UNIVERSIDAD CATÓLICA ANDRÉS BELLO.

FACULTAD DE INGENIERÍA.

ESCUELA DE TELECOMUNICACIONES.

DISEÑO DE UN SISTEMA DE CONTROL DE OPERACIÓN DE UN AUTOMÓVIL ALQUILADO A TRAVÉS DE RFID

TRABAJO ESPECIAL DE GRADO

Presentada ante la:

UNIVERSIDAD CATÓLICA ANDRÉS BELLO.

Como parte de los requisitos para optar por el título de:

INGENIERO EN TELECOMUNICACIONES.

REALIZADO POR: Butrón S., Carlos M.

Rodríguez M., Carolina

TUTOR: Ing. Iñaki Mendizabal

FECHA: Caracas, 26 de Febrero 2010.

UNIVERSIDAD CATÓLICA ANDRÉS BELLO.

FACULTAD DE INGENIERÍA.

ESCUELA DE TELECOMUNICACIONES.

DISEÑO DE UN SISTEMA DE CONTROL DE OPERACIÓN DE UN AUTOMÓVIL ALQUILADO A TRAVÉS DE RFID

REALIZADO POR: Butrón S., Carlos M.

Rodríguez M., Carolina

TUTOR: Ing. Iñaki Mendizabal

FECHA: Caracas, 26 de Febrero 2010.

UNIVERSIDAD CATÓLICA ANDRES BELLO

FACULTAD DE INGENIERIA

ESCUELA DE INGENIERIA DE TELECOMUNICACIONES

Diseño de un sistema de control de operación de un automóvil alquilado a través de RFID

Este jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: 19 (Diecinueve puntos)

JURADO EXAMINADOR

Firma: **Firma:** **Firma:**

Nombre: LUIS FERNANDO **Nombre:** **Nombre:** IÑAKI MENDIZABAL

REALIZADO POR: Butrón S., Carlos M.

Rodríguez M., Carolina

PROFESOR GUIA: Ing. Mendizabal Iñaki

FECHA: Caracas, x de 2009.

Dedicatoria

Este proyecto representa una gran meta que he logrado culminar con éxito, por esto quiero dedicárselo a mis Padres José y María Josefina, que con su ejemplo, esfuerzo y dedicación lograron inculcarme valores que logre aplicar durante mi carrera.

A mis hermanas Heidi y Susana, por apoyarme siempre en cada momento, y darme fuerzas para levantarme si en algún momento no podía, A mi Abuela por ser un ejemplo a seguir, por ser alguien a quien admiro y por darme razones cada día para sentirme orgullosa de ser su nieta, y a mis Tías Cachi, Crisa y Mireya por siempre tenderme su mano amiga y tener un consejo para cualquier ocasión.

A mi novio y compañero de TEG por apoyarme en todo momento y trabajar siempre en equipo para culminar este proyecto con éxito

Carolina Rodríguez Méndez

Todo los esfuerzos realizados para culminar el Trabajo Especial de Grado, fueron inspirados en mi Abuela, cuya crianza me llenó de ganas de saber de todo un poco, en mi Madre, que con su sabiduría y cariño, me inculcó el amor a la familia, en mis tíos, que con su paternidad me mostraron el hombre que debía ser, a mis primos y hermanos que me ayudaron a disfrutar cada momento de mi vida, a mi novia que mejoró mi forma de ver al mundo y en todos aquellos que me tendieron su mano.

Gracias a la inspiración que provocaron en mi, pude culminar esta etapa, por eso mi trabajo se lo dedico a ustedes.

Carlos Butrón Salgado

Agradecimientos

Queremos agradecerle a Dios por permitirnos culminar esta meta con éxito.

Agradecemos a nuestros familiares, por su apoyo para culminar este logro y ser Ingenieros, por sus conocimientos para ayudarnos a tomar decisiones en nuestras vidas a nivel profesional y personal y por su fuerza para caminar con nosotros cursando una carrera y acompañarnos en el camino hasta culminarlo.

A el Ing Iñaki Mendizabal, tutor del Trabajo Especial de Grado por impartirnos sus conocimientos para realizar el proyecto con el mayor de los éxitos.

A los Ing. Carlos Arteta e Ing. José Gabriel Teriús por el apoyo brindado y por sus orientación y guía durante el desarrollo del trabajo.

Resumen

DISEÑO DE UN SISTEMA DE CONTROL DE OPERACIÓN DE UN AUTOMÓVIL ALQUILADO CON RFID

Butrón Salgado, Carlos Miguel

cartelecom@yahoo.es

Rodríguez Méndez, Carolina

rmcarola@gmail.com

El Diseño de un Sistema de Control de Operación de un Automóvil Alquilado con RFID, contempla el diseño y la elaboración de un sistema de monitorización que obtenga, analice y envíe los datos del computador de un auto de manera inalámbrica a una etiqueta RFID y está se encargue de radiar su información cuando el lector RFID se lo ordene.

Los datos recolectados del automóvil son: velocidad, revoluciones por minuto, nivel de gasolina, temperatura del motor y kilometraje recorrido con una luz indicadora de mal funcionamiento prendida, y la información que tiene la etiqueta son las violaciones de ciertos umbrales que varían según el parámetro. Esta información podrá ser usada para evaluar el manejo del conductor, realizar el mantenimiento adecuado del carro, y hasta multar al usuario si este excedió los límites establecidos.

El sistema está formado por dos subsistemas, por los cuales transita la información recolectada a través de la etiqueta RFID. El primero se encuentra en el automóvil y está formado por: el ELMSCAN5 que es el encargado de comunicarse con el computador del automóvil bajo los protocolos OBDII, el lector RFID que se encarga de guardar los datos en una etiqueta RFID y por el DsPIC30F4011 que dirige todos los procesos. El segundo subsistema se encuentra en la empresa que alquila automóviles y está conformado por el lector RFID que

lee la información de la etiqueta y un computador que le muestra la información del monitoreo a los usuarios.

Palabra Clave: RFID, OBDII, Automóvil y Monitorización.

Índice

Dedicatoria I
Agradecimientos II
Resumen III
Índice V
Índice de Figuras VII
Índice de Tablas VII
Introducción 1
I.- Planteamiento del Proyecto 3
I.1 Planteamiento del Problema 3
I.2 Objetivos 4
I.2.1 Objetivo General 4
I.2.2 Objetivos Específicos 5
I.3 Justificación 5
I.4 Alcances y Limitaciones 6
II.- Marco Referencial 9
II.1 *Radio Frecuency Identification* 9
II.1.1 Definición 9
II.1.2 Descripción del Sistema 11
II.1.2.1 *Hardware* 11
II.1.2.1.1 Etiqueta 11
II.1.2.1.2 Tipos de Etiquetas 12
II.1.2.1.2.1 Alimentación 12
II.1.2.1.2.2 Frecuencia de operación 14
II.1.2.1.2.3 Capacidad de lectura-escritura 14
II.1.2.1.3 Lectores 14
II.1.2.2 *Software* 16
II.1.2.2.1 *Software* del Sistema 17
II.1.2.2.2 *Middleware* 18
II.1.2.2.3 *Software* del Usuario Final 18
II.1.3 Clasificación de los Sistemas RFID 18
II.1.3.1 Clasificación según rango de información y la capacidad de procesamiento de datos 19
II.1.3.1.1 Clasificación de los sistemas de un solo bit 20
II.1.3.1.2 Clasificación de los sistemas de n bits 20
II.1.3.1.2.1 Clasificación según el tipo de memoria 22
II.1.3.1.2.2 Clasificación según los diferentes procedimientos para el envío de datos 23
II.1.4 Regulación y Estándares 23
II.1.4.1 Regulación 23
II.1.4.2 Estandarización 24
II.1.4.2.1 EPC 25
II.2 *On Board Diagnostic* (OBD) 28
II.2.1 OBD-I 28
II.2.2 OBD-II 29
II.2.2.1 PID 31
II.2.2.2 Protocolos OBD-II 32

II.2.3 Interfaz ELMSCAN 33
II.3 Microcontroladores 33
II.3.1 Definición 33
II.3.2 Clasificación de los microcontroladores 34
II.3.2.1 Según bus de datos 34
II.3.2.2 Según sus recursos 34
II.3.2.3 Según el set de instrucciones 35
II.3.2.3.1 Con instrucciones CISC 35
II.3.2.3.2 Con instrucciones RISC 35
II.3.2.4 Según su arquitectura interna 35
II.3.2.4.1 Arquitectura de Von Neumann 35
II.3.2.4.2 Arquitectura de Harvard 36
II.3.2.5 Según el fabricante 36
II.3.2.5.1 PIC de Microchip 36
II.3.2.5.2 AVR de Atmel 36
III.- Metodología y Desarrollo 37
III.1 Proceso de investigación y documentación 37
III.2 Diseño e Implementación 37
III.2.1 Descripción del Sistema 39
III.2.2 Selección del *Hardware* 41
III.2.2.1 Dispositivos de Administración 41
III.2.2.2 Lectores 42
III.2.2.3 *Tags* 43
III.2.3 Selección del *Software* 45
III.2.3.1 *Software* del DsPIC30F4011 45
III.2.3.2 *Software* de la Aplicación 45
III.2.4 Selección de la solución para la comunicación con el automóvil 46
III.2.5 Diseño del Sistema 46
III.3 Pruebas y Optimización del Proyecto 48
III.3.1 Prueba con el Carro 48
III.3.1.1 Prueba con el Carro y el Computador 48
III.3.1.2 Prueba con el Carro con el DsPIC30F4011 49
III.3.2 Pruebas con el Lector RFID 49
III.3.2.1 Pruebas con el Lector RFID y la Computadora, vía puerto serial . 50
III.3.2.2 Pruebas con el Lector RFID y el Computador, vía puerto ethernet51
III.3.2.3 Pruebas con el Lector RFID y DsPIC30F4011, vía puerto serial .. 52
III.3.3 Pruebas con el MAX232 53
III.3.4 Pruebas con el programador del DsPIC30F4011 54
III.4 Análisis de Resultados 54
IV.- Resultados 57
IV.1 Sistema 57
IV.1.1 Interfaz gráfica 61
V.- Conclusiones y Recomendaciones 65
V.1 Conclusiones 65
V.2 Recomendaciones 66
Bibliografía 69
Apéndices 71

Índice de Figuras

Figura 1 *Hardware* del Sistema RFID 11
Figura 2 Modelo de *tags* 12
Figura 3 *Software* del Sistema RFID 16
Figura 4 Esquema de los diferentes principios de operación de los sistemas RFID. 18
Figura 5 Conector hembra J1962 30
Figura 6 ELMSCAN 5 33
Figura 7 Estructura básica de los microcontroladores 34
Figura 8 Estructura de microcontroladores con arquitectura Von Neumann 35
Figura 9 Estructura de microcontroladores con arquitectura de Harvard 36
Figura 10 Diagrama del Sistema De Control 39
Figura 11 Prueba con el ELMSCAN 5 48
Figura 12 Prueba con el lector RFID y el puerto serial 50
Figura 13 Prueba con el lector RFID y el puerto ethernet 52
Figura 14 Especificaciones del MAX 232 53
Figura 15 Sistema de Control 57
Figura 16 Menú principal del Sistema de Control 59
Figura 17 Configuración de la dirección IP de la computadora de la empresa..60
Figura 18 Interfaz de la base de datos y del sistema de control 61

Índice de Tablas

Tabla I Relación entre las frecuencias y las aplicaciones de RFID 14
Tabla II Normas ISO para RFID 25
Tabla III Estándares EPC 27
Tabla IV Representación de los pines del conector OBD-II 31
Tabla V Protocolos según los pin 31
Tabla VI Comparación de los dispositivos de Administración 42
Tabla VII Comparación de los Lectores RFID 43
Tabla VIII Comparación de los *tags* RFID 44
Tabla IX Distribución de los datos en el *tag* 47
Tabla X Experimentos con el protocolo OBD 49
Tabla XI Comandos utilizados en la prueba con el puerto serial 51
Tabla XII Conexión del programador de microcontroladores 54
Tabla XIII Parámetros del DsPIC30F4011 58
Tabla XIV Herramientas del menú principal del sistema de control 60
Tabla XV Herramientas de la interfaz de la base de datos 61
Tabla XVI Herramientas del Sistema de Control 62

Introducción

Todas las empresas buscan ofrecer a sus usuarios, los mejores servicios a nivel de calidad y precio, cuidando sus interés y los del cliente. Esto es necesario para que las empresas que prestan servicios como es el caso del alquiler de vehículos, puedan cumplir los fines de la organización y posicionarse entre las mejores del mercado.

En Venezuela y en muchos otros países del mundo estás empresas mencionadas son fundamentales para la evolución y el desarrollo del turismo, ya que brindan al turista la posibilidad de movilizarse con facilidad en el entorno que van a conocer, por lo que estos las consideran una opción como medio de transporte. El turista se interesa en poseer un vehículo con el que puedan transitar, que se encuentre en buenas condiciones a nivel de mecánica y electrónica, por lo que las empresas deben ofrecer vehículos que dispongan de las características que busca el turista.

La compañía debe asegurar, que el automóvil entregado sea una fuente de seguridad y confort, para esto la empresa tiene que cumplir con el mantenimiento del vehículo, y los usuarios no deben abusar de su servicio.

En Venezuela, el último punto es el más difícil de hacer realidad, ya que las leyes no siempre son aplicadas a la perfección y sin ninguna excepción, lo que induce a muchos de los usuarios a romper las reglas, como estar a altas velocidades en zonas donde está prohibido; por lo que estas empresas consideran que no en todos los casos se puede confiar en la bondad y en la rectitud del carácter de los usuarios, ya que si esto fuera posible los accidentes automovilísticos por fallas del conductor, no representarían una de las principales causas de mortalidad en el país.

Es aquí donde este Trabajo Especial de Grado, intenta beneficiar a varios aspectos de la sociedad, al turista y a dichas empresas, ya que busca brindar una solución factible para la monitorización de los parámetros del carro, y así permitirles a las empresas evaluar el manejo del conductor, mantener el mantenimiento adecuado, y hasta multar al conductor o al responsable del

vehículo durante el tiempo de alquiler, si este excedió los límites de velocidad, entre otras violaciones.

Durante el desarrollo del proyecto se pretende presentar un sistema sencillo para el usuario de la empresa, que muestre los parámetros tomados del vehículo, haciendo uso de una interfaz en su puesto de trabajo entrelazadas con herramientas electrónicas en el vehículo que se comunican con el computador del mismo.

Con dispositivos electrónicos y algunas tecnologías como *Radio Frecuency Identification* (RFID) y *On Board Diagnostic* (OBD), se puede acceder a esta información, y a su vez se puede almacenar y presentar adecuadamente todos los datos necesarios.

Identificación por Radiofrecuencia (RFID) es un sistema de identificación inalámbrica que tuvo sus orígenes en la Segunda Guerra Mundial, gracias a esta tecnología, hoy en día se optimizan los procesos de inventario en las empresas más importantes del mundo. El sistema está formado por una etiqueta y un lector. La etiqueta generalmente esta enlazada a un producto y el lector es el encargado de leer y escribir todos los datos en la etiqueta.

El OBD (*On Board Diagnostic*) es un sistema de monitoreo de vehículo, que se creó con la intención de reducir las emisiones de gases del mismo, informando al usuario que existe un mal funcionamiento de algún sector, y a su vez ayudar a los técnicos automotrices en la identificación y diagnóstico de las fallas de funcionamiento. Actualmente se utiliza su evolución llamada OBD-II, la cual se explica a lo largo del proyecto, como parte fundamental para el diseño.

En este proyecto se presenta el desarrollo de un sistema de control que monitoriza los automóviles alquilados, utilizando las tecnologías anteriormente mencionadas.

I.- Planteamiento Del Proyecto

En el primer capítulo del Trabajo Especial de Grado, se presenta una descripción de las razones por las cuales se seleccionó el tema, tomando en cuenta los objetivos, los alcances y las limitaciones que definen la estructura del proyecto.

I.1 Planteamiento del Problema

Hoy en día el turismo para muchos países, es tan fundamental como lo es el petróleo para Venezuela. Es tan relevante que en una de las potencias petroleras del mundo como lo es Dubái, consideran que si este se acabara, su principal fuente de ingreso sería el turismo.

En Venezuela existen innumerables maravillas naturales, pero el primer paso para encaminar el turismo en el país, es concientizar a las personas y a las empresas de su importancia.

Las empresas que alquilan automóviles son unas de las primeras que deben ser capacitadas para el turismo, ya que facilitan el transporte al viajero. Estas tienen ciertos criterios para mantener el automóvil en el mejor estado posible, lo que beneficia tanto a la empresa como a los clientes.

Entre las principales inquietudes de cualquier empresa está la conservación de sus bienes. En el caso específico de las arrendadoras de automóviles, estas deben preocuparse por los siguientes hechos:

1. Los accidentes automovilísticos. Estos representan una pérdida para la empresa y además para la sociedad, ya que es una de las principales causas de mortalidad en el país; provocados principalmente por el exceso de velocidad.

2. El uso del automóvil. Conducir a altas velocidades y revolucionar el motor continuamente por encima de las 6000 RPM, disminuiría la vida útil del mismo.

3. El descuido del usuario. Cada automóvil cuenta con un tablero donde se encuentran los indicadores de la temperatura del motor, del nivel de la

gasolina y del mal funcionamiento de algún componente, si el usuario ignora estos parámetros podría producir fallas en el carro.

Con el análisis del uso que el conductor le da al automóvil; refiriéndose a esto como la evaluación de parámetros como la velocidad, las revoluciones por minuto, la temperatura y el kilometraje, se podría implementar un sistema de multas basado en la violación de estos parámetros y por consiguiente los usuarios tomarán sus previsiones para no pagar montos excesivos, además se podría personalizar el mantenimiento del automóvil según su uso y esto lograría que a largo plazo se incrementara el tiempo de vida de los automóviles, y la seguridad de los otros usuarios.

A pesar de estas ventajas las empresas no monitorean parámetros como los mencionados anteriormente para llevar un control del uso del carro alquilado por parte del beneficiario. Esto ocurre porque es necesario tener conocimientos en el área de la electrónica para poder obtener los datos ya mencionados, y la mayoría de los empleados de estas empresas no tienen dichos conocimientos.

La ingeniería en telecomunicaciones resuelve el problema de monitorización y agrega un beneficio importante, el cual es que los datos recolectados sean accesibles de forma inalámbrica.

Las soluciones para la transmisión de datos de forma inalámbrica dadas por la ingeniería en telecomunicaciones se encuentran en las siguientes tecnologías Wi-Fi, Bluetooth, IrDA, telefonía móvil, RFID, entre otras.

Por las características del sistema la tecnología seleccionada es la identificación por radiofrecuencia, ya que tiene ciertas ventajas que serán explicadas en el capítulo III

I.2 Objetivos

En este segmento del Trabajo Especial de Grado se presentan las metas que se pautaron para el desarrollo y la elaboración del sistema.

I.2.1 Objetivo General

Diseñar un sistema de almacenamiento y transmisión de parámetros de operación relevantes de un automóvil alquilado con tecnología RFID.

I.2.2 Objetivos Específicos

- ✓ Seleccionar la tecnología adecuada para la realización del proyecto.
- ✓ Escoger un dispositivo de almacenamiento y los parámetros relevantes para la aplicación
- ✓ Diseñar el sistema de almacenamiento y transmisión basándose en los estándares de la tecnología escogida.
- ✓ Diseñar análisis y presentación de los datos obtenidos.
- ✓ Implementar el sistema de almacenamiento, procesamiento, transmisión y presentación de la data adquirida.

I.3 Justificación

La realización de este proyecto se justifica en el hecho de que la monitorización de un vehículo automotor es muy útil en distintos campos. En este punto se presentan las ventajas que trae la implementación del proyecto en una compañía arrendadora de automóviles, entre ellas están:

1. Establecer un sistema de multas basado en los parámetros almacenados en el carro. Gracias a estas multas los usuarios serán más precavidos al conducir y por ende los automóviles estarán en mejores condiciones.
2. Los mecánicos conocerán el estado del automóvil al leer el dispositivo. Con este *software* el personal especializado en mecánica automotriz, sabrá cual es el estado del carro en el instante que el automóvil llega a la empresa.
3. Control del inventario automotor de la empresa. Las tarjetas RFID además de tener algunos parámetros del carro, contendrán la matrícula vehicular. Este hecho facilita la conexión entre la base de datos de la tarjeta RFID y la de la empresa, por esta razón el *software* permite tener la información del estado de cada vehículo que se encuentra en el lugar.

Además, si se relacionan la base de datos de la etiqueta RFID con la de los entes encargados del tránsito terrestre del país, este proyecto puede ser incluido en peajes y otros eslabones de la organización, obteniendo las siguientes ventajas:

1. Un oficial de tránsito puede observar el historial del automóvil con solo leer los parámetros guardados en la etiqueta RFID.
2. Los peajes pueden obtener la información de los automóviles, en tiempo real y de forma inalámbrica.
3. La gestión de multas será automatizada y facilitará el control de los conductores multados.
4. La localización de automóviles robados será más sencilla, produciendo así una reducción del índice delictivo del país.

I.4 Alcances y Limitaciones

Con este Trabajo de Especial de Grado se propone una solución para empresas de alquiler de vehículos con aplicación de la tecnología RFID. La aplicación permite tener información acerca del uso y abuso de los vehículos alquilados, de manera práctica.

El dispositivo de almacenamiento recibe los datos del medidor cada cierto tiempo, pero este tiene un espacio limitado para datos, por lo tanto la memoria solo contiene información de algunos parámetros durante un periodo de tiempo limitado, y el *tag* solo tendrá los datos recopilados más relevantes.

El software que utilizará el empleado de la empresa solo leerá una etiqueta a la vez

Este Trabajo Especial de Grado incluye la construcción de un sistema que transmite los datos almacenados mediante la tecnología antes indicada, a un computador personal.

Los equipos seleccionados para la realización del proyecto, satisfacen cada una de las necesidades del sistema, estas se especifican con detalle a lo largo del desarrollo del Trabajo Especial de Grado, sin embargo entre las más importantes encontramos que son de un tamaño ideal para colocarlos cercanos al puerto *On*

Board Diagnostic (OBD) y no interfiera con el espacio del conductor, soportan temperaturas por encima de la temperatura ambiente, ya que en el sitio que se colocan se producen temperaturas mayores a esta, son de baja potencia ya que la distancia entre el lector y el *tag* no sobrepasa algunos cm, son software libre lo que facilita la elaboración de los programas, los dispositivos no necesitan altos voltajes de alimentación , entre otros.

En la Universidad se cuenta con dispositivos RFID, que son los *tags* Alien 9540 y el lector Alien 9900, cuyas especificaciones se detallan en el Capítulo II; estos dispositivos cumplen con las necesidades técnicas del sistema; es decir, los *tags* son de lectura-escritura, los dispositivos están desarrollados en software libre, entre otras características; sin embargo sus dimensiones sobre todo las del lector RFID y su alimentación no son las ideales para el diseño.

Gracias a que los dispositivos que tiene la UCAB cuentan con las consideraciones técnicas mínimas para el diseño, pueden utilizarse sin ningún inconveniente en la realización del proyecto piloto, ya que no alterarán los resultados técnicos sino solo la presentación física; la cual puede solventarse con los equipos recomendados para el proyecto.

II.- Marco Referencial

Este Trabajo Especial de Grado, consiste en crear un sistema que pueda transmitir vía inalámbrica parámetros del automóvil alquilado, como velocidad, revoluciones por minuto (RPM), kilometraje recorrido con una luz indicadora de falla, gasolina y temperatura.

El proyecto básicamente consiste en dos partes, la primera es la recolección de los datos del automóvil y la segunda la transmisión de esos datos. Para cumplir estas tareas se utilizó el protocolo *On Board Diagnostic* (OBD) para obtener los parámetros del carro y la tecnología *Radio Frecuency Identification* (RFID), para transmitir la información cuando sea necesario. Los principios básicos de estas tecnologías se describirán a continuación.

II.1 *Radio Frecuency Identification*

A través de los años las tecnologías avanzan rápidamente, optimizando los equipos a utilizar con la intención de cubrir las necesidades del mercado que deseen atacar.

Las bases que fundamentan el sistema RFID se remontan a aplicaciones que se le daban a unos radares para evaluar el acercamiento de aviones durante la II Guerra Mundial.

El problema presentado durante la II Guerra Mundial, permite tomarlo como antecedente para RFID; ya que no había manera de identificar de que bando eran los aviones; por lo que el ejército alemán descubrió que si los pilotos balanceaban sus aviones al volver a la base cambiaría la señal de radio reflejada de vuelta, y de esa manera podrían distinguir a los aviones alemanes de los que no lo eran, logrando identificarlos y ser la base fundamental para el desarrollo de teorías, como el comportamiento de un RFID pasivo, entre otros.

II.1.1 Definición

La identificación por radio frecuencia, es una tecnología de identificación inalámbrica que usa un dispositivo móvil y uno generalmente fijo, capaces de almacenar, transmitir y recibir datos.

Los dispositivos principales que conforman el sistema RFID son la etiqueta y el lector. La etiqueta tiene como función principal almacenar datos y el lector es el encargado de leer y escribir en la etiqueta. (Engineers, 2008)

El funcionamiento del sistema se puede describir en los siguientes pasos:

1. El lector por medio de su antena emite ondas electromagnéticas a una frecuencia que puede ir desde los 50 KHz hasta los 2.5 GHz, y tiene como función recibir y procesar los datos guardados en la etiqueta
2. La antena de la etiqueta convierte la señal electromagnética enviada por el lector en energía eléctrica. Generalmente es utilizada como fuente de poder (esto solo pasa en las etiquetas pasivas, las activas y las semi-pasivas tienen una fuente de poder en su estructura y por eso son capaces de iniciar una conversación), y a través de ella se envía la información que contiene en su memoria a través del aire.

La comunicación entre el lector y la etiqueta descrita anteriormente tiene determinadas características como son: la velocidad de transmisión, la frecuencia de operación, la cantidad de información enviada, la separación entre la etiqueta y el lector.

Para ilustrar el párrafo anterior se puede decir que la separación entre el emisor y el receptor al usar una etiqueta activa puede ser de 30 m y al usar una etiqueta pasiva el alcance es menor de 9 m. (García Candel, 2006)

La información que generalmente transporta esta tecnología está relacionada con las características de un producto, por esta razón la descripción del funcionamiento del sistema RFID encontrada en los libros es muy parecida a la mostrada a continuación: la comunicación en este sistema se logra mediante la transmisión de datos entre un producto y un computador utilizando ondas de radio. (Engineers, RFID Engineers, 2008)

Es importante saber que usualmente la tecnología RFID es utilizada para optimizar el proceso de registrar los productos en una base de datos (inventario de productos), pero puede ser aplicada en muchos campos, como por ejemplo el rastreo de maletas, monitoreo de ganado, entre otros.

II.1.2 Descripción del Sistema

El análisis del sistema RFID se puede dividir en dos partes, el *hardware* y el software.

El *hardware* de un sistema, son todos los dispositivos físicos que lo conforman. En el caso de la tecnología RFID, el *hardware* está compuesto básicamente por la etiqueta, las antenas y el lector como puede observarse en la figura 1.

El *software* de un sistema, son todas las aplicaciones informáticas que lo conforman. En el caso de la tecnología RFID, existen 3 tipos de *software*, el del sistema, el *middleware* y el del usuario final (Ver figura 3).

II.1.2.1 *Hardware*

Figura 1 *Hardware* del Sistema RFID (Fuente Propia)

II.1.2.1.1 Etiqueta

La etiqueta RFID, también denominada *TRANSPONDER* o *TAG*, es la encargada de almacenar, recibir y enviar información. En algunos casos estos dispositivos son programables, y esto dependerá de sus características.(Ver Figura 2)

La etiqueta debe tener la capacidad de comunicarse de forma inalámbrica y de guardar los datos obtenidos durante la comunicación. Para cumplir estas funciones, la etiqueta está formada por una antena y por un chip. (ALiens, 2009)

Figura 2 Modelo de *tags* (ALiens, 2009)

Chip: es un elemento semiconductor capaz de guardar información del producto al que esta enlazado.

La información del producto es enviada por el lector y es guardada en la etiqueta. Para que este procedimiento sea exitoso el dispositivo debe contar con una memoria no volátil en donde se guardan los datos, una ROM que sirve para guardar las instrucciones básicas para su funcionamiento y una RAM que tiene la cualidad de rápido acceso.

El dispositivo también cuenta con módulos electrónicos que sirven para procesar la señal (filtros, moduladores) y en algunos casos también tienen una fuente de alimentación.

Antena: es el dispositivo capaz de convertir las señales eléctricas en electromagnéticas y viceversa, por eso es el punto de entrada y de salida de toda la información que circula por el sistema.

Hay que recordar que las dimensiones de una antena están relacionadas con la frecuencia de operación del sistema. Por esta razón las antenas utilizadas por la tecnología son de un tamaño muy reducido (en rangos de altura de 1 a 3 cm y de ancho de 2 a 10 cm), y esto permite hacerlas en circuitos impresos para facilitar su integración a la etiqueta. (García Candel, 2006)

II.1.2.1.2 Tipos de Etiquetas

Se pueden clasificar en tres grupos, según la alimentación, según la frecuencia de operación y según la capacidad de lectura-escritura. (Samá Casanova & Ciudad Herrera, 2007)

II.1.2.1.2.1 Alimentación

Las etiquetas activas son aquellas que poseen su propia fuente de energía, que se utiliza para darle corriente a sus circuitos integrados y para enviar su señal al lector.

Gracias a su fuente de energía tienen la capacidad de transmitir señales con mayor potencia que las etiquetas pasivas, lo que indica que son más eficientes a la hora de la transmisión en entornos dificultosos para la radiofrecuencia. A la misma vez pueden cubrir distancias mayores pudiendo generar respuestas claras a partir de recepciones débiles.

Esta clasificación de etiquetas tienen rangos efectivos de decenas de metros (10m-100m), una capacidad de almacenamiento hasta de un 1 Mb pero posee una vida útil limitada que según los proveedores es de hasta 10 años. (Engineers, 2008)

Por sus características estos son utilizados en aplicaciones donde la distancia entre el emisor y el receptor es grande, como por ejemplo en autopistas.

Las etiquetas pasivas no poseen ninguna fuente de alimentación eléctrica. La señal que les llega de los lectores induce una pequeña corriente eléctrica, suficiente para operar el circuito integrado CMOS de la etiqueta, lo que les impide iniciar la comunicación.

Debido a lo mencionado anteriormente tienen distancias de uso práctico de unos pocos metros, dependiendo de la frecuencia de funcionamiento, el diseño y tamaño de la antena.

Como no tienen ningún módulo de alimentación estas etiquetas son más pequeñas que las etiquetas activas.

Las etiquetas semi-pasivas al igual que las activas poseen una fuente de alimentación propia, que se utiliza principalmente para alimentar al microchip para que pueda medir algunos parámetros.

La energía contenida en las ondas electromagnéticas es tomada por las etiquetas para poder emitir señales hacia el lector, como en una etiqueta pasiva. Un uso alternativo para la batería es almacenar información propagada desde el lector para emitir una respuesta en el futuro.

Estas etiquetas generalmente integran sensores de humedad, vibración, luz, radiación, temperatura y componentes atmosféricos como el etileno. Este tipo de

etiquetas suelen durar más que las activas, como también poseen una gran confiabilidad (García Candel, 2006)

II.1.2.1.2.2 Frecuencia de operación:

El rango de frecuencia en la que trabajará el sistema definirá la velocidad de transmisión, este puede estar dividido en 4 frecuencias muy generales: baja, alta, ultra y microondas, y dependerá de la aplicación específica, y pueden verse en la Tabla I. Mientras la frecuencia sea más alta la velocidad aumentará pero el desvanecimiento de la onda empeorará, aspecto importante que se debe tomar en cuenta.

Clasificación	Rango de Frecuencia	Aplicación
Baja	120KHz a 134KHz	Identificación de Animales
Alta	13,56MHz	Control de Acceso
Ultra	868 MHz y 956 MHz	Control de Acceso y Peaje
Microondas	2,5GHz	Control Automotriz

Tabla I Relación entre las frecuencias y las aplicaciones de RFID (Vega Anderson, Karina, & Mendizábal, 2009)

II.1.2.1.2.3 Capacidad de lectura-escritura:

Algunas etiquetas son solo de lectura o de lectura-escritura, estas últimas pueden usarse para varias aplicaciones entre ellas están: la monitorización de la temperatura y el seguimiento de un dispositivo y las de solo lectura tienen un campo de aplicación más restringido como por ejemplo la identificación de un producto.

Es importante destacar que tienen una memoria interna en bytes, para poder almacenar los datos que puede variar de decena de bytes a millares de bytes, todo depende del modelo. Se hace referencia a esto, porque según la capacidad de memoria en bytes se relaciona la función de lectura o lectura y escritura.

Los *tags* que tienen la característica de poder reprogramarse cada cierto tiempo, cuentan con un parámetro fundamental que son los ciclos de escritura, el cual indica la cantidad de veces que al *tag* se le puede modificar su información.

II.1.2.1.3 Lectores:

Un lector RFID o interrogador, es el elemento encargado de transmitir y recibir ondas electromagnéticas, para cumplir las funciones de detección, lectura y escritura de las etiquetas que están en su alrededor.

Entre las características de un lector RFID están:

- ✓ Sensibilidad: permite detectar señales procedentes de las etiquetas de -60dbm de potencia, ya que es la mínima potencia registrada por ellos.
- ✓ Selectividad: Debe tener la capacidad de seleccionar la señal procedente de los etiquetas dentro de una gran cantidad de señales recibidas. Este tópico es de mucha importancia ya que las frecuencias RFID trabajan cerca de las frecuencias de telefonía, por lo que podría existir interferencia.
- ✓ Alcance Dinámico: debe detectar y seleccionar señales procedentes, al mismo tiempo, de varias etiquetas RFID, que se encuentren a distancias diferentes, con lo que las potencias de etiquetas pueden diferir en un factor mayor de 10000.
- ✓ Operatividad en entornos Densos de lectores RFID. Debería poder soportar la interferencia de otros lectores RFID
- ✓ Inter Operatividad Multi-Fabricante: debe trabajar con todas las etiquetas que trabajen a su frecuencia sin importar el fabricante.
- ✓ Trabajar bajo normativas: debe tener la selección correcta del ancho de frecuencia a trabajar como la potencia máxima.

Conociendo los parámetros más importantes según su variación, existen diferentes tipos de lectores RFID entre los que encontramos: (Engineers, 2008)

Lectores RFID fijos: son aquellos que salieron en los principios de este sistema y que en la actualidad siguen siendo de mucha utilidad. Estos tienen tres subgrupos:

- ✓ Lectores básicos, con la simple funcionalidad de envío de señal y recepción de la respuesta de la etiquetas. Para 1 ó 2 antenas.
- ✓ Lectores funcionales; con funciones adicionales como regulación de la potencia de la antena, código EPC, filtrado, etc. Generalmente tienen aproximadamente 30 dB de salida en el conector.
- ✓ Lectores *Performance*; con funciones adicionales de conectividad XML, memoria flash, alta velocidad de proceso, ejecución de aplicaciones en Java, gestión de otros dispositivos.

Lectores RFID móviles: son aquellos que permiten su movilización tales como los industriales, carretillas o como dispositivos de lectura manual.

Lectores RFID de mano: existen diferentes modelos con y sin lector de barcode. Dispositivos para entornos duros, con Windows CE, bluethooth, wifi con y sin teclados, todos compatibles con RFID *data suite*. (Engineers, Dispositivos: Antenas RFID, Lectores RFID y Portales RFID, 2008)

II.1.2.2 *Software*

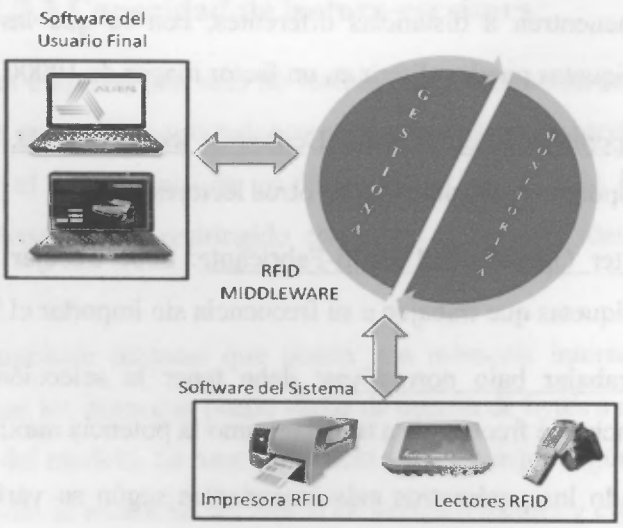


Figura 3 *Software* del Sistema RFID (Fuente Propia)

II.1.2.2.1 *Software* del sistema

El *Software* del Sistema RFID es la aplicación que se encuentra en la memoria del lector, y se encarga de controlar las operaciones básicas para establecer y mantener una comunicación exitosa entre el lector y la etiqueta.

Las operaciones que controla este *software* son las siguientes:

- ✓ Lectura-escritura: Es una de las principales operaciones del sistema. El lector pide permiso para leer o escribir sobre la etiqueta. En el caso de que se quiera leer, la etiqueta busca la información en su memoria y la transmite, y en el caso de que se quiera escribir, el lector envía los datos y la etiqueta los almacena en su memoria.

  También se puede escribir en la etiqueta desde el *software* utilizado por el usuario.

- ✓ Anticolisión: Esta operación es muy importante cuando en el campo de radiación del lector están múltiples etiquetas. El *software* debe gestionar el proceso de tal forma que las etiquetas no se interfieran entre sí. Generalmente cada etiqueta espera un tiempo aleatorio para responder la petición del lector.

- ✓ Detección y Corrección de errores: Esta operación debe ser capaz de detectar si los datos obtenidos durante la comunicación lector-etiqueta son correctos y si no lo son, debe corregir cualquier tipo de error.

- ✓ Seguridad: Esta operación debe ser capaz de cifrar los datos para que solo lo puedan leer el lector y la etiqueta, autorizar el intercambio de información solo con los dispositivos de interés y autenticar cada elemento del sistema para que no todos los dispositivos puedan leer o escribir en las etiquetas.

  Para lograr este objetivo tanto la etiqueta como el lector deben cooperar para que el protocolo ejecutado sea exitoso.

II.1.2.2.2 *Middleware*

Este *software* se encuentra en los equipos que se encargan de monitorizar el estado de los dispositivos y de gestionar el flujo de información, es decir, es un puente entre el sistema RFID y la aplicación del usuario final.

Monitorizar el sistema consiste en avisar de alguna forma que los dispositivos no están funcionando correctamente y gestionar el flujo de datos consiste en codificar, recolectar, procesar y filtrar la información transmitida entre las etiquetas y los lectores para que pueda ser transmitida y entendida por la aplicación del cliente. (García Candel, 2006)

II.1.2.2.3 *Software* del Usuario Final

Este es el *software* que se encuentra en la computadora del usuario y recibe la información de la etiqueta procesada y normalizada a través del lector y del *middleware* RFID.

El *software* del usuario final es la aplicación encargada de procesar y presentar los datos obtenidos de la etiqueta al usuario y este depende del tipo de información recolectada.

II.1.3 Clasificación de los Sistemas RFID

Los sistemas RFID se clasifican según: el sistema de memoria que incorpora el *transponder*, el rango de información y la capacidad de procesamiento que tiene el *transponder*.

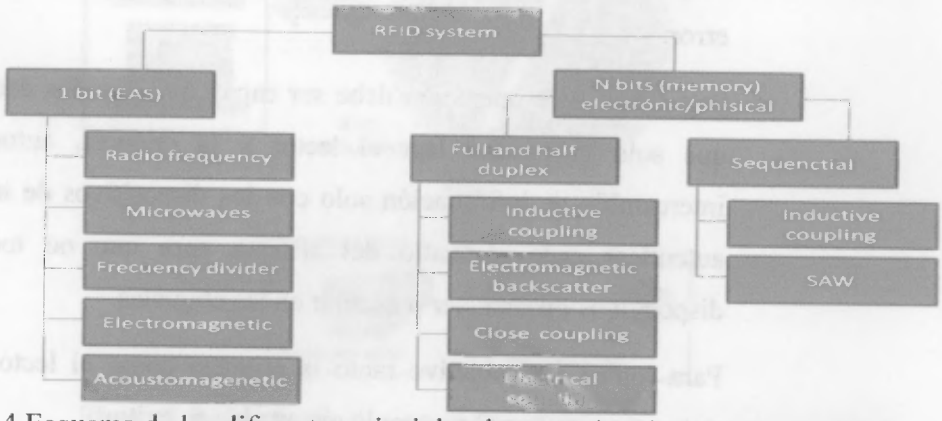


Figura 4 Esquema de los diferentes principios de operación de los sistemas RFID.

(Samá Casanova & Ciudad Herrera, 2007)

II.1.3.1 Clasificación según rango de información y la capacidad de procesamiento de datos

Es importante estudiar el rango de información y la capacidad de procesamiento de datos que ofrece el *transponder*, así como el tamaño de su memoria de datos. Realizando esta clasificación obtenemos un amplio espectro de variantes que se dividen en sistemas *Low-end*, *Mid-range* y *High-end*.

- ✓ Sistemas *Low-end*: son sistemas que reconocen la presencia de un artículo en la zona de alcance del lector, este grupo lo conforman principalmente los sistemas EAS (*Electronic Article Surveillance*), aparte de estos también encontramos los *transponders* de solo lectura, estos tienen un número de serie en su memoria que no puede ser cambiado.

  Estos sistemas son útiles para aplicaciones que necesitan cantidades de información pequeñas y son capaces de trabajar en todo el rango de frecuencias que opera RFID.

- ✓ Sistemas *Mid-range*: son sistemas que poseen una memoria con una capacidad de hasta 100Kbyte EEPROM o SRAM, lo que permite la escritura en la memoria.

  Los *transponders* de este tipo son capaces de procesar comandos simples de lectura para la selectiva lectura/escritura, aparte de soportar procesos de anticolisión. Por seguridad en estos sistemas se usan procedimientos de encriptación de datos y autenticación. Estos sistemas son capaces de trabajar en todo el rango de frecuencias que opera RFID.

- ✓ Sistemas *High-end*: contienen microprocesadores y un sistema de funcionamiento de tarjeta inteligente. Por seguridad en estos sistemas se usan procedimientos de encriptación de datos y autentificación. Estos sistemas operan en una frecuencia de 13.56 MHz.

Generalmente los sistemas RFID tienen un rango entre pocos y centenares de Kbyte como capacidad de información; sin embargo existen sistemas que solo necesitan un bit de información, lo necesario para detectar la presencia o la ausencia del *transponder*. Esto nos permite clasificar los *transponder* a aquellos que solo utilizan un bit y los que utilizan N bits como se muestra en la figura 4.

II.1.3.1.1 Clasificación de los sistemas de un solo bit

Los sistemas de un solo bit pueden clasificarse según su principio de funcionamiento:

- ✓ Procedimiento RF: basado en unos circuitos resonantes LC ajustados a una determinada frecuencia de resonancia.
- ✓ *Electromagnetic types*: que usan campos magnéticos muy fuertes en el rango VLF (10 Hz-20KHz), los elementos de seguridad contienen una línea metálica que sufren una saturación magnética ya que está sometida a un campo magnético muy potente y alternante, lo que origina unos armónicos a la frecuencia básica del lector.
- ✓ Sistema acústico magnético: están conformados por pequeñas cajas de plástico que contienen dos líneas metálicas, una de ellas produce vibraciones al pasar por un campo magnético debido a que no se encuentra conectada directamente a la caja. La amplitud generada en esta vibración varía según la frecuencia del campo electromagnético alterno producido por el lector, lo que corresponde con la frecuencia de resonancia de línea metálica.

Una de las principales aplicaciones es el EAS para la protección de objetos en tiendas y negocios. (Samá Casanova & Ciudad Herrera, 2007)

II.1.3.1.2 Clasificación de los sistemas de n bits

En el caso de los sistemas que utilizan N bits, ellos necesitan de un microchip electrónico como sistema portador de datos. Para leer o escribir en estos sistemas se realiza una transferencia de datos entre lector y *transponder*, esta transferencia puede seguir tres procesos:

- ✓ Sistemas *half duplex*: en este proceso el lector es quien inicia la comunicación. El *transponder* responde en *broadcast* cuando detecta el campo RF, el campo generado por la etiqueta es bastante débil comparado con el que genera el lector, por lo que este debe ser capaz de reconocer las dos señales. En la práctica la transferencia de datos se realiza por modulaciones con portadora o subportadora, pero también con armónicos de la frecuencia de transmisión del lector. En este sistema a diferencia de otros la transferencia de datos entre el lector y el *transponder*, se alterna con la comunicación entre ellos. Suelen usar modulaciones de carga con o sin subportadora, y armónicos.

- ✓ Sistemas *full duplex*: tienen un comportamiento muy similar al *half duplex*, la diferencia es que la comunicación *transponder*/lector se realiza al mismo tiempo que la comunicación lector/*transponder*. Incluye procedimientos más complicados ya que en la transferencia de datos se realiza mediante una fracción de frecuencia del lector, en subarmonicos o en frecuencias completamente distintas, no armónicos.

Estos sistemas utilizan como principios de funcionamiento para la transmisión y recepción de datos, el acoplamiento inductivo, *backscatter, close coupling y electrical coupling.*

- ✓ Sistemas Secuenciales: En este proceso el *transponder* es alimentado por pulsos, lo que indica que el campo generado por el lector se enciende y se apaga en intervalos periódicos. De esta manera el *transponder* se comunica con el lector en los intervalos en los que él no se está comunicando. El problema con estos sistemas es que en los intervalos que se corta la comunicación se pierde mucha energía, sin embargo puede mejorarse con una alimentación externa.

Estos sistemas utilizan como principios de funcionamiento para la transmisión y recepción de datos, el acoplamiento inductivo y SAW (*Surface Acoustic Wave*); basado este último en el efecto piezoeléctrico y una dispersión en la superficie de las ondas acústicas a pequeña velocidad. (Samá Casanova & Ciudad Herrera, 2007)

II.1.3.1.2.1 Clasificación según el tipo de memoria

Existe otro elemento importante para la clasificación de los sistemas RFID que es el tipo de memoria que usa el *transponder*, EEPROMs, FRAMs o SRAMs.

- ✓ EEPROMs: son las siglas de *Electrically-Erasable Programmable Read-Only Memory*. Es un tipo de memoria ROM (*Read Only Memory*) no volátil, que permite que pueda ser programada, borrada y reprogramada eléctricamente.

  Estas son las memorias más utilizadas en acoplamiento inductivo. Tiene un número limitado de ciclos de escritura (100.000 y 1.000.000) y a su vez un alto consumo de energía durante la operación de escritura.

- ✓ FRAMEs: son las siglas de *Ferromagnetic Random Access Memory*. Es muy parecida a las EEPROMs; sin embargo tiene un consumo de energía de 100 veces menor y un tiempo de escritura de 1.000 veces menor.

- ✓ SRAMs: son las siglas de *Static Random Access Memory*, o Memoria Estática de Acceso Aleatorio. Es un tipo de memoria basada en semiconductores que es capaz de mantener los datos (mientras esté alimentada) sin necesidad de un circuito de actualización (no se descargan). Sin embargo, son memorias volátiles, lo que permite que pierdan la información si se les interrumpe la alimentación eléctrica. Estas facilitan rápidamente el acceso a los ciclos de escritura; pero necesitan de un suministro de energía ininterrumpido de una batería auxiliar. (Samá Casanova & Ciudad Herrera, 2007)

En sistemas programables la lectura, escritura y la autorización se realizan mediante lógica interna. Mediante máquinas de estado generalmente, se pueden realizar secuencias complejas, pero no posibilita cambios en el programa sin realizar cambios en el *layout*. El uso de microprocesadores mejora este problema, incluyendo *software* para cada aplicación.

II.1.3.1.2.2 Clasificación según los diferentes procedimientos para el envío de datos

- ✓ Reflexión o *backscatter*: cuando la etiqueta entra en el campo electromagnético del lector, es capaz de reflejar parte de la potencia recibida con el fin de comunicarse con el lector. El lector es capaz de recibir esta señal sin problemas porque tiene una acoplador bi-direccional que permite diferenciar su señal, de la señal de la etiqueta.
- ✓ *Load modulation*: esta comunicación se basa en el siguiente hecho, la etiqueta es vista desde el lector como una impedancia Z. La etiqueta transmite su información variando su impedancia de acuerdo al flujo de datos salientes, gracias a esta técnica la etiqueta puede comunicarse con el lector.
- ✓ Sub-armónicos: los *tags* que trabajan con este método se liberan del problema de la interferencia causada por la señal del lector, transmitiendo sus mensajes en una frecuencia que es múltiplo de la frecuencia del lector, es decir, la etiqueta trabaja con los sub-armónicos de la frecuencia del lector.

II.1.4 Regulación y Estándares

II.1.4.1 Regulación

Actualmente en ningún país existe alguna administración, que se encargue de regular a nivel global la tecnología RFID, sino que cada país tienen sus propios órganos designados a regular de un modo individual el uso de las potencias y las frecuencias permitidas en su propio territorio. (García Candel, 2006).

GS1 Venezuela es el representante oficial de EPCglobal en el país. EPCglobal es una organización que regula el sistema del Código Electrónico de Producto pero trabaja con la tecnología RFID. Esta compañía le dio a GS1 Venezuela la tarea de administrar, mercadear y apoyar el desarrollo y la puesta en marcha de la Red EPCglobal en Venezuela.

CONATEL es el ente encargado de asignar el uso del espectro radioeléctrico en Venezuela. Sin embargo, no existe una banda UHF para el uso

de la tecnología EPC/RFID, por lo que GS1 Venezuela, comenzó a realizar contactos con CONATEL, para que asigne dicha banda en la medida de lo posible.

II.1.4.2 Estandarización

Es fundamental para el desarrollo del proyecto conocer un poco sobre los estándares con los que trabaja RFID.

"Un estándar es una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad." (Wales & Sanger)

Las normativas fundamentales para el sistema RFID, tratan de seguir ciertos objetivos que permitan:

- ✓ La Unificación: Para permitir la interoperabilidad a nivel internacional.
- ✓ La Simplificación: Se trata de reducir los modelos quedándose únicamente con los más necesarios
- ✓ La Especificación: Se persigue evitar errores de identificación creando un lenguaje claro y preciso.

Los requerimientos principales definidos en los estándares RFID son los siguientes:

- ✓ Reconocer e identificar *tags* RFID en el radio de acción del lector.
- ✓ Gestionar el problema de la colisión.
- ✓ Determinar el modo de presentación de los datos.
- ✓ Determinar el tamaño de la memoria del *tag*.
- ✓ Determinar qué partes de la memoria del *tag* son accesibles.
- ✓ Definir el modo en que se transmiten y reciben datos del *tag*.
- ✓ Modulación y codificación de la señal.
- ✓ Proveer autentificación de lectura y escritura.
- ✓ Compatibilidad internacional.
- ✓ Proveer mecanismos de seguridad e integridad de los datos y las comunicaciones.

Los organismos de estandarización para RFID, ISO y EPCglobal, han desarrollado diferentes estándares según las necesidades en un momento dado. En la tabla (II) se presentará un resumen de las normas ISO. (García Candel, 2006)

NORMAS ISO	DESCRIPCIÓN
11748	Identificación de animales por radiofrecuencia. Contiene la estructura de codificación del ID para la identificación de animales por radiofrecuencia. No especifica los protocolos de transmisión entre el dispositivo transpondedor (*tag*) y el transceptor (lector).
11785	Identificación de animales por radiofrecuencia. Especifica cómo se activa el transpondedor y como la información almacenada se transmite hacia el transceptor. Especifica las características técnicas del protocolo de comunicación entre lector y *tag*.
14223	Identificación avanzada de animales. Además de permitir el almacenamiento de un código de identificación, permite almacenar información adicional, e implementa métodos de autenticación. Este estándar consiste en tres partes: · *Part 1: Radio Frequency Identification of Animals, Advanced transponders - air interface.* · *Part 2: Radio Frequency Identification of Animals, Advanced transponders - code and command structure.* · *Part 3: Radio Frequency Identification of Animals, Advanced transponders – applications.* Es una extension de la ISO 11784 e ISO 11785. Define el interfaz aire entre el *tag* y el lector.
10536	Describe el modo de operación de las tarjetas inteligentes de acoplamiento cercano, con rango de operación de hasta 1 cm. Implica el acoplamiento capacitivo e inductivo a 4,9152 MHz. Debido al corto rango de operación no tiene un uso muy amplio en el mercado.
14443	Describe el modo de operación de las tarjetas de proximidad. Este estándar soporta un rango de lectura de hasta 10 cm. ISO 14443 tiene dos variantes: ISO 14443-A y ISO 14443-B. Ambas operan sobre misma frecuencia pero con parámetros diferentes (modulación, codificación y anticolisión).
15693	Estándar voluntario para la identificación RFID de contenedores. Es un estándar dual, pasivo, y de sólo lectura que incluye las frecuencias 850-950 y 2400-2500 Mhz.
10374	Estándar voluntario para la identificación RFID de contenedores. Es un estándar dual, pasivo, y de sólo lectura que incluye las frecuencias 850-950 y 2400-2500 Mhz.
/IEC 18000	Estándar RFID para la gestión de objetos. Define una línea de trabajo para los protocolos de comunicación comunes a nivel internacional en el uso de RFID. El estándar se especifica en varias partes: · *Part 1, Generic Parameters for Air Interface Communication for Globally Accepted Frequencies.* · *Part 2, Parameters for Air Interface Communication below 135 kHz.* · *Part 3, Parameters for Air Interface Communication at 13.56 MHz.* · *Part 4, Parameters for Air Interface Communication at 2.45 GHz.* · *Part 5, Parameters for Air Interface Communication at 5.8 GHz.* · *Part 6, Parameters for Air Interface Communication - UHF Frequency Band*

Tabla II Normas ISO para RFID (García Candel, 2006)

II.1.4.1 EPC

El código electrónico de producto, nació de un consorcio formado por EAN *International*, representado en 103 países. La intención de crear el EPC fue para promover un concepto de tecnología que pretendía cambiar la actual cadena

de suministro por otra con un estándar abierto y global, que permita la identificación en tiempo real de cualquier producto, en cualquier empresa de cualquier parte del mundo.

Las especificaciones del EPC se pueden clasificar de la siguiente manera:

- ✓ Especificaciones para los lectores: protocolo para el interfaz aire y comunicaciones lógicas con las etiquetas.
- ✓ Especificaciones para las etiquetas: referentes a los datos almacenados en ellas, a los protocolos de comunicación con el lector y la parte de RF que permite la comunicación.

Para principios del año 2005, EPCglobal publicó las especificaciones de la última versión de EPC, el EPC *Generation 2*, versión 1.0.9.

Esta última publicación es el estándar adaptado a nivel mundial en el uso de los sistemas de RFID ya que se ha realizado para cumplir con las necesidades de los consumidores. Para poder suplir las necesidades mencionadas EPCglobal, además de incluir especificaciones no observadas en otras regulaciones realizadas anteriormente, ha pretendido homogeneizar los principales estándares existentes.

En la tabla III podemos observar los estándares que se tienen como pre-requisito en EPC Gen2, los cuales son los más importantes existentes en la actualidad. Un dato muy importante es que se incluye la norma EN 302 208 de la ETSI, lo que representa un gran paso para una estandarización única entre Europa y USA, es decir: el EN 302 208 y el EPC *Generation 2* se complementan el uno al otro. (Samá Casanova & Ciudad Herrera, 2007)

Es importante destacar que las normas ISO, basan sus desarrollos en la aplicación que se le requiera dar al sistema, como a su vez la frecuencia, la codificación y la modulación a usar. En el caso de este Trabajo Especial de Grado y la solución que se plantea en el capítulo de resultados, las normas ISO I14443-A son las usadas por las tarjetas de proximidad como lo es en el caso del *Tag* Mifare. El lector Mifare Kimaldi KRD13M es compatible con dicha norma permitiendo que sea exitosa la comunicación entre los dos dispositivos. (Electronics).

EPCglobal™: EPC™ Tag Data Standards
EPCglobal™ (2004): FMCG RFID Physical Requirements Document (draft)
EPCglobal™ (2004): Class-1 Generation-2 UHF RFID Implementation Reference (draft)
European Telecommunications Standards Institute (ETSI), EN 302 208: Electromagnetic compatibility and radio spectrum matters (ERM) – Radio-frequency identification equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W, Part 1 – Technical characteristics and test methods
European Telecommunications Standards Institute (ETSI), EN 302 208: Electromagnetic compatibility and radio spectrum matters (ERM) – Radio-frequency identification equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W, Part 2 – Harmonized EN under article 3.2 of the R&TTE directive
ISO/IEC Directives, Part 2: Rules for the structure and drafting of International
ISO/IEC 3309: Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures – Frame structure
ISO/IEC 15961: Information technology, Automatic identification and data capture – Radio frequency identification (RFID) for item management – Data protocol: application interface
ISO/IEC 15962: Information technology, Automatic identification and data capture techniques – Radio frequency identification (RFID) for item management – Data protocol: data encoding rules and logical memory functions
ISO/IEC 15963: Information technology — Radiofrequency identification for item management — Unique identification for RF tags
ISO/IEC 18000-1: Information technology — Radio frequency identification for item management — Part 1: Reference architecture and definition of parameters to be standardized
ISO/IEC 18000-6: Information technology automatic identification and data capture techniques — Radio frequency identification for item management air interface — Part 6: Parameters for air interface communications at 860–960 MHz
ISO/IEC 19762: Information technology AIDC techniques – Harmonized vocabulary – Part 3: radio-frequency identification (RFID)
U.S. Code of Federal Regulations (CFR), Title 47, Chapter I, Part 15: Radiofrequency devices, U.S. Federal Communications Commission

Tabla III Estándares EPC (García Candel, 2006)

II.2 ON BOARD DIAGNOSTIC (OBD)

II.2.1 OBD-I

On Board Diagnostic según sus siglas en inglés o diagnóstico a bordo; es un sistema de monitoreo de vehículo.

Para Abril de 1985, *California Air Resources Board* (CARB), aprobó las normas del sistema OBD; con la intención de reducir notablemente las emisiones de gases a la atmósfera y evolucionar para vigilar algunos sectores del automóvil que puedan encontrarse con fallas en algún momento determinado, permitiendo al técnico y quizás al usuario, obtener los reportes de un auto-diagnóstico hecho por el sistema. La cantidad de información de diagnóstico disponible por medio de OBD ha variado ampliamente desde sus comienzos. (Vega Anderson, Karina, & Mendizábal, 2009)

Los objetivos básicos de este sistema son dos:

- ✓ Mejorar las emisiones de fallos del sistema, informando al usuario que existe un mal funcionamiento de algún sector del vehículo.
- ✓ Ayudar a los técnicos automotrices en la identificación y reparación del sector que se encuentre fallando.

La primera generación de diagnostico a bordo se dio a conocer como OBD-I. Este estándar era muy general; sin embargo permitía la monitorización de los siguientes sistemas: la medición de combustible, recirculación de gases de combustión (EGR) y emisiones relacionadas a componentes eléctricos. A parte de cumplir con las exigencias dadas por la CARB, se requirió almacenar los códigos de diagnostico DTC (*Data Trouble Code*) de fallas identificando las áreas de fallas. (Samá Casanova & Ciudad Herrera, 2007)

Después de una gran cantidad de estudios de la OBD- I, se encontró que estos sistemas no eran capaces de detectar emisiones relacionadas con algún componente a menos que este fallara por completo. A medida que iban realizando pruebas notaron que este sistema no tenía gran éxito ya que producía dificultades técnicas con la obtención de información.

II.2.2 OBD-II

Para mediados de los años 90, se generaron nuevas leyes y requerimientos conocidos actualmente como OBD-II, para intentar complementar las fallas del primer sistema OBD. Con este nuevo estándar ofrecían un control casi completo sobre el motor, además de detectar fallas de carrocerías, del chasis y la comunicación con la red de control de diagnostico del vehículo. (Diagnostic, 2008)

OBD II es la segunda generación de sistemas de "Diagnostico a Bordo". El sistema se incluye en vehículos de pasajeros, camionetas pick up y vehículos deportivos. El sistema OBD II monitorea virtualmente cualquier componente que pueda afectar el buen desempeño de las emisiones. Esto asegura que el vehículo se mantenga ambientalmente amigable durante su vida útil, asistiendo a los técnicos cuando se les presenten problemas relacionados con el diagnostico de emisiones. Si un problema es detectado, el sistema OBDII encenderá una luz de advertencia en el tablero para dar a entender al conductor que existe una falla y que se requiere un servicio para corregir el problema. (García Candel, 2006)

Lo más importante de este estándar es la compatibilidad que tiene con una gran variedad de marcas de fabricación de vehículos, ya que la forma de leer los códigos de diagnóstico del computador no dependerá de esa marca; esto beneficiará al técnico de reparación ya que no necesitará ningún adaptador para realizar la conexión. Esto es posible ya que el sistema OBD-II posee especificaciones estándar para el Conector de Diagnóstico (DLC).

La forma de trabajar de OBD-II consiste en monitorear pulsaciones eléctricas de diversos componentes del sistema de emisiones y comparar los datos para determinar si están funcionando correctamente. Los sensores de oxígeno antes y después del convertidor catalítico evalúan el desempeño del mismo. El espacio entre componentes como los inyectores de combustible y los sensores de oxígeno es muy importante porque el OBDII calcula el tiempo y el movimiento de los gases de escape a través del sistema cuando hace sus comparaciones.

Los nuevos requerimientos para OBD-II son:

- ✓ El uso de un computador a bordo para monitorizar las condiciones en que están los componentes electrónicos y para encender el led indicador de malfuncionamiento (MIL, por sus siglas en ingles) si los componentes fallan o los niveles de emisión se exceden de los límites permitidos.
- ✓ El conector de diagnóstico (DLC) y la localización del mismo es estándar, y además se permite el acceso con escáneres genéricos. También los Códigos de Diagnóstico (DTC) son según definiciones estándar.
- ✓ Si las emisiones HC, CO o Nox (Hidrocarburos, óxidos de carbono y óxidos de nitrógeno) exceden ciertos límites (1,5 es normalmente el nivel permitido por el Procedimiento de Pruebas Federal) se encenderá el MIL.
- ✓ Estandarización de sistemas eléctricos, términos de componentes y acrónimos.

El conector que generalmente trae el equipo es el DLC es un conector J1962 tipo hembra de 16 pines, como se observa en la figura 5. La salida de los pines del conector está definida de acuerdo a SAEJ1962 como se muestra en las tablas IV y V: (Müller, 2007)

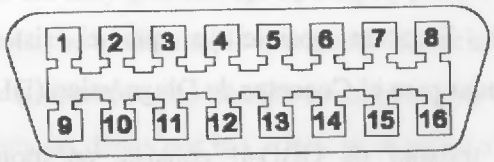


Figura 5 Conector hembra J1962 (Trapp, 2009)

Es importante destacar que el Pin 15 sólo se usa para mayores implementaciones de ISO 9141, todos los demás pines de otros fabricantes son reservados y no utilizados para el estándar OBD-II.

Pin 2 - J1850 *Bus+*
Pin 4 - *Chassis Ground*
Pin 5 - *Signal Ground*
Pin 6 – *CAN High* (J-2284)
Pin 7 – ISO 9141-2 K *Line*
Pin 10 – J1850 *Bus*
Pin 14 – CAN *Low* (J-2284)
Pin 15 – ISO 9141-2 L *Line*
Pin 16 - *Battery Power*

Tabla IV Representación de los pines del conector OBD-II (Trapp, 2009)

Pin (Señal)	Pin	Pin (Señal)	Pin (Señal)	Pin (+12 V)	Protocolos
--	4 + 5	7	15 *	16	ISO 9141-2
2	4 + 5	--	10	16	PWM J1850
2	4 + 5	--	--	16	VPW J1850
--	4 + 5	6	14	16	CAN Bus

Tabla V Protocolos según los pin (Trapp, 2009)

II.2.2.1 PID

OBD-II PID´s: Los parámetros de identificación son códigos usados para solicitar datos al vehículo. Estos parámetros se basan en los estándares SAE, específicamente en el J2012 quien muestra el esquema de cómo se deben presentar las fallas a través de 5 bits; utilizando el primer bit como limitador de la parte del carro donde se encuentra la falla. (Diagnostic, 2008)

Existen 9 modos de operación regidos por el estándar OBD-II SAE J1979 que son los siguientes:

1. Mostrar los datos actuales
2. Mostrar la trama de datos
3. Mostrar los códigos de diagnóstico almacenados

4. Borrar los códigos de diagnóstico almacenados
5. Resultados de la prueba, supervisión del sensor del oxígeno
6. Resultados de la prueba, supervisión de otros componentes del sistema (ej. Catalizador, EVAP)
7. Mostrar los códigos de diagnóstico pendientes (los detectados durante el ciclo que conduce actual o pasado)
8. Control de operación de los componentes o sistemas de a bordo
9. Solicitud de información del vehículo

II.2.2.2 PROTOCOLOS OBD-II

Los protocolos usados en el OBD-II son: (Vega Anderson, Karina, & Mendizábal, 2009)

- ✓ ISO9141-2: es un protocolo asíncrono en serie de 10,4 Kbaud. Los niveles de señal de inactividad son elevados. Los mensajes de este protocolo son los mismos mensajes del protocolo J1850 pero la interface física es diferente son hasta 12 bytes, excepto los delimitadores marco. El conector debe tener contactos metálicos en los pines 4, 5, 7, 15 (opcional) y 16.
- ✓ J1850 PWM (*"Pulse Width Modulated"*): es un protocolo de 41.6kbps utilizado principalmente en los vehículos Ford. El conector debe tener contactos metálicos en los pines 2, 4, 5, 10 y 16.
- ✓ J1850 VPW (*"Variable Pulse Width"*): es un protocolo de 10.4Kbps utilizado principalmente en los vehículos de *General Motors* y puede tener una longitud de bus de hasta 35 metros (con 32 nodos). El conector debe tener contactos metálicos en los pines 2, 4, 5 y 16.
- ✓ J2284 CAN (*"Controller Area Network"*): es un protocolo de alta velocidad y desempeño de 250kbit/sec o 500kbit/sec. CAN se está convirtiendo en el bus usado más ampliamente para comunicaciones en vehículos. El conector debe tener contactos metálicos en los pines 4, 5, 6, 14 y 16.

A parte de los protocolos mencionados, OBD II cuenta con un sistema de redes de baja velocidad y costo denominado LIN, y se utiliza como un complemento de las redes de CAN para integrar los dispositivos de sensores inteligentes o actuadores en coches de hoy. Las subredes LIN se conectan con las redes CAN de los vehículos por medio de "*gateways*".

II.2.3 Interfaz ELMSCAN

Es una herramienta de análisis compatible con OBD-II; esta permite leer los parámetros de funcionamiento del vehículo de una manera práctica y fácil. Tiene funciones de detección automática de protocolos y a través de un *software* interroga al carro según los parámetros que desea obtener del carro (Ver Figura 6). (Scantool.net, 2009)

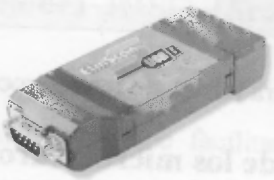


Figura 6 ELMSCAN 5

Este dispositivo es compatible con los siguientes protocolos OBD-II:

- ✓ ISO15765-4 (CAN)
- ✓ ISO14230-4 (Protocolo de palabras clave 2000)
- ✓ ISO 19141-2
- ✓ J1850 VPW
- ✓ J1850 PWM

Es importante destacar que el protocolo de salida es el USB Virtual COM Port, que tiene una velocidad de transmisión adaptable de 9600 ó 38400 y necesita una tensión de funcionamiento de 12V.

II.3 MICROCONTROLADORES

II.3.1 Definición

Es un circuito integrado que está compuesto por tres unidades fundamentales de un computador: unidad central de procesador que es el

encargado de interpretar las instrucciones y procesar los datos contenidos, memoria es la que se encarga de retener los datos por un intervalo de tiempo y periféricos de E/S (entrada/salida), que son las unidades para comunicarse con otros dispositivos externos. (Ver Figura 7)

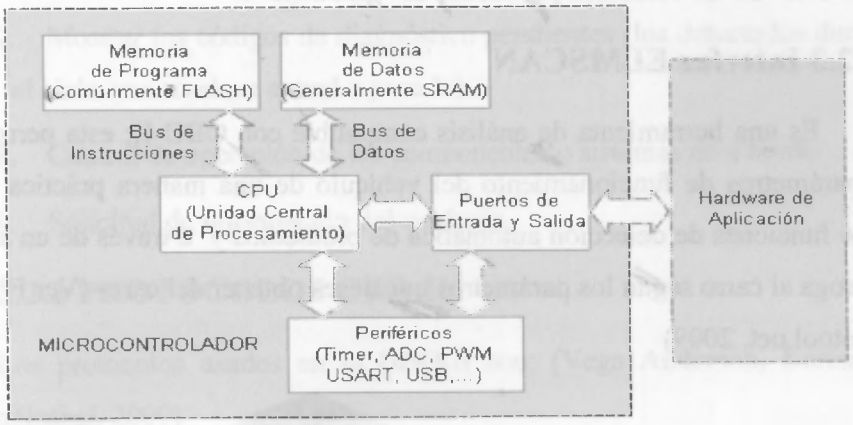


Figura 7 Estructura básica de los microcontroladores (S/A, 2007)

II.3.2 Clasificación de los microcontroladores

Los microcontroladores se clasifican:

II.3.2.1 Según bus de datos

El bus de datos es un sistema digital que transfiere datos en un dispositivo o en varios como el computador. Existen dos tipos de bus, clasificados según el método de envío de información que son: bus paralelo en donde los datos son enviados por bytes al mismo tiempo, con la ayuda de varias líneas que tienen funciones fijas, y bus serie en donde los datos son enviados, bit a bit y reconstruyen por medio de registros o rutinas de *software*.

En esta clasificación los microcontroladores pueden tener 8 bits, 16 bits o 32 bits; aquí es importante que a mayor longitud del bus de datos, mayor será la eficiencia en operaciones con datos grandes, pero aumenta la complejidad del circuito impreso y por ende su costo.

II.3.2.2 Según sus recursos

Esta clasificación se refiere a los periféricos que cuenta el microcontrolador y a todas las características que facilitan las conexiones con otros dispositivos. Existen microcontroladores con módulos CAN, con módulos UART, módulos para comunicarse al computador vía puerto USB, entre otros.

II.3.2.3 Según el set de instrucciones

II.3.2.3.1 Con instrucciones CISC *(Complex Instruction Set Computer)*

Son las instrucciones con las que se programaba desde los comienzos de los microcontroladores, son complejos ya que sus instrucciones deben ser muy elaboradas, por lo que se necesita una alta eficiencia en el microcontrolador.

II.3.2.3.2 Con instrucciones RISC *(Reduced Instruction Set Computer)*

Son instrucciones muy sencillas que facilitan la programación en el microcontrolador, y a su vez no se necesita que el microcontrolador tenga una alta eficiencia en el procesamiento de los datos.

II.3.2.4 Según su arquitectura interna

II.3.2.4.1 Arquitectura de Von Neumann

Estos contienen una memoria única que constituye tanto el segmento de memoria de programa como el de datos y un solo bus de comunicación entre la memoria y el procesador, permitiendo solo una comunicación a la vez. (Ver Figura 8)

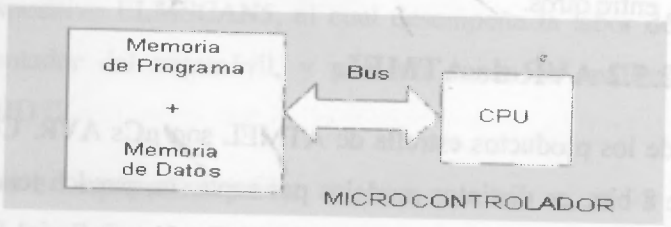


Figura 8 Estructura de microcontroladores con arquitectura Von Neumann (S/A, 2007)

II.3.2.4.2 Arquitectura de Harvard

Esta arquitectura contiene dos memorias, una que contiene el programa y otra para almacenar los datos, como puede observarse en la Figura 9. De este modo el CPU puede tener acceso simultáneo a ambas memorias utilizando buses diferentes. Actualmente los microcontroladores siguen esta arquitectura.

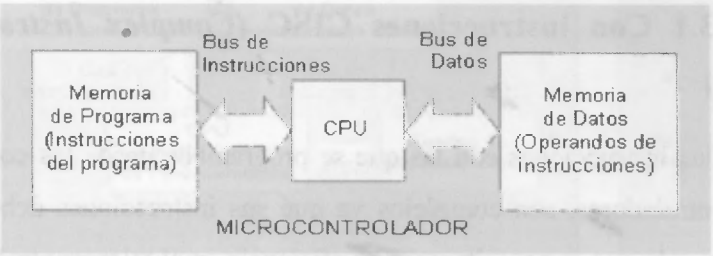


Figura 9 Estructura de microcontroladores con arquitectura de Harvard (S/A, 2007)

II.3.2.5 Según el fabricante

Existen muchas marcas de microcontroladores en el mercado, de las cuales se mencionan a continuación algunos de los fabricantes más reconocidos, de los que se ubiquen fácilmente los componentes y las herramientas de desarrollo:

II.3.2.5.1 PIC de MICROCHIP

Su facilidad de uso, abundante información y libre disposiciones de herramientas de *software* proporcionada por ellos misma, los han posicionado en el mercado como uno de los primeros. En lo que respecta a este proyecto se utiliza la familia de los DsPIC´s, que contiene altas capacidades de memoria y recursos útiles para las aplicaciones como: módulos UART, puertos configurables como E/S pin a pin, entre otros.

II.3.2.5.2 AVR de ATMEL

Uno de los productos estrella de ATMEL son µCs AVR. Comparado con otros µCs de 8 bits, en distintos modelos por supuesto, pueden tener memoria de programa flash reprogramable, capacidad ICSP (In Circuit Serial Programming), puertos configurables como E/S pin a pin, interfaces de comunicación serial RS232 e I2C, módulos generadores de onda PWM, etc.

III.- Metodología y Desarrollo

Durante este capítulo se presentan las distintas actividades desarrolladas en cada etapa y como han sido aplicadas a lo largo del proyecto. Las etapas que conformaron el desarrollo del Trabajo Especial de Grado son:

1. Proceso de investigación y documentación
2. Diseño e implementación
3. Pruebas y optimización
4. Análisis de resultados

III.1 Proceso de investigación y documentación

En esta etapa se investigó la teoría necesaria para el desarrollo del proyecto, abarcando principalmente las tecnologías RFID y OBDII ya que son el núcleo del trabajo. Las actividades que se realizaron durante esta etapa fueron:

1. Estudio de los protocolos y estándares RFID
2. Estudio de las características técnicas de los equipos RFID
3. Análisis del funcionamiento del instrumento de medición OBDII

El estudio de la tecnología RFID comenzó con en el análisis de las estructura del sistema de identificación por radiofrecuencia, después de conocer las características del *software* y del *hardware* del sistema, la investigación se concentró en los protocolos y en la estandarización realizada por la EPC para esta tecnología.

Después de adquirir los conocimientos anteriormente mencionados, se estudió el dispositivo ELMSCAN5, el cual desempeña la labor de comunicarse con el computador del automóvil, y por esta razón se analizaron todos los protocolos OBD II.

III.2 Diseño e Implementación

Este segmento representa el núcleo del Trabajo Especial de Grado, ya que es donde se resuelve el problema planteado por medio de la implementación de un sistema basado en las tecnologías que se analizaron en la etapa anterior.

Es importante justificar la selección de la tecnología RFID, destacando características relevantes que no son útiles para el sistema de control planteado en el Trabajo Especial de Grado, de las tecnologías más comunes en el área de las telecomunicaciones como WiFi, Bluetooth, IrDA y sistemas de telefonía móvil. Entre esas características encontramos:

1. Bluetooth, el máximo número de comunicaciones que puede manejar un terminal al mismo tiempo, es muy limitado, más específicamente una red bluetooth puede tener como máximo 8 miembros (Dunne, Roche, & O'Loghlin).

2. IrDA, el cual es un estándar que define la transferencia de datos usando el espectro infrarrojo. En este tipo de tecnología se necesita que exista línea de vista entre el receptor y el transmisor. (S/A)

3. Los sistemas de telefonía móvil son eficaces pero su gran desventaja es que sus servicios están ligados a la cobertura que tiene la operadora en una zona determinada, es decir, si no hay cobertura el proyecto queda inhabilitado. Otro punto importante es que estos sistemas tienen tarifas por conexión, por lo tanto la empresa deberá pagar una mensualidad para disfrutar del servicio.

4. WiFi, al utilizar esta tecnología inalámbrica se necesita una infraestructura tecnológica para poder establecer una conexión (puntos de acceso y router) en los sitios de interés. (System, 2006)

Por estas razones se estudio la tecnología RFID la cual supera todas las desventajas nombradas anteriormente. RFID es un sistema de identificación que está revolucionando los procesos de inventario en el mercado mundial, y entre sus principales características están:

1. El lector RFID tiene la capacidad de leer varios *tags* simultáneamente.

2. El lector y el *tag* RFID no necesitan tener línea de vista, por la frecuencia y la polarización de las antenas.

3. El sistema RFID no necesita de un área específica de cobertura, ya que es un enlace punto a punto con gran movilidad.

4. No necesita de gran infraestructura, ya que se requieren solo los elementos básicos, el lector y el *tag*.

III.2.1 Descripción del Sistema

Para que la selección de dispositivos fuera un proceso exitoso se hizo un estudio de las funciones que debía desempeñar el Sistema de Control de Operación y la conclusión de este análisis trajo como resultados, un esquema que se observa en la figura 10, donde se muestran cada módulo del proyecto.

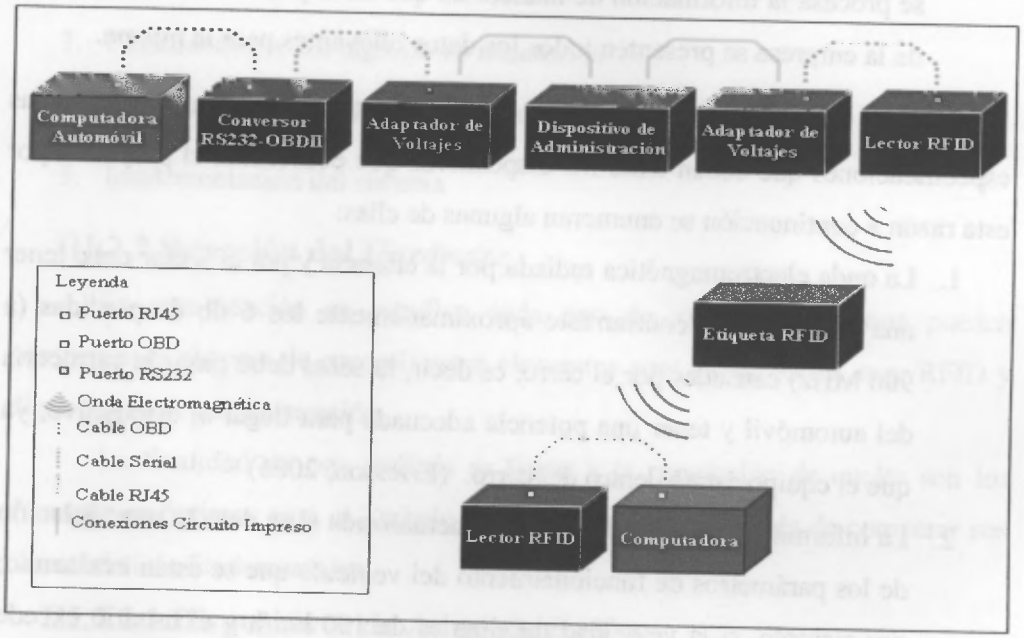


Figura 10 Diagrama del Sistema De Control (Fuente Propia)

Entre las funciones que debe realizar el sistema mostrado en la figura, se encuentran:

1. Monitorear los parámetros del automóvil. Para este proceso se necesita una unidad de administración que sea capaz de obtener los datos del computador del automóvil.
2. Seleccionar la información. El dispositivo de administración mencionado en el punto anterior, debe analizar los datos recolectados y codificarlos de una forma adecuada para grabarlos en la etiqueta RFID.
3. Controlar el proceso de escritura del lector RFID. Esta es la última función que debe cumplir el dispositivo de administración, y esta consiste en

mandar una serie de comandos a través de una conexión serial RS232, que le indiquen al lector RFID que código debe escribir en la etiqueta.

4. Mostrar los datos. De esta función se debe encargar el computador final, el que posiblemente usarán los empleados de la empresa. Básicamente esta etapa del sistema, consiste en leer los parámetros que tiene la etiqueta RFID que se encuentra en el vehículo, a través del lector que está conectado al computador vía ethernet. Una vez se obtengan estos valores se procesa la información de manera tal que en la pantalla del computador de la empresa se presenten todos los datos relevantes para la misma.

Las funciones básicas del sistema son tan importantes como las especificaciones que deben tener los dispositivos que conforman el proyecto, por esta razón a continuación se enumeran algunas de ellas:

1. La onda electromagnética radiada por la etiqueta y por el lector debe tener una potencia que contrarreste aproximadamente los 6 db de pérdidas (a 900 MHz) causadas por el carro, es decir, la señal debe pasar la carrocería del automóvil y tener una potencia adecuada para llegar al dispositivo, ya que el equipo estará dentro del carro. (Ericsson, 2006)
2. La información de las etiquetas será actualizada si se detecta una violación de los parámetros de funcionamiento del vehículo que se están evaluando; por ejemplo, si la velocidad máxima es de 100 km/h y el usuario excede esta velocidad, se grabará en el *tag* una infracción. Por esta razón las etiquetas deben tener varios ciclos de escritura para que permita almacenar la información cada vez que sea actualizada. Se espera que los *tags* tengan una vida útil de aproximadamente un año (365 días). Si se tiene un máximo de 5.000 ciclos de escritura, se estima que se puedan realizar 13 actualizaciones diarias.
3. Las etiquetas deberán tener una capacidad de almacenamiento para guardar los parámetros obtenidos del vehículo; como lo son la velocidad, revoluciones por minuto, entre otros.
4. El lector debe contener puertos de comunicación, preferiblemente RS-232, ya que los microcontroladores poseen módulos UARTs, los cuales facilitan la comunicación mediante este estándar.

5. El lector deberá trabajar con protocolos abiertos. Esto facilitará la convergencia de las soluciones diseñadas con el sistema RFID.

Después de describir del sistema se puede mencionar las actividades que se realizaron durante esta etapa para el diseño y la implementación del proyecto:

1. Selección de los dispositivos RFID adecuados para la solución del problema.
2. Selección del dispositivo administrador del sistema de control
3. Elaboración de los algoritmos necesarios
4. Diseño del sistema de control
5. Implementación del sistema

III.2.2 Selección del *Hardware*

En esta sección se estudian cada uno de los elementos que pueden conformar el sistema de control; estos elementos son: lector RFID, *tags* RFID y dispositivos de administración.

La finalidad de este análisis es llegar a la conclusión de cuales son los dispositivos óptimos para el Trabajo Especial de Grado, después de comparar sus características fundamentales.

III.2.2.1 Dispositivos de Administración

Para elegir el dispositivo de administración se estudiaron los siguientes parámetros:

- ✓ MIPS, estas son las siglas de Mega Instrucciones Por Segundo, este parámetro indica la velocidad de procesamiento de los microprocesadores.
- ✓ Memoria RAM, esta es la memoria de acceso aleatorio y es la utilizada para procesar toda la información del dispositivo.
- ✓ Memoria del Programa: es el espacio de memoria que se dispone para programar el dispositivo.
- ✓ I/O *Pins*: son el número de pines que tiene el dispositivo.
- ✓ *Timers*: son los relojes que puede utilizar el usuario para controlar eventos temporales dentro del sistema, en este parámetro se indica el número de *timers* que tiene el dispositivo.

- ✓ UART: *Universal Asynchronous Receiver-Transmitter*, este es el dispositivo que permite realizar una comunicación serial de manera automática, en este parámetro se indica la cantidad de UARTs que tiene el dispositivo. Se debe considerar que el sistema debe contar como mínimo con 1 UART porque debe comunicarse con el lector RFID por medio del puerto serial.

Los parámetros de los equipos a evaluar se muestran en la siguiente tabla:

EQUIPOS PARÁMETROS	PIC 16F877A	DSPIC30F4011	PIC18F4520	SE TE-486
MIPS	2	30	10	No aplica
Memoria RAM (en Bytes)	368	2048	1536	32 o 46 M
Memoria de Programa (en Bytes)	7 K	48K	32K	16 MB
I/O *Pins*	16	30	36	No aplica
Timer	3	7	3	1
UART	0	2	0	2
Disponibilidad en Venezuela	Si	Si	Si	No

Tabla VI Comparación de los dispositivos de Administración (Inc., 2009)

*SE: Sistemas Embebidos

Por la cantidad de UARTs, la disponibilidad en Venezuela y la memoria de programa, el dispositivo de administración elegido es el DsPIC30F4011. El costo de este equipo son BsF 60

III.2.2.2 Lectores

Para elegir el lector se estudiaron los siguientes parámetros:

- ✓ Alcance: es la distancia entre el lector y la etiqueta, donde el lector es capaz de leer y escribir. En el sistema esta distancia debe ser de 3 metros si la etiqueta está fija en algún lugar del carro y de unos centímetros si la etiqueta no está fija.
- ✓ Lectura y Escritura: es la capacidad que tiene el lector para leer y/o escribir. Para el sistema el lector debe ser capaz de leer y escribir la etiqueta.
- ✓ Fuente de alimentación: es el voltaje que necesita el lector para funcionar. El voltaje que proporciona la electrónica del carro es de 15 V, por lo tanto

el voltaje de alimentación del lector debe ser menor o igual que el voltaje del carro.

- ✓ *Tags* compatibilidad: en este parámetro se habla de las etiquetas con las que trabaja el lector.
- ✓ Puertos de comunicación: son todas las opciones que tiene el lector para comunicarse. El sistema está planificado para usar el puerto serial del lector para establecer la comunicación.
- ✓ Frecuencia: es la frecuencia de operación del lector.

Los parámetros de los equipos a evaluar se muestran en la siguiente tabla:

EQUIPOS / PARÁMETROS	Alien's 9900	Alien's 8800	Kimaldi KRD13M	SYRD245-1N	SYDRD245-CF-1
ALCANCE	-	-	5 cm	13 metros	10 metros
LECTURA	SI	SI	SI	SI	SI
ESCRITURA	SI	SI	SI	NO	NO
FUENTE DE ALIMENTACIÓN	110 o 240 VAC	110 o 240 VAC	5VDC	7.5 a 28 VDC	3.3 VDC
TAGS COMPATIBILIDAD	EPC Gen 2, ISO 18000-6c	EPC Gen 2, ISO 18000-6c	ISO 14443-A	*Tags* Activos de la misma Compañía	*Tags* Activos De la misma Compañía
PUERTOS DE COMUNICACIÓN	RS-232 LAN TCPI / IP	RS-232 LAN TCPI / IP	RS232	RS-232, RS-485 Ethernet, Wiegand	CompactFlash
FRECUENCIA	902,75 MHz - 927,25 MHz	865,6 MHz - 867,6 MHz	13,5 MHz	2.4 a 2.8 GHz	2.4 a 2.8 GHz

Tabla VII Comparación de los Lectores RFID (ALiens, 2009) (Technology, 2007)

III.2.2.3 *Tags*

Para elegir las etiquetas que se utilizan en el sistema, se estudiaron los siguientes parámetros:

- ✓ Tipo de etiqueta: este parámetro describe si la tarjeta tiene o no, alimentación incorporada.
- ✓ Ciclos de escritura: este número describe la cantidad de veces que se puede escribir el *tag*. La etiqueta debe durar por lo menos 1 año.

- ✓ Memoria: este campo indica el espacio disponible para guardar los datos recolectados del automóvil. Para el sistema la memoria debe ser suficiente para guardar los datos del carro.
- ✓ Alcance: es la distancia entre el lector y la etiqueta, donde el lector es capaz de leer y escribir. En el sistema esta distancia debe ser de 3 metros si la etiqueta está fija en algún lugar del carro y de unos centímetros si la etiqueta no está fija.
- ✓ Frecuencia: es la frecuencia de operación de la etiqueta.

Los parámetros de los equipos a evaluar se muestran en la siguiente tabla:

EQUIPOS PARAMETROS	Alien's 9540	SYTAG245-2C	*Tag* Mifare	Thin PVC Card	Flex
Tipo de etiqueta	Pasiva	Activa	Activa	Pasiva	Pasiva
Ciclos de escritura	10000	-	100000	-	-
Memoria	96 bits	4kbytes 32kbytes (opcional)	752 byte	96 bits	256 bits
Alcance	Corto	Mediano	10 cm	2M de grabación 3,5M para lectura	2 a 3 cm
Frecuencia	840 MHz - 960 MHz	2.4 a 2.48 GHz	13, 5 MHZ	860 MHz 960 MHz	134,2 kHz

Tabla VIII Comparación de los *tags* RFID (ALiens, 2009) (Electronics) (Technology, Kimaldi, 2007)

Al comparar parámetros como el alcance, la memoria de almacenamiento y el número de ciclos de escritura de las etiquetas y el alcance, el tamaño, la fuente de alimentación, la capacidad de lectura y escritura del lector, se eligió el lector Kimaldi que tiene un costo de BsF 1303 con las etiquetas Mifare que tienen un costo de BsF 24.

Aunque el dispositivo escogido es el óptimo para esta aplicación, el lector utilizado es el Alien 9900 porque es el mas accesible, ya que se encuentra en la UCAB, y solo no cumple con los parámetros correspondientes al tamaño y al voltaje de alimentación y se puede decir que no altera el rendimiento del sistema

piloto. Los costos correspondientes a estos equipos son para el lector con las antenas de BsF 6850 y de los *tags* BsF 0,602

III.2.3 Selección del *Software*

Para el sistema se debe programar tanto la aplicación del DsPIC30F4011 como la aplicación del computador, por eso para cada uno de los dispositivos se debe escoger un lenguaje de programación que facilite los objetivos del proyecto.

III.2.3.1 *Software* del DsPIC30F4011

Básicamente la empresa MICROCHIP permite que el dispositivo se programe bajo el lenguaje ensamblador o C. El ensamblador o ASM, es caracterizado por otorgarle al programador un control a muy bajo nivel del dispositivo, es decir, se trabajan con los registros y acumuladores que controlan al microcontrolador, en cambio el lenguaje C, es caracterizado por ofrecer una programación de alto nivel, el cual facilita el control del dispositivo.

El lenguaje escogido fue C, ya que otorga al programador ciertas comodidades características de los lenguajes de alto nivel. Permite que el programa sea más eficiente con la manipulación de procedimientos y librerías y por último es un lenguaje que es conocido por los estudiantes de Ingeniería de Telecomunicaciones de la UCAB.

III.2.3.2 *Software* de la Aplicación.

A diferencia de un microcontrolador, en un computador se pueden trabajar con muchísimos lenguajes de programación entre los cuales están, ensamblador, C, C++, C#, Visual Basic, Java, NET, entre otros.

Uno de los parámetros que rigen la elección de este *software* es que la empresa ALIEN, entrega en el CD de instalación una guía para los desarrolladores de herramientas RFID, y en ella se encuentran las librerías con las funciones del lector, pero estás solo están en Visual Basic, Java y NET.

El lenguaje NET, se descartó porque fue desarrollado para facilitar las aplicaciones que usen conexiones con servidores y que necesiten conexiones a la red, pero el sistema no requiere estas propiedades.

El lenguaje escogido es el Visual Basic ya que permite el manejo de gráficas y tablas el cual es fundamental para el desarrollo del proyecto, además es

compatible con los sistemas operativos Microsoft, los cuales son los más populares en el mercado.

III.2.4 Selección de la solución para la comunicación con el automóvil

Para comunicarse con el automóvil, se estudiaron los antecedentes del proyecto, en uno de ellos se hizo la comunicación por medio de un microcontrolador que fue programado para utilizar un solo protocolo OBDII y en el otro se utilizó un dispositivo llamado ELMSCAN5 el cual es compatible con múltiples protocolos OBDII, y sus interfaz de comunicación externa es serial.

Ya que el ELMSCAN 5 está disponible en la Universidad y representa una solución compatible con varios modelos de automóviles, se seleccionó para que haga la función de convertir de RS232 (señales emitidas por el DsPIC30F4011) a OBDII y viceversa.

III.2.5 Diseño del Sistema

El *hardware* del sistema fue seleccionado en el paso anterior, está conformado básicamente por los siguientes dispositivos:

- ✓ DsPIC30F4011 que desempeña la labor de recolectar los datos del ELMSCAN5 y transmitirlos al lector RFID.
- ✓ MAX232 que realiza la función de transformar los voltajes de TTL a RS232 y viceversa.
- ✓ El ELMSCAN5, el cual se encarga de comunicarse con el computador del automóvil.
- ✓ El KRD 13M kimaldi y *Tags* Mifare, que fueron sustituidos por el lector ALIEN 9900 y por la etiqueta ALIEN 9540, ya que no alterarán los resultados técnicos.

El *software* del sistema se divide en dos, el del computador y el del DsPIC30F4011, en la etapa anterior se escogieron los lenguajes de programación, y en esta se estudiaron los parámetros que debían ser leídos del computador del automóvil y los posibles formatos de los datos en la etiqueta, es decir, la distribución de la información en la memoria del *tag*.

El formato de la información que se encuentra en la etiqueta define la relación entre el *software* del computador y el del DsPIC30F4011, sin la existencia de las reglas para guardar la información, el sistema no podría procesar la información correctamente.

La etiqueta debe tener todos los datos que le permita a la empresa arrendadora evaluar el uso del automóvil por parte del usuario, por esta razón los parámetros estudiados fueron: velocidad, revoluciones por minuto, nivel del tanque de gasolina, temperatura del motor y distancia recorrida con una luz indicadora de mal funcionamiento encendida.

El problema es que el espacio de memoria de la etiqueta RFID es de 96 bits (24 caracteres hexadecimales), y no se pueden guardar todas las velocidades recolectadas por el dispositivo de administración, por esta razón en cuanto a la velocidad, revoluciones por minuto, nivel del tanque de gasolina y temperatura del motor se almacenan solo el número de veces que el usuario pasa un umbral determinado y en el caso de la distancia recorrida con una luz indicadora de mal funcionamiento encendida, no existe ninguna compresión. Entre las opciones estudiadas están las siguientes:

	Opción 1	Opción 2	Opción 3
Matrícula Vehicular	X_1 - X_{11}	X_1 - X_{11}	X_1 - X_{11}
Velocidad (Rango 1)	X_{12} – X_{14} (Solo 2 bits)	X_{12} - X_{13}	X_{12} - X_{14}
Velocidad (Rango 2)	X_{14} - X_{16} (Solo 2 bits)	X_{14} - X_{15}	-
RPM (Rango 1)	X_{17} – X_{19} (Solo 2 bits)	X_{16} - X_{17}	X_{15} - X_{17}
RPM (Rango 2)	X_{19} – X_{21} (Solo 2 bits)	X_{18} - X_{19}	-
Nivel de Gasolina	X_{22}	X_{20} – X_{21}	X_{18} - X_{19}
Temperatura del Motor	-	-	X_{20}
Kilometraje con una luz indicadora de mal funcionamiento encendida	X_{23} – X_{24}	X_{22} – X_{24}	X_{21} – X_{24}

Tabla IX Distribución de los datos en el *tag* (Fuente Propia)

Cada opción ofrece un valor máximo de violaciones registradas, como por ejemplo, el umbral de la velocidad puede ser violado como máximo 1023 veces en la opción 1 y 255 en la opción 2 y 3.

III.3 Pruebas y Optimización del Proyecto

En esta etapa, se estudian todos los dispositivos y se observan los parámetros que se deben tomar en cuenta para optimizar el funcionamiento del sistema. La única actividad planificada en la metodología fue, realizar las pruebas necesarias.

III.3.1 Prueba con el Carro

El dispositivo que se utiliza para leer los parámetros del automóvil es el ELMSCAN5, este tiene un puerto OBDII y uno serial, los documentos anexados en el CD de instalación del programa ScanTool.net, fueron de gran utilidad para conocer las características técnicas del equipo.

III.3.1.1 Prueba con el Carro y el Computador

Gracias al *datasheet* del ELMSCAN5, se obtuvo el formato de transmisión serial del equipo, el cual es 8N1, es decir, que cada paquete posee 8 bits de datos, 1 bit de parada y no posee bit de paridad, y también se obtuvo que la velocidad de transmisión por defecto era de 38400 bps.

La finalidad de esta prueba era conocer las características de transmisión del equipo, y familiarizarse con el *software* que lo controla, denominado ScanTool.net.

La conexión se puede observar en la siguiente figura

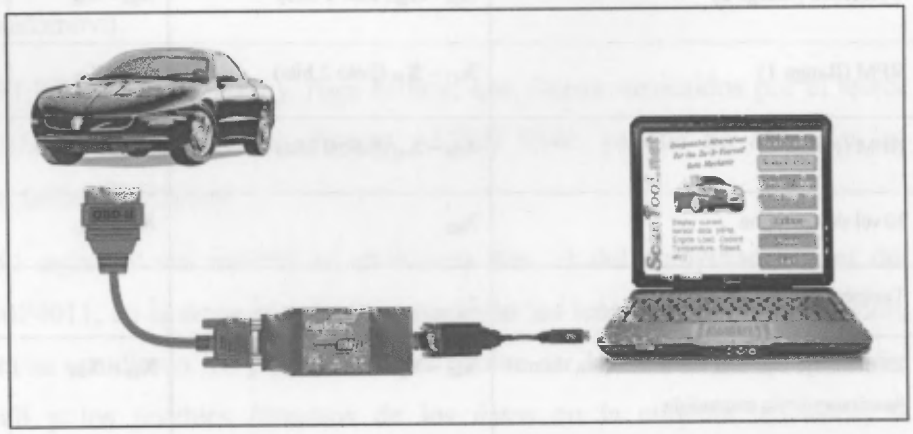


Figura 11 Prueba con el ELMSCAN 5 (Fuente Propia)

Entre las pruebas realizadas estuvieron la obtención de la velocidad, las revoluciones por minuto, el nivel de combustible, la temperatura del refrigerante, entre otros, por medio del *software* del ELMSCAN5, después se hizo la conexión serial por medio de Hyperterminal.

Por último se analizaron las tramas de la conexión serial mediante el *software* Serialmonitor, el cual permite tener una visión más detallada de la comunicación del equipo.

III.3.1.2 Prueba con el Carro con el DsPIC30F4011

Después de configurar todos los parámetros del UART, se programó en el microcontrolador con el lenguaje C, la transmisión y la recepción de algunos parámetros. A continuación se muestra una tabla resumiendo la información importante de este experimento.

Parámetro	Código de Petición	Código de Respuesta
Velocidad	010D	41 0D XX
Revoluciones por Minuto	010C	41 0C XX XX
Nivel de Gasolina	012F	41 2F XX
Temperatura del Refrigerante	0105	41 05 XX
Kilometraje con un luz indicadora de mal funcionamiento encendida	0121	41 21 XX XX

Tabla X Experimentos con el protocolo OBD (Fuente Propia)

Toda la comunicación se realiza bajo la codificación ASCII, entre los problemas observados, está el hecho de que si el automóvil se prende, se apaga y se prende, el ELMSCAN5 deja de funcionar correctamente, la solución es, reiniciar el dispositivo con el comando ATZ.

Cabe destacar que cuando el carro no esta prendido y se envía uno de los códigos mostrados en la tabla anterior, el responde con un "UNABLE TO CONECT" y si se prende, se apaga y se prende el automóvil envía el siguiente mensaje "*NO DATA*".

III.3.2 Pruebas con el Lector RFID

Entre los documentos utilizados para conocer el manejo del lector RFID, está el *Guide, Interface Reader*, que se encuentra en el CD de instalación proporcionado por la compañía ALIEN. Este documento proporciona información

de utilidad para los desarrolladores de aplicaciones, ya que contiene la descripción de todas las interfaces (E/S, Ethernet y RS232), explicación de los modos de operación y todos los comandos con los que trabaja el lector.

III.3.2.1 Pruebas con el Lector RFID y el Computador, vía puerto serial

Entre los datos obtenidos del documento mencionado está, que el formato de comunicación serial usado por el lector es 8N1, también se encontró que la tasa de transmisión por defecto es de 115200 bps y que no existe ningún mecanismo de control de flujo de datos.

La finalidad de esta prueba era conocer las características de transmisión del equipo, y familiarizarse con el *software* Alien RFID Gateway. Se realizo la conexión para esto como se muestra en la figura

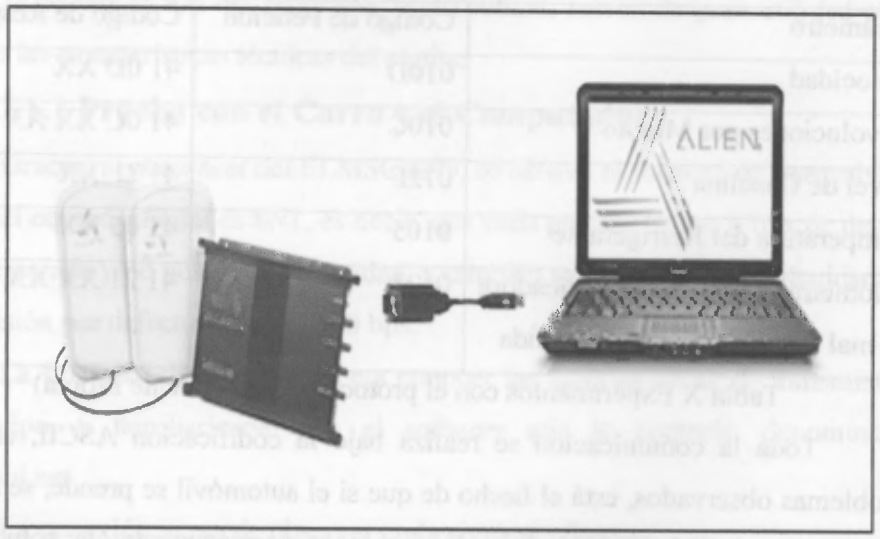


Figura 12 Prueba con el lector RFID y el puerto serial (Fuente Propia)

En esta experiencia por medio del *Alien* RFID *Gateway*, se escribió y se leyó el código EPC de varias tarjetas, se cambió la atenuación del dispositivo y después se realizaron los mismos pasos pero con la herramienta Hyperterminal.

Por último se utilizó un monitor del puerto serial llamado Serialmonitor, el cual permite ver todos los datos que pasan por el puerto DB9, esto permite revisar el proceso de inicialización del lector y de escritura y lectura de la etiqueta.

Como resumen, los comandos más importantes que se utilizaron durante esta prueba, se anexan en la siguiente tabla, en la que se presenta un breve resumen de las funciones que realizan cada uno.

Comandos	Función
Set BaudRate	Configura la velocidad de transmisión del puerto serial, los valores permitidos son: 9600-19200-38400-57600-115200.
Set RFlevel	Configura la atenuación del equipo, va desde 0 hasta 150, cada decena representa 1 db. Es una herramienta que permite manejar la zona de cobertura de la antena.
G2READ	Es una forma de leer la información de la etiqueta pero leyendo toda la información de la etiqueta, no solo el código EPC. Los parámetros son los siguientes: Banco (0-3), Posición del primer *word* que se desea leer y número de *words* que se desea leer.
G2WRITE	Es una forma de es escribir la información de la etiqueta pero leyendo toda la información de la etiqueta, no solo el código EPC. Los parámetros son los siguientes: Banco (0-3), Posición del primer *word* que se desea escribir y datos.
REBOOT	Permite reiniciar el dispositivo con el fin de que configurar algunos parámetros cambiados como por ejemplo la velocidad de transferencia.
ProgramEPC	Permite escribir el código EPC de la etiqueta, se deben mandar los caracteres hexadecimales en pares y separarlos por un espacio.
Get Taglist	Permite leer el código EPC de la etiqueta, este valor es enviado en grupos de 4, separados por un espacio.

Tabla XI Comandos utilizados en la prueba con el puerto serial (Fuente Propia)

III.3.2.2 Pruebas con el Lector RFID y el Computador, vía puerto ethernet

A través del *software* Visual Basic y la conexión como se muestra en la figura 13, se analizaron unos ejemplos propuestos por la empresa Aliens RFID

(*software* de los elementos de RFID) , en donde se podía estudiar los parámetros para la conexión del lector con el computador, y las funciones de programar y leer *tags*.

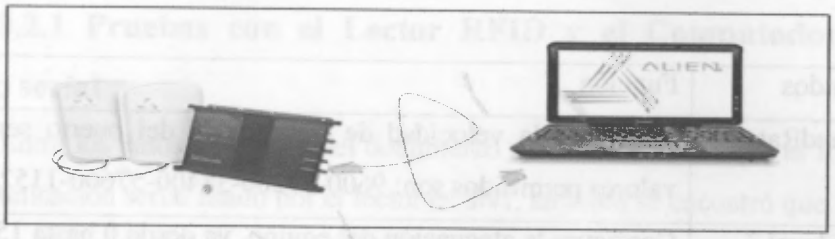


Figura 13 Prueba con el lector RFID y el puerto ethernet (Fuente Propia)

En el proceso de analizar la conexión se observó, que para que esta sea exitosa se debe encontrar el lector y el computador en la misma red la cual es 192.168.1.0/24, y el gateway por defecto de la computadora debe ser la 192.168.1.100. Entre otros parámetros que trae por defecto el lector RFID es el usuario y el password que son "alien" y "password" respectivamente.

Para el caso de la lectura y escritura del *tag*, en el *software* de Visual Basic se necesita hacer referencia a la librería API para RFID de Alien, que contiene funciones directas que en conjunto con otras herramientas del programa, permiten leer y escribir en el *tag*. Entre estas funciones se selecciona solo el tagID y es este el que se procesa y se modifica.

En los procesos anteriormente mencionados es necesario encontrarse conectado, por lo que se verifica siempre la conexión del computador con el lector a través de la función Reader.IsConnected.

III.3.2.3 Pruebas con el Lector RFID y DsPIC30F4011, vía puerto serial

Después de configurar todos los parámetros del UART, se programó en el microcontrolador el algoritmo mostrado a continuación, en lenguaje C:

1. Configurar el formato de la información recibida. Esto se hace para que el lector solo envié el código EPC de la etiqueta. Se usan los siguientes comandos "set taglistformat = custom" el lector responde con "TagListFormat = Custom", luego "TagListCustomFormat = %i" y el lector responde con el mismo comando.

2. Obtener la información de la etiqueta. Esto se logra con el comando "get taglist". Si no hay etiquetas al alcance de la antena, el lector responderá de la siguiente manera "(No Tags)" y de lo contrario mandará el código EPC de la etiqueta que tiene 24 caracteres hexadecimales en grupos de 4 separados por un espacio.
3. Escribir la información en la etiqueta. Esto se logra mandando el siguiente comando "ProgramEPC = XX XX XX XX XX XX XX XX XX XX XX XX" el código EPC se debe mandar en grupos de dos caracteres hexadecimales. Si la etiqueta está fuera del alcance del lector, él devolverá el siguiente comando "Error 134: *No Tag found.*" pero si el procedimiento no tiene problemas el lector responderá con el mismo comando enviado.

Como la conexión entre los dos dispositivos se hace por medio del puerto serial, se conectaron las salidas del UART del microcontrolador a un puerto DB9, solo se conectaron las líneas de transmisión, de recepción y de tierra.

La prueba consistió en revisar si el código EPC se escribía y si al remover la tarjeta, el dispositivo esperaba hasta que una etiqueta estuviera la alcance del lector.

III.3.3 Pruebas con el MAX232

El integrado MAX232 es el dispositivo que te permite pasar de voltajes TTL a RS232 y viceversa, para la elaboración del Trabajo Especial de Grado, este dispositivo es el acoplador entre las interfaces del microcontrolador, del puerto OBD del automóvil y del lector RFID.

La prueba consistió en realizar la conexión obtenida del *datasheet* del dispositivo, la conexión realizada fue la siguiente:

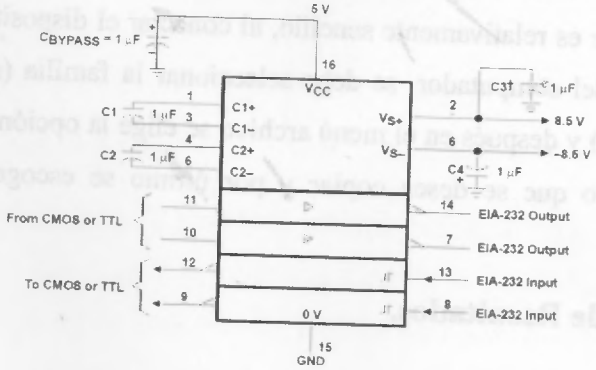


Figura 14 Especificaciones del MAX 232 (Texas Instruments, 2003)

En la primera prueba se verificó el cambio de TTL a RS232. Esta prueba consistió en introducir un voltaje de 5 V en el pin 10 y en observar el voltaje del pin 7, en esta caso la lectura del multimetro era de -8,5 V y al colocar un voltaje de 0 V en el pin 10, el voltaje del pin 7 cambiaba a 8,5 V.

La segunda prueba consistió en verificar como el voltaje cambia de RS232 a TTL. Para esto se conecto el pin 7 con el 8, y se observaron los voltajes en el pin 9, cuando se cambiaba el voltaje de 0 a 5 V en el pin 10.

III.3.4 Pruebas con el programador del DsPIC30F4011

Para grabar la información del programa en el dsPIC30F4011, se usó el *software* IPROGRAM 1.2.0.0 de la compañía ROSO Electric Supply, y el programador USB de la misma compañía.

Las conexiones se describen en la siguiente tabla:

Pin Programador	Pines DsPIC
1 (MCL)	1 (MCLR)
2 (+ 5 V)	11, 21, 32, 40 (Vdd)
3 (GND)	12, 20, 31, 39 (Vss)
4 (PGD)	25 (PGD)
5 (PGC)	26 (PGC)

Tabla XII Conexión del programador de microcontroladores (Fuente Propia)

Como se puede observar, la configuración es sencilla y se debe comprobar que las conexiones sean correctas porque de lo contrario, es imposible que el programa detecte el microcontrolador.

El *software* es relativamente sencillo, al conectar el dispositivo por medio del puerto USB del computador, se debe seleccionar la familia (que en nuestro caso es dsPIC30F) y después en el menú archivo se elige la opción importar, para obtener el archivo que se desea copiar y por último se escoge la opción de escribir.

III.4 Análisis de Resultados.

En esta etapa, se analizan los resultados obtenidos en las pruebas realizadas para poder hacer las conclusiones necesarias sobre el funcionamiento del sistema. La actividad que está ligada a esta etapa es el análisis exhaustivo de los resultados, por consecuente los resultados de las pruebas con el carro y/o con el sistema RFID se observan más detalladamente en el capítulo IV de resultados

IV.- Resultados

IV.1 Sistema

En el diagrama mostrado en la figura 15, se observan todos los equipos y sus conexiones, los cuales cumplen con determinadas funciones que permiten que el sistema realice la monitorización del automóvil.

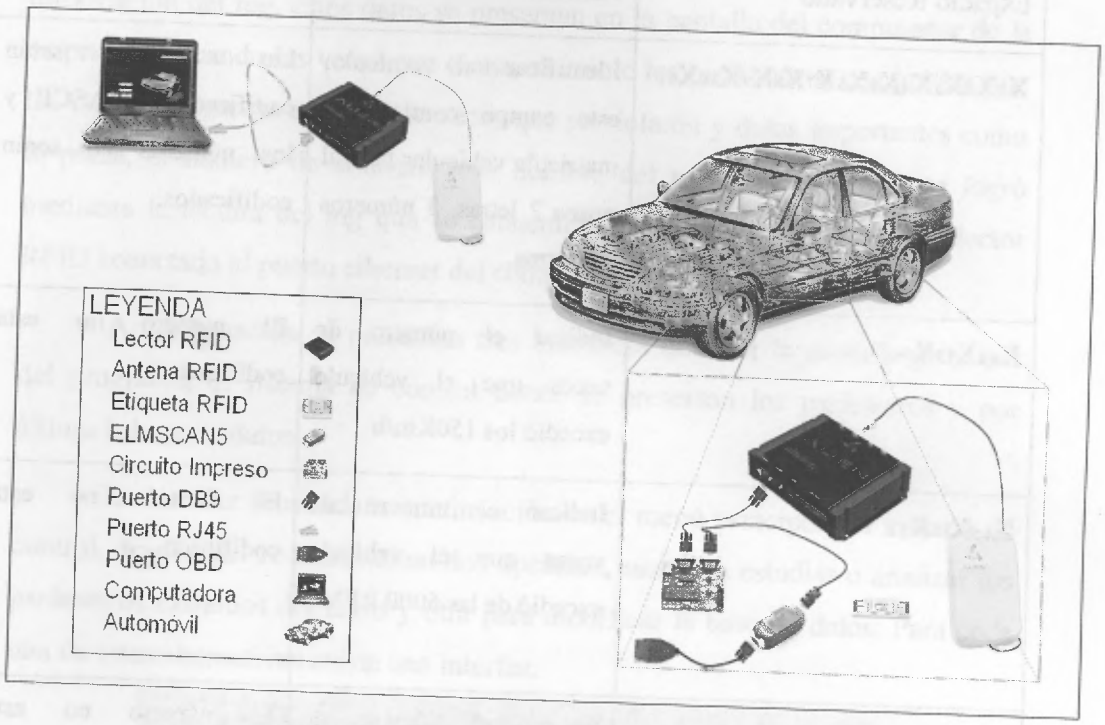


Figura 15 Sistema de Control (Fuente Propia)

A continuación se especifican los componentes utilizados:

- ✓ Adaptadores de Voltaje y Aislamiento: MAX 232
- ✓ Dispositivo de Administración: DsPIC30F4011
- ✓ Lector RFID: KRD 13M kimaldi (Sustituido por ALIEN 9900)
- ✓ Etiqueta RFID: *Tags* Mifare (Sustituido por ALIEN 9540)
- ✓ Conversor de RS232 – OBD II: ELMSCAN5

Cabe destacar que el lector y la etiqueta escogida son la mejor opción pero los dispositivos Alien que se encuentran en la universidad, nos brindan una solución que en cuanto a *performance*, la diferencia no es notable.

En la tabla mostrada a continuación se encuentran los parámetros que el DsPIC30F4011 adquiere de la computadora del automóvil y el formato de la información que reside en la memoria de la etiqueta.

EPC=$X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}X_{15}X_{16}X_{17}X_{18}X_{19}X_{20}X_{21}X_{22}X_{23}X_{24}$

Espacio Reservado	Función	Formato
$X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}$	Identificar al vehículo, este campo contiene la matricula vehicular la cual posee 2 letras, 3 números y 2 letras.	Las letras serán codificadas en ASCII y los números no serán codificados.
$X_{12}X_{13}X_{14}$	Indicar el número de veces que el vehículo excedió los 150Km/h	El número no está codificado.
$X_{15}X_{16}X_{17}$	Indicar el número de veces que el vehículo excedió de las 6000 RPM.	El número no está codificado.
$X_{18}X_{19}$	Indicar el número de veces que nivel del tanque de gasolina del vehículo este por debajo del 15 %.	El número no está codificado.
X_{20}	Indicar el número de veces que la temperatura del motor exceda los ° C.	El número no está codificado.
$X_{21}X_{22}X_{23}X_{24}$	Indicar el kilometraje recorrido con una luz de mal funcionamiento encendida	El kilometraje no está codificado.

Tabla XIII Parámetros del DsPIC30F4011(Fuente Propia)

Además de los equipos mencionados anteriormente se necesita un inversor de voltaje de 45 W, el cual permite convertir los 12VDC del automóvil a 110VDC para alimentar el sistema.

IV.1.1 Interfaz gráfica

La interfaz gráfica es la etapa del sistema en donde se procesa la información del *tag*, estos datos se presentan en la pantalla del computador de la empresa, indicando las veces que el usuario violó los parámetros, una gráfica que los relacione con las cantidad de veces que se violaron y datos importantes como la placa, el número de contrato y el nombre del usuario. Todo esto se logró mediante la lectura del *tag* que se encuentra en el vehículo, a través del lector RFID conectado al puerto ethernet del computador de la empresa.

A continuación se presentan tres interfaces que son la pantalla principal del programa, el sistema de control donde se presentan los parámetros y por último la base de datos.

La interfaz mostrada a continuación es el menú principal del sistema de control, desde aquí se seleccionan dos opciones, una para estudiar o analizar los parámetros extraídos del carro y otra para modificar la base de datos. Para cada una de estas alternativas existe una interfaz.

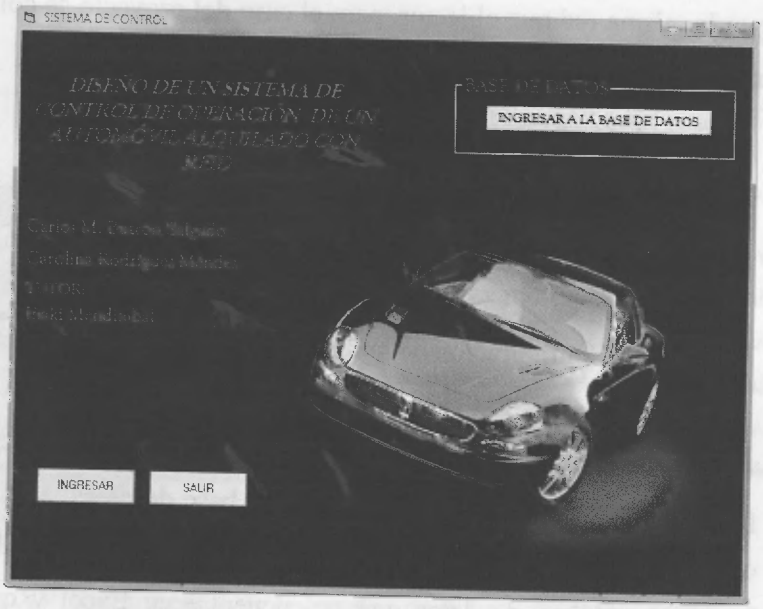


Figura 16 Menú principal del Sistema de Control (Fuente Propia)

HERRAMIENTAS	FUNCIÓN
INGRESAR	Permite al usuario conectarse al lector RFID, y abrir la ventana del sistema de control donde se estudian los parámetros del vehículo
SALIR	Permite al usuario salir del sistema
INGRESAR A LA BASE DE DATOS	Permite al usuario tener acceso a la base de datos para modificar o agregar nuevos datos del vehículo a alquilar

Tabla XIV Herramientas del menú principal del sistema de control (Fuente Propia)

En la interfaz principal se programó la conexión con el lector RFID al darle click al botón ingresar, esto se logra a través de las instrucciones:

```
Set Reader = New clsReader
On Error Resume Next
Reader.ConnectAndLogin "192.168.1.100", 23, "alien", "password"
```

En la última línea se puede observar la dirección IP del lector, el puerto, el usuario y la contraseña para poder establecer la conexión, parámetros que se encuentran por defecto en el lector.

Es necesario que la dirección de la computadora de la empresa se encuentre en la misma red que el lector, para el caso del proyecto se colocó como se muestra en la figura 17.

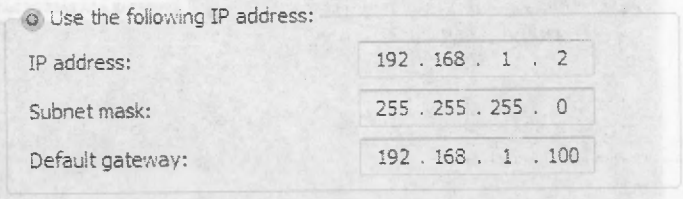


Figura 17 Configuración de la dirección IP de la computadora de la empresa

La siguiente figura muestra las ventanas a las que se pueden acceder desde el menú principal, la imagen de la izquierda es la base de datos, aquí se pueden ingresar o modificar información relevante del vehículo que se está alquilando y almacenarlos en una base de datos que se encuentre en Excel y que se debe almacenar con el nombre "basededatos" y la imagen de la derecha es el sistema de

control en donde se muestran las violaciones realizadas por el conductor. En está interfaz se exponen el número de violaciones de forma numérica y gráfica, además relaciona la placa guardada en la etiqueta con la base de datos de la empresa para que se visualicen los datos del usuario.

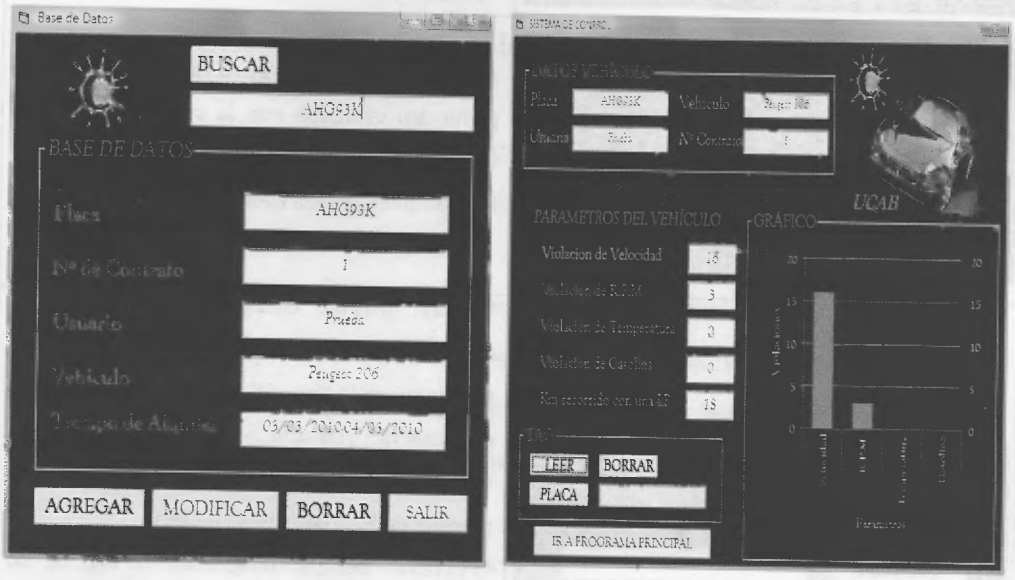


Figura 18 Interfaz de la base de datos y del sistema de control (Fuente Propia)

HERRAMIENTAS	FUNCIÓN
MODIFICAR	Permite ingresar y modificar datos de la base de datos
SALIR	Permite regresar al menú principal del sistema de control
BASE DE DATOS Nº de Contrato Usuario Vehículo Placa Tiempo Alquiler	En este bloque se tienen los datos de interés para la empresa sobre el vehículo alquilado. Se debe ingresar en cada botón en blanco el dato correspondiente a lo indicado en cada uno, para luego al tocar el botón de modificar se almacene en la base de datos de Excel

Tabla XV Herramientas de la interfaz de la base de datos (Fuente Propia)

HERRAMIENTAS	FUNCIÓN
DATOS VEHÍCULO Placa Vehículo Usuario Nº Contrato	En este bloque de la interfaz se presentan datos relevantes para el usuario de la empresa que se encuentran en el *tag* y en la base de datos como lo son: placa, vehículo, usuario y nº de contrato
GRÁFICO LEYENDA	Aquí se muestra un gráfico que relaciona cada parámetro evaluado con la cantidad de veces violados. Se presenta a su vez una leyenda para indicar que significa cada parte del gráfico
TAG LEER PLACA	"LEER" permite obtener los datos que se encuentran en el tag para procesarlos y en conjunto con la base de datos se llene el bloque de datos del vehículo, y a su vez se observen todos los parámetros del mismo, como se muestra en los siguientes bloques. "PLACA" permite programar el *tag*, si se alquila un vehículo se debe ingresar la placa en el cuadro de al lado y si se está regresando el *tag* se debe reiniciar colocando ceros en lugar de la placa.
Violación de Velocidad	Muestra la cantidad de veces que se violó el parámetro de velocidad
Violación de R.P.M.	Muestra la cantidad de veces que se violó el parámetro de revoluciones por minuto
Violación de Gasolina	Muestra la cantidad de veces que se violó el parámetro de gasolina
Violación de Temperatura	Muestra la cantidad de veces que se violó el parámetro de temperatura
Km recorrido con una LP	Muestra el kilometraje recorrido con un luz indicadora en el tablero de alguna falla
IR A PROGRAMA PRINCIPAL	Botón que permite regresar al programa principal

Tabla XVI Herramientas del Sistema de Control (Fuente Propia)

En la interfaz del sistema de control se realizaron las instrucciones para lectura y escritura del *tag*, y para el procesamiento de los datos en la etiqueta. El botón de leer no solo permite obtener los datos a través de la siguiente instrucción

```
If Reader.IsConnected = True Then
    Me.Caption = "Connected = " & Reader.IsConnected
    TagID = Mid(Reader.TagList, 1, 24)
    Text8.Text = TagID
```

Sino también muestra los valores en cada uno de los cuadros asignados para cada parámetro y muestra un gráfico indicando el comportamiento de los mismos durante el tiempo de alquiler. Al mismo tiempo se programó para que buscara en la base de datos todos los datos relacionados a la placa que se encuentra en el *tag* y que se muestra en el cuadro de datos del vehículo.

En el botón de la placa se utiliza la instrucción para escribir en el *tag* la información colocada en el cuadro en blanco al lado de dicho botón, esa instrucción es la siguiente:

```
If Reader.IsConnected = True Then
Text8.Text = data
  Dim status As String
   status = Reader.ProgramTag(data)
```

V.- Conclusiones y Recomendaciones

V.1 Conclusiones

El Sistema de Control de Operación de un Automóvil Alquilado con RFID, está dirigido a las empresas que deben controlar el uso de un vehículo automotor, permitiendo implementar sistemas de multas y de planificación del mantenimiento del vehículo, basándose en los parámetros mostrados en la aplicación final.

Todos los equipos utilizados para el proyecto fueron seleccionados según características teóricas, sin embargo con la integración de ellos se logró cumplir con todos los objetivos propuestos, por lo que se podría decir que el diseño del Sistema de Control se llevo a cabo sin problemas obteniendo buenos resultados

La aplicación final está realizada bajo el lenguaje de programación Visual Basic 6.0, y la comunicación se realiza a través del puerto de red. La interfaz del sistema de control tiene tres ventanas, la primera permite acceder al análisis de la información del *tag* o a la base de datos, la segunda ventana permite leer la información guardada en la etiqueta RFID, escribir la matrícula del vehículo y visualizar mediante gráficas las infracciones realizadas, y en la tercera ventana se puede modificar la base de datos donde se encuentra la información de los vehículos.

El sub-sistema que se encuentra en el vehículo está formado, por el DsPIC30F4011, el ELMSCAN5 y el lector RFID Alien 9900. El diseño propuesto incluye el lector Kimaldi ya que es la mejor opción para el Trabajo Especial de Grado, pero se decidió utilizar el equipo de Alien porque este se encontraba en la Universidad y no altera el rendimiento del sistema.

El DsPIC30F4011 está programado bajo el lenguaje C, y utiliza los dos UARTs disponibles para establecer la comunicación con el lector RFID y el ELMSCAN5, la transmisión de datos se realiza bajo el formato 8N1 y las velocidades son de 115200 y 38400 bps respectivamente.

Gracias a que los lenguajes de programación no son muy complicados es importante destacar que los parámetros que se obtienen del vehículo y que se

muestran en las herramientas del sistema pueden ser modificados de acuerdo a los requerimientos de la empresa.

La etiqueta RFID utilizada es la Alien 9540, tiene un espacio de 96 bits y contiene la información acerca del número de violaciones del umbral de la velocidad, del umbral de las revoluciones por minuto, del umbral del nivel del tanque de gasolina, del umbral de la temperatura del motor y del kilometraje recorrido con una luz indicadora de mal funcionamiento prendida, también es capaz de identificar al automóvil ya que se cuenta con un espacio para la matrícula vehicular.

Los umbrales pueden ser cambiados según los propósitos de las empresas, mediante un programador que sea compatible con el DsPIC30F4011.

Durante el desarrollo del Trabajo Especial de Grado, se observó que el Sistema RFID es una tecnología que puede expandirse a un sin fin de aplicaciones dependiendo prácticamente del criterio del usuario que la utilice. En el caso de este proyecto esta tecnología se pudo adaptar sin problema con todas las herramientas utilizadas.

V.2 Recomendaciones

- En el caso tal de implementar el proyecto se recomienda comprar los equipos seleccionados para el mismo:
  - ✓ *Tag* Mifare. Es compatible con el lector Kimaldi y contiene mayor capacidad de memoria para recopilar otros datos.
  - ✓ Lector Kimaldi. Tiene el tamaño más adecuado para colocarlo en el vehículo y tiene el puerto RS-232 para comunicarse con el dispositivo administrador, el DsPIC30F4011.
- Al instalar el *software* se recomienda leer el manual del mismo, y a su vez capacitar a los empleados de la empresa sobre el manejo de los equipos RFID.
- Es necesario capacitar a los empleados sobre el sistema de control antes de utilizarlo.

- Para aplicar un sistema de control de tránsito o la localización de automóviles robados, se pueden utilizar los lectores de mano: 13.56 MHz. RFID Handheld Reader / Writer, 140 Rugged PDA para los equipos recomendados y 902 MHz RFID Gen 2 UTE Portable Reader Writer para los equipos utilizados
- La adaptación de un GPS al dispositivo de administración, dependerá de los parámetros de la empresa o de la aplicación que se le de al sistema

Bibliografía

Adrían de Pérez, T. (2005). Recuperado el 20 de Septiembre de 2009, de http://prof.usb.ve/tperez/docencia/3413/contenido/ModBinsinPe.pdf

ALiens. (2009). *Aliens technology*. Recuperado el 2 de Octubre de 2009, de http://www.alientechnology.com/

Diagnostic, O. b. (17 de Abril de 2008). Recuperado el 12 de Septiembre de 2009, de http://www.onboarddiagnostics.com/page03.htm

Electronics, K. (s.f.). *Kimaldi*. Recuperado el 2 de Octubre de 2009, de http://www.kimaldi.com/productos/sistemas_rfid/lectores_rfid_y_tags_activos/tags_rfid_activos/tag_rfid_activo_sytag245_2c

Engineers, D. R. (2008). Recuperado el 8 de Febrero de 2009, de RFID Engineers: http://www.dipole.es/RFID/rfid.html

Engineers, D. R. (2008). *Dipole*. Recuperado el 10 de Febrero de 2009, de http://www.dipole.es/Lectores_RFID/Lectores_RFID_Lector.html

Engineers, D. R. (2008). *Dispositivos: Antenas RFID, Lectores RFID y Portales RFID*. Recuperado el 10 de Febrero de 2009, de http://www.antenasrfid.com/Dispositivos_RFID/Dispositivos_RFID.html

Ericsson. (08 de 08 de 2006). GSM Radio Network Tuning. 10. Caracas.

García Candel, V. (2006). *Estudio de la identificacion por radio frecuencia (RFID) y desarrollo del software relacionado con el control de la cadena de suministro* . Málaga.

Müller, G. (30 de Noviembre de 2007). *OBD Home Page*. Recuperado el 12 de Septiembre de 2009, de http://www.obd-ii.de/tech_conn.html

S/A. (2007). *Curso de microcontroladores*. Recuperado el Noviembre de 2009, de Pheltron: http://pheltron.com/es/CursoMicros/mod1/images/BITMAP/Gen/MCU%20Harvard.GIF

Samá Casanova, E., & Ciudad Herrera, J. M. (2007). *Estudio, diseño y simulación de un sistema RFID basado en EPC*.

Scantool.net. (2009). *Scantool*. Recuperado el 2 de Octubre de 2009, de http://www.scantool.net/elmscan-5-usb.html

Technology, K. (2007). *Kimaldi*. Recuperado el 2 de Octubre de 2009, de http://www.kimaldi.com/

Technology, K. (2007). *Kimaldi*. Recuperado el 3 de octubre de 2009, de http://www.kimaldi.com/productos/tarjetas_plasticas/tarjetas_rfid

Trapp, D. (2009). *Explorador de Vehiculo*. Recuperado el 3 de 10 de 2009, de http://www.vag-com.esànol.com/OBD-2/index.htmñ

Vega Anderson, L., Karina, V. M., & Mendizábal, I. (Febrero de 2009). Diseño de un sistema de monitorización inalámbrico en los vehículos de transporte de carga y contenedores de equipaje en los terminales aéreos. Trabajo Especial de Grado, UCAB. Caracas.

Apéndice A Datasheet DsPIC30f4011

MICROCHIP

dsPIC30F4011/4012

dsPIC30F4011/4012 Enhanced Flash 16-Bit Digital Signal Controller

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *"dsPIC30F Family Reference Manual"* (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

High-Performance, Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture with flexible addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 48 Kbytes on-chip Flash program space (16K instruction words)
- 2 Kbytes of on-chip data RAM
- 1 Kbyte of nonvolatile data EEPROM
- Up to 30 MIPS operation:
  - DC to 40 MHz external clock input
  - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 30 interrupt sources:
  - 3 external interrupt sources
  - 8 user-selectable priority levels for each interrupt source
  - 4 processor trap sources
- 16 x 16-bit working register array

DSP Engine Features:

- Dual data fetch
- Accumulator write-back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle
- ±16-bit, single-cycle shift

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
  - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports 4 Frame modes)
- I^2C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 1 CAN module, 2.0B compliant

Motor Control PWM Module Features:

- 6 PWM output channels:
  - Complementary or Independent Output modes
  - Edge and Center-Aligned modes
- 3 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-Bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

dsPIC30F4011/4012

Analog Features:

- 10-Bit Analog-to-Digital Converter (A/D) with 4 S/H inputs:
  - 1 Msps conversion rate
  - 9 input channels
  - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

Special Digital Signal Controller Features:

- Enhanced Flash program memory:
  - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
  - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)

Special Digital Signal Controller Features (Cont.):

- Flexible Watchdog Timer (WDT) with on-chip, low-power RC oscillator for reliable operation
- Fail-Safe Clock Monitor operation detects clock failure and switches to on-chip, low-power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
  - Sleep, Idle and Alternate Clock modes

CMOS Technology:

- Low-power, high-speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low-power consumption

dsPIC30F Motor Control and Power Conversion Family*

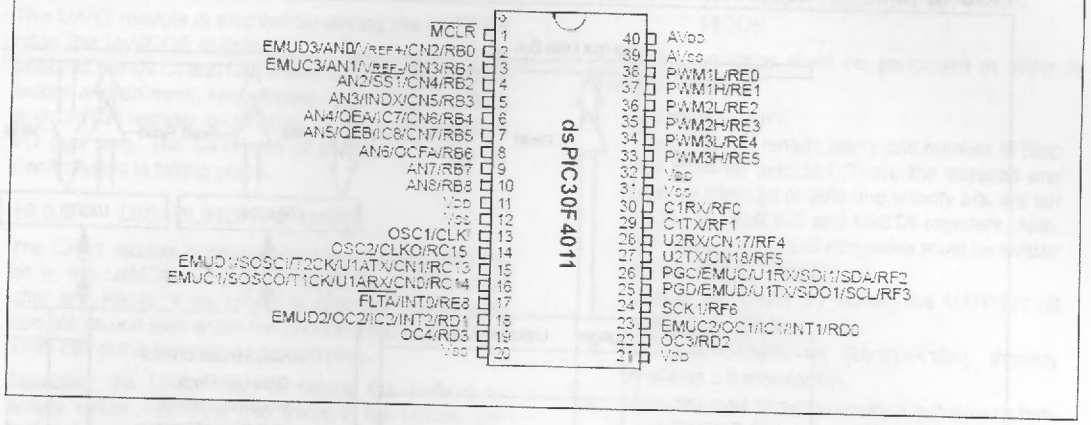
Device	Pins	Program Mem. Bytes Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp Std PWM	Motor Control PWM	10-Bit A/D 1 Msps	Quad Enc	UART	SPI	I²C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	-
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

* This table provides a summary of the dsPIC30F6010 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

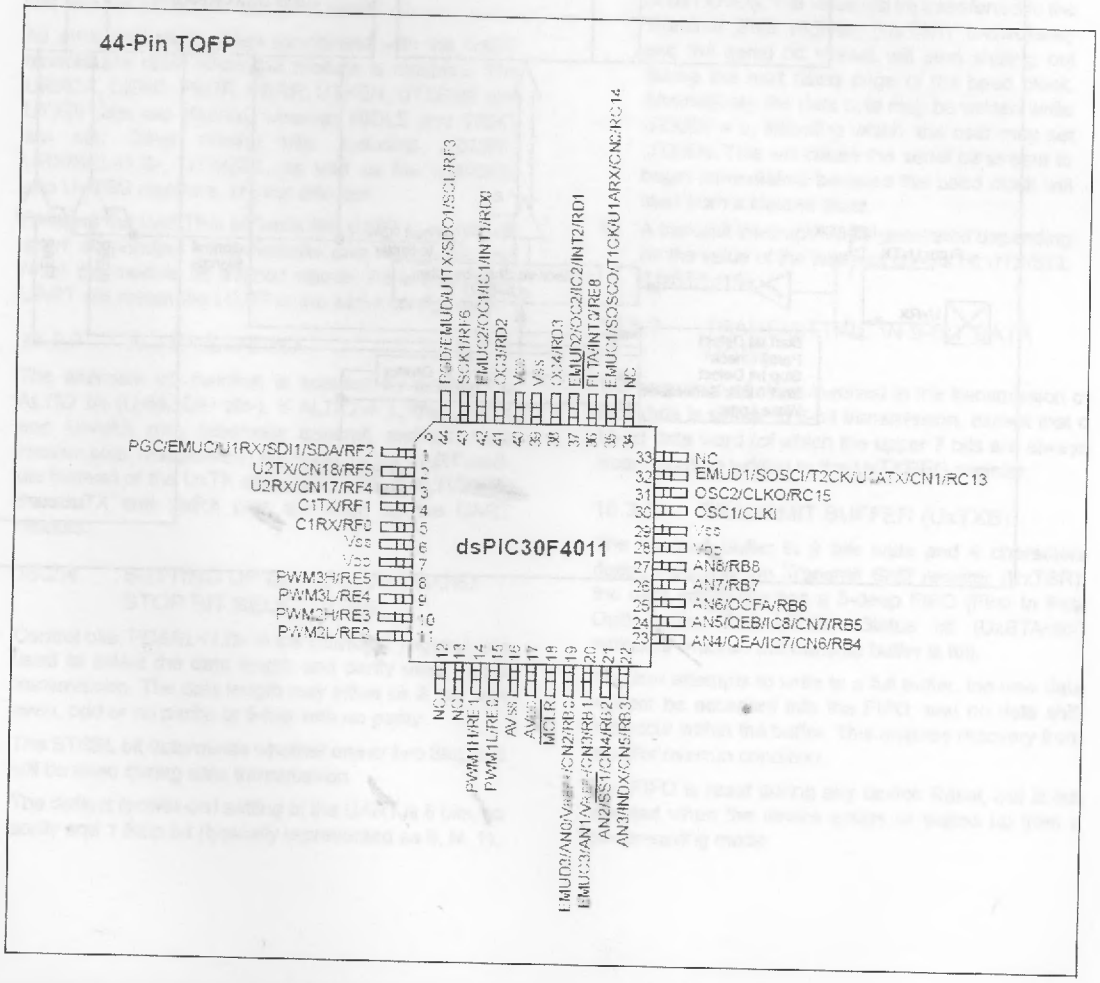
dsPIC30F4011/4012

Pin Diagrams

40-Pin PDIP



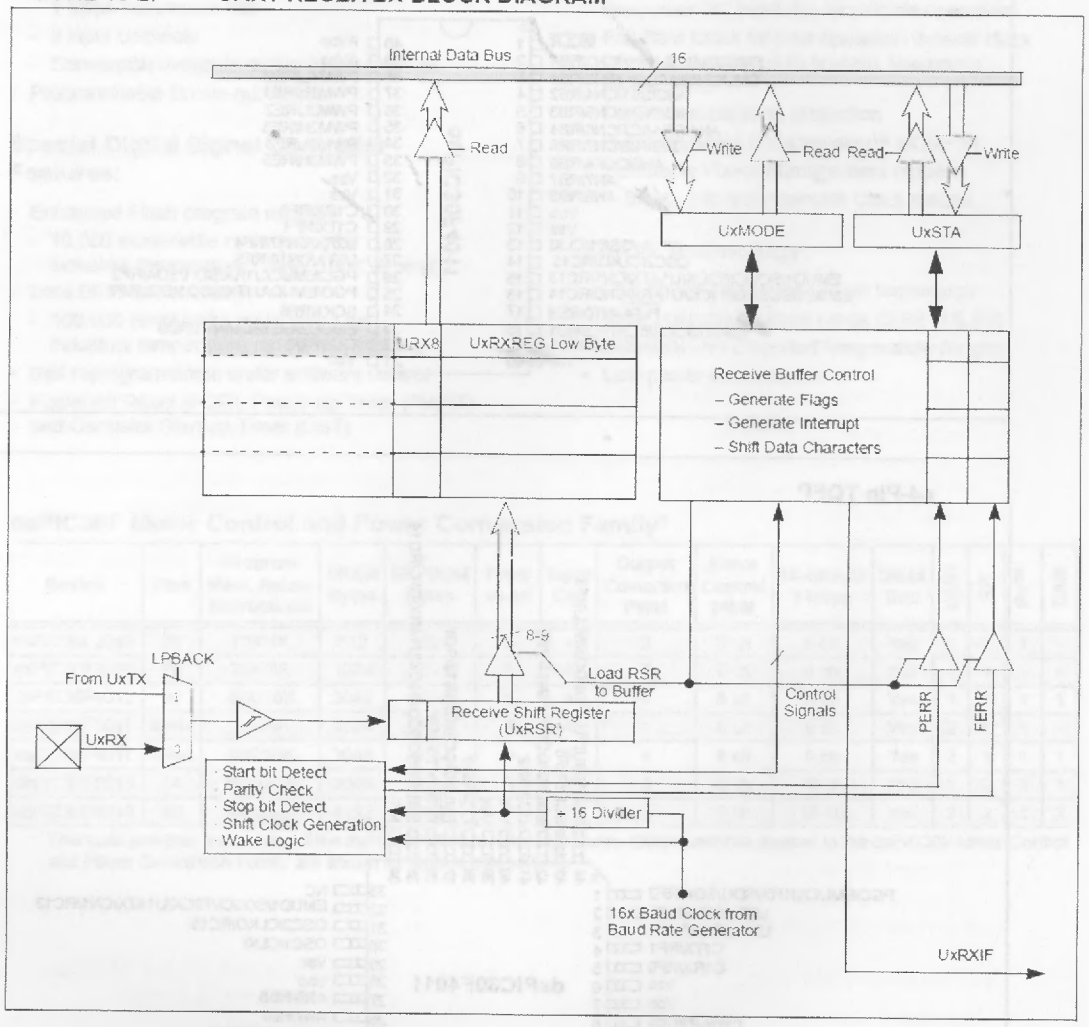
44-Pin TQFP



Apéndice B Uart DsPIC30f4011

dsPIC30F4011/4012

FIGURE 18-2: UART RECEIVER BLOCK DIAGRAM



18.2 Enabling and Setting Up UART

18.2.1 ENABLING THE UART

The UART module is enabled by setting the UARTEN bit in the UxMODE register (where x = 1 or 2). Once enabled, the UxTX and UxRX pins are configured as an output and an input, respectively, overriding the TRIS and LATCH register bit settings for the corresponding I/O port pins. The UxTX pin is at logic '1' when no transmission is taking place.

18.2.2 DISABLING THE UART

The UART module is disabled by clearing the UARTEN bit in the UxMODE register. This is the default state after any Reset. If the UART is disabled, all I/O pins operate as port pins under the control of the LATCH and TRIS bits of the corresponding port pins.

Disabling the UART module resets the buffers to empty states. Any data characters in the buffers are lost and the baud rate counter is reset.

All error and status flags associated with the UART module are reset when the module is disabled. The URXDA, OERR, FERR, PERR, UTXEN, UTXBRK and UTXBF bits are cleared, whereas RIDLE and TRMT are set. Other control bits, including ADDEN, URXISEL<1:0>, UTXISEL, as well as the UxMODE and UxBRG registers, are not affected.

Clearing the UARTEN bit while the UART is active will abort all pending transmissions and receptions and reset the module as defined above. Re-enabling the UART will restart the UART in the same configuration.

18.2.3 ALTERNATE I/O

The alternate I/O function is enabled by setting the ALTIO bit (UxMODE<10>). If ALTIO = 1, the UxATX and UxARX pins (alternate transmit and alternate receive pins, respectively) are used by the UART module instead of the UxTX and UxRX pins. If ALTIO = 0, the UxTX and UxRX pins are used by the UART module.

18.2.4 SETTING UP DATA, PARITY AND STOP BIT SELECTIONS

Control bits, PDSEL<1:0> in the UxMODE register, are used to select the data length and parity used in the transmission. The data length may either be 8 bits with even, odd or no parity, or 9-bits with no parity.

The STSEL bit determines whether one or two Stop bits will be used during data transmission.

The default (power-on) setting of the UART is 8 bits, no parity and 1 Stop bit (typically represented as 8, N, 1).

18.3 Transmitting Data

18.3.1 TRANSMITTING IN 8-BIT DATA MODE

The following steps must be performed in order to transmit 8-bit data:

1. Set up the UART:
   First, the data length, parity and number of Stop bits must be selected. Then, the transmit and receive interrupt enable and priority bits are set up in the UxMODE and UxSTA registers. Also, the appropriate baud rate value must be written to the UxBRG register.
2. Enable the UART by setting the UARTEN bit (UxMODE<15>).
3. Set the UTXEN bit (UxSTA<10>), thereby enabling a transmission.
4. Write the byte to be transmitted to the lower byte of UxTXREG. The value will be transferred to the Transmit Shift register (UxTSR) immediately and the serial bit stream will start shifting out during the next rising edge of the baud clock. Alternatively, the data byte may be written while UTXEN = 0, following which, the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
5. A transmit interrupt will be generated depending on the value of the interrupt control bit UTXISEL (UxSTA<15>).

18.3.2 TRANSMITTING IN 9-BIT DATA MODE

The sequence of steps involved in the transmission of 9-bit data is similar to 8-bit transmission, except that a 16-bit data word (of which the upper 7 bits are always clear) must be written to the UxTXREG register.

18.3.3 TRANSMIT BUFFER (UxTXB)

The transmit buffer is 9 bits wide and 4 characters deep. Including the Transmit Shift register (UxTSR), the user effectively has a 5-deep FIFO (First In First Out) buffer. The UTXBF Status bit (UxSTA<9>) indicates whether the transmit buffer is full.

If a user attempts to write to a full buffer, the new data will not be accepted into the FIFO, and no data shift will occur within the buffer. This enables recovery from a buffer overrun condition.

The FIFO is reset during any device Reset, but is not affected when the device enters or wakes up from a power-saving mode.

dsPIC30F4011/4012

18.9 Auto-Baud Support

To allow the system to determine baud rates of received characters, the input can be optionally linked to a selected capture input (IC1 for UART1, IC2 for UART2). To enable this mode, the user must program the input capture module to detect the falling and rising edges of the Start bit.

18.10 UART Operation During CPU Sleep and Idle Modes

18.10.1 UART OPERATION DURING CPU SLEEP MODE

When the device enters Sleep mode, all clock sources to the module are shut down and stay at logic '0'. If entry into Sleep mode occurs while a transmission is in progress, then the transmission is aborted. The UxTX pin is driven to logic '1'. Similarly, if entry into Sleep mode occurs while a reception is in progress, then the reception is aborted. The UxSTA, UxMODE, UxBRG, transmit and receive registers and buffers, are not affected by Sleep mode.

If the WAKE bit (UxMODE<7>) is set before the device enters Sleep mode, then a falling edge on the UxRX pin will generate a receive interrupt. The Receive Interrupt Select Mode bit (URXISEL) has no effect for this function. If the receive interrupt is enabled, then this will wake-up the device from Sleep. The UARTEN bit must be set in order to generate a wake-up interrupt.

18.10.2 UART OPERATION DURING CPU IDLE MODE

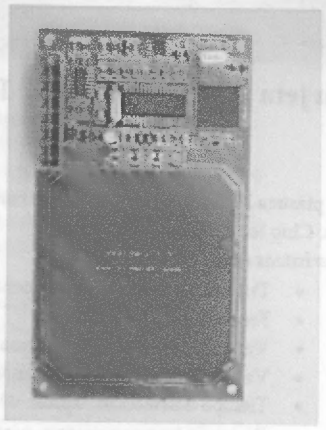
For the UART, the USIDL bit selects if the module will stop operation when the device enters Idle mode, or whether the module will continue on Idle. If USIDL = 0, the module will continue operation during Idle mode. If USIDL = 1, the module will stop on Idle.

Apéndice C Especificaciones Lector MIFARE KRD13M

Lector Mifare® Kimaldi® KRD13M

Lector / grabador de 13.56 MHz Mifare® compatible con la normativa ISO 14443A.

- Lector / grabador de tarjetas y tags sin contacto Mifare.
- Antena incorporada en el circuito impreso.
- La distancia de lectura es de hasta 5 cm. dependiendo del tag.
- Salidas: para 2 leds y 1 beeper.
- Programacion: comandos serie.
- Conexión: RS-232.
- Lector ideal para integracion.
- Posibilidad de suministrarse con carcasa y con distintos encapsulados.
- Fabricación OEM.
- Aplicaciones típicas: control de acceso, transporte, multi-aplicación, monedero, TPV's, vending, kioscos ...



Características técnicas

Voltaje:	5 VDC +/- 10%.
Consumo aprox.:	200 mA.
Interface:	RS-232, protocolo binario.
Baud rate:	9600, 19200 configurable.
Salidas:	señal TTL para control de 2 leds y 1 beeper.
Frecuencia:	13.56 MHz.
Distancia lectura:	Hasta 5 cm. dependiendo del tag
Normativa:	ISO 14443A.
Tags soportados:	Mifare standard (1 Kbyte), Mifare ultralight, Mifare 4 Kbyte.
Dimensiones:	70 mm ancho x 121 mm alto x 13mm profundidad.
Peso:	84 gr.

Apéndice D Especificaciones de la Tarjetas Proximidad Mifare 13,56Mhz

Kimaldi

Tarjetas proximidad 13,56 Mhz

Tarjeta prox. 13,56 Mhz. Lectura / Escritura modelo Mifare 1K

Tarjeta plástica PVC laminada por ambas caras. Tamaño ISO estandar: 85,7 x 54 mm. Grosor inferior a 0,9 mm. Chip lectura / escritura Mifare.

Características chip:

- Transferencia de datos: sin contacto.
- Frecuencia: 13,56 Mhz.
- Velocidad transferencia lectura: 106 Kbits/s.
- Velocidad transferencia escritura: 106 Kbits/s.
- Tiempo transacción: aprox. <100 ms.
- Distancia lectura/escritura: hasta 100 mm.
- Capacidad memoria total: 1024 bytes x 8 bit EEPROM.
- Memoria disponible: 752 bytes.
- Distribución de la memoria: 16 sectores con 4 bloques, cada bloque: 16 bytes.
- Integridad datos y tratamiento multi-tarjeta en un campo: Anti-colisión / El sistema selecciona una tarjeta activa y coloca todas las otras en una cola hasta que la transacción de la tarjeta activa se ha completado.
- Vida estimada del chip: 100.000 ciclos de escritura. 10 años de retención de datos.

Características de seguridad:

- Número de serie único (4 bbytes - ISO 14443)
- Rápido protocolo anti-colisión.
- Derechos de acceso individual para cada bloque de 48 bits.
- Llave individual para cada bloque.
- Cálculo de transferencia de autentificación mutua.
- Encriptación de datos durante la transacción.
- Cálculo checksum CRC.
- Código de transporte.

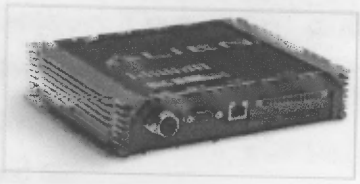
Apéndice E Especificaciones de los equipos Alien

ALIEN

ALR-9900 Enterprise RFID Reader

High Performance, Easy to Deploy, Easy to Manage

The Alien® ALR-9900 Enterprise Reader enables users to deploy manageable, robust, best-in-class EPC Gen 2 RFID solutions for supply chain, manufacturing and asset management applications. Based on the popular ALR-9800, the ALR-9900 adds enhanced interference rejection, monostatic antenna connections and a much smaller footprint. The ALR-9900 delivers high read rates in demanding applications in an easy-to-integrate, easy-to-manage package.



- High read rates for demanding applications
- Enhanced noise rejection for reliable data capture in noisy environments
- Easy integration with RFID software
- EPC Gen 2 interoperable
- Small footprint (8" x 8") and easy monostatic antenna layout
- Manageable and upgradeable

Interoperable and Broadly Supported

Alien pioneered the network-ready EPC RFID reader with the widely supported Alien Reader Protocol. The ALR-9900 is supported by key RFID platforms including Microsoft BizTalk RFID, IBM WebSphere 5.0, Oat Systems, and Oracle. Proven support for SAP through 3rd-party middleware is also available. A well-documented SDK featuring .NET and Java libraries enables easy, custom interfaces to control the reader if desired.

The ALR-9900 includes support for remote firmware management and remote monitoring via the Simple Network Management Protocol (SNMP), which enables the network to monitor real-time health, revision and status information from the reader.

Powerful Interfaces for Effective Integration

The Alien Reader Protocol features Autonomous Mode, a programmable state machine that enables the reader to operate independently, based on external triggers, timing or software inputs. This flexible system leads to best-in-class read rates by enabling users to precisely control the parameters for timing, protocols, antenna usage and other critical variables without network latency.

A flexible general-purpose input-output (GPIO) system enables tight integration with external sensors and actuators for effective integration with existing business processes. High capacity, optically isolated

ALR-9900 Enterprise RFID Reader

High Performance, Easy to Deploy, Easy to Manage

Model Number	ALR-9900
Architecture	XScale processor, Linux, 64 Mbytes RAM, 64 Mbytes Flash
Supported RFID Tag Protocols	EPC Gen 2; ISO 18000-6c
Reader Protocols	Alien Reader Protocol, SNMP; firmware upgradeable
LAN Protocols	DHCP, TCP/IP, NTP
Dense reader management	Dense Reader Mode, Event triggering
Frequency	902.75 MHz - 927.25 MHz
Channels	50
Channel Spacing	500 kHz
RF Power	Max 4 watts EIRP with Alien Antenna
Power	Tri-voltage AC/DC power converter; 45 Watts maximum - 120 or 240 VAC
Communications	RS 232 (DB 9 F), LAN TCP/IP (RJ-45)
Antennas	4 ports; interrogator topology; circular or linear polarization; reverse polarity TNC
General Purpose Inputs/Outputs	4 inputs, 8 outputs, optically isolated, 0.5 amp, requires external power source of no more than 24 volts
Dimensions	(L) 8.2" x (W) 8.3" x (D) 1.8"
Weight	1.5 kg (3.4 lb)
Operating Temperature	-20°C to +60°C (-4°F to +122°F)
Dust and Moisture	IP53
LED Indicators	Power, Link, Active, Ant 0-3, CPU, Read, Sniff, Fault (red)
Software SDK	Java and .NET APIs
Compliance Certification	Emissions: FCC Part 15 Safety: UL 60950

Copyright © 2007 Alien Technology Corporation. All rights reserved.
Alien, Alien Technology, and the Alien Technology logo are registered trademarks of Alien Technology Corporation in the USA and other countries. Other trademarks are the property of their respective owners.
Updated 2007

Anexo A Código del *Software* en Visual Basic 6.0 para trabajar con el Lector RFID

```
'Conexión con el lector
Private Sub ingresar_click(Index As Integer) ' Función para ingresar al sistema de control
 Set Reader = New clsReader
On Error Resume Next
   Reader.ConnectAndLogin "192.168.1.100", 23, "alien", "password" ' Verifica dirección IP, puerto, usuario y password
      If Reader.IsConnected = True Then 'Verifica conexión
      Me.Caption = "Connected = " & Reader.IsConnected
       Reader.AntennaSequence = "0"
       Reader.Persistime = 5
       Unload Me
       inicio.Hide
       VEHICULO.Show 'Abre interfaz del sistema de control
   Else
     MsgBox "Problemas con la conexión. Intentar de nuevo", vbOKOnly, "Error... "
    End If
   End Sub
```

Procedimientos para escribir y leer en el lector, codificando y decodificando los datos

```
Private Sub Command3_Click()
Dim programa As Integer
programa = MsgBox("¿Desea regresar al programa principal?", vbYesNo, "Cerrando...")
If programa = 6 Then
Text1.Text = ""
```

```
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""
Reader.Disconnect ' Desconecta la conexión con el lector
Set Reader = Nothing
inicio.Show
VEHICULO.Hide
End If
End Sub
Private Sub leer_Click()
Dim tag As String
Dim TagID As String
Dim placa As String
Dim rpm As String
Dim vel As String
Dim gas As String
Dim temp As String
Dim km As String
Dim vrpm As Long
Dim y(1 To 3) As String
Dim x(1 To 3) As Long
Dim vvel As Long
Dim i As Long
Dim f(1 To 7) As String
Dim o(1 To 14) As String
Dim a(1 To 2) As String
```

```
Dim b(1) As String
Dim parametro As Long
Dim letra As String
Dim re(1) As String
Dim revisar As String

'On Error Resume Next
 If Reader.IsConnected = True Then
   Me.Caption = "Connected = " & Reader.IsConnected
   revisar = Mid(Reader.TagList, 1, 4)
   re(1) = Mid(revisar, 1, 1)
   If re(1) = "T" Then
   TagID = Mid(Reader.TagList, 5, 34)
   Else
   If re(1) = "(" Then
   MsgBox "No Detecta ningún tag", , "TAG"
   Else
   TagID = Mid(Reader.TagList, 1, 29)
  ' TagID = "0000 0000 0000 0000 0000 0000"
   End If
   End If
   placa = Mid(TagID, 1, 13)

   b(1) = Mid(placa, 1, 1)
   If b(1) = "0" Then
   f(1) = Mid(placa, 2, 2)
   a(1) = Mid(placa, 4, 1)
   a(2) = Mid(placa, 6, 1)
   letra = Join(a, "")
   f(2) = Mid(letra, 1, 2)
   f(3) = Mid(placa, 7, 2)
   f(4) = Mid(placa, 9, 1)
```

```
    f(5) = Mid(placa, 11, 1)
    f(6) = Mid(placa, 12, 2)
    f(7) = ""
    Else
    f(1) = Mid(placa, 1, 2)
    f(2) = Mid(placa, 3, 2)
    f(3) = Mid(placa, 6, 1)
    f(4) = Mid(placa, 7, 1)
    f(5) = Mid(placa, 8, 1)
    a(1) = Mid(placa, 9, 1)
    a(2) = Mid(placa, 11, 1)
    letra = Join(a, "")
    f(6) = Mid(letra, 1, 2)
    f(7) = Mid(placa, 12, 2)
    End If
Dim n As Long
n = 1
While n < 8
Select Case f(n)
Case ""
  o(n) = f(n)
Case 41
  o(n) = "A"
Case 42
  o(n) = "B"
Case 43
  o(n) = "C"
Case 44
  o(n) = "D"
Case 45
  o(n) = "E"
Case 46
```

```
o(n) = "F"
Case 47
o(n) = "G"
Case 48
o(n) = "H"
Case 49
o(n) = "I"
Case "4A"
o(n) = "J"
Case "4B"
o(n) = "K"
Case "4C"
o(n) = "L"
Case "4D"
o(n) = "M"
Case "4E"
o(n) = "N"
Case "4F"
o(n) = "O"
Case 50
o(n) = "P"
Case 51
o(n) = "Q"
Case 52
o(n) = "R"
Case 53
o(n) = "S"
Case 54
o(n) = "T"
Case 55
o(n) = "U"
Case 56
```

```
    o(n) = "V"
  Case 57
    o(n) = "W"
  Case 58
    o(n) = "X"
  Case 59
    o(n) = "Y"
  Case 60
    o(n) = "Z"
  Case 0 To 9
    o(n) = f(n)
  If n = 3 Then
  Select Case f(7)
  Case 1 To 9
    o(7) = f(7)
  Case "A" To "Z"
    o(7) = ""
  End Select
  End If
  End Select
  n = n + 1
  Wend
      Dim vplaca As String
      vplaca = Join(o, "")

Set xlApp = Excel.Application
Set xlWB =
xlApp.Workbooks.Open(FileName:="C:\Users\Krola\Documents\basededatos.xls
")
Set xlSH = xlWB.Worksheets(1)
Application.DisplayAlerts = False
'On Error Resume Next
```

```
Cells.Find(What:=vplaca, After:=ActiveCell, LookIn:=xlFormulas, LookAt _
:=xlWhole, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:= _
False, searchformat:=False).Activate
If Err.Number <> 0 Then
MsgBox "No existe coincidencia", , "Error"
Else
Do While ActiveCell <> Empty
ActiveCell.Offset(1, 0).Select 'si la celda contiene datos, pasa a la fila siguiente
Loop
Dim texto As String
Do While ActiveCell.Row <> 2 And ActiveCell <> vplaca
ActiveCell.Offset(-1, 0).Select
Loop
If ActiveCell.Row = 1 Then
MsgBox "ERROR"
Else
Text1.Text = ActiveCell
ActiveCell.Offset(0, 1).Select
Text4.Text = ActiveCell
ActiveCell.Offset(0, 1).Select
Text2.Text = ActiveCell
ActiveCell.Offset(0, 1).Select
Text3.Text = ActiveCell
End If
End If
xlWB.Close True, "C:\Users\Krola\Documents\basededatos.xls"
xlApp.Quit
Set xlSH = Nothing
Set xlWB = Nothing
Set xlApp = Nothing

Set xlApp1 = Excel.Application
```

```
Set xlWB1 =
xlApp1.Workbooks.Open(FileName:="C:\Users\Krola\Documents\Grafico3.xls")
Set xlSH1 = xlWB1.Worksheets(1)
    rpm = Mid(TagID, 14, 4)
    y(1) = Mid(rpm, 1, 1)
    y(2) = Mid(rpm, 3, 1)
    y(3) = Mid(rpm, 4, 1)
i = 1
While i < 4
Select Case y(i)
Case "A"
  x(i) = 10
Case "B"
  x(i) = 11
Case "C"
  x(i) = 12
Case "D"
  x(i) = 13
Case "E"
  x(i) = 14
Case "F"
  x(i) = 15
Case 0 To 9
x(i) = CLng(y(i))
End Select
i = i + 1
Wend
Dim v(1 To 3) As String
Dim z(1 To 3) As Long
    vrpm = x(1) * 256 + x(2) * 16 + x(3)
    Text5.Text = vrpm
    Range("B3").Select
```

```
    ActiveCell.FormulaR1C1 = vrpm
    vel = Mid(TagID, 18, 4)
    v(1) = Mid(vel, 1, 1)
    v(2) = Mid(vel, 2, 1)
    v(3) = Mid(vel, 4, 1)

Dim g As Long
g = 1
While g < 4
Select Case v(g)
Case "A"
  z(g) = 10
Case "B"
  z(g) = 11
Case "C"
  z(g) = 12
Case "D"
  z(g) = 13
Case "E"
  z(g) = 14
Case "F"
  z(g) = 15
Case 0 To 9
z(g) = CLng(v(g))
End Select
g = g + 1
Wend
    vvel = z(1) * 256 + z(2) * 16 + z(3)
    Text9.Text = vvel
    Range("B2").Select
    ActiveCell.FormulaR1C1 = vvel
Dim t(1) As String
```

```
Dim q(1) As Long
    temp = Mid(TagID, 24, 1)
    t(1) = temp
Select Case t(1)
Case "A"
   q(1) = 10
Case "B"
   q(1) = 11
Case "C"
   q(1) = 12
Case "D"
   q(1) = 13
Case "E"
   q(1) = 14
Case "F"
   q(1) = 15
Case 0 To 9
q(1) = CLng(t(1))
End Select
    Text10.Text = q(1)
     Range("B4").Select
    ActiveCell.FormulaR1C1 = q(1)
Dim s(1 To 2) As String
Dim r(1 To 2) As Long
    gas = Mid(TagID, 22, 2)
    s(1) = Mid(gas, 1, 1)
    s(2) = Mid(gas, 2, 1)
Dim p As Long
p = 1
While p < 3
Select Case s(p)
Case "A"
```

```
    r(p) = 10
  Case "B"
    r(p) = 11
  Case "C"
    r(p) = 12
  Case "D"
    r(p) = 13
  Case "E"
    r(p) = 14
  Case "F"
    r(p) = 15
  Case 0 To 9
  r(p) = CLng(s(p))
  End Select
  p = p + 1
  Wend
  Dim vgas As String
      vgas = r(1) * 16 + r(2)
      Text11.Text = vgas
      Range("B5").Select
      ActiveCell.FormulaR1C1 = vgas
  Dim k(1 To 4) As String
  Dim j(1 To 4) As Long
      km = Mid(TagID, 26, 4)
      k(1) = Mid(km, 1, 1)
      k(2) = Mid(km, 2, 1)
      k(3) = Mid(km, 3, 1)
      k(4) = Mid(km, 4, 1)
  Dim l As Long
  l = 1
  While l < 5
```

```
Select Case k(l)
Case "A"
  j(l) = 10
Case "B"
  j(l) = 11
Case "C"
  j(l) = 12
Case "D"
  j(l) = 13
Case "E"
  j(l) = 14
Case "F"
  j(l) = 15
Case 0 To 9
j(l) = CLng(k(l))
End Select
l = l + 1
Wend
Dim vkm As String
    vkm = j(1) * 4096 + j(2) * 256 + j(3) * 16 + j(4)
    Text6.Text = vkm
    Range("B6").Select
    ActiveCell.FormulaR1C1 = vkm

Dim matrizDatos(1 To 4, 1 To 2)
Dim h As Integer
For h = 1 To 4
 ' Los valores de la columna A llenan la primera serie de las
 ' matriz. Si estos valores son cadenas, se convierten
 ' las etiquetas de las filas.
 matrizDatos(h, 1) = xlApp1.Worksheets(1) _
 .Range("A" & h + 1).Value
```

```
' Después los valores de la columna B llenan la segunda.
  matrizDatos(h, 2) = xlApp1.Worksheets(1) _
  .Range("B" & h + 1).Value

  Next h

MSChart1.ChartData = matrizDatos
xlWB1.Close True, "C:\Users\Krola\Documents\Grafico3.xls"
xlApp1.Quit
Set xlSH1 = Nothing
Set xlWB1 = Nothing
Set xlApp1 = Nothing
 Else
    MsgBox "error de conexión"
End If

End Sub
Private Sub placa_Click()
'On Error Resume Next
Dim v(1 To 35) As String
Dim w(1 To 7) As String
Dim m(1 To 14) As String
Dim d(1 To 7) As String
Dim data As String
Dim valor As String
Dim j As Integer
   valor = Text7.Text
w(7) = Mid(valor, 7, 1)
  w(1) = Mid(valor, 1, 1)
Select Case w(1)
Case "A" To "Z"
```

```
  w(1) = w(1)
Case "Ñ"
   MsgBox "No se acepta el caracter Ñ", , "ERROR..."
   GoTo Error
Case Else
   MsgBox "El primer caracter debe ser una letra en mayuscula", , "ERROR..."
   GoTo Error
End Select
   w(2) = Mid(valor, 2, 1)
Select Case w(2)
Case "A" To "Z"
   w(2) = w(2)
Case "Ñ"
   MsgBox "No se acepta el caracter Ñ", , "ERROR..."
   GoTo Error
Case Else
   MsgBox "El segundo caracter debe ser una letra en mayuscula", , "ERROR..."
   GoTo Error
End Select
   w(3) = Mid(valor, 3, 1)
Select Case w(3)
Case "A" To "Z"
   w(3) = w(3)
   j = 1
Case 0 To 9
   w(3) = w(3)
   j = 2
Case Else
   MsgBox "El tercer caracter debe ser un número o una letra en mayuscula", ,
"ERROR..."
   GoTo Error
End Select
```

```
  w(4) = Mid(valor, 4, 1)
Select Case w(4)
Case 0 To 9
  w(4) = w(4)
Case Else
  MsgBox "El cuarto caracter debe ser un número", , "ERROR..."
  GoTo Error
End Select
  w(5) = Mid(valor, 5, 1)
Select Case w(5)
Case 0 To 9
  w(5) = w(5)
Case Else
  MsgBox "El quinto caracter debe ser un número", , "ERROR..."
  GoTo Error
End Select
  w(6) = Mid(valor, 6, 1)
Select Case w(6)
Case "A" To "Z"
  w(6) = w(6)
Case "Ñ"
  MsgBox "No se acepta el caracter Ñ", , "ERROR..."
  GoTo Error
Case Else
  MsgBox "El sexto caracter debe ser una letra en mayuscula", , "ERROR..."
  GoTo Error
End Select
  w(7) = Mid(valor, 7, 1)
Select Case w(7)
Case ""
If j = 1 Then
  w(7) = w(6)
```

```
    w(6) = w(5)
    w(5) = w(4)
    w(4) = w(3)
    w(3) = w(2)
    w(2) = w(1)
    w(1) = "0"
  Else
   If j = 2 Then
    MsgBox "Verifique la placa", , "ERROR..."
    GoTo Error
   End If
  End If
  Case "A" To "Z"
    If j = 2 Then
    w(7) = w(7)
    Else
    MsgBox "La placa debe ser de 6 digitos", , "ERROR..."
    GoTo Error
    End If
  Case "Ñ"
    MsgBox "No se acepta el caracter Ñ", , "ERROR..."
    GoTo Error
  Case Else
   If j = 1 Or j = 2 Then
    MsgBox "Verifique la placa", , "ERROR..."
    GoTo Error
    End If
  End Select

  Dim u As Long
  u = 1
  While u < 8
```

```
Select Case w(u)
Case "A"
   m(u) = 41
Case "B"
   m(u) = 42
Case "C"
   m(u) = 43
Case "D"
   m(u) = 44
Case "E"
   m(u) = 45
Case "F"
   m(u) = 46
Case "G"
   m(u) = 47
Case "H"
   m(u) = 48
...
Case "Z"
   m(u) = "5A"
Case 0 To 9
m(u) = w(u)
End Select
u = u + 1
Wend
Dim plac As String
   plac = Join(m, "")
  v(1) = Mid(plac, 1, 1)
  v(2) = Mid(plac, 2, 1)
  v(3) = " "
  v(4) = Mid(plac, 3, 1)
  v(5) = Mid(plac, 4, 1)
```

```
v(6) = " "
v(7) = Mid(plac, 5, 1)
v(8) = Mid(plac, 6, 1)
v(9) = " "
v(10) = Mid(plac, 7, 1)
v(11) = Mid(plac, 8, 1)
v(12) = " "
v(13) = Mid(plac, 9, 1)
v(14) = Mid(plac, 10, 1)
v(15) = " "
v(16) = Mid(plac, 11, 1)
v(17) = "0"
v(18) = " "
v(19) = "0"
v(20) = "0"
v(21) = " "
v(22) = "0"
v(23) = "0"
v(24) = " "
v(25) = "0"
v(26) = "0"
v(27) = " "
v(28) = "0"
v(29) = "0"
v(30) = " "
v(31) = "0"
v(32) = "0"
v(33) = " "
v(34) = "0"
v(35) = "0"
data = Join(v, "")
```

```
 Dim status As String
 Dim revisar As String
 Dim re(1 To 4) As String
If Reader.IsConnected = True Then
 Me.Caption = "Connected = " & Reader.IsConnected
revisar = Mid(Reader.TagList, 1, 4)
  re(1) = Mid(revisar, 1, 1)
 If re(1) = "(" Then
   MsgBox "No detecta ningún tag", , "TAG"
 Else
   status = Reader.ProgramTag(data)
 End If
 Else
   MsgBox "Problemas de conexión", , "Conexión"

 End If
Error:
End Sub
```

Anexo B Código del DsPIC30f4011

```
#define __dsPIC30F4011__
#include <p30fxxxx.h>
#include <uart.h>

_FOSC(CSW_FSCM_OFF & XT_PLL8); // 4MHz Crystal with PLLx8 = 8
MIPS
_FWDT(WDT_OFF);                // Watch dog Timer OFF
_FBORPOR(PBOR_OFF & MCLR_EN);  // Brown Out OFF, MCLR Enabled
_FGS(CODE_PROT_OFF);           // Code Protect OFF
//Variables Goblales

// Reader

char BRX0[100];            // Buffer de recepción
char BRX1;
char * index=BRX0;         // Apuntador al buffer de Recepción
unsigned int i0=0;         // Bandera que indica el fin de la recepción (espera el
caracter mayor q)
unsigned int i1=0;         // Bandera que indica error en el procedimiento
char BTX0[] = {0x0D,'\0'};
char BTX1[] = {'S','e','t','
','t','a','g','I','i','s','t','f','o','r','m','a','t','=','c','u','s','t','o','m','\r','\n',0x00};
char BTX2[] =
{'t','a','g','I','i','s','t','c','u','s','t','o','m','f','o','r','m','a','t','=','%','i','\r','\n',0x00};
char BTX3[] = {'g','e','t',' ','t','a','g','I','i','s','t',0x0D,0x00};
char BTX4[] = {'P','r','o','g','r','a','m','E','P','C','='};
char BTX5[] =
{'0','0',0x20,'0','0',0x20,'0','0',0x20,'0','0',0x20,'0','0',0x20,'0','0',0x20,'0','0',0x20,'0','
0',0x20,'0','0',0x20,'0','0',0x20,'0','0',0x20,'0','0',0x0D,0x00};

//FORMULAS
char BufferRX2[] = {0x0D,'4','1',' ','0','C',' ','F','0',' ','0','0',0x20,0x0D};
char BufferRX3[] = {0x0D,'4','1',' ','0','D',' ','F','0',0x20,0x0D};
char BufferRX4[] = {0x0D,'4','1',' ','0','1',' ','F','0',' ','F','0',0x20,0x0D};
char BufferRX5[] = {0x0D,'4','1',' ','0','1',' ','F','0',0x20,0x0D};
char BufferRX6[] = {0x0D,'4','1',' ','0','1',' ','F','0',0x20,0x0D};
char BufferRX7[80];
char * pointer=0;
unsigned int CRPM = 0;
unsigned int CV = 0;
unsigned int CG = 0;
unsigned int CT = 0;
unsigned int FK = 0;

// Interrupciones
```

```
void __attribute__((__interrupt__)) _U1TXInterrupt(void)   //Rutina de
interrupción de transmisión
{
        IFS0bits.U1TXIF = 0;
}
void __attribute__((__interrupt__)) _U1RXInterrupt(void)   //Rutina de
interrupción de recepción
{
        IFS0bits.U1RXIF = 0;
        *index = (char)U1RXREG;
                if (*index=='>')
                        {
                        i0=1;
                        }
                if (*index=='N')
                        {
                        i1=1;
                        }
        index++;
}

// Funciones

// reader
void Send2readerI (char *A)
        {
        i0=0;
        i1=0;
        index=&BRX1;
        putsUART1(A); //Mandado
        while(BusyUART1()){}
        while (i0==0)
        {
        index=&BRX1;
        }
return;
        }


void Send2reader (char *A)
        {
        E1:
  i0=0;
        i1=0;
        index=BRX0;
        putsUART1(A); //Mandado
        while(BusyUART1()){}
```

```
        while (i0==0)
        {}
        if (i1==1)
        {
        goto E1;
        }
return;
        }

//Formulas

unsigned int convascii2d (unsigned int a)
        {
        if (a<0x3A)
                {
                a=a-0x30;
                }
        else
                {
                a=a-0x37;
                }
return a;
        }

unsigned int convd2ascii (unsigned int a)
        {
        if (a < 10)
                {
                a=a+0x30;
                }
        else
                {
                a=a+0x37;
                }
return a;
        }

char * buscar (char * a)
        {
        char * pointer;
        pointer=a;
        while(((*pointer != '4') || (*(pointer+1) != '1')) && ((*pointer != 0x20) ||
(*(pointer+1) != 0x0D)))
                {
                pointer++;
                }
        return pointer;
        }
```

```
void calcular (void)
        {
        unsigned int acumulador1=0;
        unsigned int acumulador2=0;
        unsigned int acumulador3=0;
        unsigned int acumulador4=0;
        char * pointer;

RPM:
        pointer=(buscar(BufferRX2));
        if (*pointer==0x20)
                {
                goto velocidad;
                }
        pointer=pointer+6;
        acumulador1=*pointer;
        pointer++;
        acumulador2=*pointer;
        pointer=pointer+2;
        acumulador3=*pointer;
        pointer++;
        acumulador4=*pointer;
        acumulador1=convascii2d(acumulador1);
        acumulador2=convascii2d(acumulador2);
        acumulador3=convascii2d(acumulador3);
        acumulador4=convascii2d(acumulador4);
        acumulador1=acumulador1*4096+acumulador2*256+acumulador3*16+ac
umulador4;
        if (acumulador1 > 24000)
                {
                CRPM++;
                }

velocidad:
        pointer=buscar(BufferRX3);
        if (*pointer==0x20)
                {
                goto kilometraje;
                }
        pointer=pointer+6;
        acumulador1=*pointer;
        pointer++;
        acumulador2=*pointer;
        acumulador1=convascii2d(acumulador1);
        acumulador2=convascii2d(acumulador2);
        acumulador1=acumulador1*16+acumulador2;
        if (acumulador1 > 150)
```

```
            {
            CV++;
            }
kilometraje:
        pointer=buscar(BufferRX4);
        if (*pointer==0x20)
            {
            goto gasolina;
            }
        pointer=pointer+6;
        if ((*pointer != '0')||(*(pointer+1) != '0')||(*(pointer+3) !=
'0')||(*(pointer+4)!= '0'))
            {
            FK++;
            }
gasolina:
        pointer=BufferRX5;
        pointer=buscar(BufferRX5);
        if (*pointer==0x20)
            {
            goto temperatura;
            }
        pointer=pointer+6;
        acumulador1=*pointer;
        pointer++;
        acumulador2=*pointer;
        acumulador1=convascii2d(acumulador1);
        acumulador2=convascii2d(acumulador2);
        acumulador1=acumulador1*16+acumulador2;
        if (acumulador1 > 38)
            {
            CG++;
            }
temperatura:
        pointer=buscar(BufferRX6);
        if (*pointer==0x20)
            {
            goto end;
            }
        pointer=pointer+6;
        acumulador1=*pointer;
        pointer++;
        acumulador2=*pointer;
        acumulador1=convascii2d(acumulador1);
        acumulador2=convascii2d(acumulador2);
        acumulador1=acumulador1*16+acumulador2;
if (acumulador1 > 113)
            {
```

```
                CT++;
                }
end:
        return;
        }

void analisis (void)
        {
        char * index2=BTX5;
        char * index3=BTX5;
        char * index4=BTX5;
        unsigned int acumulador1=0;
        unsigned int acumulador2=0;
        unsigned int acumulador3=0;
e0:
        if (CRPM > 0)
                {
                index2=&BTX5[16];
                index3=&BTX5[18];
                index4=&BTX5[19];
                acumulador1=BTX5[16];
                acumulador2=BTX5[18];
                acumulador3=BTX5[19];
                acumulador1=convascii2d(acumulador1);
                acumulador2=convascii2d(acumulador2);
                acumulador3=convascii2d(acumulador3);

        acumulador1=acumulador1*256+acumulador2*16+acumulador3+CRPM;
                CRPM=0;
                goto e1;
                }
        if (CV > 0)
                {
                index2=&BTX5[21];
                index3=&BTX5[22];
                index4=&BTX5[24];
                acumulador1=BTX5[21];
                acumulador2=BTX5[22];
                acumulador3=BTX5[24];
                acumulador1=convascii2d(acumulador1);
                acumulador2=convascii2d(acumulador2);
                acumulador3=convascii2d(acumulador3);

        acumulador1=acumulador1*256+acumulador2*16+acumulador3+CV;
                CV=0;
                goto e1;
                }
        if (CG > 0)
```

```
            {
            index3=&BTX5[25];
            index4=&BTX5[27];
            acumulador1=BTX5[25];
            acumulador2=BTX5[27];
            acumulador1=convascii2d(acumulador1);
            acumulador2=convascii2d(acumulador2);
            acumulador1=acumulador1*16+acumulador2+CG;
            CG=0;
            if (acumulador1>254)
                {
                *index3='F';
                *index4='F';
                goto ee;
                }
            goto e2;
            }
        if (CT > 0)
            {
            index4=&BTX5[28];
            acumulador1=BTX5[28];
            acumulador1=convascii2d(acumulador1);
            acumulador1=CT+acumulador1;
            CT=0;
            if (acumulador1>14)
                {
                *index4='F';
                goto ee;
                }
            goto e3;
            }
        goto ee;
e1:
        if(acumulador1 > 4094)
            {
            *index2='F';
            *index3='F';
            *index4='F';
            goto ee;
            }
        else
            {
            if (acumulador1 > 255)
                {
                acumulador2=acumulador1/256;
                acumulador2=convd2ascii(acumulador2);
                *index2=acumulador2;
                acumulador1=acumulador1%256;
```

```
                }
e2:
                if (acumulador1 > 15)
                        {
                        acumulador2=acumulador1/16;
                        acumulador2=convd2ascii(acumulador2);
                        *index3=acumulador2;

                        acumulador1=acumulador1%16;
                        }
e3:
                acumulador1=convd2ascii(acumulador1);
                *index4=acumulador1;
                }
        goto e0;
ee:
        if (FK > 0)
                {
                index2=buscar(BufferRX4);
                index2=index2+6;
                index3=&BTX5[30];
                *index3=*index2;
                index2++;
                index3++;
                *index3=*index2;
                index2=index2+2;
                index3=index3+2;
                *index3=*index2;
                index2++;
                index3++;
                *index3=*index2;
                FK=0;
                }
        return;
        }

// retardo
void retardo (void)
{
unsigned int i =0;
unsigned int j =0;
while (i!=50)
{
while (j!=65535)
{
asm("nop");
j++;
}
```

```
j=0;
i++;
}
return;
}

int main(void)
{
TRISE=0;
LATE=0;
char * index0;
char * index1 = BTX5;
unsigned int i2=0;
unsigned int baudvalue;
unsigned int U1MODEvalue;
unsigned int U1STAvalue;
TRISE=0;
//LATE=0;

CloseUART1();
ConfigIntUART1(UART_RX_INT_EN &
                    UART_RX_INT_PR2 &
        UART_TX_INT_EN &
        UART_TX_INT_PR6);
baudvalue = 11;
U1MODEvalue =  UART_EN &
                UART_IDLE_CON &
        UART_ALTRX_ALTTX &
        UART_EN_WAKE &
        UART_DIS_LOOPBACK  &
        UART_DIS_ABAUD &
        UART_NO_PAR_8BIT  &
        UART_1STOPBIT;

U1STAvalue =   UART_INT_TX &
        //UART_TX_PIN_NORMAL &
        UART_TX_ENABLE &
        UART_INT_RX_CHAR &
        UART_ADR_DETECT_DIS &
        UART_RX_OVERRUN_CLEAR;


OpenUART1(U1MODEvalue, U1STAvalue, baudvalue);

U1STAbits.UTXBRK = 0;


Send2readerI(BTX0);
```

```
Send2reader(BTX1);
Send2reader(BTX2);

while (1)
{
LATE=1;
retardo();
LATE=2;

Send2reader(BTX3);
index1=BTX5;
index0=&BRX0[13];

        while (*index0 != 0x0D)
        {
                if (*index1 != 0x20)
                {
                        if (*index0 != 0x20)
                        {
                        *index1=*index0;
                        index0++;
                        index1++;
                        }
                        else
                        {
                        index0++;
                        }
                }
                else
                {
                index1++;
                }
        }

calcular();
analisis();
Send2reader(BTX4);

}
CloseUART1();

return;
}
```