datos, se utilice un sistema de cámaras de video o fotográficas el cual capture las placas del vehículo, y posteriormente después de haber verificado esta información se envíe a la jefatura de tránsito para una posterior sanción al infractor.

- Se recomienda comprar un ancho de banda dentro del espectro radioeléctrico del país o utilizar equipos de comunicación que trabajen dentro del rango de frecuencias de la banda ISM. Puesto que como experiencia en la realización del proyecto se encontró la dificultad de interferencia en la frecuencia utilizada, dado que los PIC's de radio frecuencia adquiridos trabajaban en la banda UHF y con frecuencias ya asignadas (433.92 MHz.).

REFERENCIAS BIBLIOGRÁFICAS

ANGULO, José; ROMERO, Susana; ANGULO, Ignacio, **Microcontroladores PIC Diseño Práctico de Aplicaciones Segunda Parte PIC16F87X**, 2ª ed., Madrid: Concepción Fernández Madrid, 2000. Capítulo 2, Arquitectura diagrama de conexiones y repertorio de instrucciones, p. 21-38, Capítulo 8, El USART: Transmisor/Receptor Síncrono/Aíncrono, p. 191-205

REYES, Carlos, **Aprenda rápidamente a programar Microcontroladores PIC 16F62X, 16F81X, 12F6XX**, 1ª ed., Gráficas Ayerve, Ecuador – 2004, Capítulo 5.9 Comunicación, p 118 - 133

CEBALLOS, Francisco, **Enciclopedia de Microsoft Visual Basic 4**, ra – ma, Madrid - 1997, Capítulo 6 Trabajando con menús 139 – 183.

LEÓN, Alberto, INDRA, Widjaja, **Redes de Comunicación**, 1ª ed., McGraw – Hill, Madrid 2002, Capítulo 3 Fundamentos de transmisión digital, p. 114 – 150

<http://www.sectra.cl/its/documentos/documentos.htm>, *Aspectos generales y metodológicos de los Sistemas Inteligentes de Transportes ITS*

<http://iaci.unq.edu.ar/materias/telecomunicaciones/archivos/2004/>, *Modulación Digital I*

<http://iaci.unq.edu.ar/materias/telecomunicaciones/archivos/2004/>, *Modulación Digital II*

<http://iaci.unq.edu.ar/materias/telecomunicaciones/archivos/2004/> , *Relación entre Modulación y Codificación*

<http://tramullas.com/documatica/2-3.html> ,*Introducción a la Documática*

<http://www.microchip.com>, *Microchip PIC16F87XA Data Sheet 28/40-pin Enhanced FLASH Microcontrollers.*

<http://www.microchip.com>, *Microchip PIC16F62XA Data Sheet 18/20-pin Enhanced FLASH Microcontrollers.*

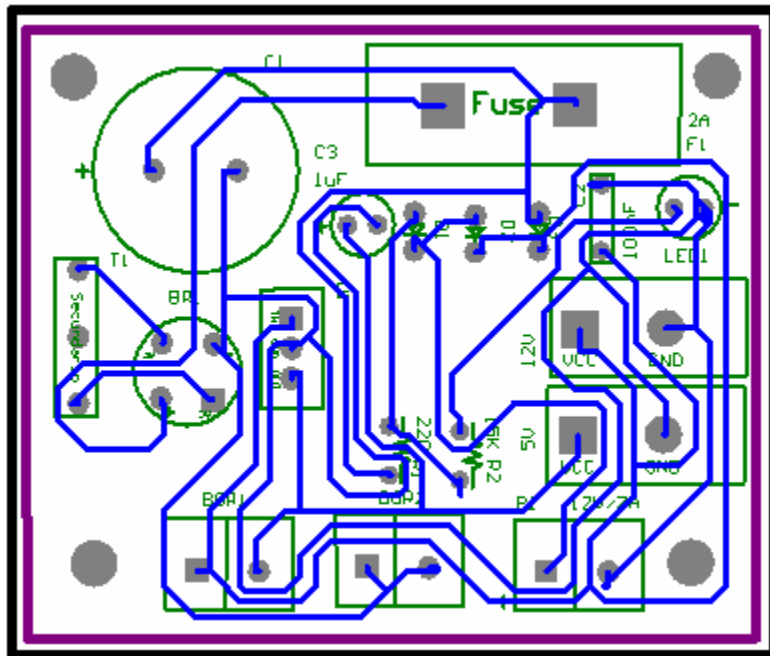
<http://www.itdelicias.edu.mx/Paginas/Sistemas/Luis/vb/>, *Comunicación de VB con Control MScmm.*

ANEXO 1

DIAGRAMA ELECTRÓNICO Y

PCB DE PLACA DE

SUMINISTRO DE ENERGÍA



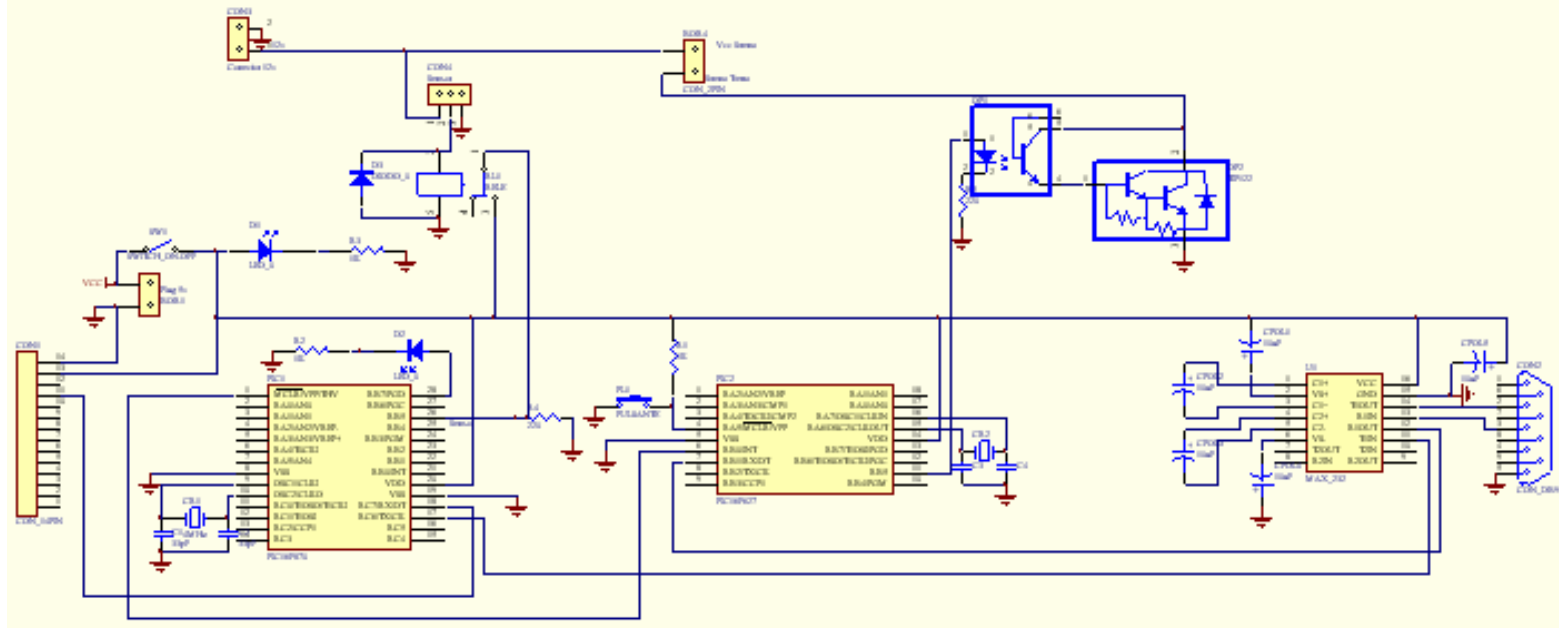
ANEXO 2

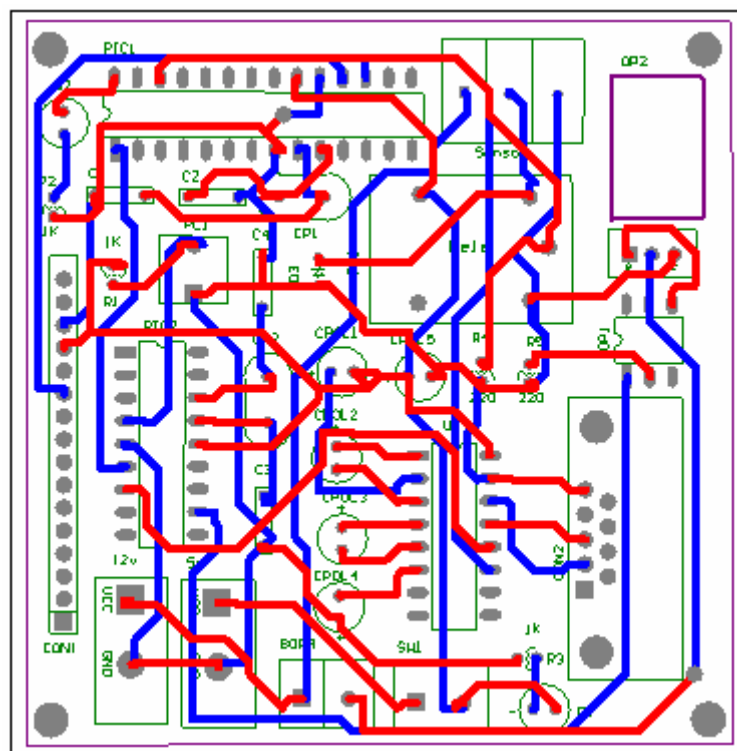
DIAGRAMA ELECTRÓNICO Y

PCB DE PLACA DE

ADQUISICIÓN DE DATOS

Esquemático de la Tarjeta de Adquisición de datos





ANEXO 3

CÓDIGOS FUENTE DEL

SOFTWARE DE TRANSMISIÓN

Y ADQUISICIÓN DE DATOS

PROGRAMA DEL MÓDULO DE TRANSMISIÓN

;Programa donde se transmitira con el formato RS232
;ultima grabacion 20 de febrero del 2005
;programa que contiene los tiempos normales del formato serial a 2400 baudios
;y se ha colocado los filtros y otros bits de parada

```
                list    p=12f675    ; list define a procesador
#include <p12f675.inc> ;especifica definicion variables del
procesador

                errorlevel -302      ; suprime mensaje 302

                ;FILTROS

D0              equ    D'65'          ;A
D1              equ    D'66'          ;B

                ; DATOS A TRANSMITIR

D2              equ    D'80'          ;P          ; DATOS A TRANSMITIR
D3              equ    D'82'          ;R
D4              equ    D'65'          ;A
D5              equ    D'49'          ;1
D6              equ    D'52'          ;4
D7              equ    D'56'          ;8

                ; PARADA

D8              equ    D'49'          ;1
D9              equ    D'50'          ;2

DATO_0 equ 0x20      ;HEX 41
DATO_1 equ 0x21      ;HEX 46
DATO_2 equ 0x22      ;HEX 4B

DATO_3 equ 0x23      ;HEX 50
DATO_4 equ 0x24      ;HEX 41
DATO_5 equ 0x25      ;HEX 46
DATO_6 equ 0x26      ;HEX 4B
DATO_7 equ 0x27      ;HEX 46

DATO_8 equ 0x28      ;HEX 4B
DATO_9 equ 0x29      ;HEX 31
```



```

        cblock 0x30
            Count          ;Contador usado para
            Count2
            BIT_CONT
            BANDERA
            auxiliar
            CONTIMER1
        endc

w_temp    EQU 0X70    ;direcciones multiplexadas en los 2 bancos
status_temp EQU 0X71

__CONFIG _CPD_OFF & _CP_OFF & _BODEN_OFF & _MCLRE_OFF &
_PWRTE_OFF & _WDT_OFF & _INTRC_OSC_NOCLKOUT

        org    0x00          ;vector de inicio
        goto  INICIO

        org    0x04          ; vector de interrupcion
        goto  INTERRUPCION

INTERRUPCION
;push

        BCF   INTCON,GIE

        movwf w_temp          ; guardo registro W
        swapf STATUS, W      ; swap status para guardar dentro de W
        bcf   STATUS, RP0     ; ---- Selecciono Banco 0 -----
        movwf status_temp    ;cargando registro STATUS
        DECFSZ CONTIMER1,F
        goto  POP

        MOVLW D'2'
        MOVWF CONTIMER1

        BSF   GPIO,5         ; HABILITO LA RF

        BSF   GPIO,2
        CALL MBIT
        BSF   GPIO,2
        CALL MBIT
        BSF   GPIO,2
        CALL MBIT
        BSF   GPIO,2
        CALL MBIT
        BSF   GPIO,2
        CALL MBIT
        BSF   GPIO,2

```

```

CALL MBIT
BSF      GPIO,2
CALL MBIT
BSF      GPIO,2
CALL MBIT
BSF      GPIO,2
CALL MBIT

```

MAIN

```

movlw DATO_0
movwf  FSR

```

REVISANDO1

```

MOVLW  D'8'
MOVWF  BIT_CONT
movf  INDF,W
movwf  auxiliar
BCF   GPIO,2           ;BIT DE INICIO
CALL MBIT

```

;subrutina de transmision

SSBIT

```

RRF      auxiliar,F
BTFSSSTATUS,C
CALL CERO
BTFSC   STATUS,C
CALL UNO
DECFSZ  BIT_CONT,F
GOTO SSBIT

```

call UNO ;bit de parada

```

incf  FSR,F
movf  FSR,W
xorlw DATO_9 + 1
btfss STATUS,Z
goto  REVISANDO1   ;siga transmitiendo codigo
BCF   GPIO,5

```

;DESHABILITO LA RF

bcf STATUS,RP0 ;banco0

POP

```

movlw 0x0B
movwf TMR1H
movlw 0xDB
movwf TMR1L

```

```
BCF      PIR1,0      ;RESETEO BANDERA DE TMR1
```

```
swapf   status_temp, W      ;  
movwf   STATUS              ; restaura registro STATUS  
swapf   w_temp, F  
swapf   w_temp, W
```

```
BSF     INTCON,GIE  
retfie
```

INICIO

```
movlw   D0  
movwf   DATO_0  
movlw   D1  
movwf   DATO_1  
movlw   D2  
movwf   DATO_2  
movlw   D3  
movwf   DATO_3  
movlw   D4  
movwf   DATO_4  
movlw   D5  
movwf   DATO_5  
movlw   D6  
movwf   DATO_6  
movlw   D7  
movwf   DATO_7  
movlw   D8  
movwf   DATO_8  
movlw   D9  
movwf   DATO_9
```

```
CLRF   GPIO
```

```
CLRF   TRISIO
```

```
bsf     STATUS,RP0      ;banco 1
```

```
movlw   B'10011000'  
movwf   OPTION_REG  
MOVLW   B'01000000'  
MOVWF   INTCON  
BSF     PIE1,0  
movlw   B'10000000'  
movwf   ANSEL  
movlw   b'11011011'  
movwf   TRISIO  
BSF     INTCON,GIE  
bcf     STATUS,RP0      ;banco0  
movlw   b'00110101'
```

```

movwf    T1CON
MOVLW   D'8'
MOVWF   CONTIMER1
movlw   0x0B
movwf   TMR1H
movlw   0xDB
movwf   TMR1L

```

REVISANDO

GOTO REVISANDO

```

UNO      BSF      GPIO,2
          MOVLW   D'1'          ;[1]
          movwf   Count2       ;[1]

          DELAY1p1
          movlw   D'90'        ; [1]
          movwf   Count        ; [1]

          DELAY400lp1
          nop          ;[1]
          decfsz   Count,F     ; [1]
          goto    DELAY400lp1 ; [2]
          ;           ; -----
          ;           ; 90 x 4 = 360us

          decfsz   Count2,F    ; [1]
          goto    DELAY1p1    ; [2]

```

RETURN

CERO

```

          BCF      GPIO,2
          MOVLW   D'1'          ;[1]
          movwf   Count2       ;[1]

          DELAY1p0
          movlw   D'90'        ; [1]
          movwf   Count        ; [1]
          DELAY400lp0
          nop          ; [1]
          decfsz   Count,F     ; [1]
          goto    DELAY400lp0 ; [2]
          ;           ; -----
          ;           ; 90 x 5 = 360us

          decfsz   Count2,F    ; [1]

```

```
goto DELAYlp0 ;[2]
```

```
RETURN
```

```
MBIT
```

```
MOVLW    D'1'      ;[1]  
movwf    Count2    ;[1]
```

```
DELAYlp  
movlw    D'90'     ; [1]  
movwf    Count     ; [1]
```

```
DELAY400lp
```

```
nop                      ; [1]  
decfsz   Count,F        ; [1]  
goto     DELAY400lp    ; [2]  
        ; -----  
        ;          90 x 4 = 360us
```

```
decfsz   Count2,F      ; [1]  
goto     DELAYlp      ; [2]
```

```
return   ; [2]  
        ; w = 1 -> 407 useg @4MHz el tiempo de bit del preambulo
```

```
end
```

```

*****
'* Name   :Receptor_1.BAS                               *
'* Author : Francisco Javier Hoyos M.                   *
'* Notice : Copyright (c) 2005                           *
'*         : All Rights Reserved                         *
'* Date   : 07/04/2005                                   *
'* Version : 1.5                                         *
'* Notes  :                                              *
*****

```

```

dato   var byte[6]
trisc.6=0 "SALIDA SERIAL Tx transmite a compu
trisc.7=1 "ENTRADA DE RECEPTOR RX recibe de tarjeta
trisb.5=1 "entrada sensor

```

```

PORTB.5=0 "sensor

```

```

INTCON=%11000000
PIE1.5=1
RCSTA=$90
SPBRG=25 ; 2400 BAUDIOS

```

```

MAIN:

```

```

if portb.5=1 then espera_datos
GOTO MAIN

```

```

ESPERA_DATO:

```

```

RCSTA.4=1
If PIR1.5=1 Then LECTURA1
goto espera_datos

```

```

LECTURA1:

```

```

RCSTA.4=0
HSEROUT ["START1"]
PAUSE 500

```

```

RCSTA.4=1           'Activa la recepcion continua USART
PIE1.5=1
HSERin[wait("AB"),str dato\6]

```

```

RCSTA.4=0
HSEROUT [str dato\6]
high portb.7
PAUSE 1000
low portb.7
pause 10
PIR1.5=0
goto main

```

Programa de transmisión de computadora a PIC utilizando interrupción

```
list p= 16f627
#include <p16f627.inc>
errorlevel    -302
```

CBLOCK 0X20

Count
Count2

ENDC

```
temp_w      equ h'70'
temp_sequ h'71'
```

```
org          0x00 ;vector reset
goto inicio  ;va a la primera instruccion del programa
org          0x04 ;vector de interrupciones
goto interrupcion
```

interrupcion

```
bcf  INTCON,GIE ;deshabilito interrupciones globales
```

;push

```
movwf temp_w
swapf temp_w, f
swapf STATUS,W
movwf temp_s
```

;;?Que interrupcion sera

```
btfss PIR1,RCIF

goto pop
BCF  RCSTA,SPEN
BCF  RCSTA,CREN
MOVLW  B'01010010'
XORWF  RCREG
BTFSSTATUS,Z
GOTO  pop
```

```

    bcf    PORTB,0           ;orden de reset
    call   tiempo1          ;1 seg
    BSF    PORTB,0
    CALL   tiempo           ; 100 ms
    bsf    PORTB,5          ; alarma 2 seg
    call   tiempo1
    call   tiempo1
    bcf    PORTB,5
    bcf    PIR1,RCIF

pop

    swapf  temp_s,W
    movwf STATUS
    swapf  temp_w,W
    bsf    INTCON,GIE       ;habilito interrupciones
    BSF    RCSTA,SPEN
    BSF    RCSTA,CREN
    retfi

inicio

    banksel    PORTB
    movlw 0x00
    movwf PORTB
    movlw 0x00
    movwf PORTA

    banksel    TRISA
    movlw B'11100000'
    MOVWF     TRISA
    movlw B'11000010'
    movwf TRISB
    MOVLW     0XFF
    movwf OPTION_REG
    BSF    PIE1,RCIE       ;habilito interrupcion por recepcion en rb1
    MOVLW     D'25'
    MOVWF     SPBRG        ; transmision de datos a 2400 baudios
    BCF    TXSTA,SYNC      ;asincrono
    BCF    TXSTA,BRGH      ;baja velocidad

    BANKSEL    RCSTA
    MOVLW     B'10110000'
    movwf RCSTA
    MOVLW     B'11000000'
    movwf INTCON
    BCF      PIR1,RCIF

MAIN

    BSF      PORTB,0       ;senal de reset

    GOTO     MAIN

```



```

nop          ; [1]
nop          ; [1]
nop          ; [1]
nop          ; [1]
decfsz Count,F      ; [1]
goto TIEMPO400lp1 ; [2]
;
;          -----
;          250 x 20 us =5000 us -> (5ms)

decfsz Count2,F      ; [1]
goto TIEMPOlp1      ; [2]

RETURN          ; [2]
;          total 2 (call) + W x (5000 + 20) + 2 (return)
;          w = 200 -> 1 seg @4MHz

```

end

ANEXO 4

CÓDIGOS FUENTE DEL

SOFTWARE DE

ADMINISTRACIÓN Y

MONITOREO

PROGRAMA DEL SOFTWARE DE ADMINISTRACIÓN

'frm Principal

```
Private Sub mcbXCliente_Click()  
frm_busqueda.Show  
End Sub
```

```
Private Sub mcNuevo_Click()  
frm_NuevoCliente.Show  
End Sub
```

```
Private Sub mePeaje_Click()  
frm_estacion.Show  
End Sub
```

```
Private Sub meProvincia_Click()  
frm_provincia.Show  
End Sub
```

```
Private Sub mpBanco_Click()  
frm_Banco.Show  
End Sub
```

```
Private Sub mpTipoCuenta_Click()  
frm_tipocuenta.Show  
End Sub
```

```
Private Sub mpTipoEvento_Click()  
frm_tipoevento.Show  
End Sub
```

```
Private Sub mpTipoVehiculo_Click()  
frm_tipovehiculo.Show  
End Sub
```

```
Private Sub mpXCliente_Click()  
frm_xcliente.Show  
End Sub
```

```
Private Sub mpXVehiculo_Click()  
frm_xvehiculo.Show  
End Sub
```

```
Private Sub msAcercade_Click()  
frmAbout.Show  
End Sub
```

```
Private Sub msSalir_Click()  
End  
End Sub
```

'frm Banco

```
Dim WithEvents adoPrimaryRS As Recordset
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean
```

```
Private Sub Form_Load()
    Dim sql As String
    sql = "select Nombre from BANCO Order by Nombre"
    Set adoPrimaryRS = New ADODB.Recordset
    adoPrimaryRS.Open sql, conecbd, adOpenStatic, adLockOptimistic
```

```
    Dim oText As TextBox
    'Bind the text boxes to the data provider
    For Each oText In Me.txtFields
        Set oText.DataSource = adoPrimaryRS
        oText.Enabled = False
    Next
```

```
    mbDataChanged = False
End Sub
```

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If mbEditFlag Or mbAddNewFlag Then Exit Sub
```

```
    Select Case KeyCode
        Case vbKeyEscape
            cmdClose_Click
        Case vbKeyEnd
            cmdLast_Click
        Case vbKeyHome
            cmdFirst_Click
        Case vbKeyUp, vbKeyPageUp
            If Shift = vbCtrlMask Then
                cmdFirst_Click
            Else
                cmdPrevious_Click
            End If
        Case vbKeyDown, vbKeyPageDown
            If Shift = vbCtrlMask Then
                cmdLast_Click
            Else
                cmdNext_Click
            End If
    End Select
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
```

```
'Private Sub adoPrimaryRS_WillChangeRecord(ByVal adReason As ADODB.EventReasonEnum, ByVal
cRecords As Long, adStatus As ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
' This is where you put validation code
' This event gets called when the following actions occur
' Dim bCancel As Boolean
```

```

'
' Select Case adReason
' Case adRsnAddNew
' Case adRsnClose
' Case adRsnDelete
' Case adRsnFirstChange
' Case adRsnMove
' Case adRsnRequery
' Case adRsnResynch
' Case adRsnUndoAddNew
' Case adRsnUndoDelete
' Case adRsnUndoUpdate
' Case adRsnUpdate
' End Select
'
' If bCancel Then adStatus = adStatusCancel
End Sub

```

```

Private Sub cmdAdd_Click()
On Error GoTo AddErr
With adoPrimaryRS
If Not (.BOF And .EOF) Then
mvBookMark = .Bookmark
End If
.AddNew
mbAddNewFlag = True
SetButtons False
End With
txtFields(0).Enabled = True
Exit Sub
AddErr:
MsgBox Err.Description
End Sub

```

```

Private Sub cmdDelete_Click()
On Error GoTo DeleteErr
With adoPrimaryRS
.Delete
.MoveNext
If .EOF Then .MoveLast
End With
Exit Sub
DeleteErr:
MsgBox Err.Description
End Sub

```

```

Private Sub cmdEdit_Click()
On Error GoTo EditErr

mbEditFlag = True
SetButtons False
txtFields(0).Enabled = True
Exit Sub

```

```

EditErr:
MsgBox Err.Description
End Sub

```

```

Private Sub cmdCancel_Click()
On Error Resume Next

```

```

SetButtons True
mbEditFlag = False
mbAddNewFlag = False
adoPrimaryRS.CancelUpdate
If mvBookMark > 0 Then
    adoPrimaryRS.Bookmark = mvBookMark
Else
    adoPrimaryRS.MoveFirst
End If
txtFields(0).Enabled = False
mbDataChanged = False

End Sub

Private Sub cmdUpdate_Click()
    On Error GoTo UpdateErr

    adoPrimaryRS.UpdateBatch adAffectAll

    If mbAddNewFlag Then
        adoPrimaryRS.MoveLast      'move to the new record
    End If

    mbEditFlag = False
    mbAddNewFlag = False
    SetButtons True
    mbDataChanged = False
    txtFields(0).Enabled = False
    Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError

    adoPrimaryRS.MoveFirst
    mbDataChanged = False

    Exit Sub

GoFirstError:
    MsgBox Err.Description
End Sub

Private Sub cmdLast_Click()
    On Error GoTo GoLastError

    adoPrimaryRS.MoveLast
    mbDataChanged = False

    Exit Sub

GoLastError:
    MsgBox Err.Description
End Sub

```

```

Private Sub cmdNext_Click()
    On Error GoTo GoNextError

    If Not adoPrimaryRS.EOF Then adoPrimaryRS.MoveNext
    If adoPrimaryRS.EOF And adoPrimaryRS.RecordCount > 0 Then
        Beep
        'moved off the end so go back
        adoPrimaryRS.MoveLast
    End If
    'show the current record
    mbDataChanged = False

    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub

Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError

    If Not adoPrimaryRS.BOF Then adoPrimaryRS.MovePrevious
    If adoPrimaryRS.BOF And adoPrimaryRS.RecordCount > 0 Then
        Beep
        'moved off the end so go back
        adoPrimaryRS.MoveFirst
    End If
    'show the current record
    mbDataChanged = False

    Exit Sub

GoPrevError:
    MsgBox Err.Description
End Sub

Private Sub SetButtons(bVal As Boolean)
    cmdAdd.Visible = bVal
    cmdEdit.Visible = bVal
    cmdUpdate.Visible = Not bVal
    cmdCancel.Visible = Not bVal
    cmdDelete.Visible = bVal
    cmdClose.Visible = bVal
    cmdNext.Enabled = bVal
    cmdFirst.Enabled = bVal
    cmdLast.Enabled = bVal
    cmdPrevious.Enabled = bVal
End Sub

Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)

    If IsNumeric(Chr(KeyAscii)) Then
        If KeyAscii <> 8 Then
            KeyAscii = 0
        End If
    End If

End Sub

```


'form Búsqueda

```
Private Sub btn_aceptar_Click()
On Error GoTo buscar_err
Dim cedula As String
Dim apellidos As String
Dim rs As New ADODB.Recordset
Dim sql As String

If opt_codigo = True Then
    cedula = Trim(txt_buscar.Text)
    sql = "select * from cliente where cedula=" & cedula & ""
Else
    apellidos = Trim(txt_buscar.Text)
    sql = "select * from cliente where apellidos like '%" & apellidos & "%' "
End If
Set rs = conecbd.Execute(sql)
'cargo la lista con los resultados

lst_clientesb.Clear
If Not rs.EOF Then
    While Not rs.EOF
        lst_clientesb.AddItem Trim(rs!cedula) & " " & Trim(rs!apellidos) & " " & Trim(rs!nombres)
        rs.MoveNext
    Wend
Else
    MsgBox "No se produjo ningún resultado..", vbInformation, "BUSQUEDA"
End If

Exit Sub
buscar_err:
MsgBox Err.Description, vbInformation, "BUSQUEDA"
End Sub

Private Sub btn_cancelar_Click()
Unload Me
End Sub

Private Sub lst_clientesb_Db1Click()
Dim item As Long
Dim cadena() As String
Dim cedula As String
Dim apellidos As String

item = lst_clientesb.ListIndex
apellidos = lst_clientesb.List(item)
cadena = Split(apellidos, " ")
cedula = Trim(cadena(0))

cargar_registro cedula

Unload Me
frm_modificarcliente.Show

End Sub
```

'frm Estación

```

Dim WithEvents adoPrimaryRS As Recordset
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean

Private Sub Combo1_Click()
Dim provincia() As String
provincia = Split(Combo1.Text, " ")
txtFields(2).Text = Trim(provincia(0))
End Sub

Private Sub Form_Load()

Dim rs As New ADODB.Recordset
Dim sql As String

Set adoPrimaryRS = New ADODB.Recordset
adoPrimaryRS.Open "select Nombre,Ubicacion,Provincia from ESTACION Order by Nombre",
conecbd, adOpenStatic, adLockOptimistic

Dim oText As TextBox
'Bind the text boxes to the data provider
For Each oText In Me.txtFields
Set oText.DataSource = adoPrimaryRS
oText.Enabled = False
Next

'cargo el combo
sql = "select * from provincia order by nombre"
Set rs = conecbd.Execute(sql)
rs.MoveFirst
While Not rs.EOF
Combo1.AddItem Str(rs!codigo) & " " & Trim(rs!Nombre)
rs.MoveNext
Wend

mbDataChanged = False
End Sub

Private Sub Form_Resize()
On Error Resume Next
lblStatus.Width = Me.Width - 1500
cmdNext.Left = lblStatus.Width + 700
cmdLast.Left = cmdNext.Left + 340
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
If mbEditFlag Or mbAddNewFlag Then Exit Sub

Select Case KeyCode
Case vbKeyEscape
cmdClose_Click
Case vbKeyEnd
cmdLast_Click
Case vbKeyHome
cmdFirst_Click

```

```

Case vbKeyUp, vbKeyPageUp
  If Shift = vbCtrlMask Then
    cmdFirst_Click
  Else
    cmdPrevious_Click
  End If
Case vbKeyDown, vbKeyPageDown
  If Shift = vbCtrlMask Then
    cmdLast_Click
  Else
    cmdNext_Click
  End If
End Select
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
  Screen.MousePointer = vbDefault
End Sub

```

```

Private Sub adoPrimaryRS_WillChangeRecord(ByVal adReason As ADODB.EventReasonEnum, ByVal
cRecords As Long, adStatus As ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  'This is where you put validation code
  'This event gets called when the following actions occur
  Dim bCancel As Boolean

```

```

  Select Case adReason
  Case adRsnAddNew
  Case adRsnClose
  Case adRsnDelete
  Case adRsnFirstChange
  Case adRsnMove
  Case adRsnRequery
  Case adRsnResynch
  Case adRsnUndoAddNew
  Case adRsnUndoDelete
  Case adRsnUndoUpdate
  Case adRsnUpdate
  End Select

```

```

  If bCancel Then adStatus = adStatusCancel
End Sub

```

```

Private Sub cmdAdd_Click()
  On Error GoTo AddErr

  txtFields(0).Enabled = True
  txtFields(1).Enabled = True
  Combo1.Enabled = True
  With adoPrimaryRS
    If Not (.BOF And .EOF) Then
      mvBookMark = .Bookmark
    End If
    .AddNew
    mbAddNewFlag = True
    SetButtons False
  End With

```

```

Exit Sub
AddErr:

```

```
    MsgBox Err.Description
End Sub
```

```
Private Sub cmdDelete_Click()
    On Error GoTo DeleteErr
    With adoPrimaryRS
        .Delete
        .MoveNext
    End With
    If .EOF Then .MoveLast
End With
Exit Sub
DeleteErr:
    MsgBox Err.Description
End Sub
```

```
Private Sub cmdEdit_Click()
    On Error GoTo EditErr
    txtFields(0).Enabled = True
    txtFields(1).Enabled = True
    Combo1.Enabled = True
    mbEditFlag = True
    SetButtons False

    Exit Sub
```

```
EditErr:
    MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
    On Error Resume Next
```

```
    SetButtons True
    mbEditFlag = False
    mbAddNewFlag = False
    adoPrimaryRS.CancelUpdate
    If mvBookMark > 0 Then
        adoPrimaryRS.Bookmark = mvBookMark
    Else
        adoPrimaryRS.MoveFirst
    End If
    mbDataChanged = False
    txtFields(0).Enabled = False
    txtFields(1).Enabled = False
    Combo1.Enabled = False
```

```
End Sub
```

```
Private Sub cmdUpdate_Click()
    On Error GoTo UpdateErr

    adoPrimaryRS.UpdateBatch adAffectAll

    If mbAddNewFlag Then
        adoPrimaryRS.MoveLast      'move to the new record
    End If

    mbEditFlag = False
    mbAddNewFlag = False
    SetButtons True
    mbDataChanged = False
```

```
txtFields(0).Enabled = False
txtFields(1).Enabled = False
Combo1.Enabled = False
Exit Sub
UpdateErr:
MsgBox Err.Description
End Sub
```

```
Private Sub cmdClose_Click()
Unload Me
End Sub
```

```
Private Sub cmdFirst_Click()
On Error GoTo GoFirstError
```

```
adoPrimaryRS.MoveFirst
mbDataChanged = False
```

```
Exit Sub
```

```
GoFirstError:
MsgBox Err.Description
End Sub
```

```
Private Sub cmdLast_Click()
On Error GoTo GoLastError
```

```
adoPrimaryRS.MoveLast
mbDataChanged = False
```

```
Exit Sub
```

```
GoLastError:
MsgBox Err.Description
End Sub
```

```
Private Sub cmdNext_Click()
On Error GoTo GoNextError
```

```
If Not adoPrimaryRS.EOF Then adoPrimaryRS.MoveNext
If adoPrimaryRS.EOF And adoPrimaryRS.RecordCount > 0 Then
Beep
'moved off the end so go back
adoPrimaryRS.MoveLast
End If
'show the current record
mbDataChanged = False
```

```
Exit Sub
```

```
GoNextError:
MsgBox Err.Description
End Sub
```

```
Private Sub cmdPrevious_Click()
On Error GoTo GoPrevError
```

```
If Not adoPrimaryRS.BOF Then adoPrimaryRS.MovePrevious
If adoPrimaryRS.BOF And adoPrimaryRS.RecordCount > 0 Then
Beep
'moved off the end so go back
```

```
    adoPrimaryRS.MoveFirst
End If
'show the current record
mbDataChanged = False
```

```
Exit Sub
```

```
GoPrevError:
    MsgBox Err.Description
End Sub
```

```
Private Sub SetButtons(bVal As Boolean)
    cmdAdd.Visible = bVal
    cmdEdit.Visible = bVal
    cmdUpdate.Visible = Not bVal
    cmdCancel.Visible = Not bVal
    cmdDelete.Visible = bVal
    cmdClose.Visible = bVal
    cmdNext.Enabled = bVal
    cmdFirst.Enabled = bVal
    cmdLast.Enabled = bVal
    cmdPrevious.Enabled = bVal
End Sub
```

```
Private Sub txtFields_Change(Index As Integer)
    On Error Resume Next
    Dim rs As New ADODB.Recordset
    Dim provincia As String
    Dim sql As String
    provincia = Trim(txtFields(2).Text)
    sql = "select * from provincia where codigo=" & provincia & ""
    Set rs = conecbd.Execute(sql)
    Combo1.Text = Str(rs!codigo) & " " & Trim(rs!Nombre)
End Sub
```

```
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
    If IsNumeric(Chr(KeyAscii)) Then
        If KeyAscii <> 8 Then
            KeyAscii = 0
        End If
    End If
End Sub

End Sub
```

'frm Login

```
Option Explicit
Private Sub btn_ingresar_Click()

    If Trim(txt_usuario.Text) = "administrador" And Trim(txt_contrasena.Text) = "12" Then
        MsgBox "Bienvenido Usuario: Administrador "
        Unload Me
        frm_principal.Show
    Else
        MsgBox "Nombre de Usuario y/o Contraseña no válidos!"
    End If
```

End Sub

```
Private Sub Form_Load()  
On Error GoTo label_err  
'Coneccion a la BD  
Dim strigconexion As String  
strigconexion = "DRIVER=Driver do Microsoft Access (*.mdb);UID = admin;FIL=MS  
Access;DefaultDir=C:\Peaje final;DBQ=C:\Peaje final\Peaje.mdb"  
Set conecbd = New ADODB.Connection  
conecbd.Open strigconexion
```

Exit Sub

```
label_err:  
MsgBox Err.Description
```

End Sub

```
Private Sub txt_contrasena_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
    btn_ingresar_Click  
End If  
End Sub
```

```
Private Sub txt_usuario_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
    btn_ingresar_Click  
End If  
End Sub
```

'frm Modificar

```
Private Sub btn_guardar_Click()  
On Error GoTo DescError  
  
With rscliente  
    !cedula = Trim(txt_cedula.Text)  
    !nombres = Trim(txt_nombre.Text)  
    !apellidos = Trim(txt_apellidos.Text)  
    !direccion = Trim(txt_direccion.Text)  
    !telefono = Trim(txt_telefono.Text)  
    !email = Trim(txt_email.Text)  
    !fecharegistro = Trim(txt_fecharegistro.Text)  
    .Update  
End With  
MsgBox "Registro Modificado exitosamente!"  
Exit Sub  
DescError:  
MsgBox Err.Description  
End Sub
```

```
Private Sub btn_guardarT_Click()  
On Error GoTo DescError
```

```
With rs cuenta
```

```
!NoCuenta = Trim(txt_cuenta.Text)
!tipocuenta = Val(txt_tc.Text)
!CodigoBanco = Val(txt_banco.Text)
!saldo = Val(txt_saldo.Text)
!CodigoCliente = Trim(txt_cedula.Text)
.Update
End With
```

```
With rsvehiculo
!NoPlaca = Trim(txt_placa.Text)
!CodigoTipoVehiculo = Val(txt_tv.Text)
!CodigoCliente = Val(txt_cedula.Text)
.Update
End With
MsgBox "Registro Modificado exitosamente!"
```

```
Exit Sub
DescError:
MsgBox Err.Description
```

```
End Sub
```

```
Private Sub cbx_banco_Click()
On Error GoTo label_err
Dim sql As String
Dim rs As New ADODB.Recordset
Dim banco As String
```

```
banco = Trim(cbx_banco.Text)
sql = "select * from banco where nombre='" & banco & "'"
Set rs = conecbd.Execute(sql)
banco = Trim(rs!CodigoBanco)
txt_banco.Text = banco
```

```
Exit Sub
label_err:
MsgBox Err.Description
```

```
End Sub
```

```
Private Sub cbx_tipocuenta_Click()
On Error GoTo label_err
Dim sql As String
Dim rs As New ADODB.Recordset
Dim tcuenta As String
```

```
tcuenta = Trim(cbx_tipocuenta.Text)
sql = "select * from tipocuenta where nombre='" & tcuenta & "'"
Set rs = conecbd.Execute(sql)
tcuenta = Trim(rs!CodigoTipoCuenta)
txt_tc.Text = tcuenta
```

```
Exit Sub
label_err:
MsgBox Err.Description
```

```
End Sub
```

```
Private Sub cbx_tipovehiculo_Click()
```



```

On Error GoTo label_err
Dim sql As String
Dim rs As New ADODB.Recordset
Dim tvehiculo As String

tvehiculo = Trim(cb_x_tipovehiculo.Text)
sql = "select * from tipovehiculo where nombre='" & tvehiculo & "'"
Set rs = conecbd.Execute(sql)
tvehiculo = Trim(rs!CodigoTipoVehiculo)
txt_tv.Text = tvehiculo

Exit Sub
label_err:
MsgBox Err.Description
End Sub

Private Sub Command1_Click()
SSTab1.Tab = 1
End Sub

Private Sub Form_Load()
On Error GoTo label_err
Dim rs As New ADODB.Recordset
Dim sql As String
'Cargar los combos de la pantalla
'tipo cuenta
sql = "select * from tipocuenta order by nombre"
Set rs = conecbd.Execute(sql)
rs.MoveFirst
While Not rs.EOF
    cb_x_tipocuenta.AddItem Trim(rs!Nombre)
    rs.MoveNext
Wend
'banco
sql = "select * from banco order by nombre"
Set rs = conecbd.Execute(sql)
rs.MoveFirst
While Not rs.EOF
    cb_x_banco.AddItem Trim(rs!Nombre)
    rs.MoveNext
Wend
'tipovehiculo
sql = "select * from tipovehiculo order by nombre"
Set rs = conecbd.Execute(sql)
rs.MoveFirst
While Not rs.EOF
    cb_x_tipovehiculo.AddItem Trim(rs!Nombre)
    rs.MoveNext
Wend
SSTab1.Tab = 0
Exit Sub
label_err:
MsgBox Err.Description

End Sub

Private Sub Form_Unload(Cancel As Integer)
'rscliente.Close
'rs cuenta.Close
'rsvehiculo.Close

```

End Sub

```
Private Sub txt_apellidos_KeyPress(KeyAscii As Integer)
If IsNumeric(Chr(KeyAscii)) Then
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If
```

End Sub

```
Private Sub txt_banco_Change()
On Error Resume Next
Dim sql As String
Dim rs As New ADODB.Recordset
Dim banco As String
```

```
banco = Trim(txt_banco.Text)
```

```
sql = "select * from Banco whereCodigoBanco=" & banco & ""
Set rs = conecbd.Execute(sql)
```

```
banco = Trim(rs!Nombre)
cbx_banco.Text = banco
```

End Sub

```
Private Sub txt_cuenta_KeyPress(KeyAscii As Integer)
If Not IsNumeric(Chr(KeyAscii)) Then
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If
```

End Sub

```
Private Sub txt_nombre_KeyPress(KeyAscii As Integer)
If IsNumeric(Chr(KeyAscii)) Then
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If
```

End Sub

```
Private Sub txt_tc_Change()
On Error Resume Next
Dim sql As String
Dim rs As New ADODB.Recordset
Dim tipocuenta As String
```

```
tipocuenta = Trim(txt_tc.Text)
```

```
sql = "select * from TipoCuenta whereCodigoTipoCuenta=" & tipocuenta & ""
Set rs = conecbd.Execute(sql)
```

```
tipocuenta = Trim(rs!Nombre)
cbx_tipocuenta.Text = tipocuenta
```

End Sub

```
Private Sub txt_telefono_KeyPress(KeyAscii As Integer)
If Not IsNumeric(Chr(KeyAscii)) Then
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If
```

End Sub

```
Private Sub txt_tv_Change()
On Error Resume Next
Dim sql As String
Dim rs As New ADODB.Recordset
Dim tipovehiculo As String
```

```
tipovehiculo = Trim(txt_tv.Text)
```

```
sql = "select * from tipovehiculo whereCodigoTipoVehiculo=" & tipovehiculo & ""
Set rs = conecbd.Execute(sql)
```

```
tipovehiculo = Trim(rs!Nombre)
cbx_tipovehiculo.Text = tipovehiculo
```

End Sub

frm Nuevo Cliente

```
Private Sub btn_guardar_Click()
On Error GoTo label_err
Dim rs As New ADODB.Recordset
Dim sql As String
Dim cedula As String
Dim nombres As String, apellidos As String
Dim direccion As String, telefono As String
Dim email As String, fecharegistro As String
```

```
cedula = Trim(txt_cedula.Text)
nombres = Trim(txt_nombre.Text)
apellidos = Trim(txt_apellidos.Text)
direccion = Trim(txt_direccion.Text)
telefono = Trim(txt_telefono.Text)
email = Trim(txt_email.Text)
fecharegistro = CStr(Format(Date, "yyyy/MM/dd"))
```

```
sql = "select * from cliente"
rs.Open sql, conecbd, adOpenStatic, adLockOptimistic
```

```
With rs
    .AddNew
    !cedula = cedula
    !nombres = nombres
```

```
!apellidos = apellidos
!telefono = telefono
!direccion = direccion
!email = email
!fecharegistro = fecharegistro
.Update
End With

btn_guardar.Enabled = False
MsgBox "Registro guardado con éxito!"
```

```
Exit Sub
label_err:
MsgBox Err.Description
```

```
End Sub
```

```
Private Sub btn_guardarT_Click()
On Error GoTo label_err
Dim rs As New ADODB.Recordset
Dim sql As String
'Para la cuenta
Dim cuenta As String
Dim tipocuenta As String
Dim banco As String
Dim saldo As String
'Para el vehiculo
Dim placa As String
Dim tipovehiculo As String
```

```
cuenta = Trim(txt_cuenta.Text)
tipocuenta = Trim(txt_tc.Text)
banco = Trim(txt_banco.Text)
saldo = Trim(txt_saldo.Text)
```

```
sql = "select * from cuenta"
rs.Open sql, conecbd, adOpenStatic, adLockOptimistic
```

```
With rs
.AddNew
!NoCuenta = cuenta
!tipocuenta = Val(tipocuenta)
!CodigoBanco = Val(banco)
!saldo = Val(saldo)
!CodigoCliente = Trim(txt_cedula)
.Update
End With
rs.Close
placa = Trim(txt_placa.Text)
tipovehiculo = Trim(txt_tv.Text)
```

```
sql = "select * from vehiculo"
rs.Open sql, conecbd, adOpenStatic, adLockOptimistic
```

```
With rs
.AddNew
!NoPlaca = placa
!CodigoTipoVehiculo = Val(tipovehiculo)
!CodigoCliente = Trim(txt_cedula)
```

```
.Update  
End With
```

```
btn_guardarT.Enabled = False  
MsgBox "Registro guardado con éxito!"
```

```
Exit Sub  
label_err:  
MsgBox Err.Description  
End Sub
```

```
Private Sub cbx_banco_Click()  
On Error GoTo label_err  
Dim sql As String  
Dim rs As New ADODB.Recordset  
Dim banco As String
```

```
banco = Trim(cbx_banco.Text)  
sql = "select * from banco where nombre=" & banco & ""  
Set rs = conecbd.Execute(sql)  
banco = Trim(rs!CodigoBanco)  
txt_banco.Text = banco
```

```
Exit Sub  
label_err:  
MsgBox Err.Description
```

```
End Sub
```

```
Private Sub cbx_tipocuenta_Click()  
On Error GoTo label_err  
Dim sql As String  
Dim rs As New ADODB.Recordset  
Dim tcuenta As String
```

```
tcuenta = Trim(cbx_tipocuenta.Text)  
sql = "select * from tipocuenta where nombre=" & tcuenta & ""  
Set rs = conecbd.Execute(sql)  
tcuenta = Trim(rs!CodigoTipoCuenta)  
txt_tc.Text = tcuenta
```

```
Exit Sub  
label_err:  
MsgBox Err.Description
```

```
End Sub
```

```
Private Sub cbx_tipovehiculo_Click()  
On Error GoTo label_err  
Dim sql As String  
Dim rs As New ADODB.Recordset  
Dim tvehiculo As String
```

```
tvehiculo = Trim(cbx_tipovehiculo.Text)  
sql = "select * from tipovehiculo where nombre=" & tvehiculo & ""  
Set rs = conecbd.Execute(sql)  
tvehiculo = Trim(rs!CodigoTipoVehiculo)  
txt_tv.Text = tvehiculo
```

```
Exit Sub
label_err:
MsgBox Err.Description
End Sub
```

```
Private Sub Command1_Click()
SSTab1.Tab = 1
End Sub
```

```
Private Sub Form_Load()
On Error GoTo label_err
Dim rs As New ADODB.Recordset
Dim sql As String
'Cargar los combos de la pantalla
'tipo cuenta
sql = "select * from tipocuenta order by nombre"
Set rs = conecbd.Execute(sql)
rs.MoveFirst
While Not rs.EOF
    cbx_tipocuenta.AddItem Trim(rs!Nombre)
    rs.MoveNext
Wend
'banco
sql = "select * from banco order by nombre"
Set rs = conecbd.Execute(sql)
rs.MoveFirst
While Not rs.EOF
    cbx_banco.AddItem Trim(rs!Nombre)
    rs.MoveNext
Wend
'tipovehiculo
sql = "select * from tipovehiculo order by nombre"
Set rs = conecbd.Execute(sql)
rs.MoveFirst
While Not rs.EOF
    cbx_tipovehiculo.AddItem Trim(rs!Nombre)
    rs.MoveNext
Wend
```

```
txt_fecharegistro = CStr(Format(Date, "dd/MM/yyyy"))
SSTab1.Tab = 0
```

```
Exit Sub
label_err:
MsgBox Err.Description

End Sub
```

```
Private Sub txt_apellidos_KeyPress(KeyAscii As Integer)
If IsNumeric(Chr(KeyAscii)) Then
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If

End Sub
```

```
Private Sub txt_cedula_KeyPress(KeyAscii As Integer)
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If
```

End Sub

```
Private Sub txt_nombre_KeyPress(KeyAscii As Integer)
If IsNumeric(Chr(KeyAscii)) Then
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If
```

End Sub

```
Private Sub txt_telefono_KeyPress(KeyAscii As Integer)
If Not IsNumeric(Chr(KeyAscii)) Then
    If KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End If
```

End Sub

'frm Reporte Por Cliente

```
Private Sub btn_aceptar_Click()
On Error GoTo errores
With frm_principal.CrystalReport2
.ReportFileName = "C:\Peaje final\Sistema\Reportes\por_cliente.rpt"
.WindowState = crptMaximized
.WindowTitle = "Reporte de Clientes"
.Connect = "DSN = peaje;uid=;pwd=;"
.SQLQuery = "SELECT " & _
    "CLIENTE.`Cedula`, CLIENTE.`Nombres`, CLIENTE.`Apellidos`, CLIENTE.`Telefono`,
CLIENTE.`Direccion`, CLIENTE.`FechaRegistro`, " & _
    "CUENTA.`NoCuenta`, " & _
    "EVENTO.`Placa`, EVENTO.`FechaHora` " & _
    "From " & _
    "(( CLIENTE` CLIENTE INNER JOIN `CUENTA` CUENTA ON " & _
    "CLIENTE.`Cedula` = CUENTA.`CodigoCliente`) " & _
    "INNER JOIN `VEHICULO` VEHICULO ON " & _
    "CUENTA.`CodigoCliente` = VEHICULO.`CodigoCliente`) " & _
    "INNER JOIN `EVENTO` EVENTO ON " & _
    "VEHICULO.`NoPlaca` = EVENTO.`Placa` "
```

```
    .Action = 1
End With
Unload Me
Exit Sub
errores:
    MsgBox Err.Description
End Sub
```

```
Private Sub btn_cancelar_Click()
Unload Me
End Sub
```

frm Acerca de

```
"From " & _  
"((`CLIENTE` CLIENTE INNER JOIN `CUENTA` CUENTA ON " & _  
"CLIENTE.`Cedula` = CUENTA.`CodigoCliente`) " & _  
"INNER JOIN `VEHICULO` VEHICULO ON " & _  
"CUENTA.`CodigoCliente` = VEHICULO.`CodigoCliente`) " & _  
"INNER JOIN `EVENTO` EVENTO ON " & _  
"VEHICULO.`NoPlaca` = EVENTO.`Placa` "
```

```
.Action = 1  
End With  
Unload Me  
Exit Sub  
errores:  
    MsgBox Err.Description  
End Sub
```

```
Private Sub btn_cancelar_Click()  
Unload Me  
End Sub
```


PROGRAMA DE MONITOREO (MONITOR)

```
Option Explicit
Dim dato(2) As String
Dim variable As String
DimCodigoEve As Integer

Private Sub Form_Load()
'Coneccion a la BD
Dim strigconexion As String
strigconexion = "DRIVER=Driver do Microsoft Access (*.mdb);UID = admin;FIL=MS
Access;DefaultDir=C:\Peaje final;DBQ=C:\Peaje final\Peaje.mdb"
Text3.Text = "<<Hora de inicio de Monitoreo: >> " & CStr(Now)
CodigoEve = 0

Set conecbd = New ADODB.Connection
conecbd.Open strigconexion
serial.PortOpen = True
Timer1.Enabled = False

End Sub

Private Sub Form_Unload(Cancel As Integer)

If serial.PortOpen = True Then
serial.PortOpen = False

End If

End Sub

Private Sub Text1_Change()
Dim Placa As String
Dim rs As New ADODB.Recordset 'objeto instanciado
Dim rs2 As New ADODB.Recordset 'objeto instanciado

Dim rsconsulta As New ADODB.Recordset
Dim sql As String
Dim cliente As String, cuenta As String
Dim Precio As Double
Dim Tvehiculo As Integer
Dim fecha As Date
If Text1.Text <> "" Then
fecha = Now
fecha = Format(Str(fecha), "yyyy/MM/dd hh:mm:ss")
```

```
Placa = Trim(Text1)
```

```
sql = "select * from Evento" 'sentencia para seleccionar tabla  
rs.Open sql, conecbd, adOpenStatic, adLockOptimistic 'ejecucion de la sentencia  
'trabajo con tabla seleccionada anteriormente
```

```
With rs
```

```
  .AddNew ' anado nuevo registro a la tabla  
  !CodigoEvento = CodigoEve 'asigno valores a cada campo de la tabla  
  !Placa = Placa  
  !fechahora = fecha  
  !codigoestacion = 1  
  .Update 'actualizo tabla
```

```
End With
```

```
' descuento de la tabla de cuentas
```

```
If Placa <> "ALARMA" Then
```

```
  sql = "select * from vehiculo where NoPlaca=" & Placa & """"  
  Set rsconsulta = conecbd.Execute(sql)  
  cliente = Trim(rsconsulta!codigocliente)  
  Tvehiculo = Val(rsconsulta!CodigoTipoVehiculo)
```

```
  sql = "select * from tipovehiculo where CodigoTipoVehiculo=" & Tvehiculo & ""  
  Set rsconsulta = conecbd.Execute(sql)  
  Precio = Trim(rsconsulta!ValorDescuento)
```

```
  sql = "select * from Cuenta where codigocliente=" & cliente & """"  
  Set rsconsulta = conecbd.Execute(sql)  
  cuenta = Trim(rsconsulta!NoCuenta)
```

```
  sql = "select * from Cuenta WHERE NOCUENTA=" & cuenta & """" 'sentencia  
para seleccionar tabla  
  rs2.Open sql, conecbd, adOpenStatic, adLockOptimistic 'ejecucion de la sentencia  
'trabajo con tabla seleccionada anteriormente
```

```
  With rs2
```

```
    !Saldo = !Saldo - Precio 'asigno valores a cada campo de la tabla  
    .Update 'actualizo tabla
```

```
  End With
```

```
End If
```

```
Text1.Text = ""
```

```
End If
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
  dato(1) = serial.Input
```

```
  serial.NullDiscard = True
```

End Sub

Private Sub serial_OnComm()

 Sleep 1000

 If serial.CommEvent = comEvReceive Then

 dato(0) = serial.Input

 If dato(0) = "START1" Then

 Debug.Print dato(0)

 Timer1.Enabled = True

 Timer1_Timer

 End If

 End If

 If dato(0) = "START1" Then

 ' cuando no llega la placa

 If Len(dato(1)) < 6 Then

 dato(1) = "ALARMA"

 serial.Output = "R"

 CodigoEve = 1 'alarma

 Text3.Text = Text3.Text + vbNewLine + "Evento: " + dato(1) + " Recibido a las: " & CStr(Now)

 Text1.Text = dato(1)

 Timer1.Enabled = False

 ' cuando llega la placa

 ElseIf Trim(dato(1)) <> "ALARMA" Then

 CodigoEve = 2 'descuento

 Text3.Text = Text3.Text + vbNewLine + " Placa: " + dato(1) + " Recibida a las: " & CStr(Now)

 Text1.Text = dato(1)

 Timer1.Enabled = False

 End If

 End If

End Sub

ANEXO 5
COMUNICACIÓN SERIAL EN
VISUAL BASIC 6.0

ANEXO 5

COMUNICACIÓN SERIAL EN VISUAL BASIC 6.0

Dentro de Visual Basic para poder utilizar la comunicación serial, es necesario agregar en el cuadro de herramientas un componente de control adicional que se denomina “Microsoft Comm Control 6.0”. Para esto hay que seguir los siguientes pasos:

1. Una vez que sea a puesto en marcha un nuevo proyecto o aplicación, hacer clic con el botón derecho del mouse en el cuadro de herramientas (ver figura E.1), y seleccionar **Componentes**.

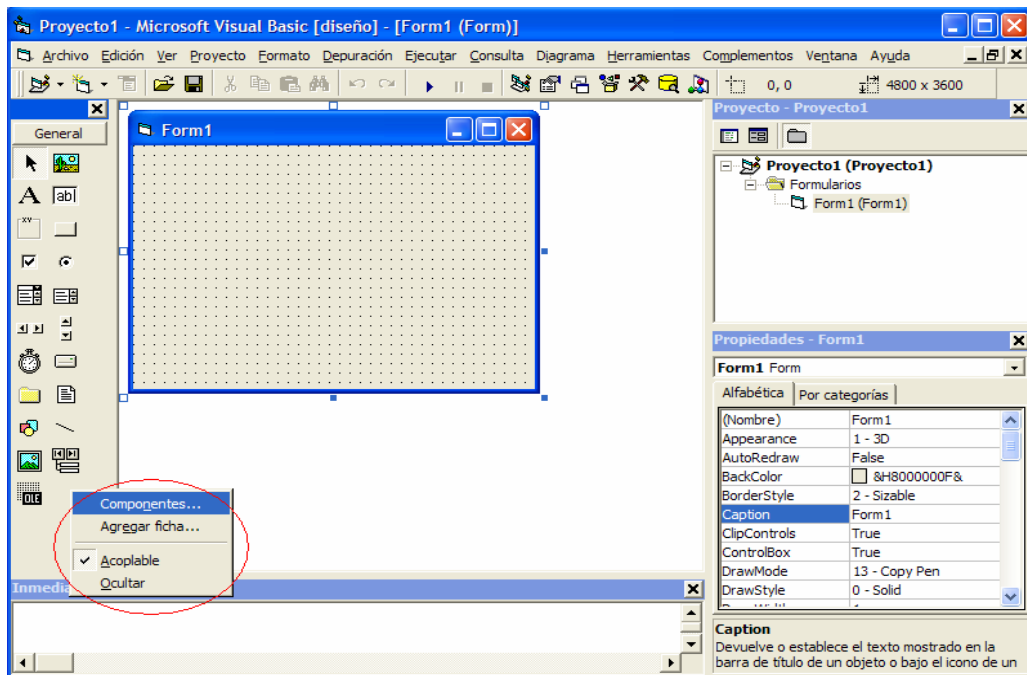


Figura. E.1. Agregar el componente para comunicación serial.

2. En la pantalla de Componentes que se abre, buscar y seleccionar el control “**Microsoft Comm Control 6.0**” (ver figura E.2). Luego hacer clic en “Aplicar”.

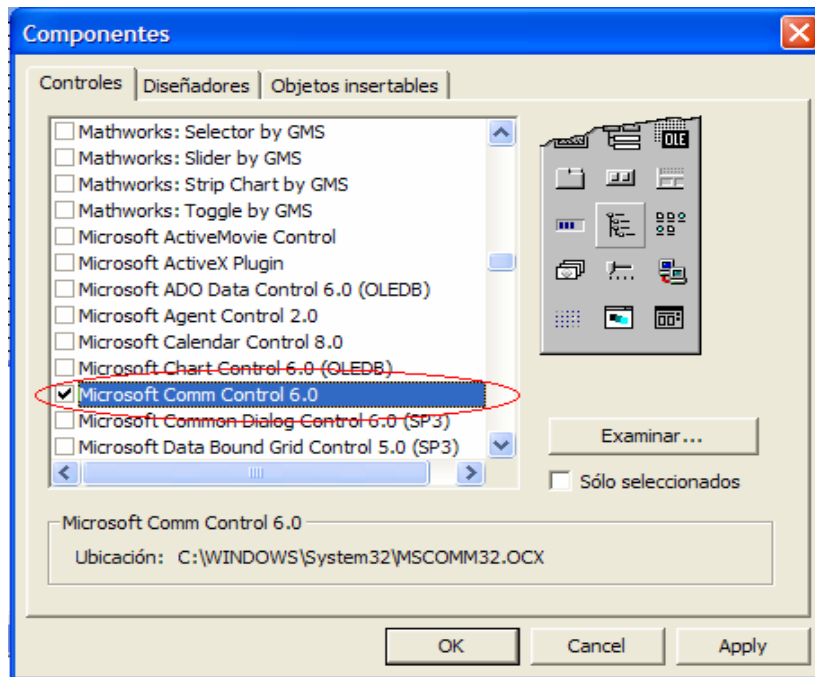


Figura. E.2. Agregando el componente para comunicación serial.

3. Debe aparecer una figura parecida a un teléfono en el cuadro de herramientas, como se muestra en la figura E.3.

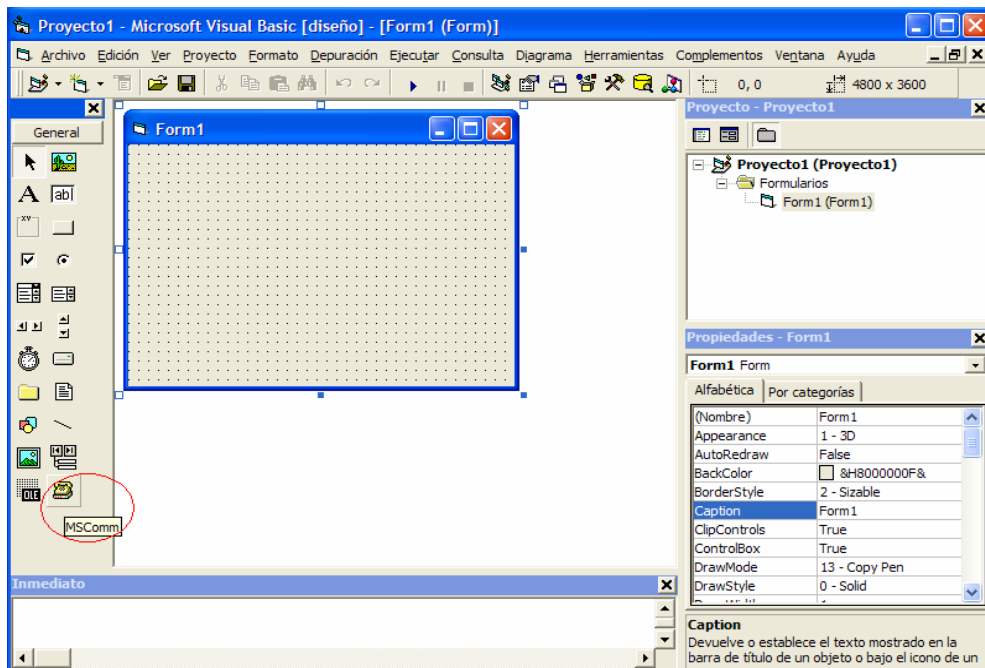


Figura. E.3. Incorporación del comando MSCComm.

4. Establecer propiedades de comunicación.

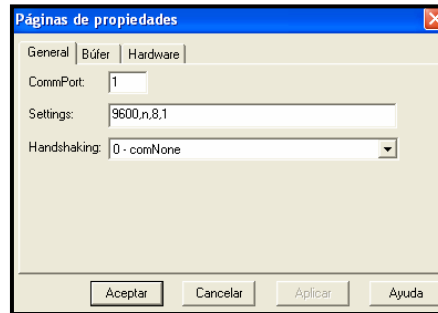


Figura E.4. Propiedades Generales del MSComm.

Las principales propiedades del MSComm están descritas en la tabla E.1.

PROPIEDADES	DESCRIPCIÓN
CommPort	Establece y devuelve el número del puerto de comunicaciones.
Settings	Establece y devuelve la velocidad en baudios, paridad, bits de datos y bits de parada en forma de cadena.
PortOpen	Establece y devuelve el estado de un puerto de comunicaciones. También abre y cierra un puerto.
Input	Devuelve y quita caracteres del búfer de recepción.
Output	Escribe una cadena de caracteres en el búfer de transmisión.

Tabla E.1. Propiedades del MSComm.

5. Establecer las propiedades de los búferes de transmisión y de recepción en tiempo de diseño.

Cuando se abre un puerto, se crean búferes de transmisión y de recepción. Para administrarlos, el control Communications le proporciona diversas propiedades que se puede establecer en tiempo de diseño a través de las páginas de propiedades del control.

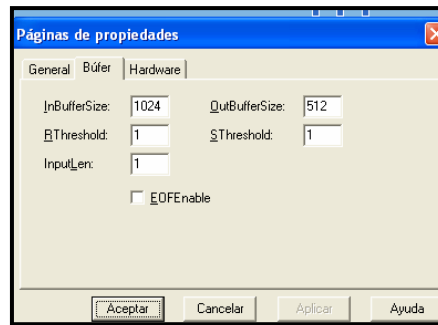


Figura E.5. propiedades del búfer.

En las propiedades **InBufferSize** y **OutBufferSize**, se debe especificar la cantidad de memoria asignada a los búferes de recepción y transmisión. Los valores establecidos de manera predeterminada son los mostrados en la figura E.5.

Las propiedades **Rthreshold** y **Sthreshold** establecen o devuelven el número de caracteres recibidos en los búferes de recepción y transmisión antes de producirse el evento **OnComm**.

Si se establece la propiedad **InputLen** a 1, para que la lectura sea de un dato a la vez, para validar el formato de bloques de datos de longitud fija.

6. Establecer los eventos en tiempo de ejecución.

Se utiliza los comandos para transmitir y recibir datos que proporciona el control **MSComm**.

Para abrir un puerto serie, se utiliza las propiedades **CommPort**, **PortOpen** y **Settings**. Así:

```
' Para abrir el puerto serie
MSComm1.CommPort = 4
MSComm1.Settings = "2400,N,8,1"
MSComm1.PortOpen = True
```


Después de especificar el puerto que se va a abrir y la forma en que se realizará la comunicación de los datos, para establecer la conexión se puede usar la propiedad **PortOpen**, que es un valor booleano, True o False. Si se establece la propiedad PortOpen a False, se cierra el puerto.

La propiedad **Input** sirve para almacenar datos en el búfer de recepción y también para obtenerlos de él. Para almacenarlo en una variable declarada previamente se escribe lo siguiente.

```
text = MSComm1.Input
```

La propiedad Output se usa para enviar comandos y datos al búfer de transmisión así, para enviar una cadena de texto se escribe lo siguiente:

```
MsComm1.Output = " R"
```

ANEXO 6

MANUAL DEL USUARIO PARA

EL PROGRAMA DE

ADMINISTRACIÓN (SIADPE)

MANUAL DEL USUARIO PARA EL SOFTWARE DE ADMINISTRACIÓN

SOFTWARE PARA ADMINISTRACIÓN DE PEAJES (SIADPE)

Al ejecutar el software lo primero que se presentará es la pantalla de ingreso de usuario y contraseña Figura F.1.

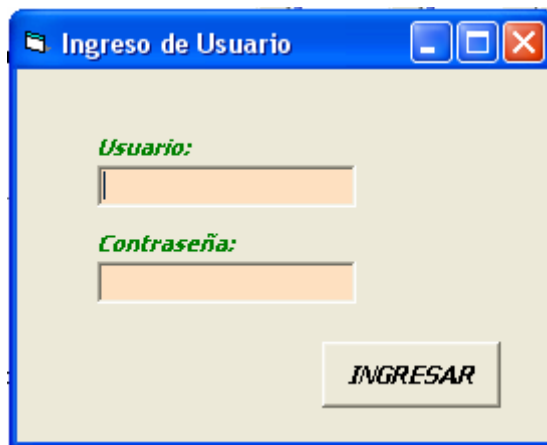


Figura F.1. Ingreso de Usuario.

Al presionar el botón **INGRESAR** se abrirá una pantalla con presentación de un menú de opciones, figura F.2.

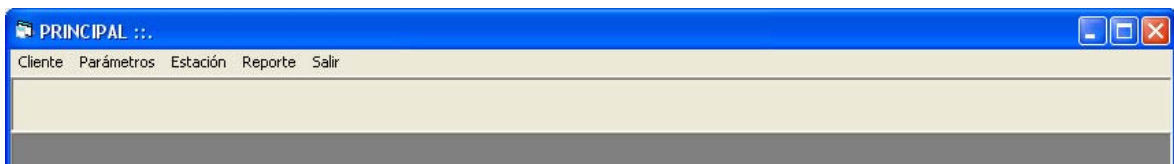


Figura F.2. Pantalla principal con opción de menú.

Desde esta pantalla nosotros podemos elegir la opción **Cliente**, la cual nos muestra un submenú con dos opciones **Nuevo** y **Buscar** (figura F.3)

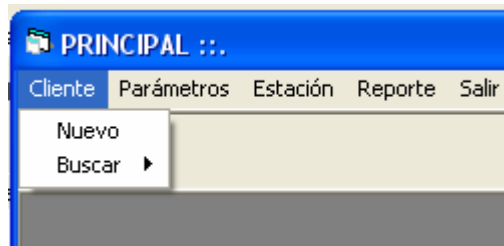


Figura F.3. Opción Cliente.

Dentro de la opción **Nuevo** del submenú, el usuario puede ingresar al sistema el registro de un cliente que por primera vez va a adquirir el servicio, donde posteriormente aparecerán las pantallas de ingreso de **Datos del Cliente**. (figura F.4, figura F.5).

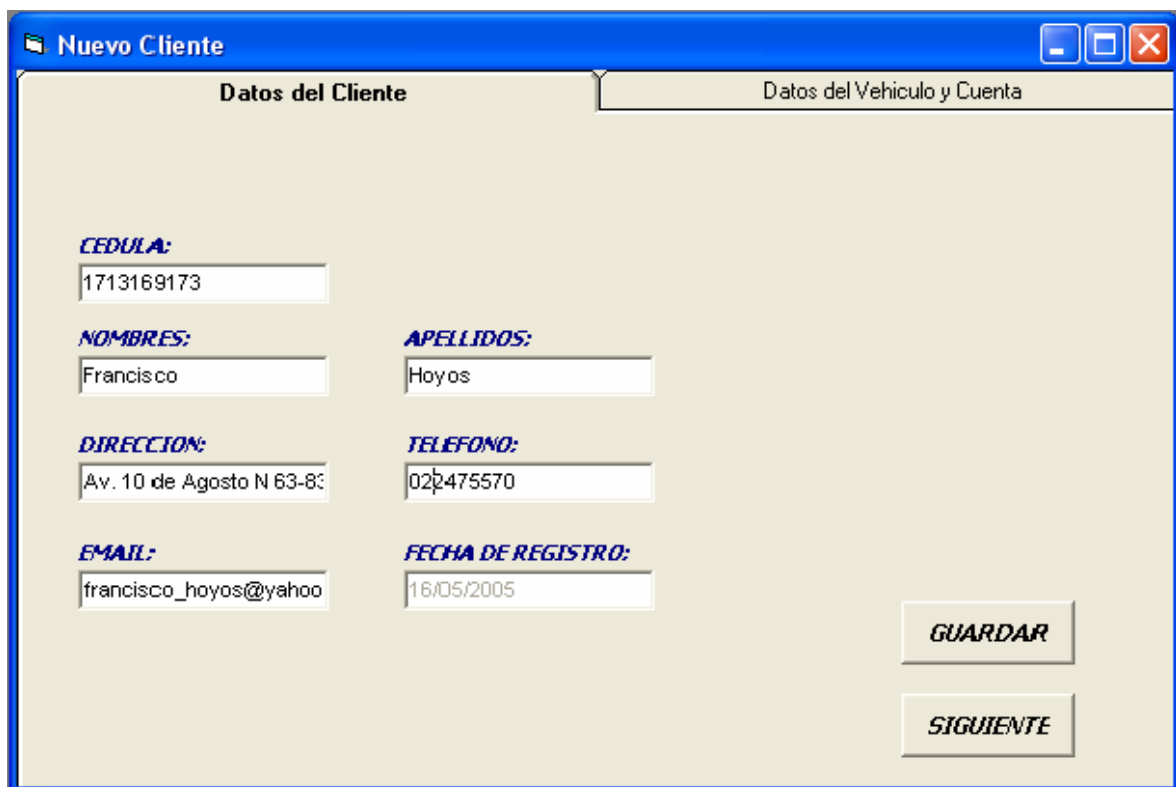
A screenshot of a software application window titled 'Nuevo Cliente'. The window has a tabbed interface with two tabs: 'Datos del Cliente' (active) and 'Datos del Vehiculo y Cuenta'. The 'Datos del Cliente' tab contains several text input fields with labels in blue: 'CEDULA:' (value: 1713169173), 'NOMBRES:' (value: Francisco), 'APELLIDOS:' (value: Hoyos), 'DIRECCION:' (value: Av. 10 de Agosto N 63-83), 'TELEFONO:' (value: 02475570), 'EMAIL:' (value: francisco_hoyos@yahoo), and 'FECHA DE REGISTRO:' (value: 16/05/2005). At the bottom right of the form, there are two buttons: 'GUARDAR' and 'SIGUIENTE'.

Figura F.4. Datos del Cliente.

Luego de haber ingresado los datos solicitados presione el botón **GUARDAR** que guarda los datos personales del nuevo cliente en la base de datos del sistema y posteriormente el botón **SIGUIENTE** el que lo llevará a la siguiente pantalla, o

simplemente presione la pestaña *Datos del Vehículo y Cuenta* ubicada en la parte superior derecha de la caja de diálogo.

The screenshot shows a software window titled "Nuevo Cliente" with two tabs: "Datos del Cliente" and "Datos del Vehículo y Cuenta". The "Datos del Vehículo y Cuenta" tab is active. It contains two main sections: "Cuenta" and "Vehiculo".

Cuenta section:

- CUENTA:** Text input field containing "1009005161".
- TIPO CUENTA:** Dropdown menu with "Corriente" selected.
- BANCO:** Dropdown menu with "Produbanco" selected.
- SALDO:** Text input field containing "500".

Vehiculo section:

- PLACA:** Text input field containing "PRA148".
- TIPO VEHICULO:** Dropdown menu with "pesado" selected.

Overlaid on the "Vehiculo" section is a smaller dialog box titled "Peaje" with a red close button. It contains the text "Registro guardado con éxito!" and an "Aceptar" button.

In the bottom right corner of the "Datos del Vehículo y Cuenta" tab, there is a button labeled "GUARDAR".

Figura F.5. Datos del Vehículo y Cuenta.

Después de haber ingresado cada uno de los datos del cliente y su vehículo presione el botón **GUARDAR** luego aparecerá una caja de diálogo indicando que el registro fue guardado con éxito en la base de datos del sistema. Con todos estos pasos cumplidos el usuario ya está registrado y puede hacer uso del servicio de peaje electrónico.

En el caso de que el cliente ya fue registrado anteriormente, y el administrador del sistema por error quiso registrarlo nuevamente el programa no lo permitirá señalando que el registro está repetido como se muestra en la figura F.6

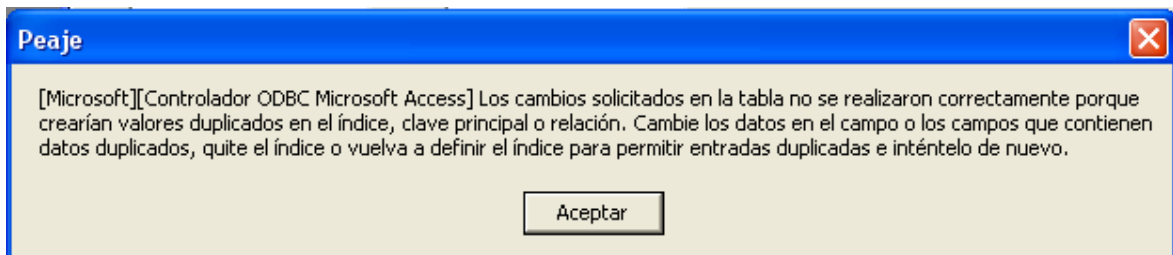


Figura. F.6. Mensaje de Error por registro repetido.

Si es el caso en que el administrador quiere hacer algún cambio en el registro o quiere añadir un nuevo vehículo al servicio de peaje electrónico, de su cliente. Él deberá elegir la opción cliente y dentro de éste el submenú **Buscar** figura F.7, que a su vez tiene dos opciones de búsqueda una **por cédula de identidad** y la otra **por apellido** del mismo.

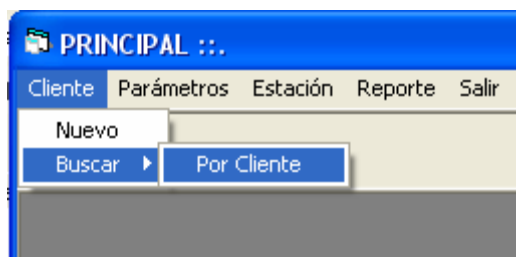


Figura. F.7. Opción de búsqueda de Clientes

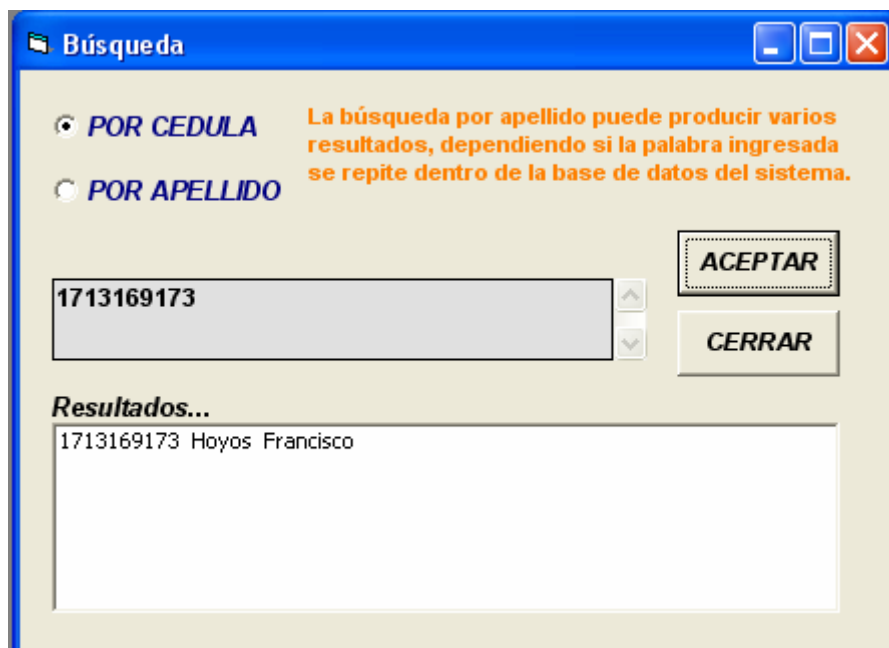


Figura F.8. Resultado de búsqueda POR CÉDULA.

En la búsqueda se maneja dos opciones de las cuales *por cédula de identidad* arroja un solo resultado como podemos observar en la figura. F.8 y varios resultados si elegimos búsqueda *por apellido*, figura F.9

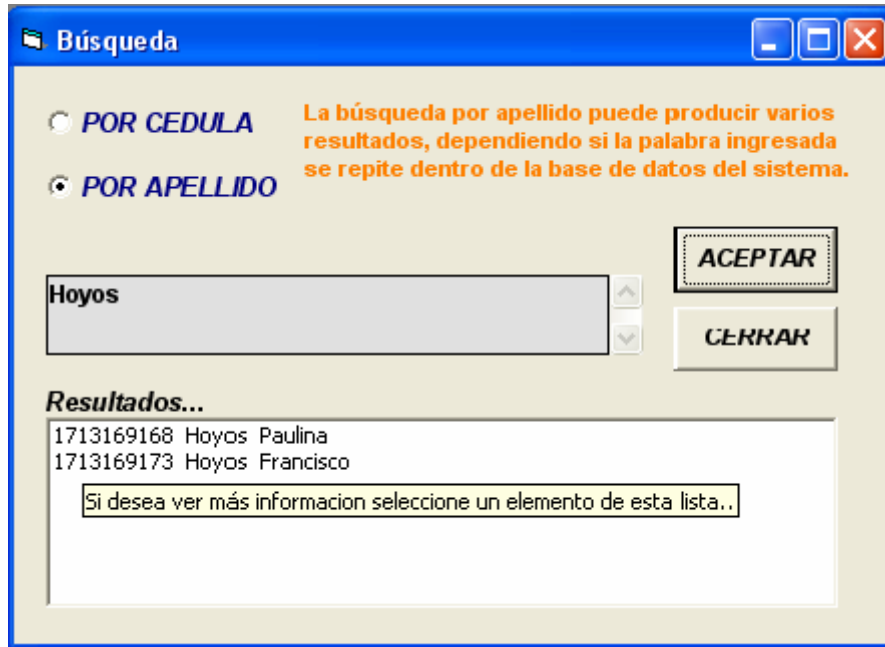


Figura F.9. Resultado de búsqueda POR APELLIDO.

Para hacer los cambios respectivos dentro del registro del cliente buscado, deberá hacer un doble clic en el resultado presentado dentro de la caja de **Resultados**. Llevándolo de esa manera nuevamente a la pantalla *Datos del Cliente* y *Datos del Vehículo y Cuenta*, figuras F.4 Y F.5 respectivamente, para revisar o cambiar el registro elegido.

Para continuar con la opción del menú principal tenemos ahora a la opción **Parámetros** la que contiene cuatro submenús dentro de él (figura F.10). En esta opción el administrador del peaje tiene la facultad de añadir, modificar, eliminar cada uno de los parámetros que afectan al sistema de peajes como son los indicados a continuación:



Figura F.10. Opción Parámetros del menú principal.

Dentro tenemos *Tipo Cuenta* figura. F.11.

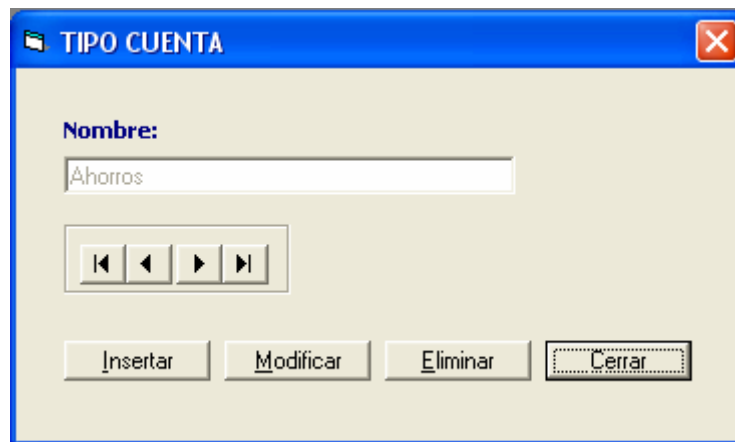


Figura F.11. Pantalla Tipo Cuenta.

En esta pantalla podemos insertar, modificar, eliminar los tipos de cuenta además podemos recorrer por todos los tipos de cuenta con la ayuda de los controles de desplazamiento que recorrerán todas las opciones.

La siguiente opción es *Tipo Vehículo* figura. F.12.



Figura F.12. Pantalla Tipo Vehículo.

En esta pantalla podemos insertar, modificar, borrar los tipos de vehículo además podemos ingresar una descripción del tipo vehículo para poder ubicar y diferenciar de mejor manera a cada uno de ellos. Los controles de desplazamiento nos ayudan a recorrer por todos los tipos de vehículo.

La siguiente opción es *Tipo Evento* figura. F.13.



Figura F.13. Pantalla Tipo Evento.

En esta pantalla podemos insertar, modificar, eliminar los tipos de eventos que puedan ocurrir en el sistema también se puede ingresar una descripción del tipo evento para

conocer de qué manera ellos se generan. Los controles de desplazamiento nos ayudan a recorrer por todos los tipos de eventos.

La siguiente opción es Banco figura. F.14.

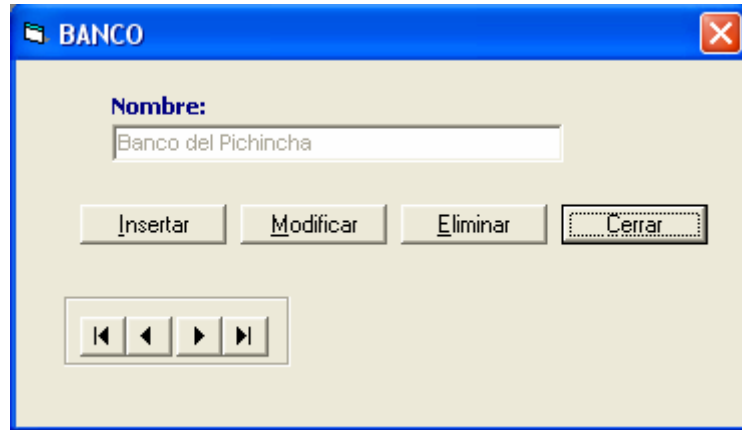


Figura F.14. Pantalla Banco.

En esta pantalla podemos insertar, modificar, eliminar los nombres de Bancos con los cuales el sistema puede trabajar para descontar el valor mensual del uso del peaje del cliente sin que este tenga que acercarse a cancelar su cuenta, solo deberá extender una solicitud de autorización para que la empresa de peajes pueda debitar de su cuenta el valor mensual de su consumo del servicio de peaje electrónico.

Además podemos recorrer por todos los nombres de los bancos añadidos al sistema con la ayuda de los controles de desplazamiento.

A continuación se muestra la opción *Estación*, una más de las opciones del menú principal la que contiene dos submenús que detallaremos a continuación. (figura F.15). En esta opción el administrador del peaje puede ampliar su sistema añadiendo dentro de su base de datos otros peajes ubicados dentro de provincia o en otras provincias.

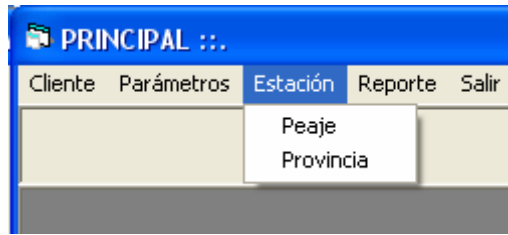


Figura F.15. Opción Estación del menú principal.

La opción *Peaje* figura. F.16.

A screenshot of a software application window titled 'PEAJE'. The window has a blue title bar with a close button (X) in the top right corner. The main area is a light beige form with the following fields:

- Nombre:** A text input field containing 'Ruminahui'.
- Ubicación:** A text input field containing 'Autopista Vía valle de los Chillos'.
- Provincia:** A dropdown menu with '1 Pichincha' selected.

Below the fields are four navigation buttons: a left arrow, a double left arrow, a right arrow, and a double right arrow. At the bottom of the form are four action buttons: 'Insertar', 'Modificar', 'Eliminar', and 'Cerrar'.

Figura F.16. Pantalla Peaje.

Dentro de esta pantalla nosotros podremos insertar, modificar, eliminar, nombres de peajes que se sumarán a nuestro sistema de peaje electrónico donde también podemos detallar su ubicación y elegir la provincia donde se encuentra.

La opción *Provincia* figura. F.17.

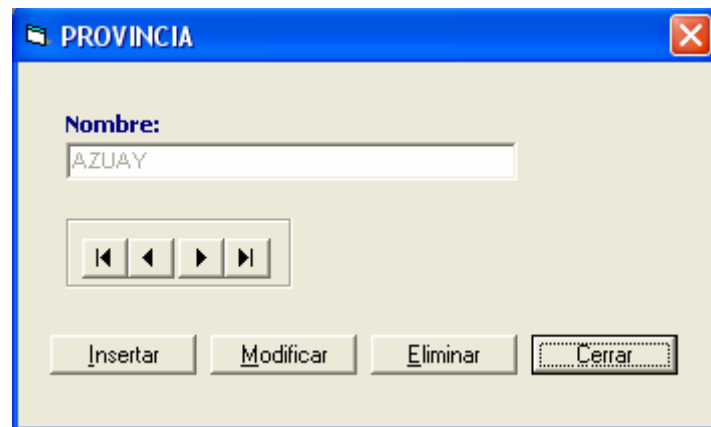


Figura F.17. Pantalla Provincia.

Dentro de esta pantalla nosotros podremos insertar, modificar, eliminar, las provincias que se sumarán a nuestro sistema de peaje electrónico.

A continuación se muestra la opción **Reporte**, una de las opciones del menú principal la que contiene dos submenús que detallaremos a continuación. (figura F.18). En esta opción el administrador del peaje puede elegir como sacar sus reportes.

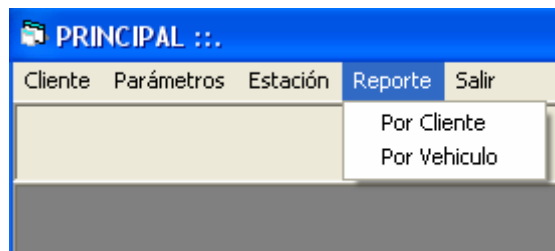


Figura F.18. Opción Reporte del menú principal.

Al elegir la opción **Por Cliente** se desplegará una caja de diálogo la cual nos pedirá la confirmación de esa opción de reporte figura. F.19.

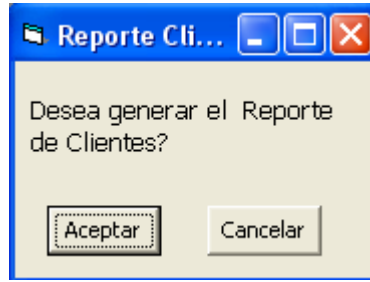


Figura F.19. Confirmación de opción del Reporte .

La cual después de haber aceptado la opción desplegará el reporte por clientes figura F.20.

Reporte de Clientes								
REPORTE DE CLIENTES								
Fecha de emisión: 16/05/2005								
Cedula	Nombres	Apellidos	Telefono	Direccion	Fecha de Registro	No de Cuenta	Fecha de Servicio	Placa
1713169168	Paulina	Hoyos	022475570	Av. 10 de Agosto N63-83	22/04/2005	1003005152	23/04/2005 23/04/2005	PGL233 PGL233
							Total de uso del servicio:	2
1803394871	Mario	Erazo		Av. Mariscal Sucre	17/03/2005	1234567890	04/04/2005 04/04/2005 04/04/2005 04/04/2005 04/04/2005 04/04/2005 22/04/2005	SRA148 SRA148 SRA148 SRA148 SRA148 SRA148 SRA148
							Total de uso del servicio:	8

Figura F.20. Presentación de Reporte por clientes.

Al elegir la opción *Por Vehículo* se desplegará una caja de diálogo la cual nos pedirá la confirmación de esa opción de reporte figura. F.21.



Figura F.21. Confirmación de opción del Reporte .

La cual después de haber aceptado la opción desplegará el reporte por vehículos figura F.22.

Reporte de Clientes								
REPORTE DE CLIENTES								
Fecha de emisión: 16/05/2005								
Cedula	Nombres	Apellidos	Telefono	Direccion	Fecha de Registro	No de Cuenta	Fecha de Servicio	Placa
1713169168	Paulina	Hoyos	022475570	Av. 10 de Agosto N63-83	22/04/2005	1003005152	23/04/2005 23/04/2005	PGI233 PGI233
							Total de uso del servicio:	2
1803394871	Mario	Erazo		Av. Mariscal Sucre	17/03/2005	1234567890	04/04/2005 04/04/2005 04/04/2005 04/04/2005 04/04/2005 04/04/2005 04/04/2005 22/04/2005	SRA148 SRA148 SRA148 SRA148 SRA148 SRA148 SRA148 SRA148
							Total de uso del servicio:	8

Figura F.22. Presentación de Reporte por vehículos.

Finalmente la última opción del menú principal *Salir*, la que contiene en su submenú dos opciones figura F.23.

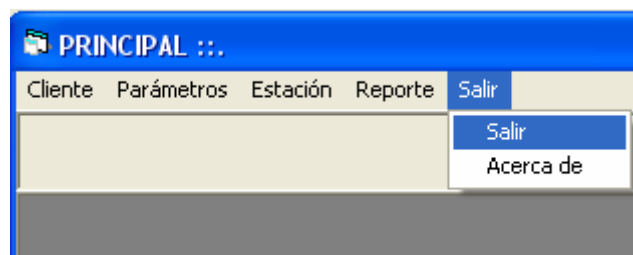


Figura. F.23. Opción Salir del menú principal

La opción **Salir** será la que finalmente nos permita salir del sistema de administración. Dentro de la opción **Acerca de** se detallan algunos aspectos informativos del sistema como por ejemplo la versión del mismo. (figura. F.24).

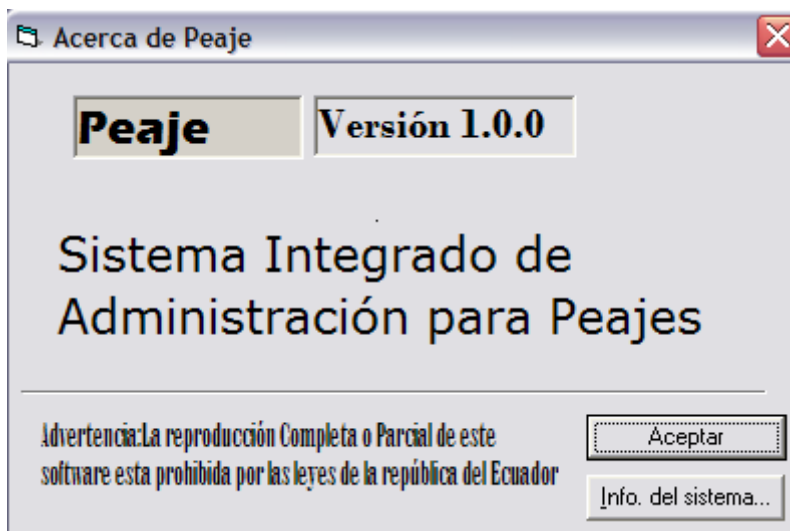


Figura. F.24. Pantalla Acerca de Peaje

ANEXO 7

MANUAL DEL USUARIO PARA

EL PROGRAMA DE

MONITOREO (MONITOR)

MANUAL DEL USUARIO PARA EL SOFTWARE DE MONITOREO

SOFTWARE PARA MONITOREO DE PEAJES (MONITOR)

Como se mencionó anteriormente se crearon dos programas en Visual Basic 6.0 con el objetivo de que el uno administre y el otro monitoree el sistema.

Se elaboraron dos software con la justificación que el software de administración no es necesario tenerlo activo todo el tiempo sino cuando se registre o modifique a un usuario, o cuando se quiera generar algún reporte; por el contrario, el software de monitoreo es necesario tenerlo activo las 24 horas, que es el tiempo en que funciona el sistema, puesto que este captura todos los eventos (alarma y descuento), que generan los usuarios.

El software de monitoreo tiene la función de monitorear el sistema de control electrónico de peaje, en el cual tenemos una única pantalla donde continuamente se están recibiendo los datos tanto de clientes (códigos) como de eventos de alarma del sistema, registrando además la fecha y hora en que ocurrió, figura G.1.



Figura. G.1. Interface del software de monitoreo

Para el correcto funcionamiento del software de monitoreo es preciso seguir los pasos que se describen a continuación:

- 1.- Montar sobre la tarjeta de adquisición de datos el módulo de recepción.
- 2.- Conectar el sensor y la sirena en la tarjeta de adquisición de datos
- 3.- Alimentar la tarjeta de adquisición de datos mediante el circuito de suministro de energía , la misma que se alimenta con la corriente de la red pública (110 V. o batería).
- 4.- Conectar el cable serial a la tarjeta de adquisición de datos y al puerto serial de la PC que contenga ya instalado el software de administración y monitoreo.
- 5.- Abrir y activar el software de monitoreo desde la PC.

ÍNDICE DE FIGURAS

Figura. 2.1. Diagrama de la arquitectura física de ITS en EE.UU.....	7
Figura. 2.2. Descripción gráfica de los componentes del sistema.....	9
Figura. 2.3. Ejemplo de Enlace Inalámbricos.....	12
Figura. 2.4. Diagrama de pines del rfPIC12F675.....	13
Figura. 2.5. Esquema del transmisor ASK típico.....	14
Figura. 2.6. Distribución de pines del rfRXD0420	18
Figura. 2.7. Descripción de pines del PIC 16F870.....	19
Figura. 2.8. Descripción de pines del PIC 16F627.....	20
Figura. 2.9. Estructura de un caracter que se transmite serialmente.....	22
Figura. 2.10. Diagrama de pines y estructura interna del MAX232	23
Figura. 2.11. Diagrama de conexión de pines del MAX232 con PIC.....	24
Figura. 3.1. Diagrama Funcional del Sistema.....	31
Figura. 3.2. Diagrama de Bloques del Sistema.....	32
Figura. 3.3. Diagrama de Bloques del subsistema de Transmisión de Datos.....	33
Figura. 3.4. rfPIC TM Development Kit.....	34
Figura. 3.5. PICKit 1 Flash Starter Kit.....	35
Figura. 3.6. Módulo Transmisor rfPIC12F675.....	36
Figura. 3.7 Descripción de conexión del rfPIC12F675 sobre el PICKit TM	37
Figura. 3.8 Interface del Software del Grabador.....	38
Figura. 3.9. Importar archivo .HEX.....	38
Figura. 3.10. Grabando el Dispositivo.....	38
Figura. 3.11. Módulo Receptor.....	39
Figura. 3.12. Esquema del transmisor ASK típico.....	40
Figura. 3.13. Diagrama de Flujo del Programa del Módulo transmisor.....	40
Figura. 3.14. Diagrama de Bloques del subsistema de Adquisición de Datos.....	48
Figura 3.15. Circuito de alimentación del bloque de Adquisición de Datos.....	49
Figura 3.16. Circuito de control de la sirena.....	51

Figura 3.17. Esquema de la configuración de los pines del PIC 16F870.....	52
Figura 3.18. Esquema de la configuración de los pines del PIC 16F627.....	52
Figura 3.19. Interfaz gráfica del Microcode Studio.....	53
Figura 3.20. Interfaz gráfica del MPLAB IDE V.6.50.....	54
Figura. 3.21. Diagrama de flujo del PIC 16f870.....	54
Figura. 3.22. Diagrama de flujo del PIC 16f627.....	55
Figura 3.23. Instrucción de recepción serial para la programación del PIC.....	57
Figura 3.24. Instrucción de transmisión serial para la programación del PIC.....	57
Figura. 3.25. Diagrama de flujo del Software MONITOR.....	58
Figura. 3.26. Interface del software de monitoreo.....	59
Figura 3.27. Tablas creadas para cada tipo de información.....	62
Figura 3.28. Configuración del tipo de datos.....	63
Figura 3.29. Clave principal.....	63
Figura 3.30. Relaciones entre tablas para reportes.....	64
Figura 3.31. Presentación de contenido de reportes.....	64
Figura. 3.32. Diagrama de bloques de la interfase grafica para la administración del sistema.....	68
Figura. 4.1. Suministro de Energía.....	69
Figura. 4.2. Sensor transmisor receptor.....	70
Figura. 4.3. Sensor de infrarrojos tipo reflex.....	70
Figura. 4.4. Módulos transmisor receptor de RF.....	71
Figura. 4.5. Tarjeta de adquisición de los datos.....	71
Figura. 4.6. Sirena 12 Vdc.....	72
Figura. 4.7. PC del sistema.....	72
Figura. 4.8. Formato del envío de datos.....	73
Figura. 4.9. Tiempo de bit de los datos.....	74
Figura. 4.10. Transmisión de la placa de vehículo.....	74

ÍNDICE DE TABLAS

Tabla. 2. 1. Banda de frecuencias de los rfPICs.....	13
Tabla. 2.2. Banda de frecuencias de los receptores.....	15
Tabla. 2.3. Ejemplo de tabla para base de datos.....	27
Tabla. 2.4. Asociación de Limitantes.....	29
Tabla. 3.1. Pines del módulo rfPIC12F675.....	36
Tabla. 3.2 Pines del módulo rfRXD0420.....	39
Tabla. 3.3. Características eléctricas del PIC 16F627 y de la SIRENA.....	50
Tabla. 3.4. Características eléctricas del opto-acoplador 4N25.....	50
Tabla. 3.5. Características eléctricas del TBJ darlington-npn tip122.....	51
Tabla. 5.1. Costos de Material Eléctrico y Electrónico.....	80
Tabla. 5.2. Costos de material de maqueta.....	80
Tabla. 5.3. Costos de Diseño y Software.....	80

GLOSARIO

A

ASK. Modulación por desplazamiento en amplitud.

B

Baudio. El número de bits de información transmitidos por segundo.

C

Centro de control. Es la estación donde se centraliza toda la información de un sistema multipunto.

D

DB. Data Base (Base de Datos)

DAQ. Data Acquisition (Tarjeta de adquisición de datos)

Darlington. Configuración de un circuito a base de transistores compartiendo colectores, formando un transistor de alta corriente; el emisor del primero ataca la base del segundo proporcionando alta impedancia de entrada y elevada ganancia.

E

EEPROM. Memoria de sólo lectura (ROM) programable, y borrable eléctricamente.

F

FCC. Federal Communication Commission (Comisión Federal de Comunicaciones de los Estados Unidos).

H

Hardware. Conjunto de todos los componentes tangibles del sistema.

HMI. Human Machine Interface (Interfaz Hombre Máquina).

I

IF. Frecuencia Intermedia

ISM. Industrial, Scientific and Medical (Bandas de Aplicaciones Industriales, Científicas y Médicas).

ITS. Intelligent Transport Systems (Sistemas Inteligentes de Transporte)

L

LO. Oscilador local.

M

Módem. Dispositivo que modula y demodula una señal.

N

NPN. Término utilizado para referirse a un semiconductor en el que una capa P está emplazada entre dos de tipo N.

O

Optoacoplador. Interfase que permite el acoplamiento de dos circuitos teniendo como medio la luz.

P

PA. Power Amplifier (amplificador de potencia)

PC. Computador personal.

PIC. Programmable Integrated Circuit (Circuito Integrado Programable). Dispositivo Digital que acepta o lee datos aplicados a cierto numero de líneas de entrada y los procesa de acuerdo a las instrucciones secuenciales de un programa almacenado en su memoria y suministra o escribe los resultados del proceso en un cierto número de líneas de salida.

PLC. Programmable Logic Controller (Controlador Lógico Programable).

Pila. Una estructura LIFO (último en entrar, primero en salir). Una pila se manipula necesariamente a través de subrutinas y manejo de interrupciones.

PNP. Término utilizado para referirse a un semiconductor en el que una capa N está emplazada entre dos de tipo P.

R

RF. Radio frecuencia.

RSSI. Indicador de potencia de señal recibida.

RS232. Estándar de comunicación entre dispositivos, donde los datos son transmitidos en forma serial como variaciones de voltaje.

S

Software. Conjunto de componentes intangibles del sistema, tales como programas y archivos.

Spread Spectrum. Técnica de modulación que dispersa su espectro sobre una gran gama de frecuencias.

T

Telemática. Rama de la electrónica moderna que combina tecnologías de telecomunicaciones e informática

TIA. Telecommunication Industry Association (Asociación de Industrias de Telecomunicaciones de los Estados Unidos).

U

USART. Universal Synchronous Asynchronous Receiver Transmitter (Transmisor Receptor Asíncrono Síncrono Universal).

V

VCO. Oscilador controlado por voltaje.

Sangolquí, Mayo del 2005

HOJA DE LEGALIZACIÓN

ELABORADO POR:

Sr. Francisco Javier Hoyos Mejía

AUTORIDADES:

TCRN. DE E.M. Ing. Marcelo Gómez
Decano de la Facultad de Ingeniería Electrónica

Dr. Jorge Carvajal
Secretario Académico de la Facultad de Ingeniería Electrónica