

Agradecimientos

Este Trabajo Especial de Grado no se hubiese podido llevar a cabo sin la ayuda incondicional que prestaron muchas personas. En primer lugar, agradecemos a el Ingeniero Iñaki Mendizábal, que ejerció las funciones de tutor de este proyecto, colaborando en todas las etapas de desarrollo del mismo.

Asimismo, agradecemos al personal de la empresa Movilnet, que colaboró en la facilitación y soporte técnico de los equipos y servicios de telecomunicaciones utilizados en las etapas de desarrollo y pruebas del sistema. En especial agradecemos al Ingeniero José María De Viana, que ejerce las funciones de Presidente de Movilnet, al Ingeniero Maximiliano Valencia, quien brindó ayuda en los aspectos técnicos, y a Julio Funes, quien nos brindó ayuda con la facilitación de los equipos y servicios requeridos para el proyecto.

Agradecemos igualmente a los Ingenieros Luis Da Costa y Joao Pinto, de la empresa Microtel, quienes, con su amplio conocimiento y experiencia en el área de telecomunicaciones, brindaron ayuda técnica con los equipos de transmisión.

De igual manera, agradecemos al Ingeniero Freddy Schlicht, de la empresa Ryokoo Motors, quien colaboró en el proyecto gracias a su experiencia en mantenimiento mecánico y diagnóstico computarizado de vehículos.

Finalmente, agradecemos a todas las personas que de alguna forma ayudaron a la exitosa culminación de este Trabajo Especial de Grado, en especial, a los profesores que dictan clases en la Escuela de Telecomunicaciones de la UCAB y a su directora Mayra Narváez, quienes brindaron su orientación en distintas etapas de desarrollo del mismo.

Elisa Huertas y Freddy Véliz, 13 de octubre de 2006

Resumen

El “Sistema Monitor Remoto Interactivo de vehículos” (SMRI) contempla el diseño y elaboración de un sistema de telemetría que permite obtener, en tiempo real, los parámetros proporcionados por el computador del vehículo, para ser utilizado por centros de servicio y mantenimiento automotriz. Este sistema está integrado por tres componentes: unidad remota, servidor y cliente. El proceso de monitorización se inicia cuando un cliente desea conocer los parámetros de un vehículo determinado, para lo cual se registra en el servidor. La unidad remota se reporta cada cierto tiempo y, al detectar que un cliente está registrado, se encarga de la extracción de los parámetros de la computadora del vehículo a través de la interfaz OBDII. La información obtenida se transmite de manera inalámbrica al servidor empleando la tecnología CDMA 2000 1X, el cual la retransmite al cliente.

Palabras clave: OBDII, diagnóstico de vehículos, monitorización inalámbrica y sistema de comunicación móvil de tercera generación.

Índice general

Agradecimientos	i
Resumen.....	ii
Introducción	1
I Planteamiento del problema.....	2
I.1 Planteamiento del problema.....	2
I.2 Objetivos.....	3
I.2.1 Objetivo general.....	3
I.2.2 Objetivos específicos	4
I.3 Alcance y limitaciones.....	4
II Marco Teórico.....	6
II.1 Qué es OBD y por qué surge	6
II.2 Protocolos OBDII	8
II.3 Descripción detallada de ISO 9141 y SAE J1979.....	10
II.3.1 ISO 9141	10
II.3.2 SAE J1979.....	14
II.4 Tecnologías de Transmisión Inalámbrica	19
II.4.1 Primera Generación.....	19
II.4.2 Segunda Generación.....	20
II.4.3 Tercera Generación.....	22
II.4.4 802.11.....	27
III Metodología	29
IV Desarrollo.....	31
IV.1 Selección del protocolo OBD.....	31
IV.2 Velocidad de los datos en ISO 9141	31
IV.3 Requerimientos de velocidad para el sistema inalámbrico.....	33
IV.4 Elección de la tecnología de transmisión inalámbrica	33
IV.5 Definición del hardware del módulo remoto	37
IV.5.1 Funciones Generales de la Unidad Remota	37

IV.5.2	Requerimientos de la Unidad Remota.....	40
IV.5.3	Definición de la Unidad de Procesamiento Principal	41
IV.5.4	Definición detallada del Hardware de la Unidad Remota.....	45
IV.6	Interacción con la unidad remota.	53
IV.7	Software de la Unidad Remota.	56
IV.8	Pruebas de la Unidad Remota	63
IV.8.1	Pruebas del software mediante MPLAB SIM.....	63
IV.8.2	Pruebas del módem CDMA.	64
IV.8.3	Pruebas de la Unidad Remota con módem simulado.....	64
IV.8.4	Pruebas de la Unidad Remota con vehículo simulado.....	65
IV.8.5	Pruebas de la Unidad Remota con vehículo real.....	67
IV.9	Diseño de las aplicaciones cliente-servidor	68
IV.9.1	Cliente	69
IV.9.2	Servidor.....	70
IV.10	Elección del lenguaje de programación para las aplicaciones cliente servidor	72
IV.11	Elección del manejador para la base de datos.....	74
IV.12	Diseño de la base de datos	75
IV.13	Pruebas del software servidor-cliente con vehículo simulado y corrección de errores	76
V	Resultados	77
VI	Conclusiones y Recomendaciones	83
VI.1	Conclusiones	83
VI.2	Recomendaciones.....	84
VII	Bibliografía	86
Apéndice 1.	Monitores de OBDII	1
Apéndice 2.	Protocolos OBD-II. Capas 1 y 2.	7
Apéndice 3.	Parámetros definidos por SAEJ1979 para la monitorización de vehículos	31

Apéndice 4. Evolución Tecnológica de los Sistemas de Telefonía Celular en Venezuela.....	40
Apéndice 5. Pruebas del módem CDMA.....	44
Apéndice 6. Pruebas con vehículo simulado.....	47
Apéndice 7. Pruebas del protocolo OBDII con sistema espía.	51

Índice de figuras

Figura 1	Protocolos OBD-II y estándares que definen cada capa.....	9
Figura 2	Arquitectura ISO 9141.....	11
Figura 3	Esquema de conexión del ECU y el equipo de diagnóstico.	11
Figura 4	Niveles de voltaje y sus respectivos valores lógicos	12
Figura 5	Ejemplo de trama ISO 9141. Orden de Envío de bits	12
Figura 6	Formato de mensajes de diagnóstico para ISO 9141.....	15
Figura 7	Tiempos solicitud/respuesta en SAE J1979 para ISO 9141	16
Figura 8	Mensajes de Petición y Respuesta para el Servicio 01	17
Figura 9	Formato de los mensajes de petición y respuesta en el servicio 3.....	18
Figura 10	Evolución de los Sistemas de Telefonía Celular	19
Figura 11	Sistemas Celulares de Tercera Generación.....	22
Figura 12	Tasa Máxima de Transmisión de Datos en CDMA.....	25
Figura 13	Tasa Máxima de Transmisión de Datos en GSM.....	27
Figura 14	Gráfico de Cobertura Digitel	35
Figura 15	Gráfico de Cobertura Movilnet.....	35
Figura 16	Módulo GPS EM-406 de GlobalSat.	38
Figura 17	Módulo EMII-800 para conexión a la red CDMA.	39
Figura 18	Solución de PC integrada modelo TS-7000 de Technologic Systems.	42
Figura 19	Microcontrolador PIC 18F4450 de 40 pines de Microchip.	42
Figura 20	Funciones y requerimientos del microcontrolador de la unidad remota....	43
Figura 21	Diagrama general de la unidad remota	46
Figura 22	Circuito de la fuente de alimentación de la unidad remota.	47
Figura 23	Configuración de interruptores del módem CDMA y módulo GPS.	48
Figura 24	Etapas de acoplamiento óptico.....	49
Figura 25	Esquema de circuito de la unidad remota.	51
Figura 26	Circuito Impreso de la unidad remota	52
Figura 27	Unidad Remota finalizada	53
Figura 28	Comandos y respuestas de la Unidad Remota.....	55

Figura 29 Diagrama de flujo del programa principal de la Unidad Remota.....	57
Figura 30 Comandos AT para establecer la conexión de datos	59
Figura 31 Diagrama de bloques de rutina <i>Interpretar Comando</i>	60
Figura 32 Método de monitorización de la Unidad Remota.....	62
Figura 33 Esquema de conexión de la Unidad Remota en modo local.....	65
Figura 34 Esquema de conexión de Unidad Remota con vehículo simulado.....	66
Figura 35 Diagrama de Interacción Vehículo-Usuario	69
Figura 36 Diagrama de flujo Cliente.....	70
Figura 37 Diagrama de flujo de software servidor.....	71
Figura 38 Diagrama general de interacción de los componentes del SMRI.....	78
Figura 39 Interfaz gráfica aplicación Servidor.....	79
Figura 40 Ventana principal aplicación Cliente.....	80
Figura 41 Ventanas de medición en tiempo real de la aplicación Cliente.....	81
Figura 42 Ventana de exploración de base de datos de aplicación Cliente.....	82

Índice de tablas

Tabla 1 Estándares 802.11	28
Tabla 2 Cuadro Comparativo entre microcontroladores.....	44

Introducción

El "Sistema Monitor Remoto Interactivo de vehículos (SMRI)" consiste en un sistema de telemetría que permite obtener, en tiempo real, los parámetros proporcionados por el computador del vehículo, para ser utilizados por centro de servicios y mantenimiento automotriz. El documento está dividido en capítulos que se describen a continuación.

El capítulo I contiene el planteamiento problema, los objetivos general y específicos propuestos para este trabajo especial de grado. Adicionalmente, se encuentra una sección de alcance y limitaciones del proyecto. El capítulo II corresponde al marco teórico, donde se realiza una breve explicación de OBDII y su funcionamiento; se incluye también una explicación de los protocolos OBDII seleccionados: ISO 9141 y SAE J1979. Posteriormente, en este capítulo, se realiza un estudio de las tecnologías inalámbricas existentes, enmarcadas en su evolución a través de los años.

La metodología seguida para el desarrollo de este proyecto se encuentra en el capítulo III, donde se exponen, brevemente, los pasos que se llevaron a cabo. Los estudios hechos, las pruebas realizadas y la descripción de los productos desarrollados se detallan en el Desarrollo, ubicado en el capítulo IV.

Finalmente, los resultados obtenidos en este Trabajo Especial de Grado se muestran en el Capítulo V. El capítulo VI expone las conclusiones del mismo y las recomendaciones para futuros desarrollos.

I Planteamiento del problema

I.1 Planteamiento del problema

El crecimiento de las redes y tecnologías inalámbricas actuales ofrecen, cada vez, mayor facilidad de incorporar sistemas de telemetría en distintos campos de la ingeniería, medicina, etc. En el área automotriz, los talleres mecánicos especializados que prestan servicios de mantenimiento y reparación de vehículos, encuentran complicado detectar ciertas fallas de funcionamiento que presentan algunos automóviles, debido a que resulta complejo simular, dentro de las instalaciones del taller, las condiciones reales que originan estas fallas, como pueden ser: pendientes inclinadas, tráfico pesado, conducción a altas velocidades, entre otros (SCHLICHT Freddy , conversación personal, Mayo 2005). Un sistema de telemetría permitiría al mecánico especializado evaluar los parámetros de funcionamiento de los vehículos en condiciones reales y, en base a éstos, realizar un diagnóstico más preciso que el obtenido mediante pruebas llevadas a cabo en el taller.

Además, una de las principales preocupaciones de los concesionarios de vehículos es el servicio post-venta que deben garantizar al cliente. Actualmente en Venezuela, estas compañías no disponen de un sistema que les permita detectar, de manera remota, deficiencias de funcionamiento de sus vehículos e informar al propietario el momento indicado para realizar los servicios de mantenimiento, basándose en parámetros reales medidos por la computadora de los mismos.

Un sistema de monitorización remota permitiría manejar estadísticas de fallas de vehículos de forma centralizada; además, facilitaría la elaboración de un historial para cada vehículo, permitiendo la realización de los diagnósticos de una manera más acertada. En caso de accidentes, un sistema de este tipo brindaría la posibilidad de

prestar un servicio de asistencia adecuado al tipo de falla presentada en el sitio donde se encuentre el vehículo.

Por otra parte, compañías con flotas de vehículos tales como: líneas de autobuses, taxis y camiones de transporte de carga, no disponen de un sistema central que les permita monitorizar el estado de las unidades y organizar la programación de los servicios de mantenimiento requeridos, sin tener que sacar sus vehículos de funcionamiento. Además, según el Cuerpo Técnico de Vigilancia de Tránsito y Transporte Terrestre (CTVTTT), una de las principales causas que originan accidentes de tránsito en el país es el trato indebido de las unidades por parte de los conductores (exceso de velocidad y conducción en vías no aptas para el tipo de vehículo). Este tipo de problema resulta imposible de detectar por la compañía administradora.

En vista de la situación expuesta, se presenta la necesidad de disponer de un sistema monitor remoto que permita examinar todos los parámetros proporcionados por la computadora del vehículo en tiempo real, de la forma más eficiente y económica posible.

I.2 Objetivos

I.2.1 Objetivo general

Diseñar el hardware y el software requerido para la implantación de un sistema monitor remoto interactivo de los parámetros de un vehículo, proporcionados por el computador del mismo a través de la interfaz OBDII.

I.2.2 Objetivos específicos

- Estudiar los sistemas de diagnóstico de vehículos (especificaciones ISO_9141).
- Estudiar requerimientos y desarrollar hardware para la adquisición de los datos.
- Estudiar, seleccionar y aplicar el sistema de transmisión inalámbrica más adecuado (CDMA 1x, GSM, GPRS, SMS, wireless 802.11,...).
- Estudiar la factibilidad de implantación de interfaz para GPS.
- Desarrollar una base de datos para facilitar la labor de diagnóstico del funcionamiento del vehículo.
- Realizar pruebas del sistema y optimización del funcionamiento de un vehículo con la colaboración de un mecánico especialista.

I.3 Alcance y limitaciones

El Sistema Monitor Remoto Interactivo (SMRI) permite la supervisión y diagnóstico de los parámetros proporcionados por vehículos en tiempo real. El proyecto contempla dos módulos de hardware, el primero para la extracción de los datos de la computadora del vehículo y el segundo es el encargado de la transmisión inalámbrica de los mismos. El primer módulo sólo es compatible con el estándar ISO-9141. Por su parte, el módulo de transmisión de la información se basa en plataformas de transmisión inalámbricas ya existentes en el mercado.

Para facilitar las labores de diagnóstico, se creó un servidor remoto al cual le son transmitidos los datos. Además, se estudió la posibilidad de incorporación de un dispositivo GPS; sin embargo, su implantación no formó parte de este proyecto, debido a que el tiempo pautado para la realización del mismo no lo permitió.

El sistema desarrollado se encarga de la extracción, transmisión y muestra de los parámetros de funcionamiento del vehículo. Sin embargo, el mismo no realiza ningún tipo de diagnóstico; éste sólo podrá ser realizado por un experto en mecánica automotriz en base a los datos mostrados por el sistema.

II Marco Teórico

Este capítulo se divide en dos secciones. En la primera, se exponen los estándares y protocolos que conforman la interfaz OBDII, la cual permite la extracción de los parámetros de funcionamiento de los vehículos. La segunda sección expone la evolución de las principales tecnologías inalámbricas disponibles actualmente, consideradas en la selección del sistema de transmisión de la Unidad Remota ubicada en el vehículo.

II.1 Qué es OBD y por qué surge

En los vehículos modernos existe una gran cantidad de módulos electrónicos repartidos en distintos puntos de los mismos, los cuales tienen como objetivo el control y la optimización del funcionamiento de las diferentes unidades mecánicas de los automóviles.

Las distintas unidades de control electrónica (denominadas ECU por sus siglas en inglés) deben comunicarse entre sí para poder optimizar el funcionamiento total del automóvil. Aprovechando la incorporación de gran cantidad de componentes electrónicos en los vehículos, la Agencia de Protección Ambiental (EPA) de los Estados Unidos de América y la *California Air Resources Board* (CARB), exigieron que ciertos parámetros de los vehículos, relacionados con las emisiones, pudiesen ser monitorizados para facilitar las prácticas de control ambiental.

Como resultado de las exigencias de implantación de sistemas de diagnóstico en los vehículos, surgieron varios estándares para regular los protocolos de comunicación entre las distintas ECU del vehículo y el sistema de diagnóstico

(*tester*). Los estándares principales que se pusieron en práctica a partir de 1996, que conforman el conjunto de protocolos OBD-II (*On Board Diagnostic System, 2nd Version*), provienen de la SAE (*Society of Automotive Engineers*) y de la ISO.

De acuerdo a Mike Thompson (THOMPSON, 2006), los monitores del sistema OBD II son pruebas específicas que realizan una serie de mediciones que le permiten a la computadora del vehículo tomar ciertas acciones para que el mismo trabaje de manera eficiente. Para proporcionar la máxima eficiencia, OBD divide el sistema completo de manejo y administración del motor en nueve subsistemas y cada uno de ellos tiene un monitor propio.

Cada monitor entra en funcionamiento bajo determinadas condiciones de operación del vehículo, es decir, un monitor determinado sólo entrará en funcionamiento cuando ocurran las condiciones que lo activan. Estas condiciones son similares entre los distintos fabricantes, sin embargo pueden presentarse variaciones.

A continuación se muestran los monitores presentes en OBD II: Eficiencia catalítica (Catalyst Efficiency); Componentes en general (Comprehensive Component Monitor, CCM); Recirculación del gas de escape; Sistema Evaporativo (Evaporative System, EVAP); Sistema de combustible (Fuel System, adaptive fuel system trim); Catalizador Calentado (Heated Catalyst); Sensor de oxígeno calentado (Heated Oxygen Sensor); Fallo de combustión (misfire ignition); Sistema de Aire Secundario (Secondary Air Injection); Termostato (Thermostat) y Aire acondicionado (Air Conditioning).

Dentro de OBD II existen dos clases de monitores, continuos y no continuos. Los monitores continuos trabajan cuando el vehículo está en funcionamiento y se producen determinadas condiciones que los activan. Los monitores no continuos entran en funcionamiento al producirse las condiciones que los activan y cuando el

resultado de una medida está fuera del rango esperado. Para ver una explicación más detallada de cada uno de los monitores ir al Apéndice 1

El objetivo principal de OBD II es mantener funcionando correctamente el convertidor catalítico y protegerlo de cualquier falla, que puede ser ocasionada por paso excesivo de combustible o un fallo de combustión. De esta manera, se reduce la cantidad de emisiones contaminantes y se liberan otros gases más amigables al medio ambiente.

Los monitores continuos son los correspondientes a: detección de fallo de combustión, sistema de combustible y componentes en general. El resto de los monitores no son críticos, por lo tanto son monitores no continuos.

El dispositivo administrador del sistema se asegura de que los monitores se ejecuten en el orden correcto, de prevenir conflictos de funcionamiento entre los monitores y de proporcionar un mecanismo de comunicación entre ellos y el equipo de medición externo.

II.2 Protocolos OBDII

Actualmente, los principales protocolos OBDII a nivel de capa física y de enlace son: SAE J1850 VPW (usada por General Motors), SAE J1850 PWM (usada por Ford) e ISO 9141 (usada por Chrysler hasta el 2002, la mayoría de los carros europeos y asiáticos actuales). Alrededor del año 2000, algunas marcas de vehículos comenzaron a producir modelos compatibles con un protocolo conocido como KWP 2000 (ISO 14230), que se establece como una mejora al ISO 9141, aunque gran cantidad de marcas de vehículos siguen siendo compatibles también con este último.

La figura 1 muestra los cuatro protocolos OBD-II, especificándose para cada uno, el estándar que define cada capa dentro del modelo OSI.

Se observa en la figura 1 que los tres primeros sistemas OBDII comparten el mismo protocolo a nivel de aplicación: SAE J1979 (técnicamente idéntico a ISO 15031-5), el cual define los mensajes entre el equipo de diagnóstico y las ECU. El KWP 2000 se diferencia en todas las capas debido a que maneja tamaños de mensajes sustancialmente mayores a los otros estándares.

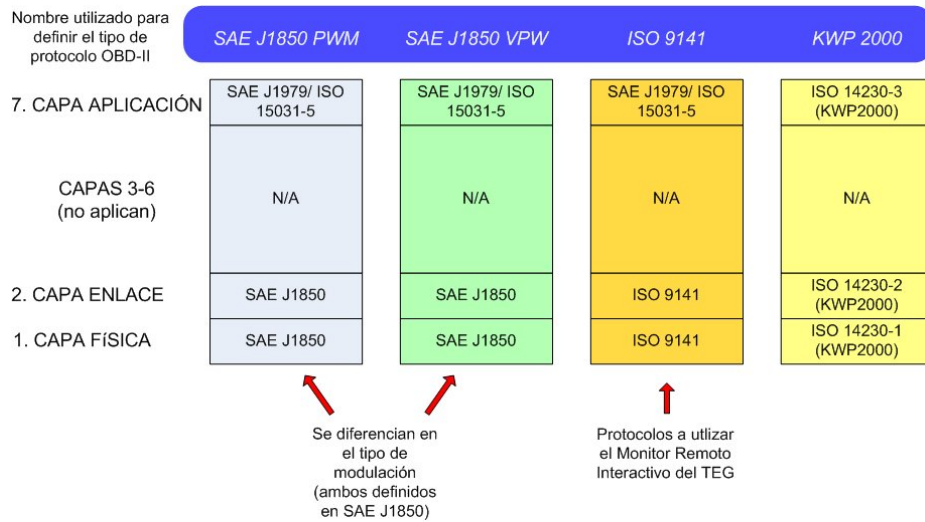


Figura 1 Protocolos OBD-II y estándares que definen cada capa

Todos los vehículos producidos y vendidos en Estados Unidos a partir del año 2008 deberán utilizar el protocolo ISO 15765-4 (CAN) como estándar para las funciones de diagnóstico, ya que ofrece, entre otras cosas, velocidades de hasta 512 Kbps. Este estándar ya ha sido adoptado en algunos vehículos. El protocolo CAN original fue creado por la empresa BOSCH para la comunicación de módulos electrónicos en industrias.

II.3 Descripción detallada de ISO 9141 y SAE J1979

En esta sección se presenta una descripción de los protocolos ISO 9141 y SAE J1979, los cuales definen las capas física-enlace y capa de aplicación respectivamente. Las definiciones para el resto de los protocolos OBDII se encuentran en el Apéndice 2.

II.3.1 ISO 9141

El estándar ISO 9141 (y su segunda revisión ISO 9141-2) define la capa física y de enlace de esta interfaz de diagnóstico. A nivel de capa de enlace, el protocolo es orientado a byte (muy parecido al protocolo RS-232), a diferencia del SAE J1850 que está orientado a mensajes con símbolos de encabezado, datos y CRC. Los mensajes de diagnóstico se forman en la capa de aplicación (SAE J1979) y son enviados byte a byte por la capa de enlace.

La arquitectura de la red ISO 9141 se muestra en la figura 2. Las líneas K y L son independientes y referenciadas a tierra. La línea K es en dos direcciones (*bidirectional*) y se usa para transmitir los mensajes de solicitud y respuesta de los parámetros; la línea L es unidireccional y solo se usa en el proceso de inicialización. A diferencia de SAE J1850, la interfaz ISO 9141-2 solo se usa para propósitos de diagnóstico, aunque también puede ser usada para otras comunicaciones (inclusive con cualquier otro protocolo) cuando no se encuentra conectado el equipo de diagnóstico.

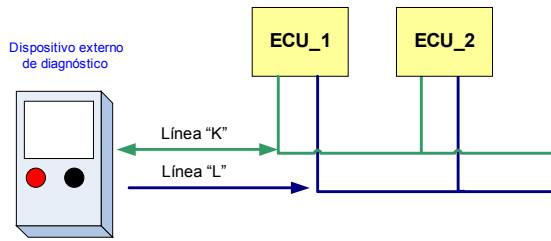


Figura 2 Arquitectura ISO 9141

En la figura 3 se muestra el esquema de conexión entre la ECU y el equipo de diagnóstico, definiendo así la capa física de esta interfaz.

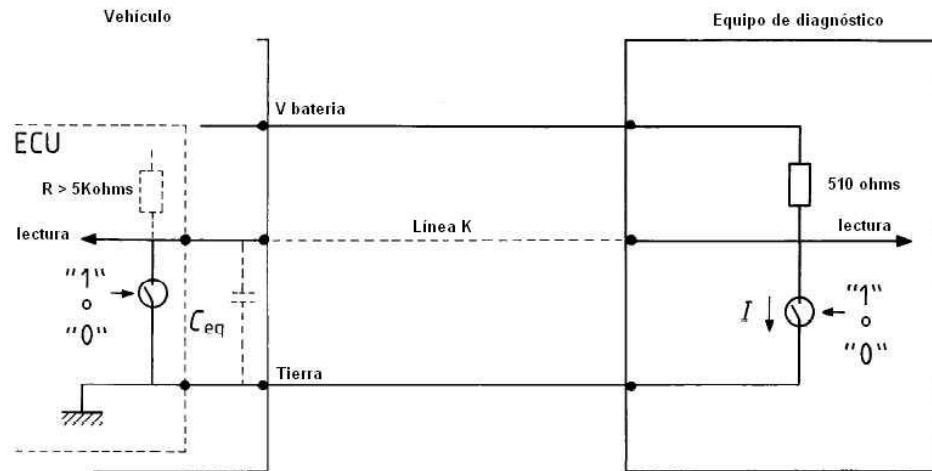


Figura 3 Esquema de conexión del ECU y el equipo de diagnóstico.

En ISO 9141 sólo existen dos símbolos: el "0" y el "1" y la codificación es NRZ. El "1" lógico se representa con un nivel igual al voltaje positivo de la batería (V_B , 12V ó 24 V) y el "0" lógico con el nivel de tierra. La figura 4 muestra el rango de valores que deben tener los símbolos transmitidos para ser interpretados como ceros o unos en el receptor. El rango denominado V_{oh} (max y mín) muestra el nivel de la señal (voltaje) a la salida del transmisor para representar el símbolo "1" lógico; V_{il} indica el rango de voltajes en el cual el receptor debe interpretar que se ha recibido un "0". De manera análoga se interpretan los otros valores.

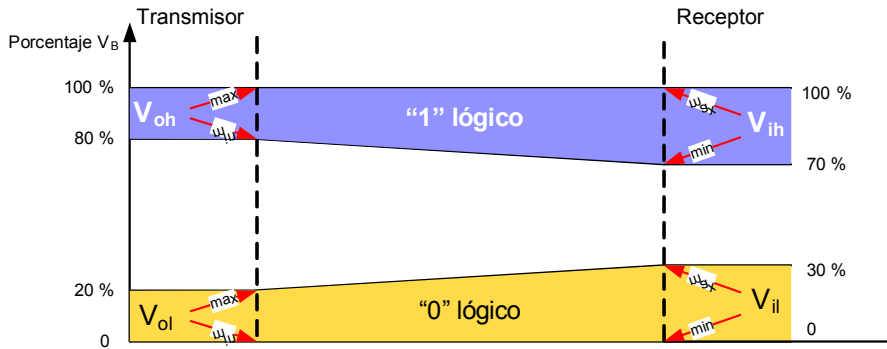


Figura 4 Niveles de voltaje y sus respectivos valores lógicos

En ISO 9141 se permite un rango muy amplio de tasas de bits: 10 bps a 10Kbps. Sin embargo, en la segunda revisión del estándar (ISO 9141-2) se fijó la tasa de bits en 10.4 Kbps, originando un tiempo de bit de 96,15 microsegundos. El tiempo máximo de transición entre dos niveles lógicos (del 20% al 80% de V_B y viceversa) se define como el 10% del tiempo de bit.

Las tramas a nivel de capa 2 son muy parecidas a las usadas en el protocolo RS-232 y están formadas por un bit de inicio (*start*), 8 bits de datos y por lo menos un bit de parada (*stop*). Cuando el bus está inactivo, el estado de las líneas “K” y “L” es “1” lógico, por lo cual el bit de *start* debe ser un “0” y el bit de *stop* un “1”. Los bits de datos se envían comenzando con el menos significativo. Véase figura 5.

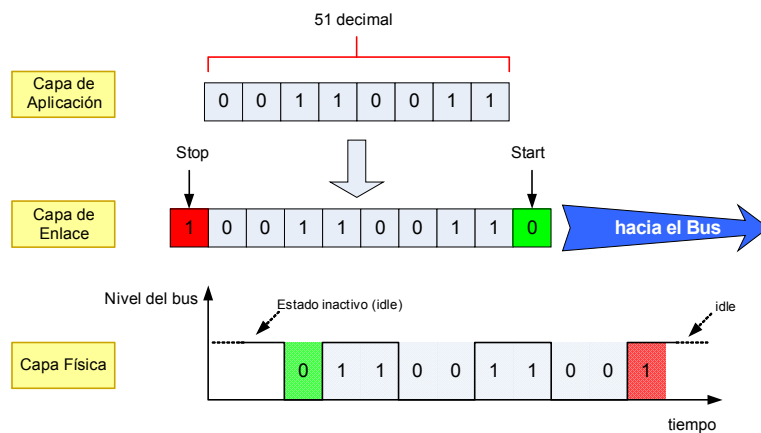


Figura 5 Ejemplo de trama ISO 9141. Orden de Envío de bits

A diferencia de SAE J1850, en el cual todos las ECU se comunican entre sí a través del mismo protocolo, el protocolo ISO 9141 sólo se usa para la comunicación entre el dispositivo de diagnóstico y las ECU. En SAE J1850, los mensajes de diagnóstico son un tipo de mensaje más, por lo cual pueden ser enviados en cualquier momento, siempre que el bus esté libre; en ISO 9141 la comunicación debe ser inicializada antes de poder llevarse a cabo. El protocolo de inicialización es el siguiente:

- a) El equipo de diagnóstico debe verificar que el bus se encuentre en estado inactivo (*idle*) durante un tiempo mínimo de 2ms y luego debe enviar un código de inicialización a una tasa de 5 bps sobre las líneas K y L simultáneamente. El código es “51” (decimal), con sus respectivos bits de *start* y *stop*. Luego la línea L es liberada.
- b) Sólo la ECU encargada de enviar las respuestas de diagnóstico contesta enviando un patrón de sincronización (01010101 binario). El tiempo que puede tardar el ECU en responder está comprendido entre 2ms y 2s. El objetivo de este mensaje es que el dispositivo de diagnóstico se pueda sincronizar a la tasa de bits del ECU. En ISO 9141-2 la tasa de bits se fijó a 10.4 Kbps, pero el byte de sincronización se sigue enviando.
- c) Después de un tiempo comprendido entre 2ms y 1,2s el ECU envía un mensaje que contiene una palabra clave (“*keyword*”) de 7 bits. El octavo bit de datos (el más significativo) funciona como bit de paridad, donde se busca que el byte posea siempre una cantidad impar de “1”. Luego, después de un tiempo comprendido entre 0,2ms y 1,2 s, se envía la segunda palabra clave. En total, el código clave consta de catorce (14) bits. Si las palabras claves son 0x08 y 0x08 indica que pueden existir uno o más ECU en la red; si las palabras claves son 0x94 y 0x94 se conoce que sólo existe un ECU en la red de diagnóstico. Estos valores tienen impacto en los tiempos entre mensajes de solicitud y respuesta definidos en SAE J1979.

- d) El dispositivo de diagnóstico confirma enviando el inverso de la segunda palabra clave.
- e) Finalmente el ECU envía el inverso del código de inicialización.

De esta manera, la interfaz de comunicación queda inicializada y lista para el intercambio de mensajes de diagnóstico. Si el bus permanece inactivo durante más de 5 segundos, se asume que la comunicación se ha finalizado. Si se deseara reanudar la comunicación, se tendría que repetir todo el proceso de inicialización.

II.3.2 SAE J1979

SAE J1979 define la capa de aplicación de OBD-II, especificando el formato de los mensajes de diagnóstico, código de los parámetros y definiendo los tipos de servicio para la evaluación del funcionamiento del vehículo. La revisión de este estándar realizada en Abril del 2002 cubre todos los protocolos OBD-II (inclusive el CAN), con el objetivo de unificarlos en capa de aplicación.

El formato de los mensajes de diagnóstico definido en SAE J1979 se adapta al de SAE J1850, debido a que este último define una estructura de trama con encabezado, campo de datos y símbolos especiales delimitadores. En cambio, ISO 9141 es un protocolo orientado a Byte, por lo que posee la flexibilidad de adaptarse a cualquier tipo de formato de mensaje en la capa de aplicación, ya que no se tiene un encabezado ni un campo de datos definido.

En la figura 6 se presenta el formato de los mensajes de solicitud y respuesta definido para ISO 9141. En realidad, todos los protocolos siguen básicamente el mismo formato, pero el valor de los bytes de encabezado varía según el estándar. En esta figura, los valores entre paréntesis indican el valor numérico del byte en notación hexadecimal.

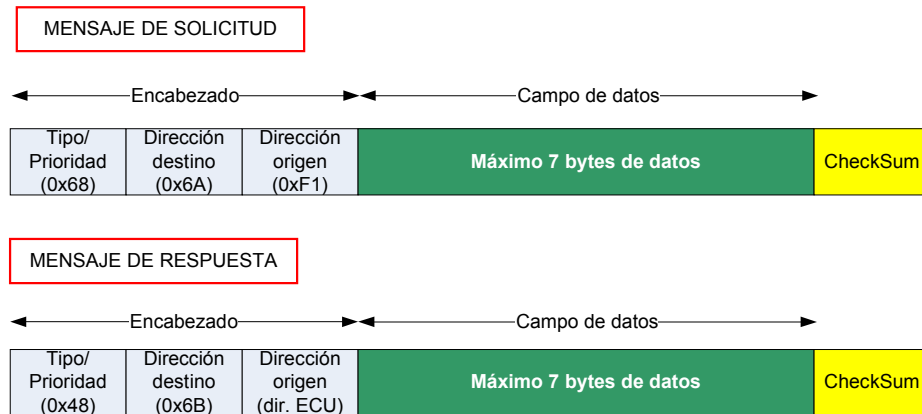


Figura 6 Formato de mensajes de diagnóstico para ISO 9141.

Se puede observar en la figura anterior que el campo de datos se definió con un máximo de 7 bytes, en concordancia con el formato de trama definido en SAE J1850. El último byte se utiliza para la verificación de errores y se define, para ISO 9141, como la suma de todos los bytes de encabezado y datos. El primer byte del campo de datos siempre corresponde al tipo de servicio.

Debido a que el tamaño del mensaje de diagnóstico es variable y el estándar ISO 9141 es orientado a byte (no posee símbolos delimitadores en el formato del mensaje), SAE J1979 define algunos lapsos de tiempo que, entre otras cosas, facilitan la detección de los mensajes en el equipo de diagnóstico y en la ECU. Además, el tamaño del mensaje específico para cada parámetro está definido en este estándar, pero la detección de los mensajes basada en esta propiedad obligaría a analizar los datos a medida que van llegando, sin certificar primero que el mensaje recibido se encuentre libre de errores.

Los lapsos de tiempo definidos en SAE J1979 para la interfaz ISO 9141 se muestran en la figura 7; estas definiciones influyen drásticamente en la velocidad de muestreo máxima de los parámetros de funcionamiento del vehículo.

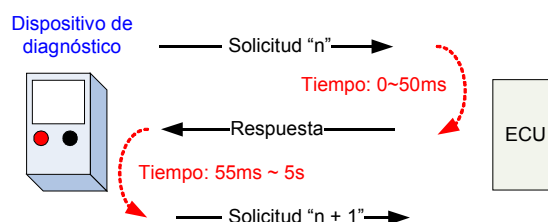


Figura 7 Tiempos solicitud/respuesta en SAE J1979 para ISO 9141

SAE J1979 define nueve tipos de servicio que puede soportar el ECU. El servicio 1 solicita los datos actuales de los sensores del vehículo; el servicio 2 da acceso a datos de los sensores almacenados en el ECU para el momento que ocurrió una falla; el servicio 3 da a conocer al usuario los códigos de falla arrojados por los monitores OBD-II ; el servicio 4 permite borrar la información de las pruebas y códigos de falla arrojados por los monitores; el servicio 5 solicita los resultados arrojados por el monitor de sensores de oxígeno; el servicio 6 da acceso a los resultados de pruebas específicas de otros monitores del vehículo; el servicio 7 permite acceder a códigos de falla presentados durante la última prueba de manejo (evalúa fallas momentáneas que pueden no aparecer en el servicio 3); el servicio 8 se define para tomar control de encendido y apagado de componentes y sistemas; el servicio 9 se define para solicitar información específica del vehículo (seriales principalmente). Los servicios 6, 8 y 9 son implementados por el fabricante y no son especificados a profundidad en el estándar.

II.3.2.1 Servicio 1: Acceso a datos en vivo del sistema a bordo

Este modo permite conocer los valores tomados por los parámetros relativos al funcionamiento del vehículo en el mismo momento que se realiza la medición. Para enviar un mensaje de petición bajo este modo se debe proporcionar un PID (*Parameter ID* o Identificador de parámetro), el cual permite especificar la información requerida al sistema de a bordo del vehículo. Los ECU son los

encargados de responder a estos mensajes con el último valor medido por el sistema, éstos no incluyen valores por defecto ni valores sustitutos en caso de que un sensor haya fallado. El formato de los mensajes de petición y respuesta para este servicio se encuentran en la figura 8.

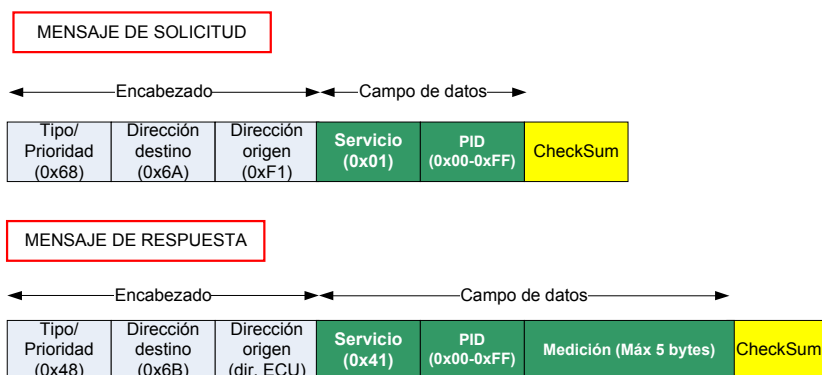


Figura 8 Mensajes de Petición y Respuesta para el Servicio 01

Los parámetros definidos por el estándar SAE J1979 para la monitorización de vehículos se listan en el Apéndice 3, donde se indica el PID correspondiente a cada parámetro, una breve descripción de la información que aporta cada uno de ellos, el número de bytes que ocupa la medición relativa a cada parámetro y los valores máximo y mínimo de dicha medición, los cuales están acompañados de las unidades métricas correspondientes a cada uno de los parámetros.

II.3.2.2 Servicio 2: Acceso a cuadro de datos congelados

El servicio 2 permite conocer los valores relacionados con las emisiones tomados por los ECU en el momento exacto de ocurrir una falla. Esta información es muy útil, ya que se puede conocer bajo cuáles condiciones ocurrió una determinada falla. El formato de los mensajes de petición y respuesta para este servicio es el mismo que en el servicio 1, sólo cambia el campo de identificador de servicio por el valor 0x02.

II.3.2.3 Servicio 3: Acceso a códigos de fallas DTC

Este servicio permite leer todos los códigos de falla arrojados por los monitores OBD. El procedimiento seguido por los equipos de diagnóstico para obtener estos códigos se realiza en dos pasos. El primer paso consiste en enviar un mensaje de petición del PID 1 bajo el servicio 1, para conocer el número de DTC disponibles por los ECU. Cada ECU que tenga almacenados uno o más DTC responderá con el número exacto de DTC que tiene almacenados; en caso de no tener información almacenada responderá con el código 0x00.

El siguiente paso es enviar un mensaje de petición de DTC bajo el servicio 3. Cada uno de los ECU enviará uno o más mensajes que contendrán un máximo de 3 DTC, si no existe información almacenada, no habrá respuesta a estos mensajes. El formato de los mensajes de petición y respuesta bajo este servicio se muestra en la figura 9.

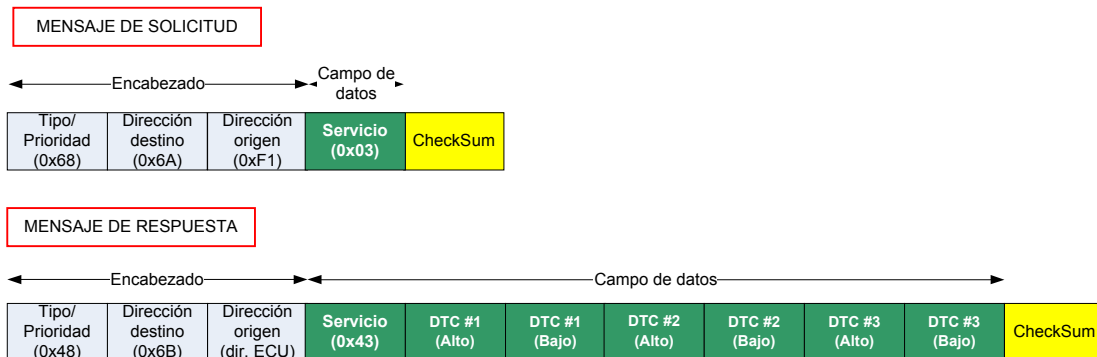


Figura 9 Formato de los mensajes de petición y respuesta en el servicio 3

Para conocer más detalles del resto de los servicios consultar el estándar SAEJ1979.

II.4 Tecnologías de Transmisión Inalámbrica

La telefonía móvil no surgió de manera espontánea, sino que sufrió una evolución que se expone, de forma no exhaustiva de acuerdo a Arturo Vera en su trabajo Sistemas Celulares de tercera generación, a continuación. Véase figura 10. Para observar cómo se produjo esta evolución en Venezuela ver Apéndice 4.

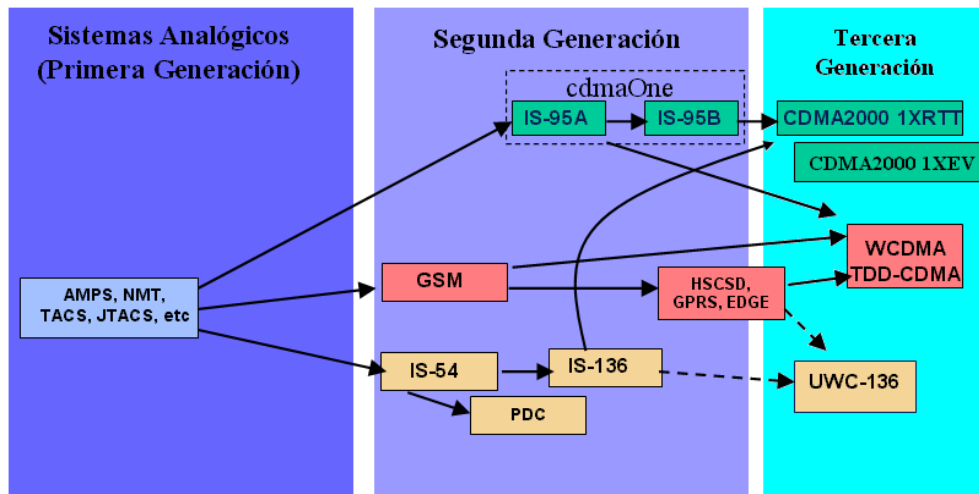


Figura 10 Evolución de los Sistemas de Telefonía Celular

(Fuente: <http://www.monografias.com/trabajos15/telefonía-celular/telefonía-celular.shtml>)

II.4.1 Primera Generación

En la primera generación de telefonía móvil celular se adoptó la técnica de acceso FDMA/FDD (Frequency Division Multiple Access / Frequency Division Duplex), la cual utilizaba el Acceso Múltiple por División de Frecuencia y dos frecuencias portadoras distintas para establecer comunicación entre el transmisor y el receptor. Las tecnologías desarrolladas dentro de la primera generación sólo podían prestar servicio de voz.

II.4.2 Segunda Generación

Como indica Alberto Sendín (Sendín, 2004) a partir de la fecha de consolidación del sistema AMPS de primera generación, es un hecho de gran relevancia a nivel mundial la separación de tendencias, de un lado los sistemas TDMA y de otro los sistemas CDMA, en el desarrollo de sistemas de comunicaciones móviles de segunda generación y siguientes.

GSM comenzó como una norma europea para unificar sistemas móviles digitales y fue diseñado para sustituir a más de diez sistemas analógicos en uso y que en la mayoría de los casos eran incompatibles entre sí.

Como lo explica Alberto Sendín (Sendín, 2004) la transición de de los sistemas GSM de segunda generación a tercera generación no se llevará a cabo de manera brusca, sino que por el contrario las redes se instalarán de manera progresiva. A este efecto, hacen aparición HSCSD, GPRS y EDGE

II.4.2.1 High Speed Circuit-Switched Data (HSCSD)

HSCSD se trata de un servicio derivado de GSM, estandarizado por la ETSI, que dedica múltiples ranuras de tiempo a un sólo usuario de forma de incrementar la tasa de datos, sin cambiar la interfaz de radio, alcanzando velocidades de 14.4 Kbps. Se pueden usar dos configuraciones: simétrica o asimétrica. Las aplicaciones típicas corresponden a elevados volúmenes de información: fax, acceso a bases de datos, imágenes, etc.

II.4.2.2 General Packet Radio System (GPRS)

GPRS, estandarizado por la ETSI dentro de GSM 2.5G, es un servicio paquetizado diseñado para: transmisión frecuente de pequeños volúmenes de datos (por ejemplo, navegación de Internet), transmisión poco frecuente de volúmenes moderados de datos (por ejemplo, acceso a archivos), servicios de transmisión punto a punto (PTP) y punto a multipunto (PTM).

A causa de estos requisitos, fue natural que GPRS eligiera un estándar TDMA de 30 KHz puesto que los sistemas analógicos existentes trabajan ya con este ancho de banda por canal. En GPRS se transmiten tres canales por cada portadora de 30 KHz. Para esta tecnología, la velocidad de transmisión en sentido descendente, se encuentra entre 18 y 53 Kbps y en sentido ascendente entre 9 y 13 Kbps.

II.4.2.3 CDMA ONE

CDMA ONE es un nombre comercial de marca registrada reservado para uso exclusivo de las empresas que son miembros de CDG (Cdma Development Group). El mismo describe un sistema inalámbrico completo que incorpora la interfaz aérea IS-95 CDMA y la norma de la red ANSI-41 para la interconexión por conmutación, además de muchas otras normas que integran el sistema inalámbrico completo.

II.4.2.4 CDMA ONE IS-95-A

CDMA ONE IS-95-A ofrece soporte a señales de voz conmutados por circuitos y datos, conmutados por circuitos o paquetes, con velocidades de hasta 14,4kbps. Debido al enfoque inicial de proveedores y operadoras en señales de voz, este sistema

ha sido utilizado sólo para voz conmutada por circuitos y, más recientemente, para un pequeño volumen de datos conmutados por circuitos.

II.4.2.5 CDMA ONE IS-95-B

CDMA ONE IS-95-B ofrece soporte a señales de voz conmutados por circuitos y datos, conmutados por paquetes. En teoría, ella provee tasas de datos de hasta 115kbps, y alcanza, generalmente, valores prácticos de 64kbps. CDMA ONE IS-95-B está siendo sustituida por CDMA2000 1X.

II.4.3 Tercera Generación

La división de los sistemas de comunicaciones móviles supera la segunda y se prolonga a la tercera generación. En la figura 11 se muestra la división de las tecnologías de tercera generación.

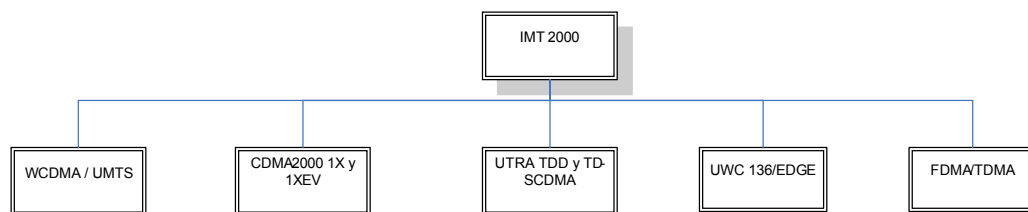


Figura 11 Sistemas Celulares de Tercera Generación

Dentro de estos sistemas amparados por la UIT bajo la denominación paraguas IMT-2000 (la cual es una recomendación UIT-R M.1457) se recogen 5 estándares, la mayoría de ellos pertenecientes a CDMA y otros TDMA. De acuerdo a Alberto Sendín (Sendín, 2004) dos de estos sistemas son los que poseen más futuro: CDMA 2000 y UMTS (W-CDMA), ambos utilizan CDMA de banda ancha.

II.4.3.1 CDMA 2000

El camino evolutivo de CDMA a IMT-2000 empieza con la propuesta de *Qualcomm* de un nuevo sistema basado en técnicas de espectro ensanchado. Esta propuesta, que luego fue estandarizada como IS-95, es el primer sistema CDMA móvil en desarrollo comercial. El acceso de multiplexado por división de códigos de banda estrecha CDMA IS-95 estipula un espaciado de portadora de 1.25MHz para servicios de telefonía. TIA (*Telecommunications Industry Association*) empezó a definir esta especificación en 1991.

CDMA2000 identifica la norma TIA para tecnología de tercera generación, que es un resultado evolutivo de CDMA ONE, el cual ofrece a los operadores que han desplegado un sistema CDMA ONE de segunda generación, una migración transparente que respalda económicamente la actualización a las características y servicios 3G, dentro de las asignaciones del espectro actual.

A fin de facilitar la migración de CDMA ONE a las capacidades de CDMA2000, ofreciendo características avanzadas en el mercado de una manera flexible y oportuna, su implementación se ha dividido en dos fases evolutivas: CDMA 2000 Fase I y CDMA 2000 Fase II.

II.4.3.2 CDMA 2000 Fase I:

Las capacidades de la primera fase se han definido en una norma conocida como 1XRTT. La publicación de la 1XRTT se hizo en el primer trimestre de 1999. Esta norma introduce datos en paquetes a 144 Kbps en un entorno móvil y a mayor velocidad en un entorno fijo. Las características disponibles con 1XRTT representan un incremento doble: en la capacidad para voz debido a que soporta 35 canales por sector y en la velocidad de datos, la cual en redes comerciales alcanza un pico de

153,6 Kbps y servicios avanzados de datos en paquetes. Además extiende considerablemente la duración de la pila.

II.4.3.3 CDMA 2000 Fase II:

La evolución de CDMA ONE, hasta llegar a las capacidades completas de CDMA 2000, continuará en la segunda fase e incorporará las capacidades de 1XRTT, usará tres portadoras de 1,25 MHz en un sistema multiportadora para prestar servicios de banda ancha de 3G.

CDMA 3XRTT proporcionará velocidad de circuitos y datos en paquete de hasta 2 Mbps, incorporará capacidades avanzadas de multimedia e incluirá una estructura para los servicios de voz y codificadores de voz 3G, entre los que figuran los datos de paquetes de “*voice over*” y de circuitos.

II.4.3.4 CDMA 2000 1XEV

Basado en el estándar 1X, el sistema 1XEV mejora la velocidad de procesamiento de datos, obteniendo velocidades de 2 Mbps sin tener que utilizar más de 1,25 MHz del espectro. Los requisitos para los operadores recién establecidos con respecto a 1XEV establecen dos fases.

En la primera fase, CDMA2000 1XEV-DO usa una portadora separada de 1.25 MHz para datos y ofrece velocidades de datos hasta 2.4 Mbps. La fase 2, CDMA2000 1X EV-DV se centra en las funciones de datos y de voz en tiempo real, así como en la mejora del funcionamiento para mayor eficiencia en voz y en datos. La velocidad de los datos en CDMA 1X-EVDV llegará hasta 3,09 Mbps

En la figura 12 se puede apreciar las diferentes fases de las redes CDMA según su velocidad de datos y aplicaciones:

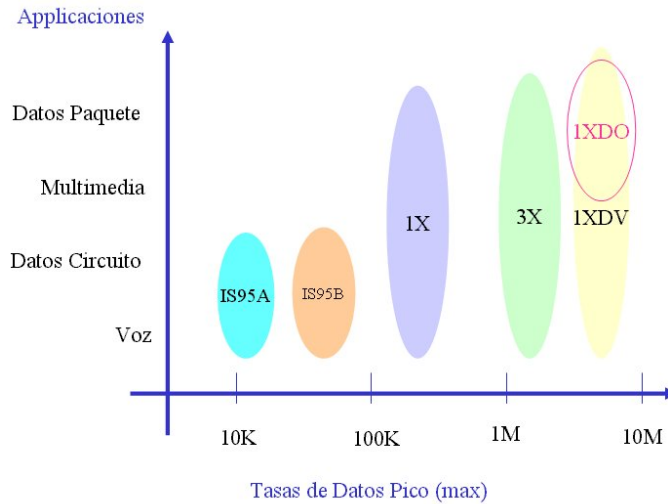


Figura 12 Tasa Máxima de Transmisión de Datos en CDMA

(Fuente: <http://www.monografias.com/trabajos15/telefonía-celular/telefonía-celular.shtml>)

II.4.3.5 UMTS

UMTS corresponde a las siglas *Universal Mobile Telecommunications System*, y es la propuesta más relevante como sistema de radio móvil europea de tercera generación. La estandarización de UMTS está siendo llevada a cabo por el ETSI.

La tecnología empleada por UMTS es WCDMA (*Wideband Code Division Multiple Access*), en operación FDD (*Frequency Division Duplex*) espectro pareado, aunque también se ha tenido en cuenta la TD/CDMA en operación TDD (*Time Division Duplex*) espectro no-pareado para uso en recintos cerrados, lo que constituye

la solución llamada UTRA. WCDMA es una técnica de acceso múltiple por división de código que emplea canales de radio con una ancho de banda de 5 MHz.

II.4.3.6 Enhanced Data Rates for GSM Evolution (EDGE)

De acuerdo a Sendín Escalona (SENDÍN, 2004), EDGE se configura como la tecnología que supera a GPRS, y se le ubica como estándar de 2.5 G, sin embargo la UIT lo sitúa como estándar 3 G.

EDGE se puede desplegar en las bandas de frecuencia 800, 900, 1800 y 1900 MHz actuales y puede servir como la vía a WCDMA. Es una solución 3G diseñada específicamente para mejorar el acceso a radio e integrarse al espectro existente, permitiendo así a que los operadores ofrezcan nuevos servicios.

EDGE ofrece servicios de Internet Móvil con una velocidad en la transmisión de datos tres veces superior a la de GPRS. El equipo de EDGE también opera automáticamente en modo GSM. EDGE es importante para los operadores con redes de GSM o GPRS que se desarrollarán en UMTS.

En el figura 13 se puede visualizar las diferentes fases de evolución de GSM según su velocidad de datos y aplicaciones:

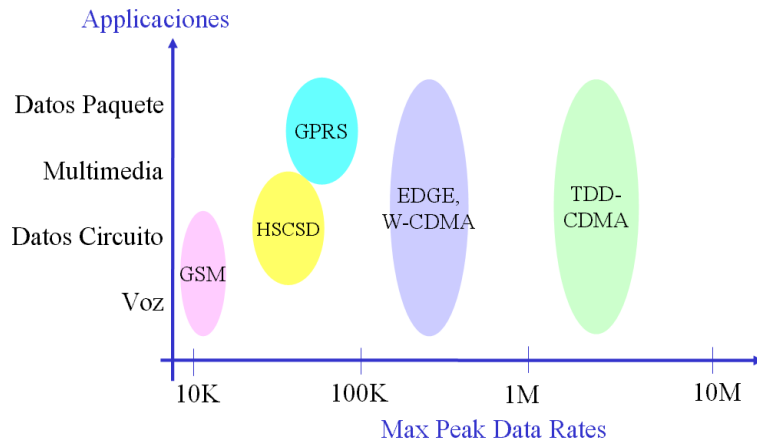


Figura 13 Tasa Máxima de Transmisión de Datos en GSM

(Fuente: <http://www.monografias.com/trabajos15/telefonía-celular/telefonía-celular.shtml>)

II.4.4 802.11

El protocolo IEEE 802.11 o WI-FI es un estándar de protocolo de comunicaciones de la IEEE que define el uso de los dos niveles más bajos de la arquitectura OSI especificando sus normas de funcionamiento en una WLAN. En general, los protocolos de la rama 802.x definen la tecnología de redes de área local. El alcance típico de 802.11 varía entre 20 y 200 metros dependiendo de si hay o no obstáculos, paredes, etc. de por medio. La tabla 1 contiene las variantes relacionadas al estándar 802.11.

802.11	Estándar WLAN original. Soporta de 1 a 2 Mbps.
802.11a	Estándar WLAN de alta velocidad en la banda de los 5 GHz. Soporta hasta 54 Mbps.
802.11b	Estándar WLAN para la banda de 2.4 GHz. Soporta 11 Mbps.
802.11e	Está dirigido a los requerimientos de calidad de servicio para todas las interfaces IEEE WLAN de radio.
802.11f	Define la comunicación entre puntos de acceso para facilitar redes WLAN de diferentes proveedores.
802.11g	Establece una técnica de modulación adicional para la banda de los 2.4 GHz. Dirigido a proporcionar velocidades de hasta 54 Mbps.

802.11h	Define la administración del espectro de la banda de los 5 GHz para su uso en Europa y en Asia Pacífico.
---------	--

Tabla 1 Estándares 802.11

Fuente: <http://www.enterate.unam.mx/Articulos/2004/agosto/redes.htm>

III Metodología

El diseño del Sistema Monitor Remoto Interactivo de vehículos (SMRI) conllevó las siguientes etapas: investigación, documentación, determinación de prestaciones y limitaciones de las tecnologías estudiadas, cálculo de requerimientos del sistema, diseño de hardware y software, procesos de pruebas y corrección de los errores observados. A continuación se enumeran cada uno de estos pasos:

- Investigación y estudio de los protocolos OBD-II: se determinaron los principales estándares y protocolos involucrados en el diagnóstico de vehículos; se estudiaron a detalle cada uno de ellos y se seleccionó el protocolo sobre el cual se iba a trabajar. Se determinaron además los parámetros disponibles para la monitorización y, tomando en cuenta las características del protocolo seleccionado, se fijaron los requerimientos del sistema inalámbrico.
- Investigación y estudio de tecnologías inalámbricas: se estudiaron las principales tecnologías de transmisión disponibles actualmente y se seleccionó la que mejor se adaptó a los requerimientos del sistema, considerando, además, aspectos como vigencia tecnológica y facilidad de acceso a la misma (disponibilidad de equipos y servicios).
- Definición del hardware de la Unidad Remota: se diseñó el hardware de la Unidad Remota necesario para cumplir con todas las tareas que la misma debía desempeñar. Esta definición incluye el hardware de comunicación inalámbrica, el procesador principal de la unidad, la interfaz con la computadora del vehículo y los demás elementos electrónicos requeridos. Finalmente, se procedió a la elaboración física de la unidad.
- Definición del método de interacción con la Unidad Remota: se definieron los formatos de comandos y respuestas para interactuar con la misma.

- Diseño del software del sistema: se diseñó el programa de la Unidad Remota mediante el lenguaje de programación seleccionado. También se diseñó el software de los programas servidor y cliente que permitiera la interacción remota con la unidad, mediante el lenguaje de programación más apto para la aplicación.
- Pruebas del Sistema: se realizaron pruebas de la Unidad Remota aislada, con módem y vehículo simulados y reales. Se realizaron pruebas de las aplicaciones cliente y servidor con múltiples usuarios, incluyendo vehículos simulados. Finalmente se realizaron pruebas del sistema completo, correcciones y optimizaciones.

IV Desarrollo

Este capítulo contiene las secciones referentes a estudios realizados, selección de protocolos y tecnologías de transmisión inalámbrica, definiciones de hardware y software, pruebas realizadas y descripción de los productos desarrollados.

IV.1 Selección del protocolo OBD

En base al estudio realizado sobre los protocolos OBD-II existentes, se decidió trabajar con ISO-9141 debido a que los vehículos que se disponían para las pruebas eran europeos y asiáticos y, por lo tanto, compatibles con este estándar. Además, el soporte especializado en mecánica automotriz con el que se contó durante la elaboración del SMRI trabajaba con un fabricante de vehículos japonés.

Otro aspecto que influyó en la selección del estándar es que ISO 9141 es el protocolo usado por la mayoría de los vehículos europeos y asiáticos y, a diferencia de las dos versiones de SAE J1850, no se identifica con ningún fabricante en específico, razón por la cual su utilización es más amplia.

IV.2 Velocidad de los datos en ISO 9141

De acuerdo a las especificaciones del estándar, la máxima velocidad de la interfaz ISO 9141 es igual a 10,4 Kbps, lo que se traduce en un tiempo de bit de aproximadamente 96,15 μ s ; en esta interfaz, cada byte está compuesto de 8 bits de datos, un bit de comienzo y un bit de parada. En SAE J1979 se define que los mensajes de diagnóstico tienen un encabezado de 3 bytes; en el caso de un mensaje

de solicitud, el campo de datos tiene un tamaño fijo de 2 bytes (correspondiente al Tipo de Servicio y PID); para los mensajes de respuesta, el campo de datos puede tener de 3 a 7 Bytes, pero la mayoría de las respuestas son de 3 y 4 bytes (incluyendo Tipo de Servicio y PID). Todos los mensajes tienen un byte de *Checksum* al final. Por lo tanto, el tiempo transcurrido entre un mensaje de solicitud y un mensaje de respuesta viene dado por:

$$tiempo_msj_solicitud = (3 + 2 + 1)bytes \times (1 + 8 + 1) \frac{bits}{byte} \times 96.15 \frac{\mu s}{bit} = 5.769ms$$

$$tiempo_msj_respuesta = (3 + 4 + 1)bytes \times (1 + 8 + 1) \frac{bits}{byte} \times 96.15 \frac{\mu s}{bit} = 7.692ms$$

Para calcular el tiempo que toma adquirir la información de un parámetro, se debe considerar la suma de: tiempo del mensaje de solicitud, tiempo que tarda la ECU en responder, tiempo del mensaje de respuesta y tiempo que debe transcurrir para realizar la siguiente solicitud (asumiendo que se desean realizar múltiples solicitudes). Según los tiempos definidos en SAE J1979 y en base a los cálculos del tiempo que demora cada tipo de mensaje (mostrado anteriormente), se puede decir que el tiempo de adquisición de un parámetro está comprendido entre:

$$(5,769 + 0 + 7,692 + 55) ms < Tiempo_un_parámetro < (5,769 + 50 + 7,692 + 55) ms$$

$$68,5 ms < Tiempo_un_parámetro < 118,5 ms$$

En conclusión, se puede asumir que el tiempo total de adquisición de un parámetro es aproximadamente 100 ms, con lo cual es posible muestrear alrededor de 10 parámetros por segundo (o 10 veces por segundo el mismo parámetro).

IV.3 Requerimientos de velocidad para el sistema inalámbrico.

Los datos obtenidos del vehículo deben ser transmitidos en paquetes IP hacia el cliente. Asumiendo que la trama de datos definida para la aplicación contiene 20 bytes por cada medición (ya que se debe incluir un encabezado, Tipo de Servicio, PID, los datos y demás campos que se requiera) y tomando 44 Bytes en encabezado TCP/IP (24 y 20 Bytes respectivamente) la velocidad en sentido ascendente, de la Unidad Remota hacia la Red, requerida para el sistema de transmisión es:

$$v = \left(\frac{(44\text{bytes} + 20\text{bytes}) \cdot 8 \frac{\text{bit}}{\text{byte}}}{\text{medición}} \right) \cdot 10 \frac{\text{mediciones}}{\text{seg}} = 5.12\text{Kbps}$$

El cálculo realizado anteriormente se corresponde a la velocidad efectiva de los datos a nivel de IP, es decir, no toma en cuenta los encabezados de las capas inferiores del modelo OSI.

IV.4 Elección de la tecnología de transmisión inalámbrica

Los aspectos considerados para la elección de la tecnología de transmisión inalámbrica que mejor se adecuara al SMRI fueron: modo de transmisión y calidad de servicio de los datos, vigencia tecnológica, disponibilidad de dicha tecnología en Venezuela, cobertura y velocidad de transmisión de los datos adecuadas a los requerimientos del SMRI.

En lo que se refiere al modo de transmisión de información se consideró el uso de mensajes cortos de texto, SMS por sus siglas en inglés, y conexión de datos. Los SMS permiten enviar cadenas de caracteres desde teléfonos móviles. De acuerdo a Sendín Escalona (SENDÍN, 2004) este servicio trabaja sobre los canales de control o

de señalización, en los momentos que no existe tráfico sobre éstos, por lo tanto no lo hace una aplicación ideal para la transmisión de datos en tiempo real.

En cuanto a calidad de servicio, este aspecto no es garantizado por los operadores para el servicio SMS, lo cual afecta notablemente la entrega inmediata debido a los retardos que se puedan presentar. Por las razones expuestas anteriormente y considerando que se requiere monitorizar la información en tiempo real, se decidió emplear una conexión de datos.

De acuerdo al estudio realizado sobre la evolución de los sistemas móviles se observa que la evolución de GSM para tercera generación será WCDMA, pasando por GPRS Y EDGE, lo cual significa que la tecnología de acceso empleada por GSM (TDMA) dejará de ser usada para dar paso a CDMA. Por lo tanto, la tecnología de transmisión del SMRI debe ser CDMA.

Al estudiar la disponibilidad de las tecnologías CDMA en Venezuela, se descartó el empleo de WCDMA, cuya implantación a nivel mundial se espera a mediados del presente año. Digitel, la operadora en Venezuela que emplea la tecnología GSM, actualmente sólo tiene al servicio de los consumidores GPRS.

Como se indicó al inicio de esta sección, otro aspecto importante en la elección del sistema de transmisión inalámbrica fue la cobertura, debido a que se desea brindar el servicio de monitorización a nivel nacional. Este aspecto impidió la escogencia de la tecnología CDMA2000 1X-EVDO, la cual se encuentra disponible sólo en algunas ciudades del país. La tecnología 802.11 también fue descartada por sus limitaciones de alcance y cobertura.

Además, estudiando la cobertura brindada por cada una de las tecnologías de transmisión inalámbrica, se hizo aún más claro trabajar con CDMA debido a que Digitel, opera sólo en la Región Central del país, y aunque puede interconectarse con

las otras operadoras de la misma tecnología, Infonet y Digicel que operan en diferentes regiones del país, no brinda cobertura en todo el territorio nacional, como se puede observar en la figura 14. Actualmente, estas tres operadoras se encuentran interconectadas, sin embargo este hecho no afecta la decisión tomada para este estudio.

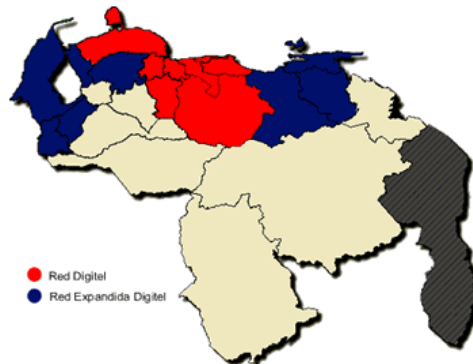


Figura 14 Gráfico de Cobertura Digicel
(Fuente: <http://www.digicel.com.ve>)

Por otro lado Movilnet, que opera con la tecnología CDMA 2000 1X, presenta un servicio de cobertura disponible a nivel nacional, como se muestra en la figura 15.



Figura 15 Gráfico de Cobertura Movilnet
(Fuente: <http://www.movilnet.com.ve>)

En cuanto a la velocidad de transmisión de datos, las redes CDMA 2000 1X soportan picos de transmisión de hasta 153,6 Kbps en el canal ascendente, llegando hasta un máximo 307 Kbps en el canal descendente. En teoría GPRS, lo más avanzado tecnológicamente de GSM que se dispone en el país, permite velocidades de hasta 144 Kbps en sentido descendente, aunque las velocidades de transmisión efectivas se encuentran entre 18 y 53 Kbps en sentido descendente y entre 9 y 13 Kbps en sentido ascendente.

Ninguna de las tecnologías estaba en posición de ventaja con respecto al resto en el aspecto de velocidad de transmisión de los datos, debido a que todas se adaptan a los requerimientos de velocidad de SMRI, ya que, como se mencionó en secciones previas, SMRI requiere de una velocidad mínima de transmisión de datos de 5.12 Kbps en sentido ascendente.

Debido a los argumentos expuestos anteriormente la tecnología de transmisión inalámbrica escogida para el SMRI fue CDMA2000 1X, por su vigencia tecnológica, disponibilidad en el país mediante las operadoras Movilnet y Movistar, brindar mayor cobertura a nivel nacional y ajustarse a los requerimientos de velocidad. Otro factor determinante para la elección del sistema CDMA fue la facilitación de equipos y líneas de prueba por parte de la empresa Movilnet.

Al escoger CDMA como sistema de transmisión inalámbrica se debe tener en cuenta que esta tecnología maneja la asignación de direcciones IP de manera dinámica y, a diferencia de GSM, los equipos móviles no se encuentran “siempre conectados”.

IV.5 Definición del hardware del módulo remoto

IV.5.1 Funciones Generales de la Unidad Remota

Para poder definir el hardware de la unidad remota se procedió a determinar, en un primer acercamiento, las funciones que éste debía realizar y por consiguiente los requerimientos que se le impondrían. Las principales funciones de la unidad remota se enumeran a continuación:

- Lectura de parámetros y códigos de falla del vehículo.
- Adquisición de coordenadas geográficas en caso de implantarse un módulo GPS.
- Transmisión de todos los valores leídos a un servidor remoto.

Lectura de parámetros y códigos de falla del vehículo.

La lectura de los parámetros y códigos de falla de los vehículos se decidió realizar mediante una interfaz compatible con OBDII, de diseño propio, incluida en el circuito de la Unidad Remota. En cambio, las funciones de adquisición de coordenadas geográficas y transmisión inalámbrica de los valores leídos debían realizarse mediante dispositivos comerciales diseñados para estos fines, por lo cual, los requerimientos de la Unidad Remota dependían directamente de los equipos seleccionados.

Adquisición de coordenadas geográficas en caso de implantarse un módulo GPS

Para la adquisición de las coordenadas geográficas, se decidió que la Unidad Remota debía brindar una interfaz compatible con el módulo GPS *EM-406*, de la

empresa *GlobalSat*. Los otros dispositivos GPS considerados fueron: el *WD-GZX4120* y el *EZGPM01* de *EAZIX*. Todos los dispositivos brindan características muy similares, pero se seleccionó el EM-406 debido a que posee mayor sensibilidad y puede ser alimentado con 5V. Este dispositivo brinda una solución rápida para implementar funciones de localización a cualquier aplicación, ya que posee todos los componentes requeridos para su operación integrados en él (inclusive la antena) y una interfaz serial para el envío de comandos y recepción de respuestas. Sus principales características son:

- Voltaje de alimentación comprendido entre 4.5V y 6.5V DC
- Interfaz serial con niveles TTL a una velocidad de 4.8 Kbps
- Sensibilidad de -159 dBm en la recepción de las señales provenientes de los satélites.

Más detalles del EM-406 pueden ser encontrados en la guía “*EM-406. GPS Receiver Engine Board. Product Guide*” disponible en la dirección Web www.usglobalsat.com. En la figura 16 se muestra una imagen del dispositivo.



Figura 16 Módulo GPS EM-406 de GlobalSat.
Fuente: www.globalsat.com

Transmisión de todos los valores leídos a un servidor remoto

El medio de transmisión escogido, como ya se explicó anteriormente, fue la red celular con tecnología CDMA. En vista a esto, se procedió a investigar los módems que las compañías operadoras en Venezuela tuviesen homologados y destinados para aplicaciones de telemetría. Los resultados de esta investigación indicaron que el único módem que utilizan estas compañías actualmente es el modelo EMII-800 de la marca *Anydata*.

El EMII-800 contiene internamente al módem DTSS-800 de la misma marca y ofrece una interfaz serial RS232 para la comunicación con el mismo; es decir, el EMII-800 es un producto que encapsula al módem interno de manera de hacerlo más manejable y fácil de conectar como periférico a un computador estándar, como se puede ver en la figura 16.



Figura 17 Módulo EMII-800 para conexión a la red CDMA.
Fuente: www.anydata.net

Las características más relevantes del EMII-800 que influyen en la definición del hardware de la unidad remota se mencionan a continuación. Se recomienda revisar el manual de utilización del EMII-800 (*“EMII-800 User Manual. Application Information”*) si se desea conocer mayores detalles.

- Voltaje de alimentación comprendido en el rango de 6V a 14V DC.
- Para un voltaje de operación de 12V, la máxima corriente demandada es de 280 mA. La corriente consumida cuando el módem se encuentra en estado inactivo es de 55mA.
- Interfaz serial RS-232 a una tasa de 115.2kbps.

IV.5.2 Requerimientos de la Unidad Remota

En base a las características del módem, módulo GPS y a las funciones generales que debía realizar la unidad remota, expuestas en el apartado anterior, se procedió a especificar los requerimientos que la unidad debe cumplir, los cuales se listan a continuación:

- La unidad remota debe operar dentro del vehículo durante un tiempo indefinido, por lo cual su alimentación eléctrica debe ser suministrada por el mismo.
- El consumo de energía eléctrica debe estar dentro de los rangos admisibles por la mayoría de los vehículos, de manera que no afecte el correcto funcionamiento del mismo.
- La unidad remota debe poseer una interfaz para comunicarse con la computadora del vehículo que, además de cumplir con los requerimientos de la capa física del ECU, asegure su integridad en caso de que ocurriese alguna falla eléctrica en la misma (falta de energía eléctrica inesperada, cortocircuito, etc.).
- La unidad remota debe manejar los protocolos OBDII a nivel de enlace y aplicación utilizados en los vehículos europeos y asiáticos principalmente (ISO 9141 y SAEJ1979). Para ello es necesario que la unidad remota pueda manejar una comunicación serial en codificación NRZ a 5bps para la

inicialización y adaptarse luego a una tasa de bits dada por un byte de sincronismo enviado por la ECU.

- La unidad remota debe poseer una interfaz serial RS-232 capaz de comunicarse a 115.2 Kbps con el módem CDMA; también debe ser capaz de manejar comandos AT para establecer la conexión remota mediante la pila de protocolos TCP/IP del módem.
- Se debe disponer de una interfaz serial TTL que permita la comunicación con el módulo GPS. Esta interfaz debe manejar una velocidad de transmisión de 4,8 Kbps.

IV.5.3 Definición de la Unidad de Procesamiento Principal

En base a los requerimientos de la Unidad Remota expuestos en el apartado anterior, se procedió a seleccionar un microprocesador capaz de gestionar la unidad remota. Existen principalmente dos vertientes en este ámbito: utilizar un microcontrolador simple y elaborar todo el hardware requerido para su operación o utilizar una tarjeta PC integrada.

Una tarjeta PC integrada está conformada, de forma resumida, por un procesador de alto desempeño (16 o 32 bits, 50~200 MHz), memoria externa de programa y RAM, puertos de comunicaciones RS-232, Ethernet, USB, etc., de forma tal que pueden ser manejadas con un sistema operativo simple (DOS o versiones simplificadas de Windows y Linux). De esta manera, el software de la unidad remota puede ser programado en lenguajes de alto nivel. La principal desventaja de una solución basada en este tipo de tarjetas es el alto costo de las mismas (aproximadamente 150 \$ US las versiones más sencillas) y el elevado consumo de energía en comparación con una solución de microcontrolador simple. En la figura 18 se puede observar un ejemplo de este tipo de soluciones.

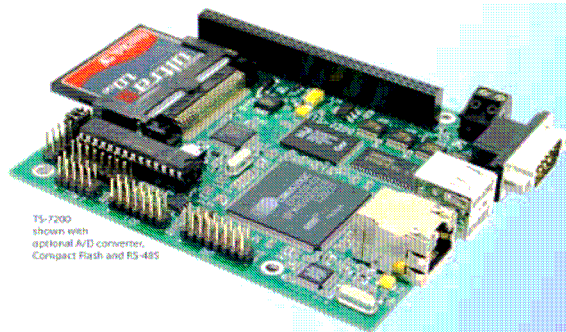


Figura 18 Solución de PC integrada modelo TS-7000 de Technologic Systems.
Fuente: www.embeddedarm.com

Por otro lado, se encuentra la posibilidad de utilizar un microcontrolador simple. Existe una amplia gama de microcontroladores disponibles en el mercado, siendo los más usados aquellos con procesadores de 8 bits. Los microcontroladores poseen internamente un procesador (CPU) y una gran cantidad de periféricos como temporizadores, convertidores A/D, UART, etc., para así facilitar su aplicación en actividades de control. Los microcontroladores de 8 bits manejan una baja velocidad de procesamiento, en comparación con los procesadores de las tarjetas PC integradas, y su grupo de instrucciones es mucho más reducido, debido a que están concebidos para realizar actividades de control y no complejos cálculos matemáticos. Las ventajas de utilizar microcontroladores simples son sus bajos costos (inferior a 10\$ US) y su fácil adquisición en el mercado venezolano. En la figura 19 se puede observar un microcontrolador simple.

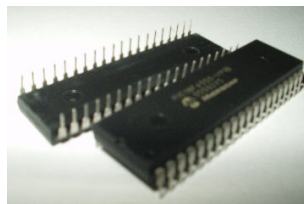


Figura 19 Microcontrolador PIC 18F4450 de 40 pines de Microchip.

Se decidió trabajar con un microcontrolador simple para la unidad remota en lugar de utilizar una tarjeta PC integrada debido a razones de costo, consumo de energía y capacidad de procesamiento requerida para la aplicación; esto último se refiere a que las actividades que debe realizar la unidad remota no requieren de complejos cálculos matemáticos ni excesiva capacidad de almacenamiento, ya que la monitorización se realiza en tiempo real. En la figura 20 se muestra de forma esquemática las principales actividades que debe realizar el microcontrolador y los requerimientos que se le imponen para cumplirlas.

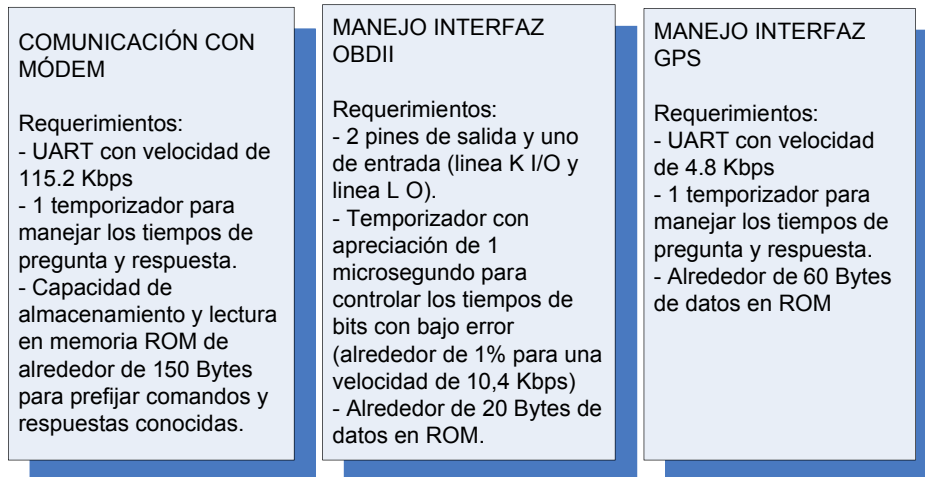


Figura 20 Funciones y requerimientos del microcontrolador de la unidad remota.

Como se puede observar en la figura anterior, la comunicación con el módem CDMA y con el módulo GPS puede ser realizada con un UART simple. El UART simple de un microcontrolador es el hardware encargado del envío y recepción de bytes de forma serial. En este periférico, debe ser previamente programada, entre otras cosas, la velocidad de operación; en caso de que se reciba un byte a una velocidad distinta a la programada, se detecta un error, pero no es posible que la interfaz se sincronice automáticamente. La comunicación con el vehículo no tiene una velocidad fija (el primer Byte de inicialización se envía a 5 bps y luego debe adaptarse a la velocidad del byte de sincronismo enviado por el ECU), por lo cual se

debe trabajar con un UART simulado en software, capaz de adaptarse a distintas velocidades utilizando un temporizador de buena apreciación.

Finalmente, el microcontrolador seleccionado para gestionar la unidad remota fue el PIC 16F877A de la empresa Microchip. La selección del microcontrolador se realizó considerando modelos semejantes de las 3 principales empresas (y arquitecturas) de microcontroladores: Microchip, Intel (producido por Atmel) y Motorola (producido por Freescale). A continuación, en la tabla 2 se muestra un cuadro comparativo de los modelos considerados:

Característica	PIC 16F877A (Microchip)	AT89C51AC2 (Atmel)	M68HC08-GP32 (Freescale)
Número de instrucciones por segundo, en MIPS (<i>Mega Instruccions Per Second</i>)	5	3.33	8
Memoria RAM (en Bytes)	368 B	256 B	512 B
Memoria ROM para datos (en Bytes)	256	2048	(la cantidad que quede libre de memoria de programa)
Memoria de Programa (en Bytes)	14 KB	32 K	32K
Puertos seriales (UART)	1	1	1
Temporizadores	3	3	1
Convertidores A/D (entradas)	8	1	8

Tabla 2 Cuadro Comparativo entre microcontroladores

Se puede observar en la tabla anterior que los microcontroladores poseen desempeños similares, pero el Motorola sobrepasa a los demás en todos los aspectos. Sin embargo, se decidió trabajar con el PIC 16F877A debido a las siguientes razones:

- Precio inferior a los demás y de fácil adquisición en el mercado Venezolano.
- Facilidad de adquisición de programadores, los cuales se utilizan para grabar el código en el microcontrolador.
- Arquitectura similar y compatible pin a pin con modelos más avanzados de Microchip, como el PIC 18F4550, lo que facilita la

migración futura, en caso de que se requieran mayores desempeños para nuevas aplicaciones.

Los microcontroladores de nueva generación de Microchip, como el PIC 18F4550, son de difícil adquisición en el mercado venezolano, al igual que sus programadores; debido a esto, no fueron considerados en la selección. Sin embargo, la arquitectura del PIC 16F877A y el PIC 18F4550 son muy similares, lo que facilitará el transporte del software diseñado para el PIC 16F877A a los de nueva generación en caso de que sea necesario.

Es importante observar que el PIC 16F877A posee un solo UART; por lo tanto, la comunicación con el módulo GPS deberá realizarse con un UART simulado por software (se prefiere usar el UART simulado en esta interfaz porque es la más lenta).

IV.5.4 Definición detallada del Hardware de la Unidad Remota

En base a todos los requerimientos y características del microcontrolador, módem y módulo GPS se procedió a realizar un diagrama general de la unidad remota, el cual se puede observar en la figura 21. La fuente brinda 5V (para alimentar al microcontrolador y al módulo GPS) y 12V (para alimentar al módem CDMA). Los relés controlan el apagado y encendido de los módulos CDMA y GPS en caso de que los mismos dejen de responder por alguna falla. El circuito de acoplamiento óptico se encarga de adaptar los niveles de voltaje entre el ECU y el microcontrolador, garantizando a su vez el aislamiento eléctrico entre ambas unidades. La comunicación entre el microcontrolador y el módulo GPS se puede realizar de manera directa, ya que ambos manejan niveles TTL; en cambio, la

comunicación con el módem CDMA debe realizarse a través de un circuito que adapte los niveles TTL a su valor equivalente en RS-232 y viceversa.

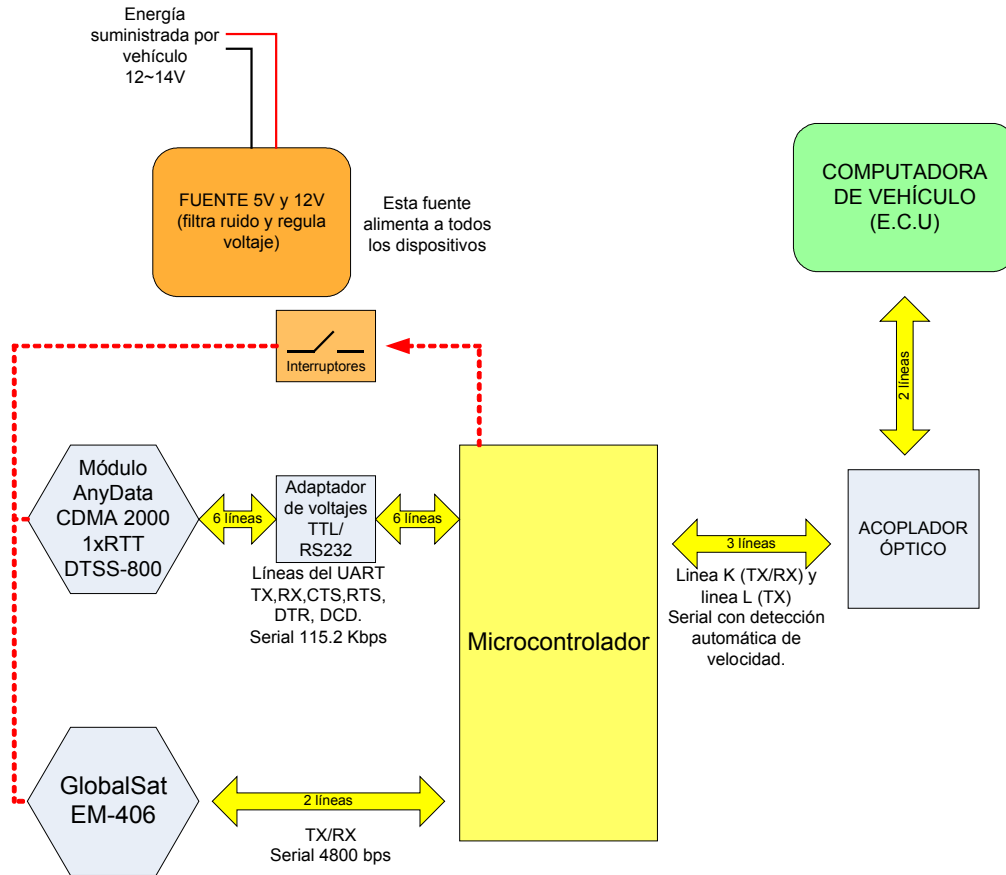


Figura 21 Diagrama general de la unidad remota

Posteriormente, se procedió definir el hardware específico para cada una de las etapas mostradas en la figura anterior. A continuación se explica detalladamente cada uno de estos circuitos.

Fuente de Alimentación

La fuente de alimentación de la unidad remota toma el voltaje suministrado por el vehículo y lo adapta a los niveles de 5V y 12V; esto es logrado mediante

reguladores LM7805 y LM7812 respectivamente. La figura 22 muestra el diagrama de circuito de la fuente. Los condensadores de 100 microfaradios ayudan a reducir el ruido que pueda tener la señal suministrada por el vehículo (debido principalmente a los disparos de las bujías) comportándose con todo el sistema como un filtro pasabajos. El diodo a la entrada ayuda igualmente a reducir el ruido y evita que los condensadores descarguen la energía almacenada contra el vehículo, en caso de que éste disminuya su voltaje, obligando a que toda la energía sea consumida dentro de la unidad remota.

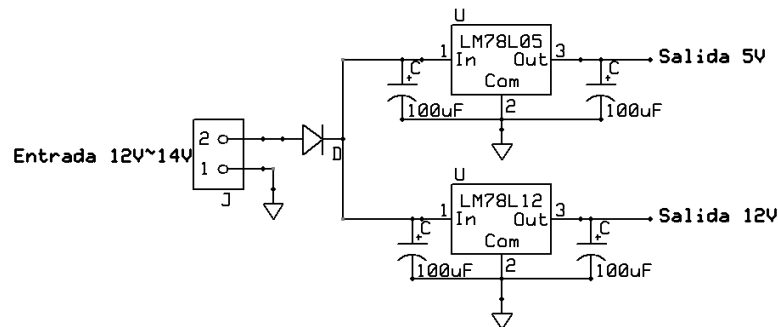


Figura 22 Circuito de la fuente de alimentación de la unidad remota.

Interruptores

Con el objetivo de disminuir las dimensiones de la unidad se decidió utilizar transistores que se colocan en corte y saturación para realizar el apagado y encendido del módem CDMA y el módulo GPS, conectados como se puede observar en la figura 23. El voltaje alto de cualquier salida del microcontrolador es, a lo sumo, el voltaje de alimentación menos 0.7V, por lo cual la corriente suministrada por el microcontrolador para colocar y mantener un dispositivo encendido es de 24mA, lo cual se encuentra por debajo del máximo de corriente que puede suministrar una salida (25mA). La corriente de base de 24mA es mayor que la corriente de saturación del colector entre la ganancia β , garantizando su estado de saturación ($I_b > I_{c_{sat}}/\beta_{dc}$).

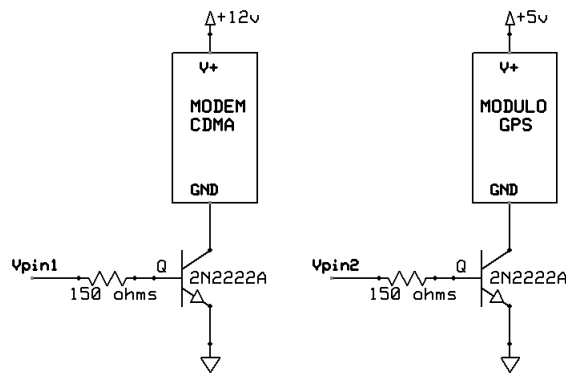


Figura 23 Configuración de interruptores del módem CDMA y módulo GPS.

Acoplador Óptico

La etapa de acoplamiento óptico permite la interoperabilidad del microcontrolador con la interfaz OBDII del vehículo (ISO 9141), garantizando a su vez un aislamiento eléctrico entre la unidad remota y la ECU; esto último asegura que la computadora del vehículo no sufrirá ningún daño en caso de que ocurra una falla eléctrica en la unidad remota. Como la línea K de OBDII trabaja en dos direcciones (*bidirectional*), se hizo necesario utilizar dos circuitos de acoplamiento óptico simultáneos, uno en cada sentido; para la línea L se requirió el acoplamiento óptico en un solo sentido. Obsérvese la figura 24.

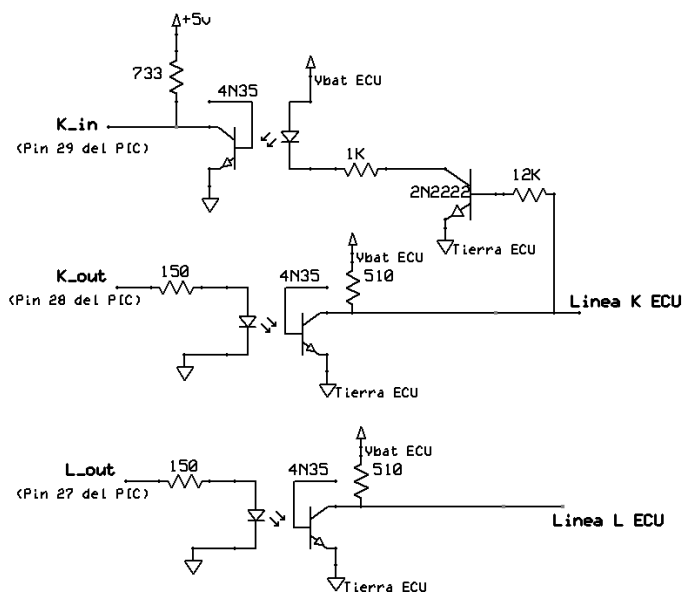


Figura 24 Etapa de acoplamiento óptico

Las resistencias de “pull-up” de la línea K y L son iguales a 510 ohmios, como lo especifica el estándar ISO 9141. Las resistencias de 150 ohmios que se encuentran a la salida de los pines 27 y 28 del microcontrolador limitan la corriente de encendido de los LED a 20 mA aproximadamente, valor que se encuentra dentro del rango de trabajo recomendado en las especificaciones del opto-acoplador. La velocidad de conmutación de estos dos opto-acopladores satisface la velocidad máxima de la interfaz ISO 9141 (10,4 Kbps) y el voltaje de saturación colector-emisor es lo suficientemente bajo para permitir la correcta detección de los bits.

La principal dificultad en la etapa de acoplamiento óptico se presentó en la recepción del PIC (o la “escucha” de la línea K). Esto se debe a que el voltaje de saturación colector-emisor del opto-acoplador aumenta a medida que aumenta la corriente de colector, es decir, a medida que la resistencia de colector disminuya. La velocidad de conmutación del opto-acoplador, en general, aumenta a medida que la resistencia de colector sea menor o a medida que el voltaje de alimentación sea mayor. En pocas palabras, a medida que aumenta la velocidad de conmutación del foto-transistor aumenta el voltaje colector emisor (no entra en saturación completa).

Se consiguió que una resistencia de 733 ohmios, lograba que el opto-acoplador tuviera un tiempo de conmutación inferior a 8,6 microsegundos y un voltaje colector emisor de 0,87 V, con lo cual se pudo manejar satisfactoriamente las velocidades de los datos provenientes de la interfaz ISO 9141 (velocidad máxima de 10,4 Kbps, es decir, tiempo de bit de 96 microsegundos.)

Adaptador de voltaje TTL/RS-232

La adaptación de los niveles TTL a niveles RS-232 y viceversa se realizó mediante el integrado MAX232, siguiendo el esquema de conexión de su especificación. Este integrado convierte 5V a -12V, 0V a 12V y viceversa; debido a que la interfaz RS-232 trabaja en lógica negada.

Esquema del circuito de la unidad remota.

En la figura 25 se presenta el esquema del circuito completo de la unidad remota realizado en el software libre *ExpressSCH*. En los pasados puntos se explicaron a detalle los módulos principales que conforman la unidad; en este diagrama se pueden distinguir estas etapas y otras nuevas como el circuito oscilador del PIC, el circuito RC de arranque en el *Reset* del PIC y algunos condensadores para disminuir ruido de alta frecuencia. Consúltese la hoja de especificaciones del microcontrolador para más detalles sobre estos circuitos.

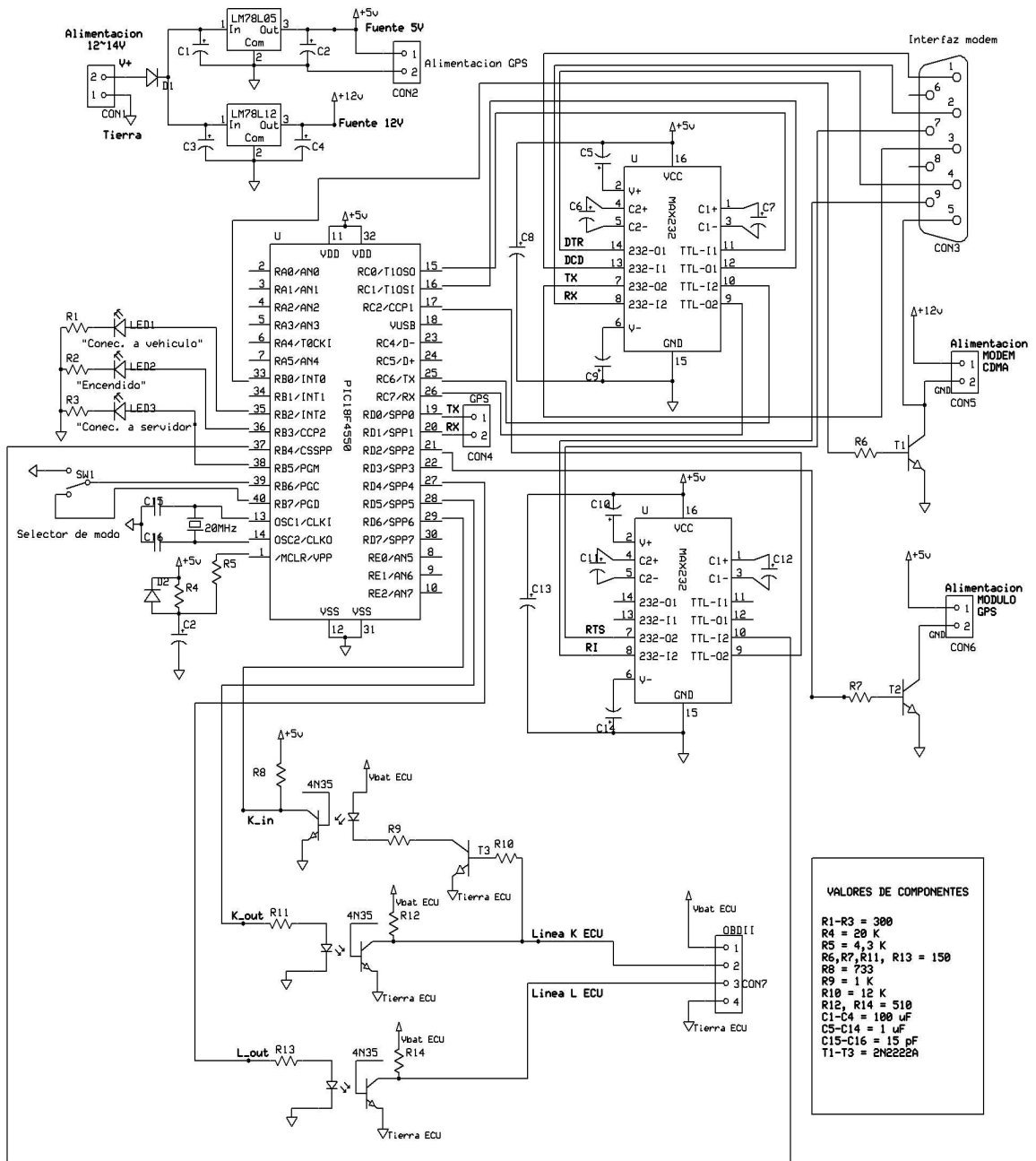


Figura 25 Esquema de circuito de la unidad remota.

Circuito Impreso de la Unidad Remota.

El arte de la tarjeta impresa para la unidad remota se realizó mediante el software libre *ExpressPCB*. El circuito impreso se realizó en dos caras, como se puede observar en la figura 26. En color rojo se muestran las líneas de la parte superior de la tarjeta y en verde las líneas de la parte inferior.

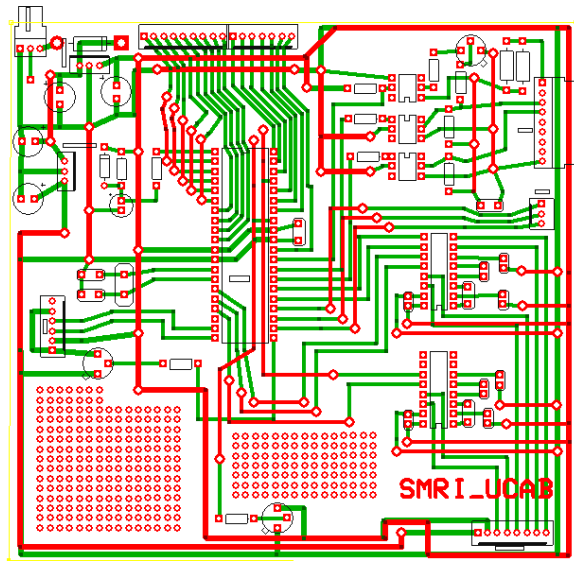


Figura 26 Circuito Impreso de la unidad remota

Elaboración física de la tarjeta

Luego de haber hecho el diagrama del circuito impreso para la tarjeta de la unidad remota se procedió a su elaboración física. Se decidió realizar la tarjeta de forma manual (no industrializada) mediante el método de papel *transfer* y posteriormente, el grabado se produce por reacción química con ácido (Percloruro de Hierro), comúnmente usado para este fin. En la figura 27 se puede observar una foto de la Unidad Remota ya finalizada (sin el módulo CDMA ni el GPS).



Figura 27 Unidad Remota finalizada

IV.6 Interacción con la unidad remota.

Con el objetivo de descentralizar la inteligencia de todo el sistema e independizar completamente la funcionalidad de la Unidad Remota y la de la aplicación cliente, se decidió que la primera debe poseer un nivel de complejidad que permita al software cliente abstraerse totalmente del funcionamiento del protocolo OBDII y enfocarse en la información. En otras palabras, la Unidad Remota debe manejar el protocolo OBDII (velocidad de la interfaz, estructura de los datos, tiempos, etc.), extraer los datos y enviarlos de una manera manejable y sencilla al cliente, el cual sólo debe encargarse de pedir los datos y presentarlos al usuario. En este sentido, se decidió que la Unidad Remota debe cumplir los siguientes aspectos:

- Recibir comandos enviados desde el cliente e interpretarlos.
- Leer el ECU, extraer la información y enviarla al cliente.
- Permitir la conexión local de un cliente (vía puerto serial), con el objetivo de facilitar el proceso de pruebas que se realizarían posteriormente y brindar al

usuario la posibilidad de monitorizar directamente el vehículo en caso que se desease.

- Los comandos recibidos y las respuestas enviadas por la Unidad Remota deben ser formados por caracteres ASCII, de esta manera se puede interactuar con la unidad mediante programas como *Hyperterminal*, al igual que se hace con la mayoría de los equipos de red configurables. El fin de un comando o de una respuesta se detecta con el carácter CR. El carácter LF puede ser enviado o no, por lo cual no debe ser utilizado para detectar el fin de un mensaje; al presionar el botón de *ENTER* del teclado en un programa editor de texto, los caracteres que se generan son CR LF, en ese orden. El carácter LF debe ser ignorado por la Unidad Remota.
- Los datos obtenidos del ECU deben ser representados en ASCII en su forma hexadecimal; es decir, si la medición de la velocidad arroja un resultado igual a 0x0f (correspondiente a 15 Km/h), el dato debe ser convertido a las letras “0” y “F”. Esto facilita el análisis de los datos en el programa cliente, ya que cada dos letras (2 bytes) representan un byte (no hay necesidad de usar caracteres delimitadores).
- Todas las letras deben ser mayúsculas.

Los mensajes que se envían y reciben en la Unidad Remota se encuentran representados en la figura 28. Éstos se encuentran clasificados en dos grupos: comandos y respuestas. Los comandos son las instrucciones que recibe la Unidad Remota por parte del cliente y las respuestas son enviadas en sentido contrario. En esta figura, los caracteres especiales como CR y LF se encuentran representados entre los símbolos “<” y “>”. Las casillas que comienzan con el símbolo “#” se refiere a que debe ir una letra del “1” al “9” o de la “A” a la “F”. Las casillas L1, L2.., L6 se refiere a que puede ir cualquier letra.

COMANDO																		
M	O	N	I	T	O	R								<CR>				
Descripción: Solicita a la Unidad Remota que muestre la tabla de pedidos que va a realizar al ECU.																		
M	O	N	I	T	O	R		#1	#1		#2	#2		<CR>				
Descripción: Solicita a la Unidad Remota que introduzca el PID indicado por #1#1 en la casilla #2#2 de la tabla de pedidos.																		
C	O	N	T	I	N	U	O		#	#				<CR>				
Descripción: Solicita a la Unidad los datos actuales (servicio 01, SAEJ1979) de los PID contenidos en la tabla. Se repite ## de veces.																		
F	A	L	L	A	S									<CR>				
Descripción: Solicita a la Unidad Remota los códigos de falla (DTC) almacenados en el ECU del vehículo.																		
C	O	N	G	E	L	A	D	O	S					<CR>				
Descripción: Solicita a la Unidad Remota que lea todos los datos congelados y almacenados en la ECU. (Servicio 02)																		
O	K													<CR>				
Descripción: Mensaje de confirmación general usado por el servidor y por el modem CDMA en múltiples procesos.																		
C	O	N	N	E	C	T								<CR>				
Descripción: Mensaje de notificación del modem que le indica a la Unidad Remota que se logró establecer la conexión TCP.																		
D	E	S	C	O	N	E	C	T	A	R				<CR>				
Descripción: Comando enviado por el servidor que solicita a la Unidad Remota que cierre la conexión TCP.																		
D	I	S	C	O	N	N	E	C	T					<CR>				
Descripción: Mensaje que envía el modem a la Unidad Remota cuando la conexión TCP no pudo ser establecida o falló.																		
RESPUESTA																		
S	M	R	I	_	U	C	A	B	>					<CR>				
Descripción: Mensaje que indica al usuario y al programa cliente que la Unidad Remota está activa y esperando comandos.																		
#1	#1		#2	#2		#3	#3	...	#10	#10				<CR>				
Descripción: Resuesta al comando <i>MONITOR<CR></i> . #1#1 indica el PID contenido en la casilla 1 de la tabla. (así sucesivamente)																		
O	K	I												<CR>				
Descripción: Mensaje de confirmación que se envía al cliente cuando se recibe correctamente un comando.																		
P	a	r	a	m	e	t	r	o	s		e	r	r	a	d	o	s	<CR>
Descripción: Hay un error en los parámetros del comando introducido																		
F	a	i	t	a	n		p	a	r	a	m	e	t	r	o	s	<CR>	
Descripción: Faltan parámetros en el comando introducido.																		
D	A	T	A		#1	#1	...	#n	#n					<CR>				
Descripción: Mensaje que contiene datos extraídos del vehículo. #1#1 indican el servicio, #2#2 el PID y los siguientes son datos.																		
E	C	U		n	o		i	n	i	c	i	a	l	i	z	a	<CR>	
Descripción: Indica que ocurrió un error en el proceso de inicialización del ECU.																		
E	C	U		C	h	k	s		e	r	r	o	r	<CR>				
Descripción: Hubo un error en el Checksum de un mensaje proveniente del ECU (el cual se descarta)																		
E	C	U		t	i	m	e	o	u	t				<CR>				
Descripción: El ECU no respondió a un mensaje de solicitud enviado por la Unidad Remota.																		
I	D		L1	L2	L3		L4	L5	L6					<CR>				
Descripción: Mensaje que envía la Unidad Remota al Servidor con el código de identificación (placa)																		

Figura 28 Comandos y respuestas de la Unidad Remota

En la siguiente sección se explicará el funcionamiento del software de la Unidad Remota, con lo que se logrará entender de forma más clara el significado de

cada uno de los comandos y respuestas anteriormente presentados. La definición de cómo se iba a interactuar con la Unidad Remota brindó la posibilidad de ir desarrollando el software de la misma y el software del Servidor-Cliente de manera simultánea. El formato exacto de los comandos y respuestas mostrados anteriormente fueron definidos durante la elaboración de ambos Software.

IV.7 Software de la Unidad Remota.

En la figura 29 se presenta el diagrama de flujo del programa principal de la Unidad Remota.

DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL DE LA UNIDAD REMOTA

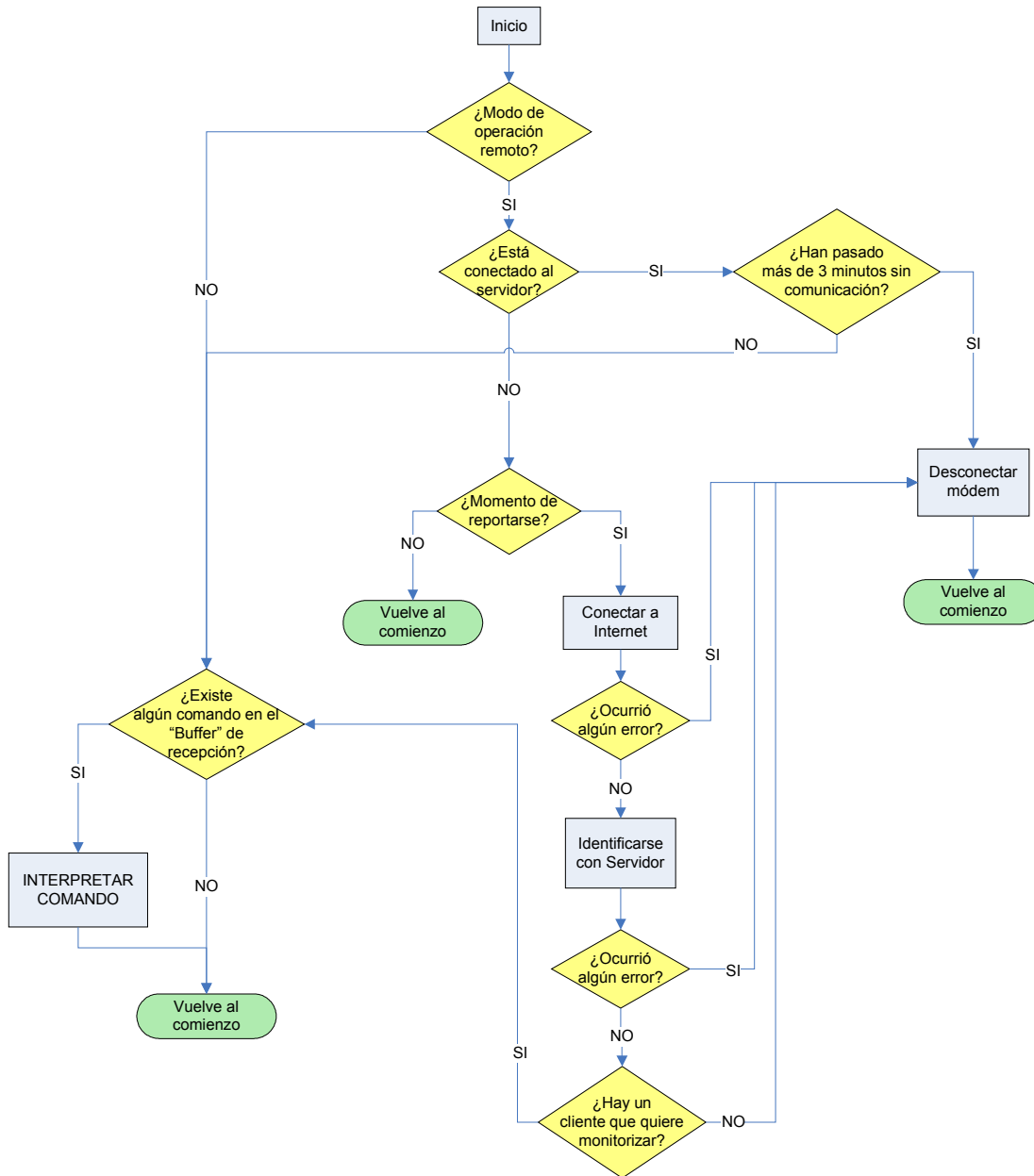


Figura 29 Diagrama de flujo del programa principal de la Unidad Remota

El programa principal de la Unidad Remota comienza verificando el modo de operación de la misma: “remoto” o “local”; esto se hace probando el estado de un interruptor externo que permite al usuario seleccionar si se desea monitorizar a través

de la red o de manera directa con un computador conectado a la Unidad. Cuando se trabaja en modo remoto, se activan temporizadores que indican el momento en que la Unidad debe reportarse con el servidor (en caso de que esté desconectada) o le indican si se debe cerrar la conexión (cuando la Unidad Remota está conectada pero pasa un cierto tiempo de inactividad y no existe intercambio de mensajes entre la Unidad y el servidor). Si se trabaja en modo local todos los temporizadores se desactivan y el programa se mantiene en espera de los comandos que se le envíen.

Si la Unidad opera en modo remoto, se consulta si la misma se encuentra conectada al servidor o no. En caso de que lo esté, se verifica que la comunicación permanezca activa; en caso contrario, se verifica el temporizador que le indica cuando es momento de reportarse con el servidor. Se prefijó que la Unidad Remota debe reportarse cada cinco minutos (5) con el servidor y que debe desconectarse del mismo si la comunicación cesa por más de tres (3) minutos.

La rutina *Conectar a Internet* se encarga de enviar y certificar la correcta aceptación de los comandos AT necesarios para que se pueda establecer la conexión TCP. Estos comandos básicamente especifican la identificación y clave del usuario, dirección IP del destino y número del puerto TCP. La secuencia de comandos AT y sus respuestas se pueden observar en la figura 30. La Unidad Remota es capaz de tolerar hasta 4 respuestas incorrectas a un comando AT antes de la respuesta esperada; en caso de que no se reciba respuesta a un comando o las respuestas recibidas sean incorrectas, el programa reintentará establecer la conexión desde el comienzo. Si el tercer intento de conexión falla, se devuelve un código de error y se llama a la rutina de desconexión del módem.

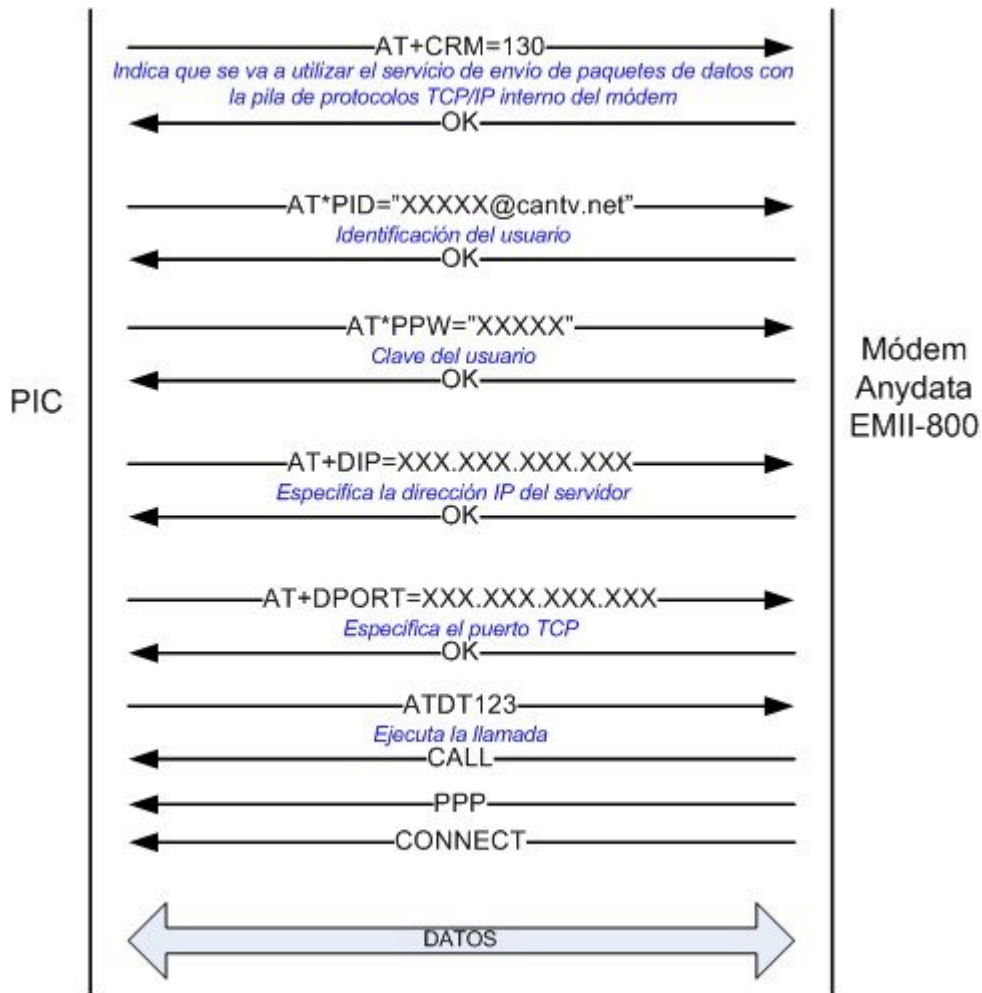


Figura 30 Comandos AT para establecer la conexión de datos
Fuente: Propia

La rutina *Desconectar módem*, en un primer paso, cierra la conexión (en caso de que esté abierta) del módem mediante una señal en el pin DTR destinado para este fin. En caso de que el módem no responda con una señal de aceptación por el pin DCD, la rutina interpreta que existe un error grave en el módem, por lo cual procede a cortar el suministro de energía durante 10 segundos y luego lo vuelve a establecer.

La rutina *Identificarse con servidor* se ejecuta cuando la conexión TCP ya ha sido establecida con el servidor. La unidad remota procede a enviar la identificación del vehículo (placa) y se queda esperando la respuesta del servidor. Si el servidor no

ha recibido ninguna solicitud de algún cliente que quiera comunicarse con ese vehículo responde “DESCONECTAR”; en ese caso la unidad remota procede a desconectar el módem y vuelve al inicio del ciclo principal. En caso de que el servidor responda “OK”, la Unidad Remota se coloca lógicamente en estado activo (conectado a servidor) y lista para recibir los comandos provenientes del servidor. En caso de que ocurra un error en la respuesta del servidor o en caso de que éste no responda, se procede a desconectar el módem y volver al estado inicial.

El procedimiento *Interpretar Comando* se encarga de identificar y validar el formato de los comandos recibidos y enviar los mensajes de confirmación o error hacia el cliente. Este procedimiento también se encarga de llamar a la rutina respectiva para cada comando. En la figura 31 se puede observar un diagrama de bloques simple que representa el funcionamiento de esta rutina.

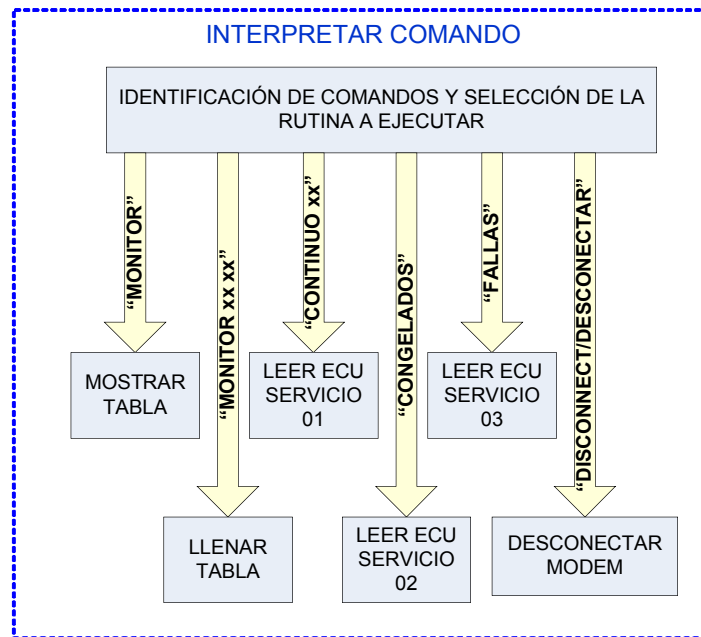


Figura 31 Diagrama de bloques de rutina *Interpretar Comando*

Para entender en qué consisten los procedimientos relacionados con el llenado y lectura de la tabla y los relacionados con la lectura del ECU es necesario explicar

primero el procedimiento general que se decidió utilizar para monitorizar los parámetros de funcionamiento del vehículo.

Como se mencionó en secciones anteriores, la interfaz ISO 9141 y el protocolo SAE J1979 permiten una monitorización de aproximadamente diez (10) muestras por segundo. Con el objetivo de aprovechar al cien por ciento la velocidad de muestreo que permite la interfaz, se descartó inmediatamente la opción de solicitar los parámetros uno por uno desde el cliente, ya que los tiempos de ida y vuelta de los paquetes entre el mismo y la Unidad Remota degradarían enormemente la velocidad de muestreo.

De acuerdo a lo descrito anteriormente, se decidió que la Unidad Remota debía poseer una tabla que contuviese todos los PID requeridos, es decir, una tabla de pedidos. Esta tabla debía ser llenada mediante comandos enviados desde el cliente, hasta que se enviara la instrucción de leer la ECU, donde la Unidad Remota procedería a solicitar todos los parámetros de la tabla el número de veces que se le indicara, para así realizar una monitorización continua y en tiempo real.

Gracias a consultas realizadas a ingenieros especializados en el área de mecánica de vehículos y diagnóstico computarizado, se decidió que el máximo número de parámetros que eran necesarios monitorizarse simultáneamente no excedían a diez (10), por lo cual se decidió que la tabla debía contener 10 casillas (una por parámetro), lo que originaría alrededor de una muestra por segundo por cada parámetro. Si por alguna razón se necesitase una resolución en tiempo mayor a la anteriormente mencionada, se pueden llenar varias casillas con el mismo PID, dando el máximo de 10 muestras por segundo si se monitoriza un solo parámetro.

El comando “MONITOR xx xx” es el encargado de llenar la tabla (véase “Interacción con la Unidad Remota”) y el comando “CONTINUO xx” da la orden de monitorizar todos los parámetros de la tabla y repetir la misma el número de veces

indicado; este comando permite que se pida la tabla completa un máximo de 255 veces, lo que corresponde a una monitorización continua durante más de cuatro (4) minutos. En caso de que se desee monitorizar por más tiempo del mencionado anteriormente se debe volver a enviar el comando “CONTINUO xx” sin tener que volver a llenar la tabla de pedidos.

En la figura 32 se puede observar un ejemplo del método de operación de la Unidad Remota para la lectura de los parámetros:

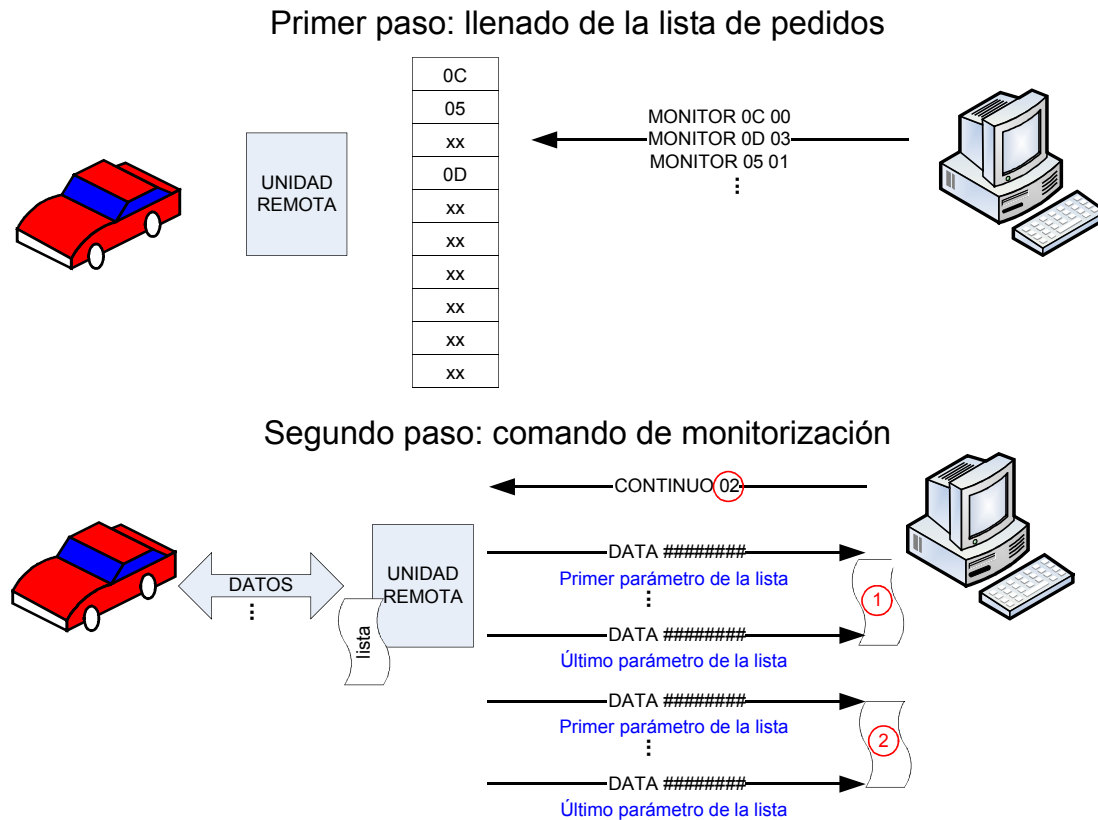


Figura 32 Método de monitorización de la Unidad Remota

Luego de haber explicado el método de monitorización de la Unidad Remota se puede entender con mayor facilidad las rutinas que son seleccionadas en el procedimiento *Interpretar Comando*. La rutina *Llenar Tabla* se encarga de colocar el PID en la casilla indicada. La rutina *Mostrar tabla* se encarga de enviar la lista de

todos los PID anotados en la tabla de pedidos. Las rutinas *Leer ECU* con Servicio 01 y 02 (invocadas con los comandos “CONTINUO ##” y “CONGELADOS” respectivamente) se encargan de solicitar al vehículo los PID anotados en la tabla, con la diferencia del Tipo de Servicio. La rutina con Servicio 03 se encarga por sí sola de preguntar si existen códigos de falla y cuántos (Servicio 01, PID 01); si existen códigos de falla, procede a leerlos (Servicio 03, sin PID) y los reenvía al cliente; si no existen códigos de falla envía “ECU timeout”.

Un aspecto importante de destacar del software de la Unidad Remota es su completa compatibilidad con ISO 9141 e ISO 9141-2, ya que su interfaz OBDII puede adaptarse a cualquier velocidad de acceso (dentro del rango indicado en los estándares) mediante una rutina de análisis del Byte de sincronismo enviado por el ECU.

IV.8 Pruebas de la Unidad Remota

IV.8.1 Pruebas del software mediante MPLAB SIM.

A medida que se iba desarrollando el software de la Unidad Remota se realizaron pruebas parciales del mismo, mediante el simulador MPLAB SIM que viene integrado al ambiente de desarrollo MPLAB IDE de Microchip. MPLAB SIM permite simular entradas asíncronas como pulsos, cambios de estados de las líneas y tramas de datos por la interfaz serial, con lo cual se pudo simular el comportamiento del módem CDMA y de la interfaz ISO 9141. MPLAB SIM fue una de las herramientas más útiles para la detección de errores en el software, a pesar de que la simulación de la interfaz ISO 9141 resultaba complicada de realizar, debido a que el mismo no permite la simulación de tramas de bits por entradas distintas a la serial, por lo cual fue necesario realizar la simulación paso a paso, llevando la contabilidad del tiempo (a través del número de instrucciones) y cambiando el estado de la línea

cada vez que fuese requerido, simulando así las tramas recibidas por la interfaz con el ECU.

IV.8.2 Pruebas del módem CDMA.

Las pruebas del módem CDMA eran de vital importancia para poder identificar comportamientos que no estuviesen documentados en los manuales del producto. Estas pruebas se realizaron conectando un computador directamente al módem y enviando los comandos AT requeridos para establecer la conexión TCP, utilizando el software *DockLight*. El programa servidor se simuló en otra computadora con acceso a Internet mediante el programa *Hyperteminal 5.1*, el cual permite el establecimiento de conexiones TCP. Las pruebas tomaron alrededor de 4 semanas debido a problemas presentados en el establecimiento de conexiones. Como resultado de las pruebas se logró identificar que las fallas presentadas eran ocasionadas por el propio módem, por lo cual se actualizó su software interno. Más detalles sobre el proceso de pruebas del módem se encuentran en el Apéndice 5.

IV.8.3 Pruebas de la Unidad Remota con módem simulado

Como se ha mencionado anteriormente, la Unidad Remota fue diseñada con la capacidad de operar en modo local, es decir, con un computador conectado a través del puerto serial. Realizando este esquema de conexión y colocando el selector de modo de operación en “remoto” se pudo simular el módem CDMA y evaluar el comportamiento de la Unidad cuando trabaja en este modo. En la figura 33 se puede observar el diagrama de conexión utilizado para esta labor. La resistencia de 50 ohmios es colocada para simular la carga del módem CDMA al ser alimentado y

proporcionar señal de tierra al computador, ya que de esta manera el transistor de apagado y encendido del módem (T1) entra en saturación.

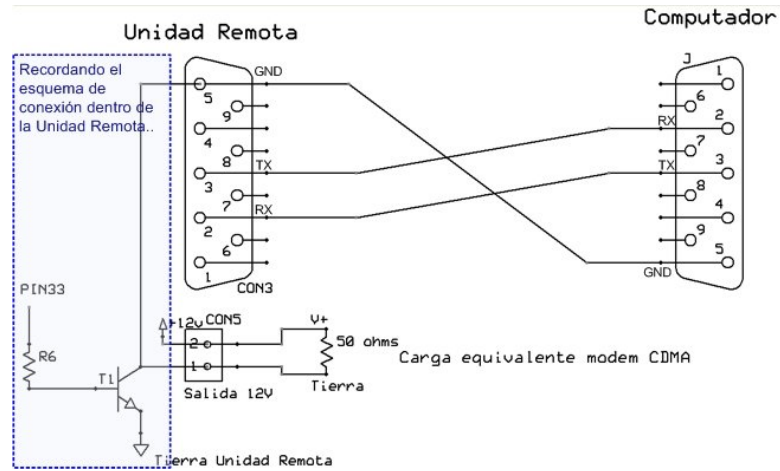


Figura 33 Esquema de conexión de la Unidad Remota en modo local

El software utilizado en el computador para establecer la comunicación por el puerto serial fue *Docklight*. Las pruebas permitieron que se realizaran comprobaciones de la correcta interpretación de los comandos y de los procesos necesarios para establecer la conexión. Se certificó igualmente la precisión de los tiempos de reporte, tiempos máximos en espera de respuestas y tiempos de inactividad. Luego se realizaron pruebas con el módem real y se obtuvieron los resultados esperados.

IV.8.4 Pruebas de la Unidad Remota con vehículo simulado.

Para poder certificar el correcto funcionamiento de todos los procesos relacionados con la lectura del ECU se decidió simular la interfaz ISO 9141 mediante un computador, empleando el puerto serial. Para ello fue necesario elaborar un hardware que permitiera adecuar los niveles de voltaje provenientes del computador a los propios de ISO 9141, siguiendo las características de la interfaz física especificada en el estándar. Las limitaciones de esta simulación fueron las siguientes:

- El byte de inicialización a 5 bps no puede ser interpretado correctamente por la interfaz serial.
- La velocidad de 10.4 Kbps (especificada en ISO 9141-2) no puede ser programada en el puerto serial del computador utilizado; por lo tanto las simulaciones se realizaron a 9,6 Kbps.

El circuito necesario y el esquema de conexión utilizado para estas pruebas se muestran en la figura 34. Se puede observar que el circuito sigue las especificaciones del estándar ISO 9141 para la ECU, excepto por el diodo Zener y el integrado MAX232, los cuales se usaron para la lectura de la línea K y la conversión a valores RS232 para la computadora.

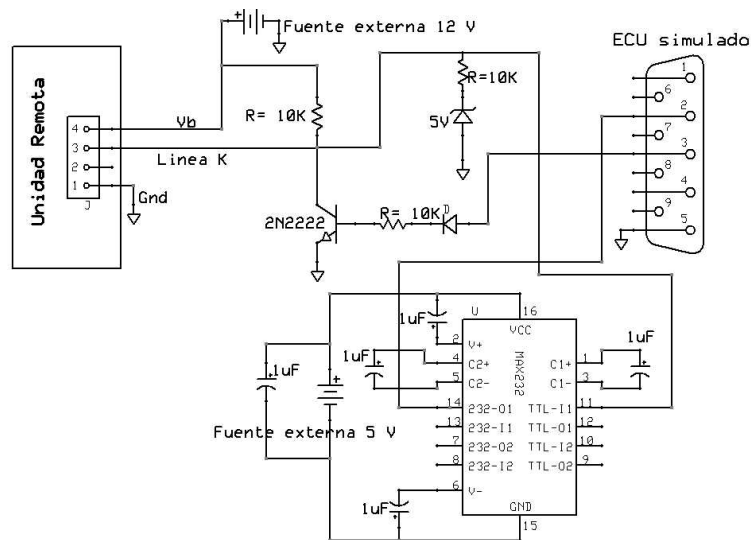


Figura 34 Esquema de conexión de Unidad Remota con vehículo simulado.

Las pruebas con el vehículo simulado permitieron comprobar el correcto envío y recepción de los mensajes de diagnóstico, además de verificar el funcionamiento de la rutina de sincronización para esa velocidad. El programa utilizado para simular el ECU del vehículo fue *Docklight* debido a que éste permite el envío de bytes específicos indicando su valor en cualquier notación: decimal,

hexadecimal u octal (otros programas como *Hyperterminal* envían solamente bytes correspondientes a caracteres ASCII, por lo que se dificulta enviar un valor numérico específico). La Unidad Remota se operó de manera local con otro computador y el programa usado para enviar los comandos fue *Hyperterminal*. Más detalles sobre estas pruebas se presentan en el Apéndice 5. Pruebas posteriores a las mostradas en el Apéndice detectaron que la Unidad Remota no lograba sincronizarse a velocidades superiores a 9.6 Kbps, lo que permitió, en conjunto con otras pruebas que se explican en secciones posteriores, determinar un error en la rutina de sincronización y realizar las correcciones posteriores.

IV.8.5 Pruebas de la Unidad Remota con vehículo real

Luego de haber realizado las pruebas con el vehículo simulado se procedió a conectar la Unidad Remota a uno real. Las primeras pruebas arrojaron errores en la inicialización de la interfaz ISO 9141. Luego de numerosas revisiones se decidió crear un “sistema espía” para monitorizar la comunicación entre un equipo de diagnóstico profesional y el vehículo, con el objetivo de determinar diferencias entre el protocolo usado por el equipo profesional y el implementado en la Unidad Remota. El sistema espía utiliza un computador y un hardware externo de acoplamiento óptico. El programa espía funciona como un osciloscopio con memoria, el cual detecta solo dos niveles de voltaje (alto y bajo), con una frecuencia de muestreo de 10 microsegundos. Este sistema guarda la información de manera codificada (almacena el tiempo entre cambios de estado) con el objetivo de comprimir el tamaño del archivo resultante.

Con los datos obtenidos del sistema espía se pudo comprobar que el vehículo analizado era compatible con ISO 9141-2 (tiempos de bit alrededor de 100 microsegundos) y que el protocolo empleado por el equipo profesional era idéntico al

empleado por la Unidad Remota (realizado en base a los estándares ISO 9141 y SAE J1979). Esta prueba sirvió de soporte para realizar una revisión más profunda del software del microcontrolador y se logró identificar un error en la rutina de sincronización. El error fue corregido y se volvieron a realizar pruebas con el vehículo simulado y el real, obteniendo los resultados esperados. Más detalles de estas pruebas en el Apéndice 6.

IV.9 Diseño de las aplicaciones cliente-servidor

El software SMRI está formado por dos aplicaciones: un cliente y un servidor. El cliente se encarga de manejar los requerimientos del usuario y brindarle la interfaz necesaria para poder comunicarse con el vehículo. El servidor se encarga de la administración de las conexiones de los clientes y los vehículos, así como también ser mediador de la comunicación entre el cliente y el vehículo correspondiente

El proceso se inicia cuando algún usuario desea monitorizar un vehículo determinado, para lo cual se requiere que el cliente se conecte vía TCP/IP al servidor; posteriormente, el proceso de registro se realiza indicando la placa del vehículo que se desea monitorizar. La unidad remota cada cierto tiempo establece conexión con el servidor, sin embargo, logra registrarse sólo cuando el cliente previamente haya manifestado la intención de monitorizar el vehículo, es decir, haya registrado la placa correspondiente en el servidor.

En caso de que el cliente ya haya registrado el vehículo, la unidad remota se registra en el sistema y se queda esperando para interactuar con el usuario. En caso contrario, la unidad remota recibe un mensaje de desconexión, y al transcurrir cierto tiempo, inicia nuevamente el proceso de conexión, como se indicó en secciones anteriores.

Cuando el vehículo y el usuario se encuentren registrados en el sistema, este último solicita al vehículo los parámetros que se encuentran disponibles para su monitorización, el cual recibe la petición y envía la respuesta. Con esta información, el cliente muestra al usuario cuáles son los parámetros disponibles. A medida que el usuario selecciona los parámetros a monitorizar, se generan solicitudes que son recibidas por el vehículo. Cuando el software cliente envía el comando “CONTINUO xx”, la unidad remota procede a enviar los parámetros solicitados.

La aplicación cliente procesa la información recibida, realiza las conversiones necesarias y se la muestra al usuario en gráficas que permiten ver la variación de cada uno de los parámetros solicitados en el tiempo para facilitar la labor de diagnóstico. En la figura 35 se indica el diagrama de interacción entre el vehículo y el usuario.

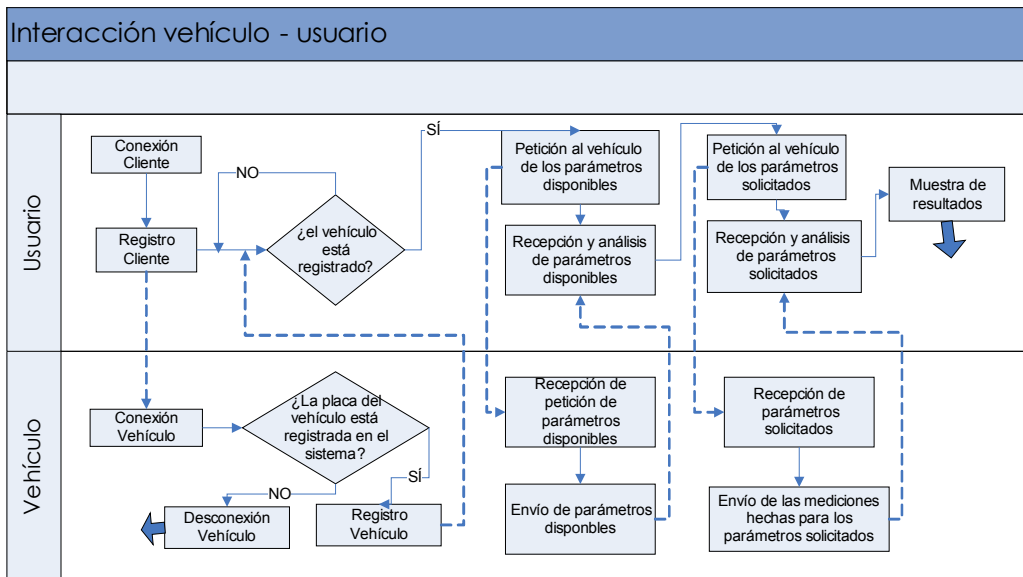


Figura 35 Diagrama de Interacción Vehículo-Usuario

IV.9.1 Cliente

En primer lugar, el cliente debe conectarse al servidor y registrar la placa del vehículo que desea monitorizar. Cuando el vehículo ya esté registrado el cliente debe

solicitarle los parámetros que se encuentran disponibles para la monitorización, al recibirlos se los muestra al usuario y a partir de este momento ya puede iniciar la monitorización.

De acuerdo a la solicitud del usuario, el vehículo responde la información solicitada, la cual es procesada por el cliente para mostrársela al usuario en forma de gráficas. A continuación, la figura 36 muestra la funciones realizadas por el cliente descritas en esta sección.

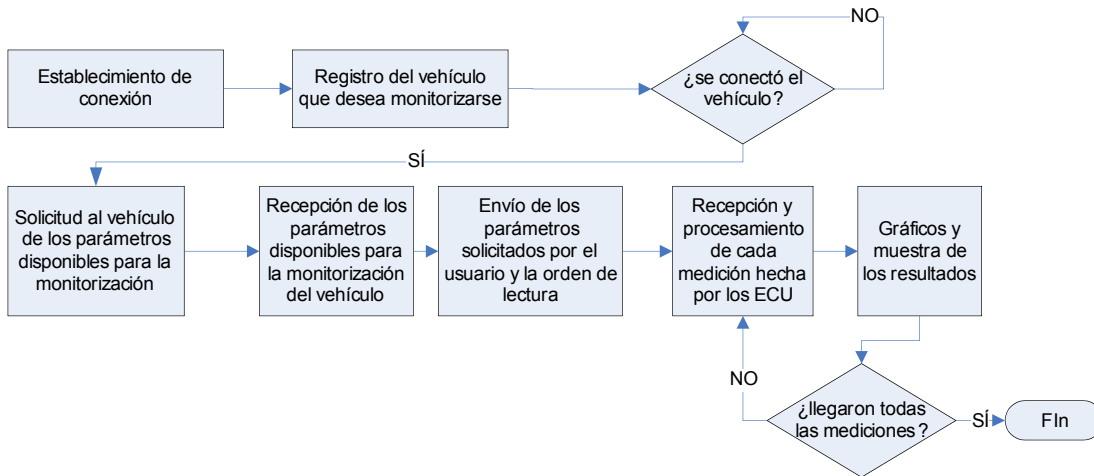


Figura 36 Diagrama de flujo Cliente

IV.9.2 Servidor

La aplicación servidor realiza las funciones que se muestran en el diagrama de flujo de la figura 37. El servidor siempre está esperando la llegada de datos, en caso de que la información que llegue sea una petición de registro en el sistema verifica si la petición pertenece a un cliente o a un vehículo.

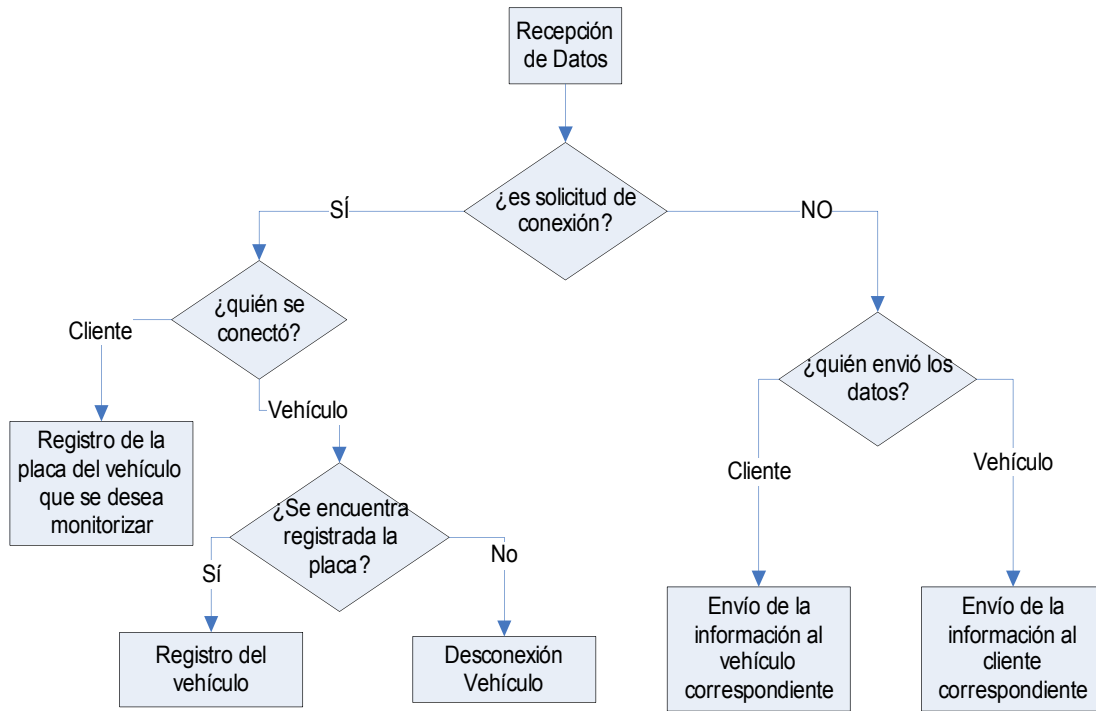


Figura 37 Diagrama de flujo de software servidor

Si se recibe una solicitud de conexión proveniente de un cliente, se registra la placa del vehículo que se desea monitorizar y se incluye el identificador del cliente (socket) en el sistema; si es un vehículo, se verifica que la placa haya sido registrada anteriormente y, en caso afirmativo, se procede a registrar el identificador del vehículo (socket) en el sistema. Una vez finalizado este proceso se tiene creado el vínculo entre el cliente y el vehículo.

Cuando los datos que llegan al sistema no son una petición de registro, la labor del servidor es ser mediador entre el emisor de los datos y el destino final. Los datos que envían los clientes deben llegar al vehículo y viceversa.

IV.10 Elección del lenguaje de programación para las aplicaciones cliente servidor

Al tener definida las funciones desarrolladas por el software SMRI se procedió a elegir el lenguaje de programación que mejor se adaptara al mismo. Los Lenguajes considerados fueron los siguientes: LabVIEW, Microsoft Visual Basic 6.0 y Microsoft Visual Studio. Net. A continuación se realiza una breve descripción de los mismos.

“LabVIEW, por sus siglas en inglés, *Laboratory Virtual Instrumentation Engineering Workbench*, es un ambiente de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de mediciones y presentaciones de datos” (National Instruments, 2006). LabVIEW es desarrollado por National Instruments y trabaja sobre una gran cantidad de plataformas Windows, UNIX, LINUX Y MAC.

“Microsoft Visual Basic 6.0 es una evolución del lenguaje Basic creado por John Kemeny y Thomas Kurtz en el Dartmouth College, el cual se convirtió en un lenguaje muy popular dentro de centros educativos y posteriormente fue adaptado para su uso en computadoras personales por Bill Gates” (HALVORSON, 1996, p. xxxiv). Desde entonces el lenguaje ha evolucionado hasta convertirse en lo que es hoy en día una herramienta para el desarrollo gráfico de aplicaciones y forma parte del paquete Visual Studio 6 de Microsoft.

Microsoft Visual Studio.Net es “ un Entorno de desarrollo integrado compartido (IDE, Integrates Development Environment) el cual permite la administración de ventanas, creación de diseñadores de formularios Web y formularios de Windows en los siguientes lenguajes: Visual Basic.NET, Visual C++ o C#” (“Así es Microsoft Visual Studio.Net”,2001, p. ix-x).

En la elección del lenguaje de programación se consideraron los requerimientos principales del software, los cuales son: establecimiento de conexiones TCP/IP y posibilidad de graficar las mediciones hechas por el vehículo. Debido a que los tres lenguajes de programación estudiados cumplían con ambos requisitos, se procedió estudiar la factibilidad técnica. En este estudio se tomaron en cuenta los siguientes aspectos: facilidad de uso, flexibilidad y curva de aprendizaje.

Los tres lenguajes son flexibles debido a que la sintaxis de los mismos resultan sencillas y de rápida asimilación, incluso si no se es un usuario experimentado en ninguno de los lenguajes. Además, todos los ambientes de trabajo ofrecen corrección de errores de sintaxis inmediata y brindan un buen soporte de ayuda.

En cuanto a la facilidad de uso, el que presenta mayor ventaja es Visual Basic, ya que con conocimientos básicos de programación se puede desarrollar cualquier programa; en el caso de LabVIEW el paradigma de programación varía notablemente con respecto a los otros dos lenguajes debido a que la programación no se realiza mediante instrucciones secuenciales sino de manera gráfica a través de módulos, asignándole a cada uno de ellos la funcionalidad deseada; para Visual Studio.net se requiere conocimientos avanzados en programación orientada a objetos.

En lo relativo a la curva de aprendizaje, Visual Basic posee la más corta con respecto al tiempo debido a la sencillez de la programación y a que no se requiere conocimientos adicionales a los adquiridos para el momento de realización del proyecto. En el caso de LabVIEW, la curva de aprendizaje se alargaba debido al cambio de paradigma de la programación, lo cual tomaría cierto tiempo de adaptación, para lograr los mismos resultados que se obtienen con los otros lenguajes. En el caso de Visual Studio. NET se requiere dominio de programación orientada a objetos, lo cual alargaba la curva de aprendizaje ya que se debía adquirir los

conocimientos necesarios en el corto tiempo que se disponía para el desarrollo del software.

Tomando en cuenta las razones anteriormente expuestas se decidió emplear como lenguaje de programación para el desarrollo del software SMRI Visual Basic 6.0, sin embargo para aplicaciones futuras se recomienda el uso Microsoft Visual Studio. Net debido a las altas prestaciones en la interfaz gráfica que brinda el mismo. Aunque LabVIEW permite cumplir con todos los requerimientos del software, no es el más adecuado para la aplicación, ya que el mismo está diseñando primordialmente para aplicaciones de instrumentación local debido a que sus funcionalidades IP son muy limitadas, especialmente para la aplicación del servidor.

IV.11 Elección del manejador para la base de datos

Para la implantación de la base de datos se consideraron los siguientes manejadores: Microsoft Access, Microsoft SQL Server y ORACLE.

Microsoft Access es un sistema de gestión de bases de datos creado y modificado por Microsoft. “Access permite guardar y recuperar, de forma separada, la información sobre los distintos tópicos de un tema; esto facilita enormemente su organización. Pero también permite establecer relaciones entre ellos y así obtener una información mucho más aprovechable.” (VIDAL, 1993, p. 27)

Microsoft SQL Server es un sistema de gestión de bases de datos relacional basado en el lenguaje SQL . “El lenguaje estructurado de consulta SQL (*Structured Query Language*), es un estándar informático corrientemente utilizado para definir, modificar y gestionar datos y controlar cómo se realizan cambios en la base de datos usando tablas, índices, claves, filas y columnas para almacenar la información.” (COFFMAN, 1999, p. 3)

ORACLE es un sistema de administración de base de datos (o RDBMS por el acrónimo en inglés *de Relational Data Base Management System*), a diferencia de los anteriores es fabricado por Oracle Corporation.

Los aspectos considerados para la elección del sistema manejador de base de datos fueron los siguientes: compatibilidad con el lenguaje de programación empleado en las aplicaciones cliente-servidor y curva de aprendizaje.

Los tres manejadores estudiados son compatibles con Visual Basic 6.0. En cuanto a la curva de aprendizaje, el que presentaba la curva más corta con respecto al tiempo era Microsoft Access, ya que, es el menos complejo de los manejadores considerados para este estudio; los otros dos requieren de conocimientos avanzados en el manejo de base de datos que no forman parte del plan de estudios de la carrera de Ingeniería de Telecomunicaciones de la UCAB.

Debido a las razones expuestas anteriormente se seleccionó Microsoft Access como manejador para la base de datos del SMRI. Sin embargo se recomienda el uso de cualquiera de los otros dos manejadores en aplicaciones que requieran acceso simultáneo a la base de datos (el sistema SMRI maneja bases de datos independientes en cada cliente y sólo se puede monitorizar un vehículo a la vez por cada aplicación cliente).

IV.12 Diseño de la base de datos

Con el objetivo de visualizar el comportamiento de un vehículo durante un período de monitorización prolongado y facilitar las labores de diagnóstico, se

decidió crear una base de datos para almacenar todas las muestras de los parámetros medidos. Esta base de datos se actualiza con cada medición recibida.

Al finalizar la monitorización en tiempo real es posible acceder a la base de datos a través de la aplicación cliente. Los datos almacenados son presentados en sus valores numéricos y en forma gráfica, para facilitar su comprensión. La base de datos, al estar realizada en Microsoft Access, es portátil y puede ser manejada posteriormente para un análisis más complejo.

IV.13 Pruebas del software servidor-cliente con vehículo simulado y corrección de errores

Una vez que se desarrolló e implantó el software se procedió a realizar pruebas de conexión, las cuales se realizaron con múltiples clientes y simulando múltiples vehículos con *hyperterminal*.

Dichas pruebas permitieron detectar errores en la extracción y tratamiento de los datos contenidos en las tramas transmitidas entre las aplicaciones cliente-servidor, así como también localizar errores en el registro y la mediación de las comunicaciones en el servidor. Finalmente, se logró un correcto establecimiento de la comunicación entre vehículos y clientes, además de la interpretación correcta de los datos.

V Resultados

El SMRI, diseñado en este Trabajo Especial de Grado, es un sistema de monitorización de vehículos que permite acceder a los parámetros de funcionamiento de los mismos de forma remota. El SMRI está formado por tres componentes principales: la Unidad Remota ubicada en el vehículo, el cliente que sirve como interfaz con el usuario y un servidor central que se encarga de gestionar la comunicación entre los componentes anteriores.

La Unidad Remota se encarga de la extracción de los parámetros de la computadora del vehículo a través de una interfaz OBD-II con acoplamiento óptico, compatible con el protocolo ISO 9141. El microcontrolador central es un PIC 16F877A. La unidad dispone de una interfaz serial RS-232, la cual brinda la posibilidad de comunicarse con un computador local o con el módem CDMA EMII-800, encargado de la transmisión inalámbrica. La alimentación eléctrica es tomada del voltaje de la batería del vehículo a través del encendedor; además, se brinda una salida de 12V para alimentar al módem CDMA. También se dispone de una interfaz para la conexión de un dispositivo GPS, aunque su implantación no formó parte de este proyecto. La Unidad Remota se reporta cada 5 minutos con el servidor para verificar si algún cliente desea monitorizar el vehículo.

El servidor se encarga de recibir las solicitudes de conexión de los clientes y establecer el enlace con los vehículos correspondientes cuando éstos se reportan. Esta aplicación fue desarrollada en Visual Basic 6.0, por lo que requiere el sistema operativo Windows 2000 o superior para operar. El servidor es capaz de gestionar la comunicación simultánea de 10 vehículos y 10 clientes.

La aplicación cliente brinda al usuario la interfaz para comunicarse con el vehículo que desea monitorizar, mediante el envío de comandos y la presentación

gráfica de las mediciones realizadas. Al igual que el servidor, esta aplicación se desarrolló en Visual Basic 6.0. Además, dispone de una base de datos implementada en Microsoft Access que permite almacenar las mediciones realizadas. Esta base de datos puede ser consultada por los usuarios desde la aplicación cliente.

En la figura 38 se muestra un diagrama general de interacción de los tres componentes del SMRI.

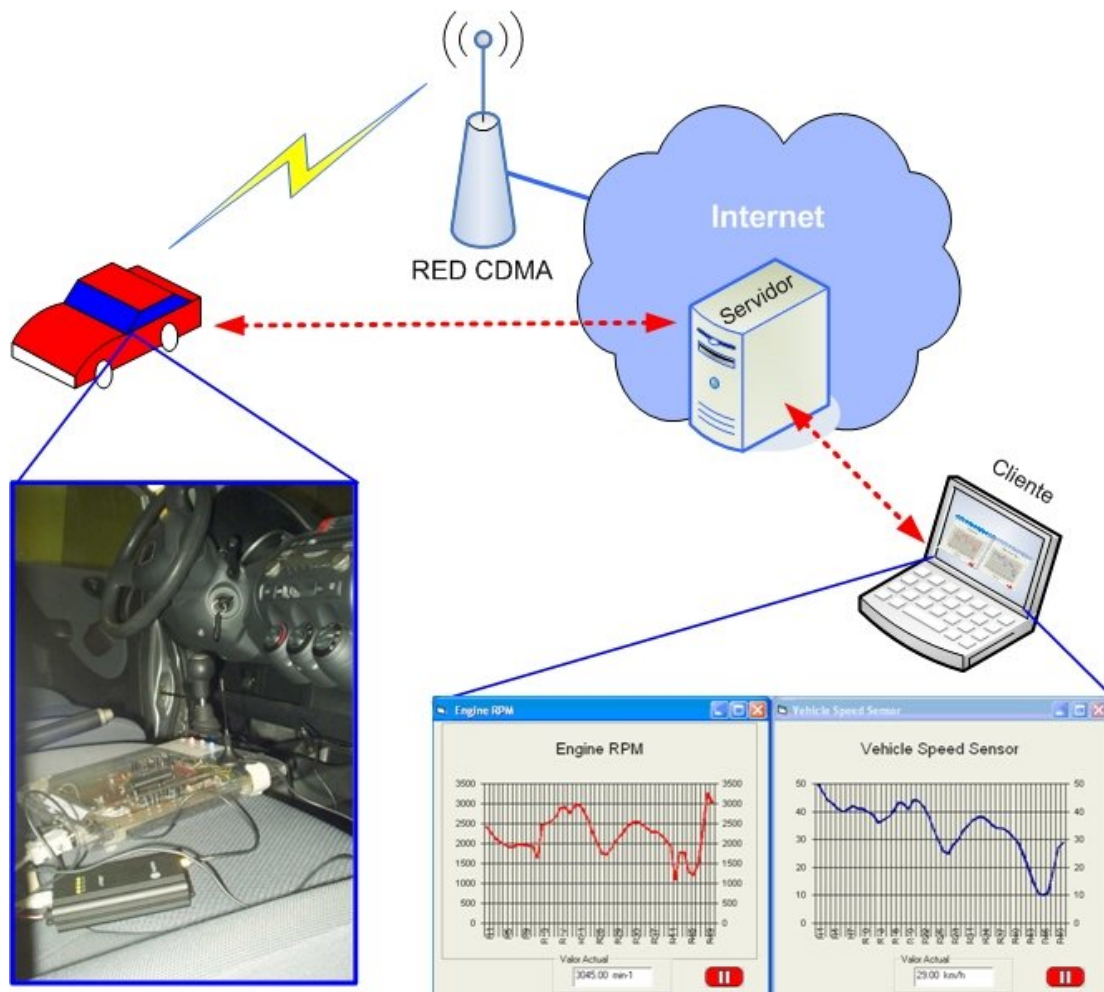


Figura 38 Diagrama general de interacción de los componentes del SMRI.

En la figura 39 se observa la interfaz gráfica de la aplicación servidor. El campo de texto superior se utiliza como medio para visualizar los mensajes transmitidos entre las unidades remotas con sus respectivas aplicaciones clientes. El marco de

Conexión permite seleccionar el puerto TCP por el cual se va a realizar la comunicación. El marco *Reporte de Conexión* permite visualizar los vehículos y clientes que están conectados en el momento indicando el número del “*Socket de Windows*” de cada uno y la placa.

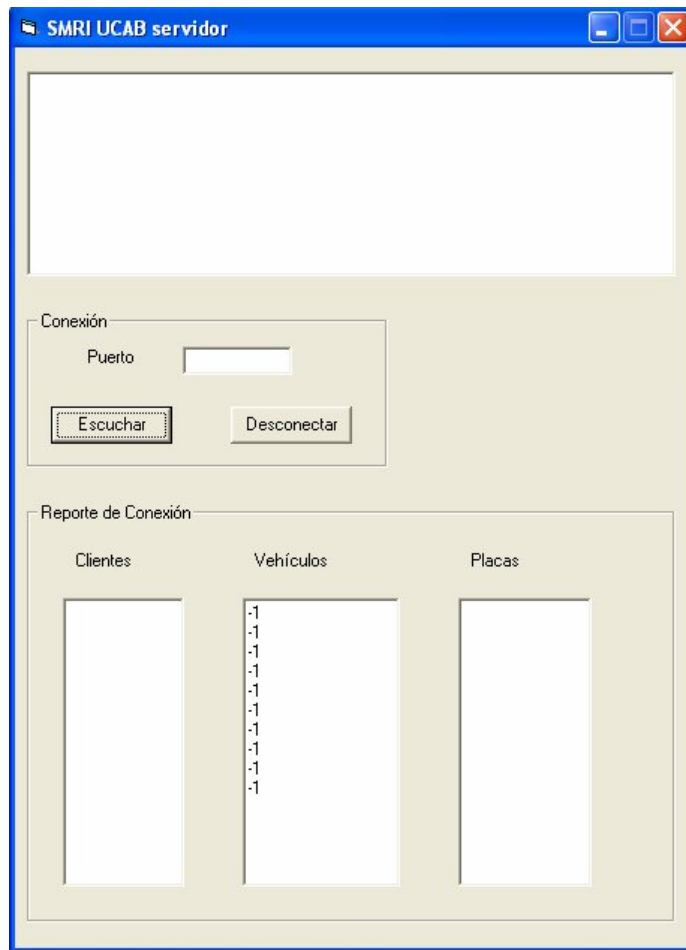


Figura 39 Interfaz gráfica aplicación Servidor.

La interfaz gráfica de la aplicación cliente está formada por dos ventanas interactivas. La ventana principal posee todos los campos necesarios para realizar la monitorización en tiempo real de los parámetros del vehículo; la segunda ventana permite explorar la base de datos de las monitorizaciones ya realizadas. En la figura

40 se puede observar la interfaz gráfica de la ventana principal de la aplicación cliente.

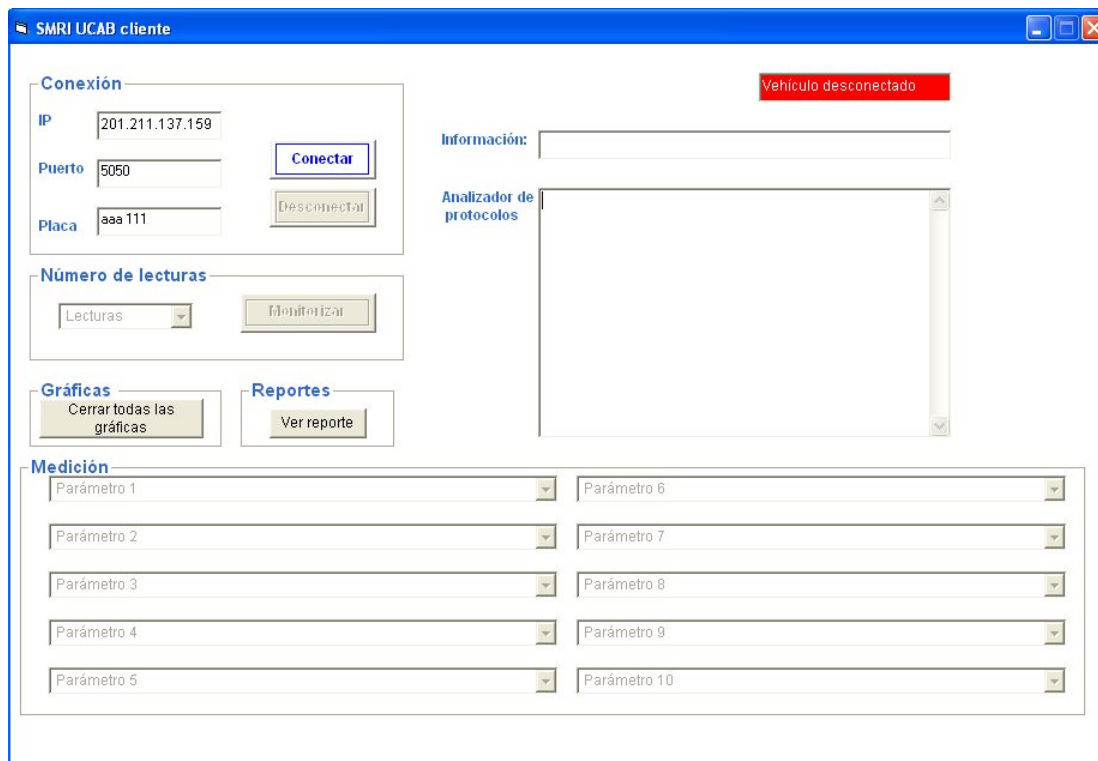


Figura 40 Ventana principal aplicación Cliente.

El marco *Conexión* permite especificar la dirección IP del servidor, el número del puerto TCP y la placa del vehículo que se desea monitorizar. El campo de texto *Información* brinda al usuario la posibilidad de conocer el estado de la aplicación: *Servidor imposible de localizar*, *Conectado a servidor*, *Localizando vehículo...*, etc. Cuando el vehículo es localizado, la aplicación envía los comandos necesarios para obtener información sobre los parámetros de monitorización que el vehículo tiene disponible. Mientras esto ocurre, las ventanas del marco *Número de lecturas* y *Medición* permanecen deshabilitadas; sin embargo, el campo de texto *Analizador de protocolos* permite visualizar la comunicación entre la aplicación Cliente y la Unidad Remota conectada al vehículo.

Una vez enviada la información de los parámetros disponibles, el usuario puede seleccionar cual de ellos desea monitorizar (máximo 10) mediante las listas ubicadas en el marco *Medición*. Es posible seleccionar el mismo parámetro en más de una ventana para mejorar la velocidad de muestreo del mismo. El marco *Número de Lecturas* permite seleccionar cuantas veces la Unidad Remota va a solicitar a la computadora del vehículo los 10 parámetros indicados en el marco *Medición*.

La monitorización del vehículo comienza cuando el usuario presiona el botón *Monitorizar*. En ese momento se abre una ventana por cada parámetro distinto que se haya seleccionado. Estas ventanas grafican las parámetros con un eje vertical autoajustable según el rango de valores de los datos recibidos y con un eje horizontal de 50 muestras. La gráfica se va trasladando de derecha a izquierda a medida que van llegando nuevas mediciones. En la figura 41 se muestran unas gráficas de mediciones realizadas en una de las pruebas del sistema.

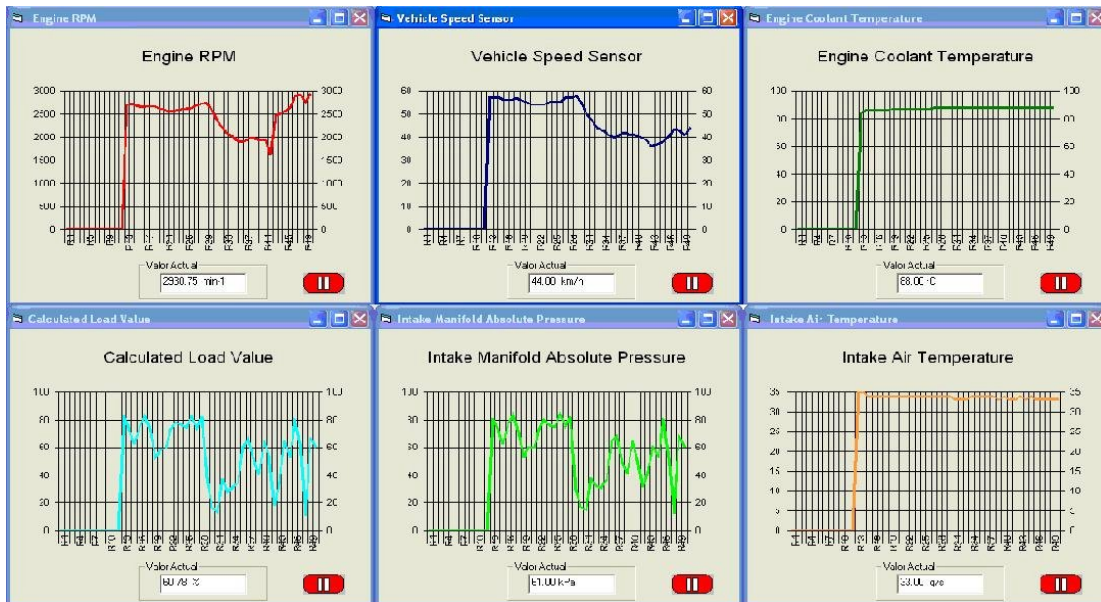


Figura 41 Ventanas de medición en tiempo real de la aplicación Cliente.

Para acceder a mediciones realizadas con anterioridad se debe presionar el botón *Ver Reportes* de la ventana principal de la aplicación cliente. Al hacerlo, se abre una

nueva ventana que permite acceder a la base de datos de la aplicación cliente. Se debe seleccionar la fecha de monitorización, la placa del vehículo, la hora y el parámetro que se desea ver. El software muestra entonces una gráfica con todos los valores de la medición respectiva y un cuadro de texto con los valores específicos. Es importante recordar que la exploración de la base de datos puede realizarse de también directamente desde *Microsoft Access*. En la figura 42 se muestra la ventana de exploración de la base de datos de la aplicación cliente.

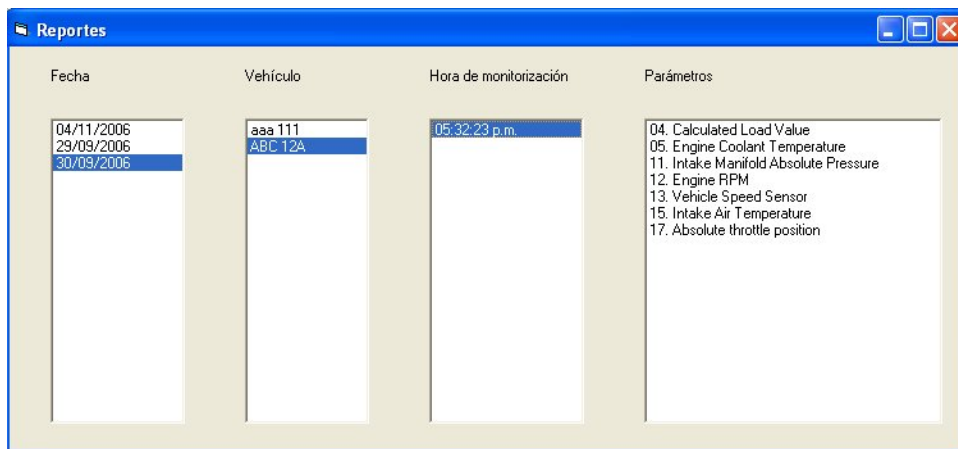


Figura 42 Ventana de exploración de base de datos de aplicación Cliente.

VI Conclusiones y Recomendaciones

VI.1 Conclusiones

El sistema SMRI logra exitosamente la monitorización inalámbrica, en tiempo real, de los parámetros proporcionados por la computadora del vehículo a través de una interfaz compatible con los estándares OBDII ISO 9141, a nivel de Capas 1 y 2 del modelo OSI, y SAE J1979 en capa de aplicación.

Se escogió CDMA 2000 1X como el sistema de transmisión inalámbrica más adecuado para soportar al sistema SMRI debido a su vigencia tecnológica, disponibilidad en el país, mayor cobertura a nivel nacional con respecto a las otras tecnologías y a debido a que brinda una velocidad para datos adecuada a los requerimientos del SMRI

Luego de estudiar los requerimientos para la adquisición de datos del computador de vehículos, se diseñó el hardware de la Unidad Remota, que cumple con las siguientes especificaciones:

- Dimensiones Físicas: 28cm x 20cm x 3.5cm
- Microprocesador: 16F877A
- Alimentación eléctrica suministrada por la batería vehículo (12V a 14V) a través del encendedor. También puede ser conectado directamente desde el cajetín de fusibles si se desea instalar el sistema de manera permanente.
- Compatibilidad con ISO 9141 e ISO 9141-2, ya que puede adaptarse a cualquier velocidad de la interfaz (hasta 10.4Kbps, de acuerdo al último estándar) mediante el byte de sincronismo enviado por la computadora del vehículo en el proceso de inicialización.

- Interfaz serial RS-232 capaz de comunicarse a 115.2 Kbps con el módem CDMA o con un computador local.
- Interfaz serial TTL a 4.8 Kbps que permite la comunicación con el módulo GPS.

La aplicación cliente del SMRI maneja una base de datos capaz de almacenar todas las mediciones realizadas por el usuario. Además, esta aplicación permite monitorizar los parámetros de un vehículo de forma gráfica en tiempo real y visualizar los datos recopilados de manera gráfica y numérica.

Finalmente, se realizaron pruebas del sistema SMRI con un vehículo destinado para este fin, modelo HONDA FIT. Además, se realizó un diagnóstico al vehículo con la ayuda de un mecánico especialista, verificando el correcto funcionamiento del mismo

VI.2 Recomendaciones

Para aplicaciones comerciales del SMRI, se recomienda utilizar el microcontrolador PIC 18F4550, debido a que sus mayores prestaciones facilitan la implantación de nuevos servicios. Adicionalmente, se recomienda para futuros desarrollos asegurar compatibilidad con el protocolo ISO 15765-4 (CAN) de OBD-II, debido a que éste se posiciona como estándar principal para diagnóstico de vehículos en los próximos años.

Las aplicaciones a futuro del SMRI incluyen su incorporación en el vehículo Minibaja UCAB, para lo cual se requiere la instalación de los sensores necesarios y una unidad capaz de leerlos y presentarlos en una interfaz ISO 9141. La Unidad Remota dispone de 8 entradas analógicas que pueden ser usadas para instalar

directamente los sensores, pero se requiere actualizar el software de la misma para soportar estas nuevas funciones.

En cuanto a aplicaciones comerciales, actualmente se está negociando, con diferentes concesionarios de vehículos y flotas de transporte, la incorporación del SMRI en el diagnóstico de las unidades.

VII Bibliografía

“CDMA Data Terminal. AnyData Proprietary Packet Data User Manual”, AnyData.net. Junio 2001.

“ISO 9141. Road Vehicles- Diagnostic systems- Requirements for interchange of digital data”, Estándar Internacional ISO, Octubre 1989.

“SAE J1979. (R) E/E Diagnostic Test Modes”, SAE, Abril 2002.

“SAE J2178. Class B data communication network messages” Partes 1-4, 1994.

“SSF 14230. Road Vehicles, Diagnostic Systems, Keyword Protocol 2000”. Partes 1-3. Fuerza de Tarea de Diagnóstico de Vehículos de Suecia. Octubre 1997.

COFFMAN, Gayle. “SQL Server 7”. Mc Graw Hill. Madrid, España. 1999

HALVORSON, Michael. “Microsoft Visual Basic 4 ”.McGraw Hill Interamericana de España. Madrid, 1996.

HUIDOBRO, José Manuel. Fundamentos de Telecomunicaciones (1ª ed.). Madrid, España. Editorial Paraninfo.2001.

National Instruments. “Aprenda sobre LabVIEW”. Fuente:

<http://digital.ni.com/worldwide/latam.nsf/b63ef100ab4b5df486256425006883b7/9de46aa55f1849c686256c3d006ff338> . 17 de septiembre 2006

RILEY, Michael. “Automotive Electronic Tests- Computers Networks take to the road”. Revista electrónica: Evaluation Engineering. Fuente:

<http://www.evaluationengineering.com/archive/articles/0398auto.htm>.

SENDÍN ESCALONA, Alberto. Fundamentos de los sistemas de comunicaciones móviles (1ª ed.). Madrid: Mc Graw Hill. 2004

THOMPSON, Mike. “Understanding OBDII Monitors”. Fuente: <http://www.deq.state.va.us/mobile/mobobdm.html>. 15 de septiembre 2006.

VERA, Arturo. “Sistemas Celulares de Tercera Generación”. Fuente: <http://www.monografias.com/trabajos15/telefonía-celular/telefonía-celular.shtml>. . 10 de noviembre 2005

VIDAL FERNANDEZ, María Carmen y PEÑA TRESANCOS, Jaime. “Microsoft Access”. Editorial Paraninfo. Madrid, España. 1993

“Así es Microsoft Visual Studio.Net”. Mc Graw Hill. Madrid, España. 2001

“Qué es ORACLE ” Fuente: <http://www.desarrolloweb.com/articulos/840.php>. 15 de septiembre 2006

Glosario

AT: comando o instrucción codificada por poder comunicarse con un módem

bps: bits por segundo.

CAN: Diagnostics on Controller Area Network

CARB: California Air Resources Board

CDMA: Code Division Multiple Access

CRC: Cyclic Redundancy Check

ECU: Electronic Control Unit

EPA: Environmental Protection Agency

GPS: Global Positioning System

ISO: International Standards Organization

OBD: On Board Diagnostic

PID: Parameter ID o identificador de parámetro

SAE: Society of Automotive Engineers

SMRI: Sistema Monitor Remoto Interactivo

SMS: Short Message Service

Socket: identificación definida por una dirección IP, un protocolo y un número de puerto.

Vb: Voltaje de la batería

WCDMA: Wideband Code Division Multiple Access

UART: Universal Asynchronous Receiver-Transmitter

Apéndice 1. Monitores de OBDII

A continuación se listan los monitores presentes en OBD II, de éstos los nueve primeros son los principales, el resto fue agregado recientemente. Seguidamente se realiza una descripción breve de los principales.

- Eficiencia catalítica (Catalyst Efficiency)
- Componentes generales (Comprehensive Component Monitor, CCM)
- Recirculación del gas de escape
- Sistema Evaporativo (Evaporative System, EVAP)
- Sistema de combustible (Fuel System, adaptive fuel system trim)
- Catalizador Calentado (Heated Catalyst)
- Sensor calentado de oxígeno (Heated Oxygen Sensor)
- Detección de fallo de disparo (misfire ignition)
- Aire Secundario (Secondary Air Injection)
- Termostato (Thermostat)
- Aire acondicionado (Air Conditioning)

Monitor de detección de fallo de disparo:

Se encarga de verificar fallas en el proceso de disparo de ignición de combustión en el cilindro, éstas se deben a la falta de ignición de la mezcla aire/combustible. Si la falla de disparo es pequeña, se graba un DTC Pendiente. Si la falla de disparo es tan grande que pudiera causar daños o permitir que las emisiones excedan 1.5 veces el estándar EPA, se graba inmediatamente un DTC guardado y se ilumina la MIL. Si la falla es severa, y capaz de dañar el convertidor catalítico, la

MIL parpadea una vez por segundo mientras se detecte la falla de disparo, y después se ilumina normalmente.

Monitor del sistema de combustible:

El monitor del sistema de combustible se encarga de incrementar o recortar el paso de combustible y lograr, de esta manera, una combustión eficiente. Este monitor es usado para verificar el programa de corrección de combustible del sistema de abordo, el cual permite o detiene la entrega de combustible para mantener la combustión de manera eficiente.

Este programa emplea dos formas de corrección: *Short-Term Fuel Trim* y *Long-Term Fuel Trim*. Estas dos formas trabajan de manera conjunta para ajustar la cantidad de combustible que reciben los cilindros para una combustión óptima.

Monitor de los componentes generales

Monitor que corre constantemente y verifica los sensores, actuadores y apagadores OBD-II. Al obtenerse valores actuales se comparan con valores fijados como correctos para determinar si el sensor, actuador o apagador es funcional y opera racionalmente.

Monitor de oxígeno

El sensor de oxígeno cumple un papel fundamental dentro del sistema de combustible. Los sensores de oxígeno se encuentran antes y después del convertidor catalítico en el tubo de escape. Sin embargo, la ubicación de estos sensores, puede variar de acuerdo al fabricante del vehículo. Cada vehículo puede tener hasta dos “bancos” (ver glosario), B1 y B2, uno para cada tubo de escape que posea el vehículo.

Se tienen cuatro sensores S1 a S4. S1 y S2 preceden al convertidor precatalítico y por lo tanto se denominan sensores *preconvertidor catalítico (pre-cat)*. S3 y S4 son sensores *postconvertidor catalítico (post-cat)*. En caso que sólo existan sensores S1 y S2, S1 será pre-cat. y S2 post-cat.

Los sensores toman muestras del escape y arrojan un valor en términos de voltaje. La computadora del motor usa el voltaje para determinar si la mezcla aire/combustible es rica o pobre, y ajusta así la cantidad de combustible que entra al motor. Debido al tiempo que toma a la computadora corregir la mezcla, y el tiempo que toma el sensor para reportar, el sensor cambia frecuentemente de rico a pobre.

Los sensores pre-cat exhiben la mayoría de cambios (cuentas cruzadas). Los post-cat deberán tener una señal plana o de amplitud relativamente baja. Un buen sensor pre-convertidor catalítico debe cambiar entre los valores de 1 volt para una mezcla rica y .1 volt para una pobre.

Monitor de temperatura y eficiencia catalítica

El convertidor catalítico se encarga de disminuir las emisiones contaminantes de los vehículos. Recibe el nombre de convertidor debido a que convierte las emisiones dañinas en gases menos dañinos y vapor de agua.

El trabajo se realiza haciendo pasar el flujo contaminado a través de un dispositivo catalítico temporizado, el cual realiza la transformación química. Por lo tanto el convertidor catalítico debe mantenerse siempre limpio y operando bajo un rango de temperatura determinado.

Monitor del Sensor calentado de Oxígeno

Este monitor no continuo evalúa el estado operacional de todos los sensores de calentamiento de oxígeno del auto.

Monitor del Sistema Evaporativo

El sistema trabaja para prevenir escape de vapores de combustible a la atmósfera. Este sistema extrae vapores del tanque al motor para ser quemados. Varios componentes forman este sistema, incluyendo canastas de carbón, solenoide de purga, tapón de gasolina y tubos conectores.

El monitor de este sistema vigila el flujo correcto de vapor de combustible al motor y presuriza el sistema para verificar si hay fugas. Este monitor trabaja en “dos vías”, es decir, requiere que una falla ocurra en dos “viajes” (ver glosario) consecutivos antes de que la computadora encienda la MIL, o luz indicadora de malfuncionamiento, y grabe un DTC en el área de DTCs guardados de la memoria de la computadora

Monitor de Sistema de Aire Secundario

El sistema de Aire secundario se usa para asistir al convertidor catalítico a quemar los gases relacionados a las emisiones. Al arrancar un motor frío, muchos componentes necesarios para la regulación de emisiones no se encuentran a temperatura operacional. La computadora del motor opera en circuito abierto.

Cuando la computadora opera en circuito abierto emplea valores predeterminados, o por defecto, para regular la mezcla de combustible y avance de chispa. En estos casos, el convertidor catalítico agiliza el paso de oxígeno en el flujo de escape por medio del sistema de aire secundario para ayudar a quemar el combustible no quemado o parcialmente quemado. Este paso de oxígeno logra que el calentamiento del convertidor catalítico se realice más rápido.

Glosario

Banco: Grupo de cilindros que alimentan a un sensor de oxígeno.

CAT: convertidor catalítico

MIL (Malfunction Indicator Lamp) o luz indicadora de mal funcionamiento, se emplea para indicar al operador que ha ocurrido una falla. Las MIL se usan sólo para señalar problemas del sistema de control de emisiones, no se emplean como recordatorio de mantenimiento,

Viaje: término que implica una serie específica de pasos requeridos para que un monitor realice y complete su prueba.

Bibliografía:

“SAE J1979. (R) E/E Diagnostic Test Modes”, SAE, Abril 2002.

Apéndice 2. Protocolos OBD-II. Capas 1 y 2.

1. SAE J1850

La norma SAE J1850 define las dos capas más bajas (del modelo OSI) del protocolo de comunicación entre las ECU y el equipo de diagnóstico, utilizado por los dos principales fabricantes de vehículos americanos: General Motors y Ford. Las características que se especifican son: arquitectura de la red, velocidades de acceso, niveles de voltaje, modulación, estructura de los mensajes, sincronización del reloj, orden de envío de los datos, etc. La única característica de capa de enlace que no se especifica en este estándar es la estructura del encabezado del mensaje, el cual se especifica en el estándar SAE J2178 y en el SAE J1979. Las principales características de este estándar se resumen en la siguiente figura.

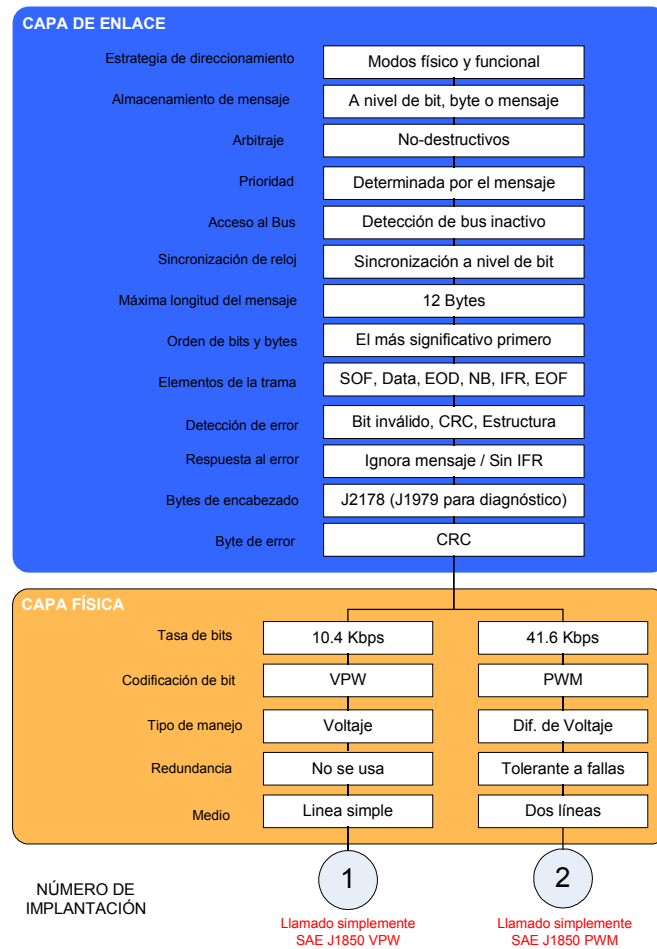


Figura 1. Mapa de SAE J1850 en el modelo OSI.

Fuente Elaboración propia

2.1 CAPA FÍSICA DE SAE J1850 (PWM y VPW).

La topología de conexión de los ECU y el dispositivo de diagnóstico de SAE J1850 es en “bus”, como se puede observar en la figura 2.

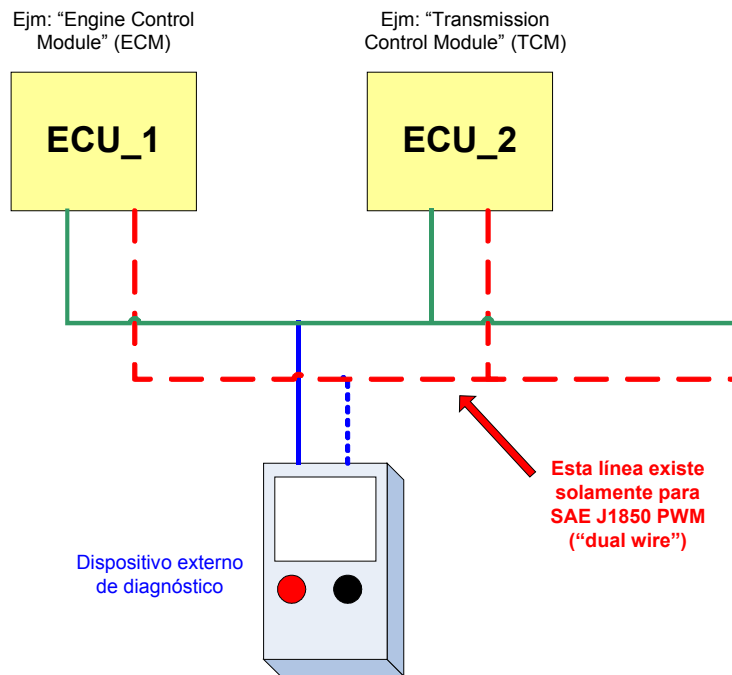


Figura 2. Topología de la red J1850. (Elaboración propia)

En la figura anterior se observa que pueden existir una gran cantidad de unidades de control electrónico (ECU) conectadas en un mismo bus. Nótese que el equipo de diagnóstico o “tester” se conecta en el mismo bus como un miembro más. Si el tipo de codificación es VPW se usa una sola línea (“single wire”) en dos direcciones (*bidirectional*); en este caso la señal se referencia a tierra. Si el tipo de codificación es PWM, se emplean dos líneas para determinar los niveles altos y bajos de la señal, en base a la diferencia de voltaje entre ambas líneas (“dual wire”).

Como ya se ha mencionado existen dos tipos de modulación diferentes especificadas en el estándar J1850:

- “Pulse Width Modulation” (PWM): Consiste en una modulación por ancho de pulso. Tanto para PWM como para VPW, existen los símbolos: “1” lógico, el “0” lógico, el “Start of Frame” (SOF), el “End of Data” (EOD), el “End of

Frame” (EOF) y el “Inter-Frame Separation” (IFS). En VPW existe además el “Normalization Bit” (NB). Cada uno de estos símbolos se representa con una forma de señal diferente.

En PWM se definen varios tiempos: TP1, TP2, TP3 ... TP9. Los valores de estos tiempos se presentan en la tabla 1. Se puede observar que estos valores están delimitados en distintos rangos para el transmisor y el receptor (uno subconjunto del otro). Esto se hace para que el receptor pueda interpretar correctamente los símbolos considerando la respuesta en frecuencia del bus.

Ancho de los pulsos en PWM (µsec)

Símbolo	Tx,min	Tx,nom	Tx, max	Rx,min	Rx,max
Tp1: Fase activa "1"	≥ 6	7	≤ 8	≥ 4	≤ 10
Tp2: Fase activa "0"	≥ 14	15	≤ 16	≥ 12	≤ 18
Tp3: Tiempo de bit	≥ 23	24	≤ 25	≥ 21	≤ 27
Tp4: Tiempo SOF / EOD	≥ 47	48	≤ 49	≥ 42	≤ 54
Tp5: Tiempo EOF	≥ 70	72	N/A	≥ 63	N/A
Tp6: Tiempo IFS	≥ 93	96	N/A	≥ 84	N/A
Tp7: SOF activo	≥ 29	31	≤ 32	≥ 27	≤ 34
Tp8: BRK activo	≥ 37	39	≤ 41	≥ 35	≤ 43
Tp9: Tiempo BRK a IFS	≥ 116	120	N/A	≥ 105	N/A

Tabla 1. Definiciones de tiempos en SAE J1850 PWM.

La representación de cada uno de los símbolos se muestra en la figura 3. Es importante destacar que no pueden existir dos transiciones de estado

pasivo a activo en un tiempo menor a TP3 (razón por la cual se hace la aclaratoria en la representación de algunos símbolos).

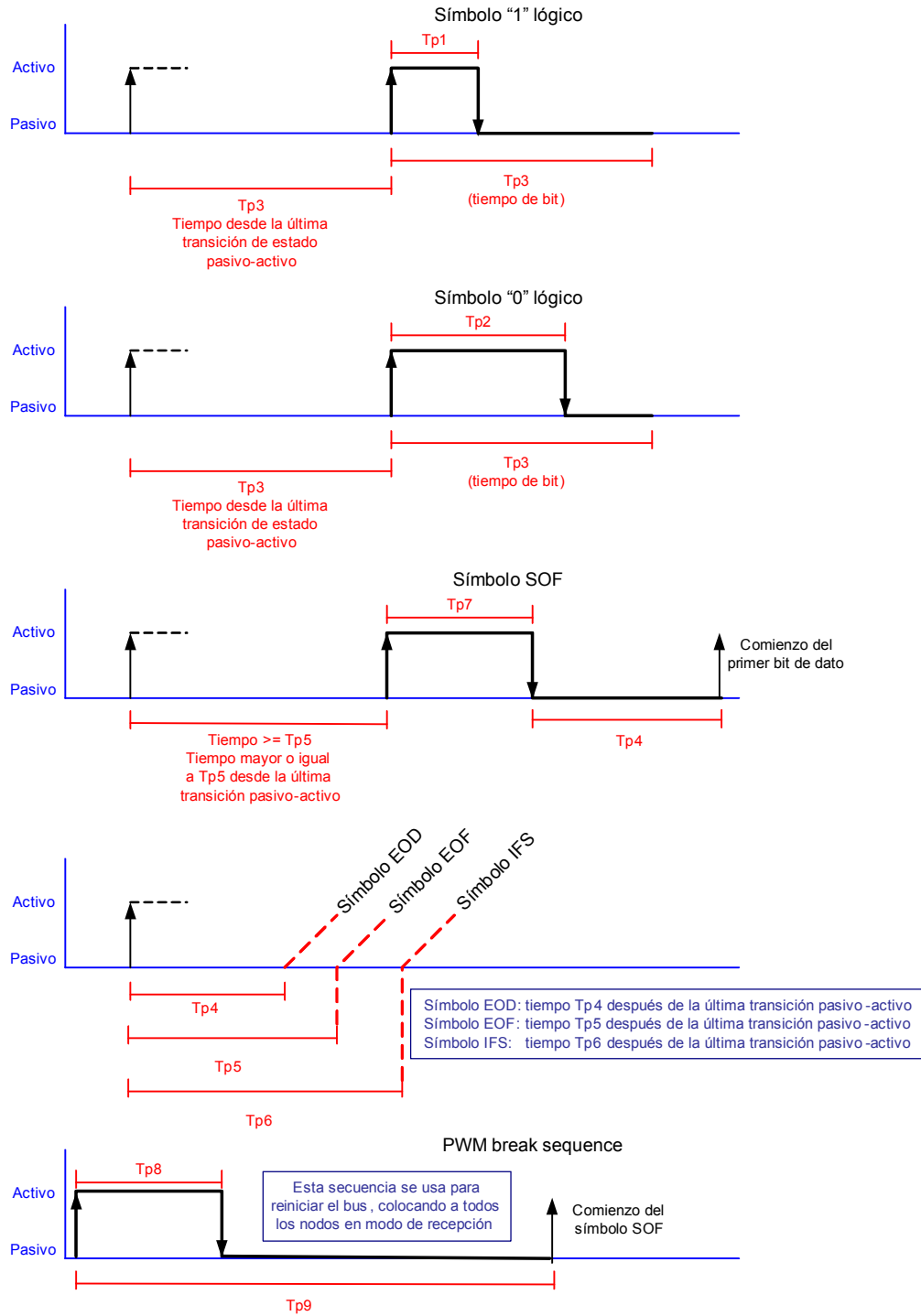


Figura 3. Símbolos SAE J1850 PWM.

La sincronización del reloj en PWM se hace en base a las transiciones de estado pasivo-activo (flancos de subida) debido a que estas transiciones son menos afectadas por la capacitancia de la línea.

- “Variable Pulse Width Modulation” (VPW): Este tipo de modulación se basa en el estado del bus para representar el siguiente símbolo. Los símbolos del “1” lógico y “0” lógico se representan con diferentes anchos de pulso, dependiendo del estado anterior del bus. Los símbolos EOD, EOF, IFS y BREAK se definen mediante el intervalo de tiempo que ha pasado desde la última transición (de manera análoga a PWM). Al igual que en PWM, se definen varios intervalos de tiempo: Tv1, Tv2,.. Tv6; estas definiciones se muestran en la tabla 2. La figura 4 muestra los diferentes símbolos en VPW.

VPW Pulse Width Times (μsec)

Symbol	Tx,min	Tx,nom	Tx, max	Rx,min	Rx,max
Tv1: Short Pulse	≥ 49	64	≤ 79	> 34	≤ 96
Tv2: Long Pulse	≥ 112	128	≤ 145	> 96	≤ 163
Tv3: SOF / EOD time	≥ 182	200	≤ 218	> 163	≤ 239
Tv4: EOF time	≥ 261	280	N/A	> 239	N/A
Tv5: BRK time	≥ 280	300	$\leq 5 \text{ msec}$	> 239	$\leq 1.0 \text{ sec}$
Tv6: IFS time	≥ 280	300	N/A	> 280	N/A

Tabla 2. Definiciones de tiempos en SAE J1850 VPW. (fuente: SAE J1850)

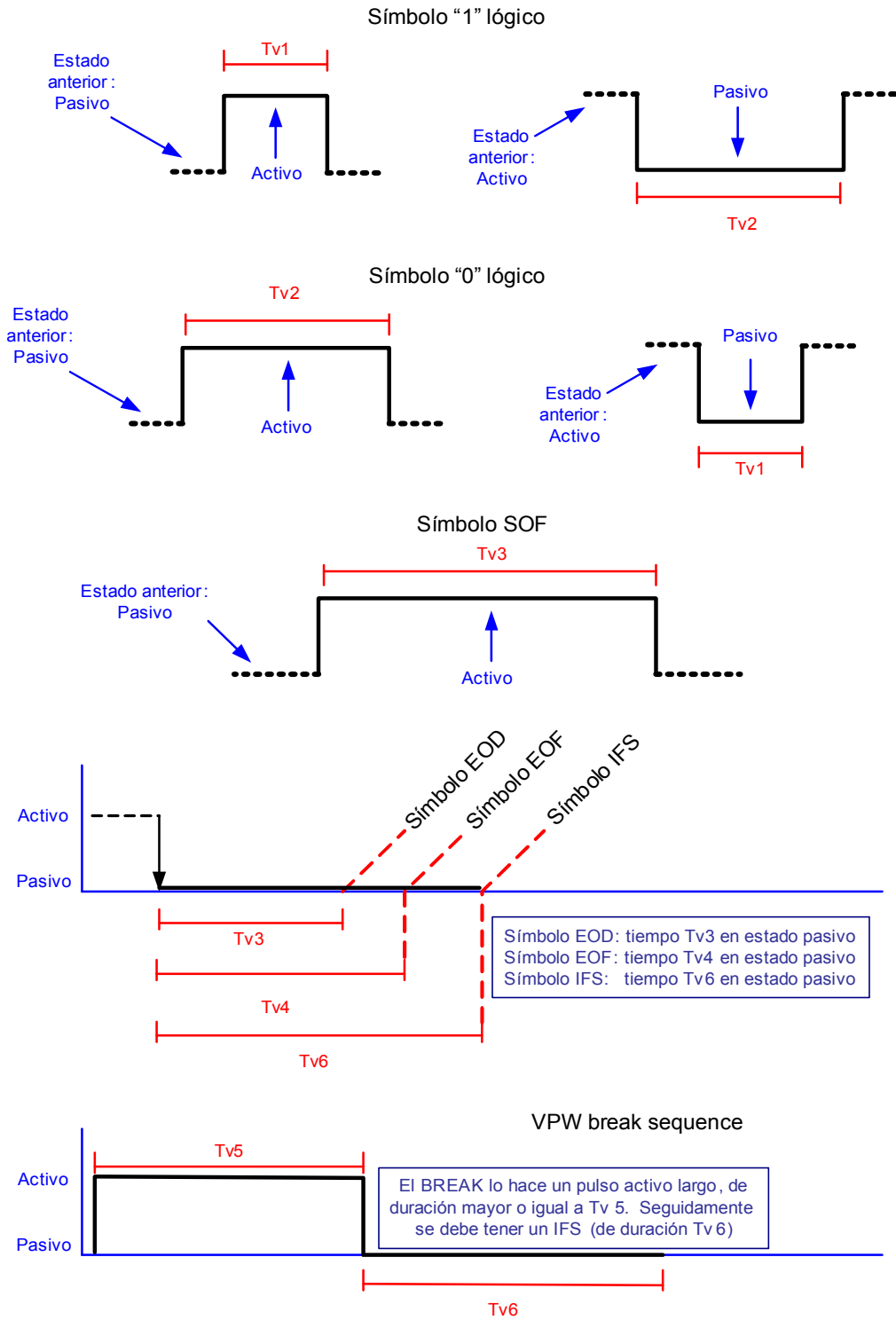


Figura 4. Símbolos SAE J1850 VPW. (Elaboración propia)

Las tasas de bits que se manejan en cada uno de los sistemas del estándar SAE J1850 son diferentes: 41,6 Kbps para PWM y 10,4 Kbps para VPW. Estos valores se obtienen directamente invirtiendo el tiempo de duración de un bit para cada esquema de codificación. Por lo tanto, para el sistema PWM:

Tiempo de bit (0 ó 1) = 24 microsegundos.

Tasa de bits = $1 / (24 \text{ microsegundos}) = 41,6 \text{ Kbps}$

Para el sistema VPW:

Tiempo de bit (0 ó 1) = 64 ó 128 microsegundos. Promedio = 96 microsegundos.

Tasa de bits = $1 / (96 \text{ microsegundos}) = 10,4 \text{ Kbps}$

Como se puede observar las tasas de bits mostradas anteriormente corresponden a velocidades de acceso, no a velocidades efectivas de datos. Las velocidades efectivas de transferencia de datos de información serán analizadas más adelante.

Las diferentes ECU conectadas al bus pueden estar en tres estados: “unpowered”, “sleeping” o “Awake”. Un ECU que está en estado “unpowered” se encuentra completamente apagado; en este estado, el ECU apagado debe permitir que la comunicación entre las demás ECU se realice de forma normal. Un ECU en estado “sleeping” se encuentra en modo de bajo consumo. Este ECU debe reconocer el cambio de la señal de pasivo a activo como indicador para pasar a estado “Awake”. Un ECU en estado “Awake” es capaz de recibir y enviar mensajes; en otras palabras, el ECU está completamente operativo. Las condiciones para que un ECU en estado operativo pase a estado “sleeping” no se especifican en SAE J1850 ya que depende de la aplicación (este tiempo es definido por el fabricante).

Otra característica de la capa física que se debe resaltar es la tolerancia a fallas. Todo ECU, en caso de falla de energía, debe garantizar que mantendrá las especificaciones de voltajes y corriente eléctrica de la red. Todo ECU debe soportar que el bus sea conectado a tierra o al voltaje de la batería. Además, en caso de que el ECU pierda conexión a tierra, debe garantizar que los otros ECU puedan seguir comunicándose entre sí a través del bus.

A continuación se presentan las características eléctricas de la interfaz SAE J1850 en las tablas 3 y 4. Los subíndices “i” y “o” indican entrada y salida respectivamente; los subíndices “l” y “h” indican bajo y alto respectivamente (se refiere a estado pasivo y activo). Por lo tanto V_{ih} indica el rango de voltajes en el cual el receptor debe interpretar que el bus está en estado activo; V_{ol} indica el rango de voltajes que debe transmitir un ECU para indicar un estado pasivo; de forma análoga se pueden interpretar los otros dos valores de voltaje. Además se presentan otros datos como resistencias y capacitancias del bus y de cada nodo (ECU). Para mayores detalles sobre las formas de onda para ambas codificaciones y tiempos de transición consúltese el documento SAE J1850.

Parámetros DC para interfaz PWM

Parámetro	Símbolo	Min	Típi.	Max	Unidad
Voltaje alto de entrada	V_{ih}	2.80	---	6.25	voltios
Voltaje bajo de entrada	V_{il}	-1.00	---	2.20	voltios
Voltaje alto de salida	V_{oh}	3.80	---	5.25	voltios
Voltaje bajo de salida	V_{ol}	0.00	---	1.20	voltios
Nivel absoluto de Voltaje de tierra	V_{go}	0.00	---	1.00	voltios
Voltaje entre los terminales Bus (+) y Bus (-)	V_{sup}	4.75	5.00	5.25	voltios
Histéresis en el receptor	V_{hys}	---	---	180	mvolts

Resistencia de la red (cada línea)	R_{load}	85	---	378	ohms
Capacitancia de la red (cada línea)	C_{load}	500	---	15,000	PF
Constante de tiempo de la red	T_{load}	---	---	1.35	μ sec
Tiempo de transición de la señal	T_t	---	---	1.75	μ sec
Resistencia de los nodos (carga unitaria, cada línea)	R_{ul}	---	2,88 0	---	ohms
Capacitancia de los nodos (carga unitaria, cada línea a tierra)	C_{ulg}	---	250	---	pF
Capacitancia de los nodos (carga unitaria, línea a línea)	C_{ulw}	---	10	---	pF
Corriente de fuga en los nodos (cada línea, estado activo)	I_{leakA}	---	---	100	μ Amp
Corriente de fuga en los nodos (cada línea, estado pasivo, nodo apagado)	I_{leakPU}	---	---	100	μ Amp
(cada línea, estado pasivo, nodo encendido)	I_{leakPP}	---	---	20	μ Amp

Tabla 3. Parámetros DC de la interfaz PWM

Parámetros DC para interfaz VPW

Parámetro	Símbolo	Min	Típi.	Max	Unidad
Voltaje alto de entrada	V_{ih}	4.25	---	20.00	volts
Voltaje bajo de entrada	V_{il}	---	---	3.50	volts
Voltaje alto de salida	V_{oh}	6.25	---	8.00	volts
Voltaje bajo de salida	V_{ol}	0.00	---	1.50	volts
Nivel absoluto Voltaje de tierra	V_{go}	0.00	---	2.00	volts
Resistencia de la red	R_{load}	315	---	1,575	ohms
Capacitancia de la red	C_{load}	2,47 0	---	16,544	pF
Constante de tiempo de la red	T_{load}	---	---	5.2	μ sec
Tiempo de transición de la señal	T_t	---	---	18.0	μ sec
Resistencia del nodo (carga unitaria)	R_{ul}	---	10,600	---	ohms
Capacitancia del nodo (carga unitaria)	C_{ul}	---	470	---	pF
Corriente de fuga del nodo	I_{leak}	---	---	10	μ Amp

Tabla 4. Parámetros DC de la modulación VPW

2.2 CAPA DE ENLACE DE SAE J1850

Uno de los principales aspectos que definen la capa de enlace es el esquema de acceso al bus mediante el mecanismo “Carrier Sense Multiple Access” (CSMA), en el cual cada nodo verifica que el bus se encuentre libre (en estado “idle”) antes de transmitir. Este método no evita que existan colisiones cuando dos o más nodos

intentan transmitir al mismo tiempo. Para la detección de colisiones, cuando un nodo se encuentra transmitiendo debe censar simultáneamente la línea y detener la transmisión si existen diferencias entre lo que está transmitiendo y lo que está “escuchando”.

La característica de **detección de colisión** (indicada en la figura 1) va directamente relacionada con lo que se denomina “bit-by-bit arbitration”. Supóngase el caso en el que dos ECU se encuentren transmitiendo simultáneamente. La detección de la colisión ocurre cuando uno de los ECU transmite un símbolo pasivo y detecta que el bus está en estado activo. Véase figura 5.

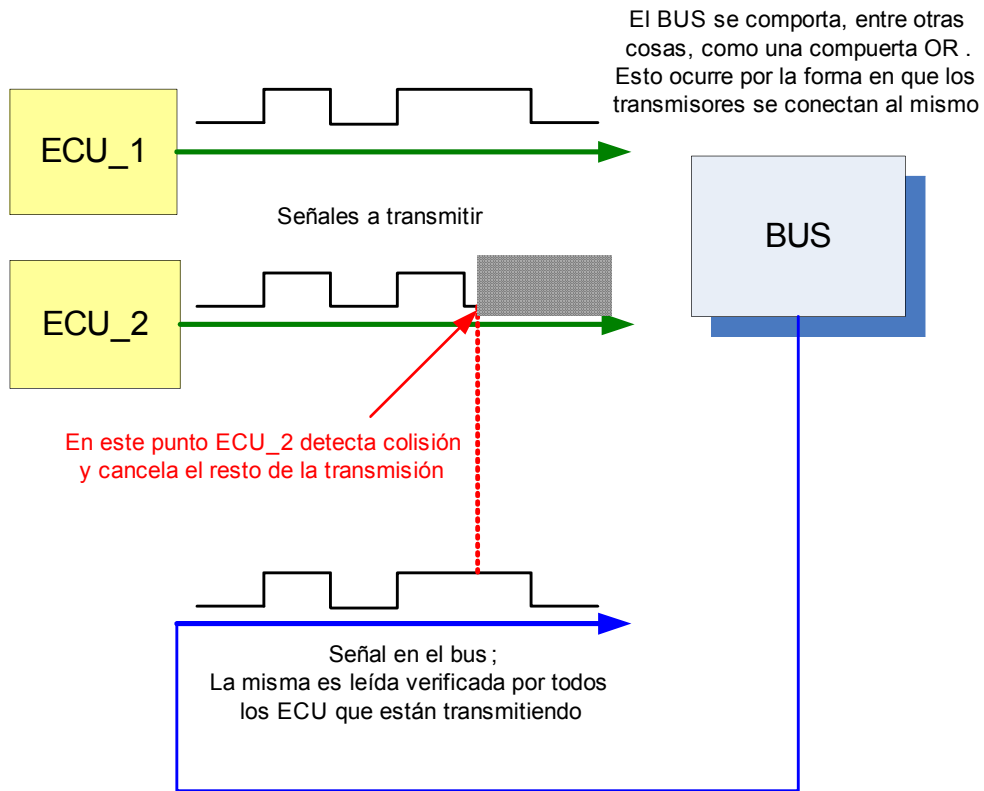


Figura 5. Ejemplo de colisión entre dos ECU. Detección de colisión.

(Elaboración propia)

Se puede observar en la figura anterior que el ECU_2 detecta la colisión y cancela la transmisión. En cambio, ECU_1 no se percata de la situación y transmite su mensaje de forma exitosa; esto ocurre debido a que ECU_1 transmite un símbolo activo y ECU_2 transmite un símbolo pasivo simultáneamente. A la predominancia de un símbolo activo sobre un símbolo pasivo se le denomina “bit-by bit arbitration” o arbitraje bit a bit. El arbitraje muchas veces se cataloga como una característica de la capa física; ciertamente, esta característica se origina por la forma como los transmisores y receptores se conectan al bus. Sin embargo, el “bit-by-bit arbitration” permite que se establezcan prioridad a los mensajes, función que indiscutiblemente pertenece a la capa de enlace.

Para entender la **prioridad de los mensajes** es necesario conocer la estructura de los mismos. Un mensaje básico (sin repuesta, tipo 0) comienza con el símbolo SOF, luego lo sigue el encabezado, los bytes de datos, el “Checksum” (CRC) y finalmente el símbolo EOF. También existen mensajes tipo 1, 2 y 3 que poseen el campo IFR (“In Frame Response”) y su CRC en caso de que el mensaje sea tipo 3. Véase figura 6

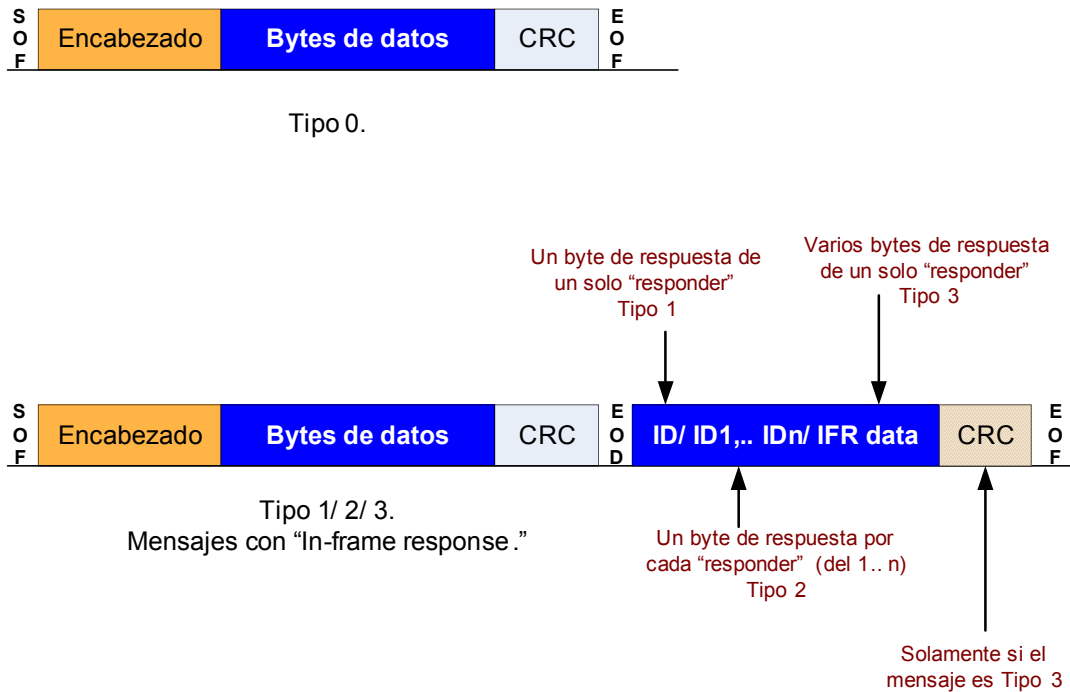


Figura 6. Tipos de "In-frame response".
(Elaboración propia)

Según las estructuras de las tramas mostradas anteriormente, los mensajes pueden ser priorizados comenzando por el encabezado y seguido por los bytes de datos. Un mensaje es de mayor prioridad que otro si, según el orden de sus símbolos en el encabezado, posee un estado activo cuando el otro mensaje está en estado pasivo. Si ambos mensajes tienen el mismo encabezado, la prioridad será definida por los bytes de datos.

El **encabezado de los mensajes** puede ser de uno (1) o tres (3) bytes. La descripción del encabezado y el campo de datos se encuentran en la norma SAE J2178, la cual consta de 4 partes. En esta norma se contempla un campo de 3 bits de prioridad para los encabezados de 3 bytes (los primeros tres bits que se envían en el bus), con lo cual se tienen 8 niveles de prioridad diferente (del 0 al 7). Véase figura 7. Los valores específicos que deben tener el encabezado y los bytes de datos de los mensajes de diagnóstico no se encuentran especificados ni en SAE J1850 ni en SAE

J2178, sino en SAE J1979. Los mensajes de diagnóstico deben ser encapsulados en mensajes de capa 2 de tres bytes de encabezado. Los valores que debe tener cada campo de los mensajes de diagnóstico serán mostrados al final de esta sección.

Encabezado de tres Bytes

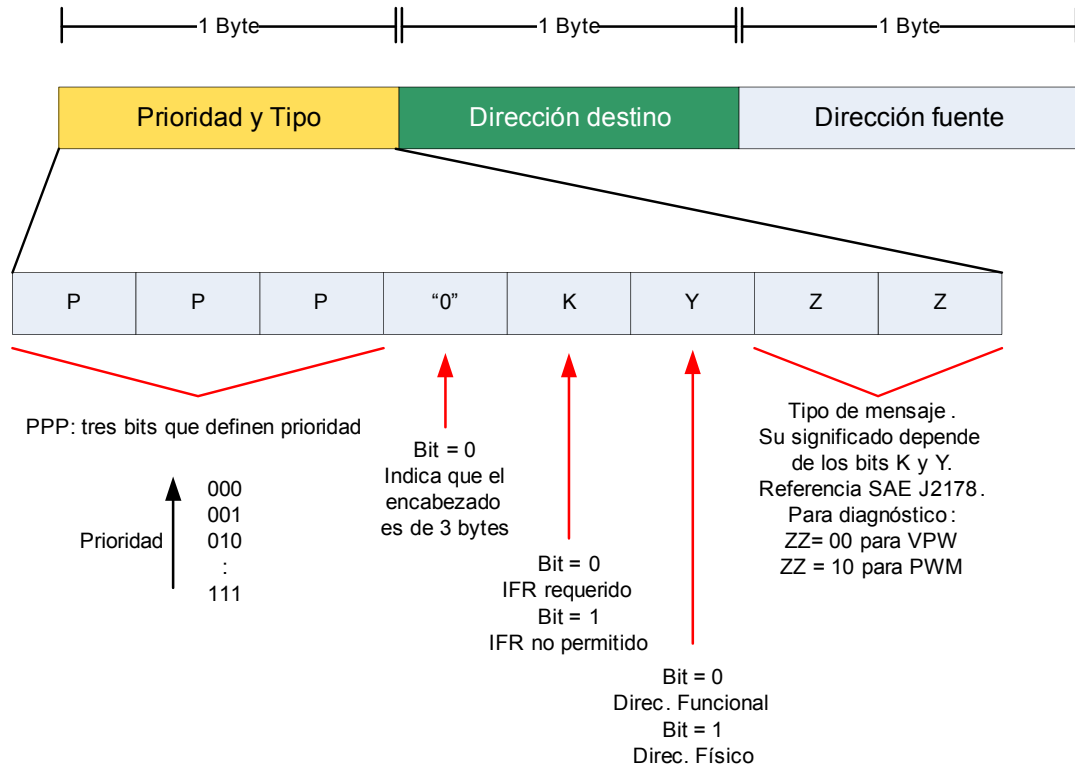


Figura 7. Estructura encabezado de tres bytes en SAE J1850 y SAE J2178.
(Elaboración propia)

El protocolo SAE J1850 define dos (2) **estrategias de direccionamiento** diferentes que pueden coexistir. El direccionamiento físico se basa en el intercambio de tramas entre dos (2) nodos en base a la “dirección física” de los mismos; esto quiere decir que cada ECU o nodo conectado al bus posee una dirección física única. El direccionamiento físico establece una comunicación entre dos nodos específicos: el que envía el mensaje y el que responde. Por otra parte, el “direccionamiento por función” sirve para establecer comunicación entre varios nodos. A cada ECU se le

asigna un número de funciones. Los mensajes van dirigidos en base a la función del ECU, por lo cual todos los ECU que tengan la misma función deberán responder.

La máxima **longitud de los mensajes** definida en SAE J1850 es de doce (12) Bytes para ambos tipos de codificación de línea (PWM y VPW). Debido a que los mensajes de diagnóstico utilizan un encabezado de tres (3) Bytes y un (1) Byte de CRC, máxima longitud disponible para datos es de ocho (8) Bytes. La **estructura de los mensajes diagnóstico**, a nivel de capa de enlace, puede observarse en la figura 8. Los valores expresados en hexadecimal fueron preestablecidos en SAE J1979 para las funciones de diagnóstico.

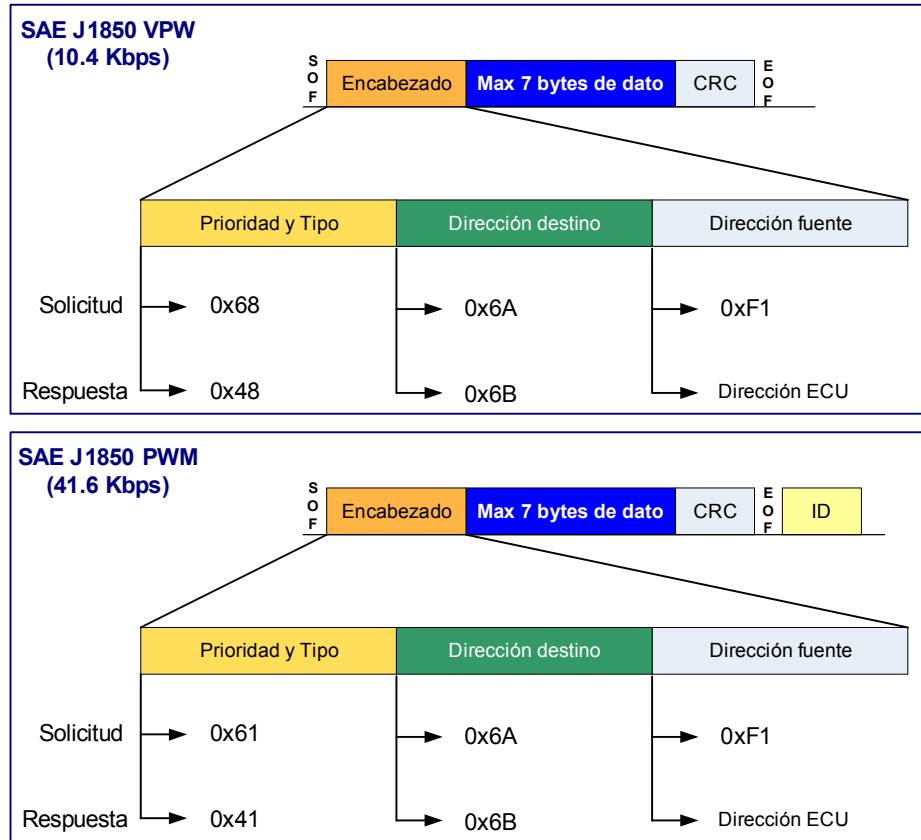


Figura 8. Mensajes de diagnóstico: solicitud y respuesta de parámetros. Capa 2.

2.3 CAPA FÍSICA Y DE ENLACE DE ISO 9141.

El protocolo ISO 9141 (y su segunda revisión ISO 9141-2) es mucho más sencillo en comparación con SAE J1850. Este protocolo es orientado a byte en la capa de enlace (muy parecido al protocolo RS-232), a diferencia del SAE J1850 que está orientado a mensajes con símbolos de encabezado, datos y CRC. Los mensajes de diagnóstico se forman en la capa de aplicación (SAE J1979) y son enviados byte a byte por la capa de enlace.

La arquitectura de la red ISO 9141 se muestra en la figura 9. Las líneas K y L son independientes y referenciadas a tierra. La línea K trabaja en dos direcciones (*bidirectional*) y se usa para transmitir los mensajes de solicitud y respuesta del estado de los parámetros; la línea L es unidireccional y solo se usa en el proceso de inicialización. A diferencia de SAE J1850, la interfaz ISO 9141-2 solo se usa para propósitos de diagnóstico, aunque también puede ser usada para otras comunicaciones (inclusive con cualquier otro protocolo) cuando no se encuentra conectado el “tester”.

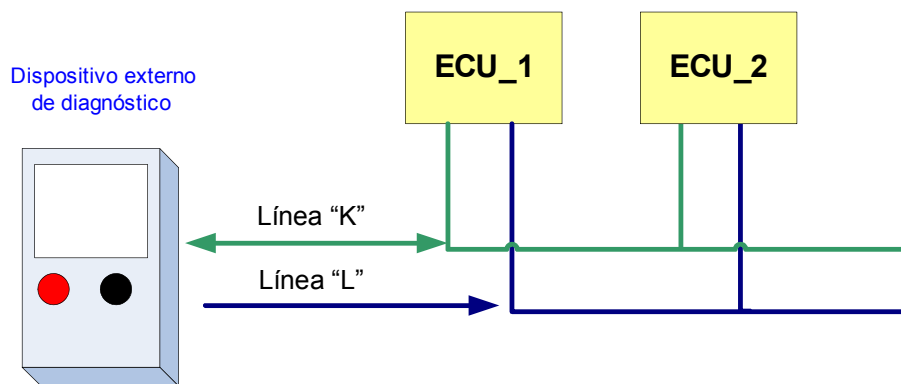


Figura 9. Arquitectura ISO 9141.

En la siguiente figura se muestra un esquema de conexión entre la ECU y el equipo de diagnóstico, con los valores de las resistencias especificados en el estándar.

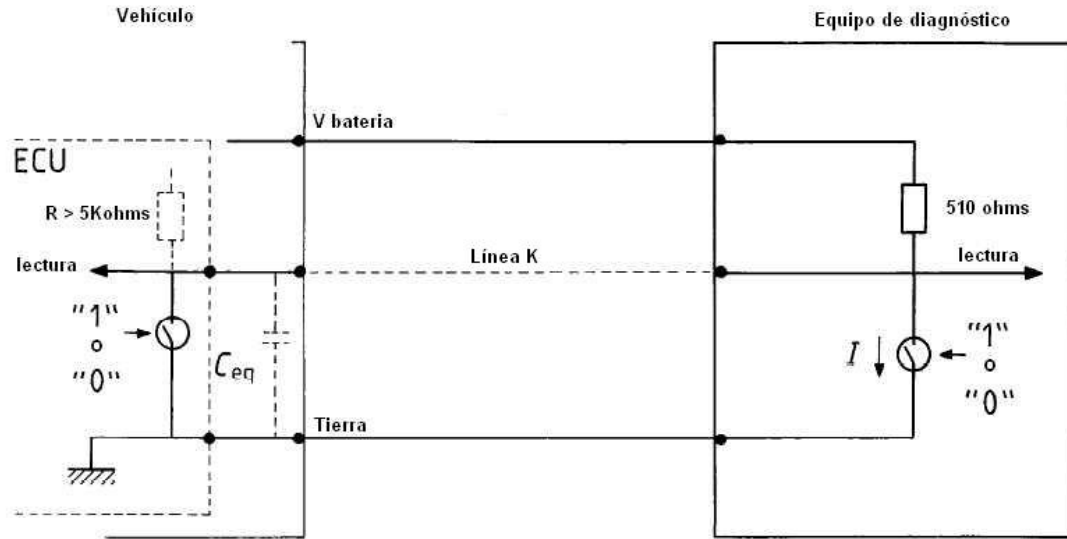


Figura 10. Esquema de conexión del ECU y el equipo de diagnóstico.

En ISO 9141 sólo existe dos símbolos: el “0” y el “1” y la codificación es NRZ. El “1” lógico se representa con un nivel igual al voltaje positivo de la batería (V_B , 12V ó 24 V) y el “0” lógico con el nivel de tierra. La figura 11 muestra el rango de valores que deben tener los símbolos transmitidos para ser interpretados como ceros o unos en el receptor. El rango denominado V_{oh} (max y min) muestra el nivel de la señal (voltaje) a la salida del transmisor para representar el símbolo “1” lógico; V_{il} indica el rango de voltajes en el cual el receptor debe interpretar que se ha recibido un “0”. De manera análoga se interpretan los otros valores.

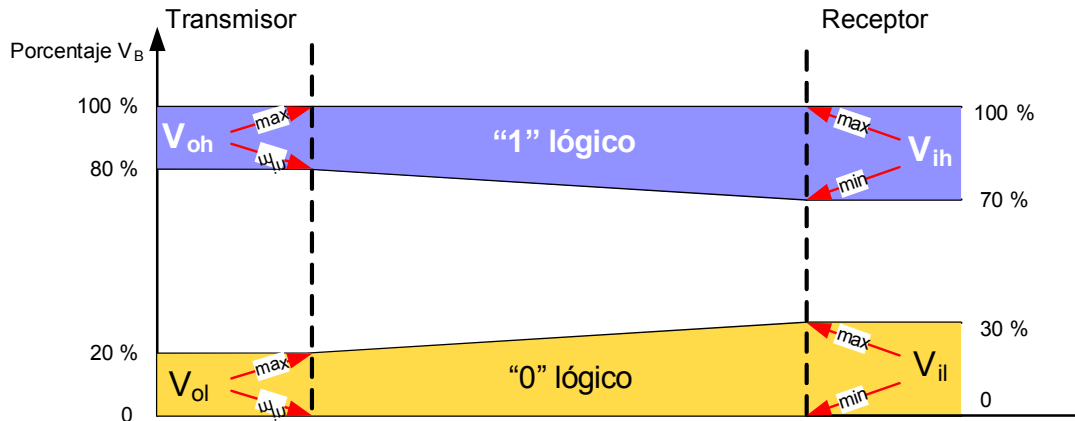


Figura 11. Niveles de voltaje y sus respectivos valores lógicos.

En ISO 9141 se permite un rango muy amplio de tasas de bits: 10 baudios a 10.000 baudios. Sin embargo, en la segunda revisión del estándar (ISO 9141-2) se fija la tasa de bits en 10.4 Kbps, originando un tiempo de bit de 96,15 microsegundos. El tiempo máximo de transición entre dos niveles lógicos (del 20% al 80% de V_B y viceversa) no debe exceder el 10% del tiempo de bit.

Las tramas a nivel de capa 2 son muy parecidas a las usadas en el protocolo RS-232 y están formadas por un bit de inicio ("start"), 8 bits de datos y por lo menos un bit de parada ("stop"). Cuando el bus está inactivo, el estado de las líneas es "K" y "L" es "1" lógico, por lo cual el bit de "start" debe ser un "0" y el bit de "stop" un "1". Los bits de datos se envían comenzando con el menos significativo. Véase figura 12.

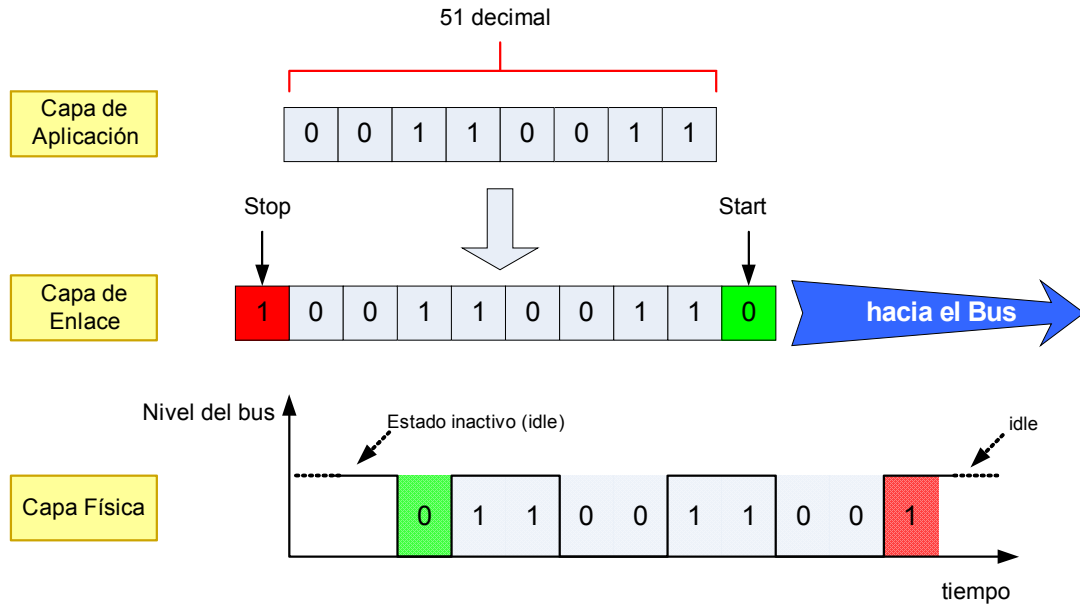


Figura 12. Ejemplo de trama ISO 9141. Orden de Envío de bits.
(Elaboración propia)

A diferencia de SAE J1850, en el cual todos las ECU se comunican entre sí a través del mismo protocolo, el protocolo ISO 9141 solo se usa para la comunicación entre el dispositivo de diagnóstico y las ECU. En SAE J1850, los mensajes de diagnóstico son un tipo de mensaje mas, por lo cual pueden ser enviados en cualquier momento, siempre que el bus esté libre; en ISO 9141 la comunicación debe ser inicializada antes de poder llevarse a cabo. El protocolo de inicialización es el siguiente:

- f) El equipo de diagnóstico debe verificar que el bus se encuentre en estado "idle" durante un tiempo mínimo de 2ms y luego debe enviar un código de inicialización a una tasa de 5 bps sobre las líneas K y L simultáneamente. El código es "51" (decimal), con sus respectivos bits de "start" y "stop". Luego la línea L es liberada.
- g) Solo la ECU encargada de enviar las respuestas de diagnóstico contesta enviando un patrón de sincronización (01010101 binario). El tiempo que

puede tardar el ECU en responder está comprendido entre 2ms y 2s. El objetivo de este mensaje es que el dispositivo de diagnóstico se pueda sincronizar a la tasa de bits del ECU. En ISO 9141-2, la tasa de bits se fijó a 10.4 Kbps, pero el byte de sincronización se sigue enviando.

- h) Después de un tiempo comprendido entre 2ms y 1.2s, el ECU envía un mensaje que contiene una palabra clave (“key word”) de 7 bits. El octavo bit de dato (el más significativo) funciona como bit de paridad, donde se busca que el byte posea siempre una cantidad impar de “1”. Luego, después de un tiempo comprendido entre 0.2ms y 1.2 s, se envía la segunda palabra clave. En total, el código clave consta de catorce (14) bits. Si las palabras claves son 0x08 y 0x08, esto indica que pueden existir uno o más ECU en la red; si las palabras claves son 0x94 y 0x94, se conoce que solo existe un ECU en la red de diagnóstico. Estos valores tienen impacto en los tiempos entre mensajes de solicitud y respuesta definidos en SAE J1979.
- i) El dispositivo de diagnóstico confirma enviando el inverso de la segunda palabra clave.
- j) Finalmente el ECU envía el inverso del código de inicialización.

De esta manera la interfaz de comunicación queda inicializada y lista para el intercambio de mensajes de diagnóstico. Si el bus permanece inactivo (“idle”) durante mas de 5 segundos, se asume que la comunicación ha finalizado. Si se deseara reanudar la comunicación se tendría que repetir todo el proceso de inicialización.

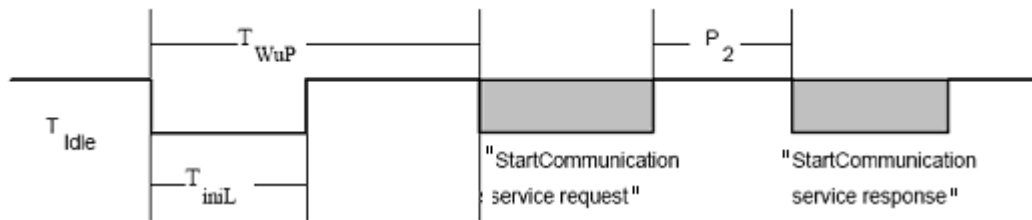
Los mensajes de aplicación (diagnóstico) son divididos y enviados byte a byte sobre ISO 9141. Los principales mensajes de diagnóstico se encuentran especificados en SAE J1979. Se puede observar que SAE J1850 e ISO 9141 son completamente diferentes. La estructura de los mensajes de diagnóstico está adecuada a SAE J1850, ya que ISO 9141 es mucho más flexible y permite soportar cualquier estructura de mensajes de aplicación. Debido a las diferencias

entre SAE J1850 e ISO 9141, SAE J1979 especifica diferentes tiempo de retardo entre mensajes de pregunta y respuesta para cada uno de los protocolos.

2.4 CAPA FÍSICA Y DE ENLACE DE ISO 14230 (KWP2000)

El protocolo KWP2000 surgió como una mejora a ISO 9141-2, por lo cual se expondrán solo las diferencias entre ambos estándares. La capa física de KWP2000 es muy similar a la ISO 9141, con la diferencia que se especifican valores diferentes de resistencias y capacitancias para vehículos de 24V. La línea “L” usada en ISO 9141 para la inicialización es prescindible en KWP2000, por lo cual toda la inicialización y posterior comunicación puede realizarse por la misma línea “K”.

La codificación en KWP2000 es NRZ y la velocidad es de 10.4 Kbps, al igual que en ISO 9141-2. El envío se hace byte a byte, pero en KWP2000 existe una estructura ya predeterminada para los datos, a diferencia de ISO 9141-2 que se adapta al protocolo de aplicación. Otra característica importante es que KWP2000 tiene una inicialización rápida de la comunicación, a diferencia de ISO 9141. La secuencia de inicialización de KWP2000 y la estructura de los mensajes se pueden ver en las figuras 13 y 14 respectivamente.



Valores de los tiempos en la inicialización rápida

		min	max
T_{iniL}	25 ± 1 ms	24 ms	26 ms
T_{WuP}	50 ± 1 ms	49 ms	51 ms

Figura 13. Secuencia de inicialización en KWP2000.

Fuente: SSF 14230-2

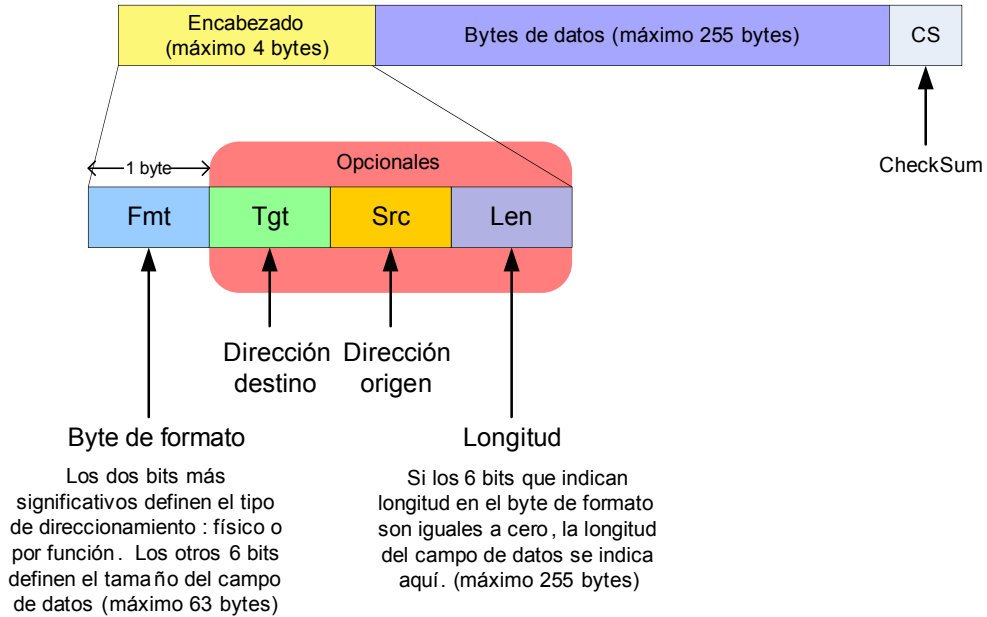


Figura 14. Formato de trama en ISO 14230- capa 2 o KWP2000

En la figura 13 se puede observar que la secuencia de inicialización de KWP2000 es diferente a la de ISO 9141. La secuencia de inicialización consiste en un pulso de valor “0” de corta duración, seguido de un mensaje denominado “Start communication service Request”. Este mensaje tiene un encabezado de tres bytes, un solo byte de dato y el Checksum. La estructura de este mensaje está definida en ISO 14230-2. Los valores para una inicialización por parte del “tester” deben ser los siguientes: 0xC1 (Fmt: indica direccionamiento por función y campo de datos de un solo byte de longitud), 0x33 (destino), 0xF1 (origen: tester), 0x81 (Service ID) y el respectivo Checksum. Los valores mostrados anteriormente se obtuvieron de los estándares ISO 14230-2 y 14230-3 y SAE J1979.

Sobre el formato de la trama de la figura 14, se soportan un gran número de servicios definidos en ISO 14230-2, 14230-3 y 14230-4. Por lo tanto, KWP2000 no era compatible a nivel de aplicación con el resto de los protocolos OBDII que se regían bajo SAE J1979 a ese nivel. Por lo tanto, la revisión de abril 2002 de SAE J1979 establece formas de soportar SAE J1979 sobre KWP2000 y sobre el nuevo protocolo ISO 15765-4 (CAN), con el fin de hacer converger todos los protocolos en la capa de aplicación. Al analizarse esto se puede observar que KWP2000 queda prácticamente igual a ISO 9141-2, diferenciándose, prácticamente, en la forma de inicialización y en el requerimiento de respuestas negativas a los servicios (y no omisión como en ISO 9141-2).

Apéndice 3. Parámetros definidos por SAEJ1979 para la monitorización de vehículos

De acuerdo al estándar SAE J1979, se determinaron los parámetros disponibles para la monitorización de vehículos, los cuales se listan en la tabla que se muestra a continuación. Esta tabla contiene el PID correspondiente a cada parámetro, una breve descripción de la información que aporta cada uno de ellos; el número de bytes que ocupa la medición relativa a cada parámetro y los valores máximo y mínimo de dicha medición, los cuales están acompañados de las unidades métricas correspondientes a cada uno de los parámetros.

PID	Parámetro	Descripción	Número de bytes de la respuesta	Valor máximo	Valor mínimo
03	Fuel System Status	Byte A:Estado del sistema de combustible 1. Byte B:Estado del sistema de combustible 2. Este estado indica si el sistema de combustible está abierto o cerrado. El sistema de combustible está abierto al arrancar el auto. Cuando los sensores de oxígeno se calientan y se reportan estables, el sistema de combustible se cierra.	2	No aplica	No aplica
04	Calculated Load Value	Valor calculado de Carga: brinda al mecánico un indicador del porcentaje de la	1	0%	100%

		capacidad del motor			
05	Engine Coolant Temperature	Temperatura del refrigerante	1	-40 °C	+215 °C
06	Short Term Fuel Trim (Bank 1)	Byte A: Entrega de combustible a corto plazo Banco 1. Byte B: Entrega de combustible a corto plazo Banco 3	2	-100 %	99.22%
07	Long Term Fuel Trim (Bank1)	Byte A: Entrega de combustible a largo plazo Banco 1. Byte B: Entrega de combustible a largo plazo Banco 3	2	-100 %	99.22%
08	Short Term Fuel Trim (Bank 2)	Byte A: Entrega de combustible a corto plazo Banco 2. Byte B: Entrega de combustible a corto plazo Banco 4	2	-100 %	99.22%
09	Long Term Fuel Trim (Bank2)	Byte A: Entrega de combustible a largo plazo Banco 2. Byte B: Entrega de combustible a largo plazo Banco 4	2	-100 %	399.22%
0A	Fuel Rail Pressure	Presión del sensor de combustible referenciada a la atmósfera	1	0 kPa	765 kPa
0B	Intake Manifold Absolute Pressure	Medida de la presión del aire de entrada	1	0 kPa	255 kPa
0C	Engine RPM	Revoluciones por minuto del motor	2	0 min ⁻¹	16383.75 min ⁻¹

0D	Vehicle Speed Sensor	Velocidad del vehículo	1	0 km/h	255 km/h
0E	Ignition Timing Advance for #1 cylinder	Avance de la sincronización de ignición para el cilindro #1	1	-64 °	63.5 °
0F	Intake Air Temperature	Indica la temperatura ambiente	1	-40 °C	+215 °C
10	Air Flow Rate from Mass Air Flow Sensor	Razón Aire-combustible	2	0 g/s	655.35 g/s
11	Absolute throttle position	Posición absoluta de la mariposa verifica el valor de la posición de la mariposa, la cual es una válvula que determina la cantidad de aire que entra al motor, para que la computadora del mismo responda rápidamente a cambios, incrementando o decrementando la cantidad de combustible según sea necesario.	1	0%	100%
12	Commanded Secondary air Status	Estado secundario del aire del comando	1	No aplica	No aplica
13	Location of Oxygen Sensors	Ubicación de los sensores de Oxígeno	1	No aplica	No aplica
14	Bank 1 Sensor 1	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B -100.00 %	Byte A 1.275 V Byte B 99.22 %

15	Bank 1 Sensor 2	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B –100.00 %	Byte A 1.275 V Byte B 99.22 %
16	Bank 1 Sensor 3	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B –100.00 %	Byte A 1.275 V Byte B 99.22 %
17	Bank 1 Sensor 4	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B –100.00 %	Byte A 1.275 V Byte B 99.22 %
18	Bank 2 Sensor 1	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B –100.00 %	Byte A 1.275 V Byte B 99.22 %
19	Bank 2 Sensor 2	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B –100.00 %	Byte A 1.275 V Byte B 99.22 %
1A	Bank 2 Sensor 3	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B –100.00 %	Byte A 1.275 V Byte B 99.22 %
1B	Bank 2 Sensor 4	Byte A: voltaje de entrada de los sensores de oxígeno. Byte B:	2	Byte A 0 V Byte B –100.00 %	Byte A 1.275 V Byte B 99.22 %
1C	OBd requirements to which vehicle is designed	Especificaciones OBd del vehículo	1	No aplica	No aplica
1D	Location of oxygen sensors	Ubicación de los sensores de oxígeno	1	No aplica	No aplica
1E	Auxiliary Input Status	Indica el estado del PTO (Power Take Off)	1	No aplica	No aplica
1F	Time Since Engine Start	Tiempo transcurrido desde el encendido del motor	2	0 sec	65,535 sec

21	Distance travelled while MIL is activated	Distancia recorrida desde la activación de un MIL	2	0 km	65535 km
22	Fuel Rail Pressure relative to manifold vacuum	Presión del carril del combustible en relación con la válvula de vacío	2	0 kPa	5177.27 kPa
23	Fuel Rail Pressure	Presión del carril del combustible	2	0 kPa	655350 kPa
24	O2 Sensor 1 Bank 1 (wide range sensor)	Sensor 1 de Oxígeno en el Banco 1. Bytes A y B: factor de equivalencia. Bytes C y D: Voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V
25	O2 Sensor 2 Bank 1 (wide range sensor)	Sensor 2 de Oxígeno en el Banco 1. Bytes A y B: factor de equivalencia. Bytes C y D: voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V
26	O2 Sensor 3 Bank 1 (wide range sensor)	Sensor 3 de Oxígeno en el Banco 1. Bytes A y B: factor de equivalencia. Bytes C y D: voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V
27	O2 Sensor 4 Bank 1 (wide range sensor)	Sensor 4 de Oxígeno en el Banco 1. Bytes A y B: factor de equivalencia. Bytes C y D: voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V
28	O2 Sensor 1 Bank 2 (wide range sensor)	Sensor 1 de Oxígeno en el Banco 2. Bytes A y B: factor de equivalencia. Bytes C y D: voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V
29	O2 Sensor 2 Bank 2 (wide range sensor)	Sensor 2 de Oxígeno en el Banco 2. Bytes A y B: factor de equivalencia. Bytes C y D: voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V

2A	O2 Sensor 3 Bank 2 (wide range sensor)	Sensor 3 de Oxígeno en el Banco 2.Bytes A y B: factor de equivalencia. Bytes C y D: voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V
2B	O2 Sensor 4 Bank 2 (wide range sensor)	Sensor 4 de Oxígeno en el Banco 2.Bytes A y B: factor de equivalencia. Bytes C y D: voltaje del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: 0 V	Bytes A y B: 1.999 Bytes C y D: 7.999 V
2C	Commanded EGR	EGR ordenado	1	0%	100%
2D	EGR Error	Error de EGR	1	-100 %	99.22%
2E	Commanded Evaporative Purge	Purgación evaporativa ordenada	1	0%	100%
2F	Fuel Level Input	Nivel de combustible	1	0%	100%
30	Number of warm-ups since diagnostic trouble codes cleared	Número de calentamientos ocurridos desde el borrado de los códigos de falla	1	0	255
31	Distance since diagnostic trouble codes cleared	Distancia recorrida desde que se desactivaron los códigos de falla	2	0 km	65,535 km
32	Evap System Vapor Pressure	Medida de la presión del vapor en el sistema de control de evaporación	2	-8192 Pa	8191 Pa
33	Barometric Pressure	Presión Atmosférica	1	0 kPa	255 kPa
34	O2 Sensor 1 Bank 1 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D:Corriente del sensor de	4	Bytes A y B: 0 Bytes C y D: - 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996

		oxígeno			mA
35	O2 Sensor 2 Bank 1 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D: Corriente del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: – 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996 mA
36	O2 Sensor 1 Bank 2 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D: Corriente del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: – 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996 mA
37	O2 Sensor 2 Bank 2 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D: Corriente del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: – 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996 mA
38	O2 Sensor 1 Bank3 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D: Corriente del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: – 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996 mA
39	O2 Sensor 2 Bank 3 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D: Corriente del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: – 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996 mA
3A	O2 Sensor 1 Bank 4 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D: Corriente del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: – 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996 mA
3B	O2 Sensor 4 Bank 4 (wide range sensor)	Bytes A y B: factor de equivalencia. Bytes C y D: Corriente del sensor de oxígeno	4	Bytes A y B: 0 Bytes C y D: – 128 mA	Bytes A y B: 1.999 Bytes C y D: 127.996 mA
3C	Catalyst Temperature B1S1	Temperatura catalítica Banco 1 Sensor 1	2	–40 °C	+6513.5 °C

3D	Catalyst Temperature B2S1	Temperatura catalítica Banco 2 Sensor 1	2	-40 °C	+6513.5 °C
3E	Catalyst Temperature B1S2	Temperatura catalítica Banco 1 Sensor 2	2	-40 °C	+6513.5 °C
3F	Catalyst Temperature B2S2	Temperatura catalítica Banco 2 Sensor 2	2	-40 °C	+6513.5 °C
42	ECU voltage	Voltaje de entrada del ECU	2	0 V	65.535 V
43	Absolute Load Value	Carga absoluta del motor	2	0%	25700%
44	Commanded Equivalence Ratio	El cociente de equivalencia indica la relación aire/combustible del motor	2	0	1.999
45	Relative Throttle Position	Posición Relativa de la mariposa	1	0%	100%
46	Ambient air Temperature	Temperatura ambiente	1	-40 °C	+215 °C
47	Absolute Throttle Position B	Posición Absoluta B de la mariposa	1	0%	100%
48	Absolute Throttle Position C	Posición Absoluta C de la mariposa	1	0%	100%
49	Accelerator Pedal Position D	Posición D del acelerador	1	0%	100%

4A	Accelerator Pedal Position E	Posición E del acelerador	1	0%	100%
4B	Accelerator Pedal Position F	Posición F del acelerador	1	0%	100%
4C	Commanded Throttle Actuator Control	Control ordenado del actuador de la válvula reguladora	1	0%	100%
4D	Minutes run by the Engine running while MIL activated	Minutos transcurrido desde la activación de un MIL	2	0 min	65535 min
4E	Time Since diagnostics codes cleared	Tiempo transcurrido desde la activación de los códigos de fallas	2	0 min	65535 min

Bibliografía:

“SAE J1979. (R) E/E Diagnostic Test Modes”, SAE, Abril 2002.

Apéndice 4. Evolución Tecnológica de los Sistemas de Telefonía Celular en Venezuela

A continuación se muestra la evolución de los sistemas de telefonía celular seguida por los operadores nacionales con mayor número de usuarios.

1. Movistar:

TELCEL BellSouth nació en 1991 como una compañía dedicada a brindar servicios de telefonía celular en Venezuela, pero poco a poco se constituyó en una empresa integral de telecomunicaciones al crear otras unidades de negocios que permiten a los clientes acceder a otras opciones como Internet y servicios de valor agregado, redes privadas de datos, centrales telefónicas, transmisión inalámbrica de datos, y localización de vehículos, entre otros.

Hoy en día Telcel es Movistar de Telefónica, en la figura 1 se observa la evolución tecnológica en sistemas celular que ha desarrollado desde su fundación.

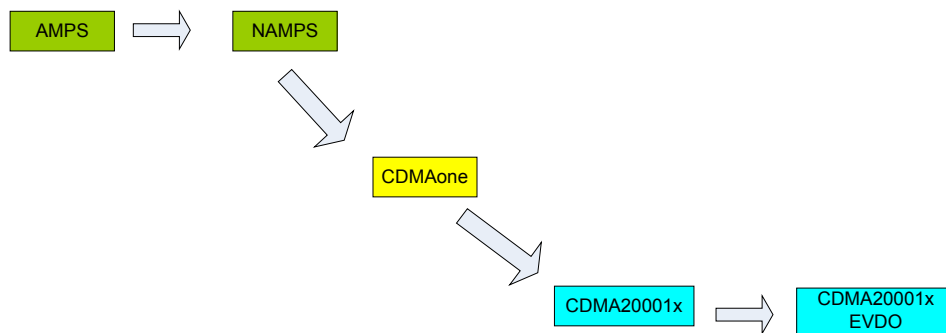


Figura 1 Evolución Tecnológica Celular en Movistar

2. Movilnet:

De acuerdo a información obtenida en la página Web oficial de Movilnet, esta compañía nace en 1992 como empresa filial de CANTV, dedicada a prestar servicios de telefonía móvil en Venezuela. En el año 1996, la empresa introduce una tecnología basada en el sistema D-AMPS y digitaliza su red a través de la incorporación de la plataforma TDMA

A finales del año 2002 toma las riendas de un nuevo proyecto tecnológico que permitió la implementación de CDMA2000 1X, la cual converge en la actualidad con la plataforma TDMA.

La red CDMA 1X ha sido desarrollada en tres fases, siendo las dos primeras ejecutadas durante los años 2001 y 2003. Esta red ha permitido ofrecer servicios de Internet móvil, entre ellos, alta velocidad de transmisión de datos de hasta 153 Kbps

Desde septiembre de 2004 se inició la tercera fase del proyecto de ampliación de la red 1X, que preveía la instalación, para el cierre del tercer trimestre 2005, de 297 nuevas celdas en distintos estados del país, de las cuales 267 radio bases ya se encuentran en operación.

Como parte del proceso de desarrollo tecnológico, en febrero de 2005 Movilnet desplegó una nueva plataforma para ofrecer servicios de datos de tercera generación. Con la puesta en marcha de esta nueva red, la operadora se convirtió en la primera empresa en implementar CDMA2000 1X-EVDO. Ver figura 2 para observar la evolución tecnológica seguida por Movilnet.

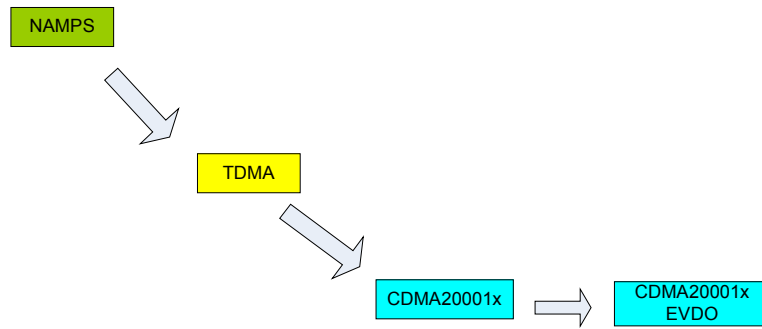


Figura 2 Evolución Tecnológica Celular en Movilnet

3. Digitel:

La historia de Digitel se inició en abril de 1997 cuando obtuvo la concesión para la región central. El 15 de Noviembre de 1998 empezó operaciones de telefonía pública en la población del Totumo, Estado Aragua.

El 24 de julio y el 7 de Septiembre iniciaron operaciones comerciales la red de telefonía rural y móvil, respectivamente. En el año 2000, Digitel construyó una nube desde estos puntos, que cubrió las ciudades Caracas y Valencia. Para finales de año se le dio cobertura a Guarenas, Guatire, Los Teques, San Antonio de los Altos, Valles del Tuy y el Litoral Central. Inicialmente se interconectó a Telcel , actualmente Movistar, y a CANTV, posteriormente a Movilnet.

La tecnología empleada por este operador es GSM e inicialmente se concentró en servicios de voz. La figura 3 muestra el camino evolutivo seguido por Digitel, de las tecnologías ahí indicadas, actualmente sólo se encuentran disponibles GSM y GPRS.

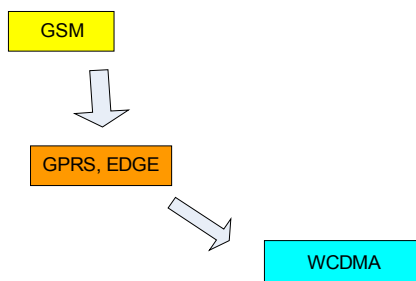


Figura 3 Evolución Tecnológica Celular en Digital

Bibliografía

- Historia de Movilnet. Fuente: <http://www.movilnet.com.ve>. (1 Dic. 2005)
- DÍAZ HUNG, Verónica. (1999). *Tendencias El año 2000 según GSM*.
<http://www.lared.com.ve/archivo/aper42-1.html>. (20 Nov. 2005)
- HUIDOBRO, José Manuel. Fundamentos de Telecomunicaciones (1ª ed.). Madrid: Paraninfo.2001

Apéndice 5. Pruebas del módem CDMA

Las pruebas del módem CDMA fueron de vital importancia para poder identificar comportamientos que no estuviesen documentados en los manuales del producto. Para ello se conectó el módem directamente a un computador (el cual simularía el comportamiento de la Unidad Remota) y se colocó un “servidor” en una computadora con acceso a Internet a través de una dirección IP pública; este servidor se encargó de esperar y aceptar la conexión TCP solicitada por el módem.

El servidor se implantó usando *Hyperterminal*, debido a que este software permite establecer conexiones TCP entre dos computadores. Se deshabilitaron todos los “firewalls” de la computadora que iba a simular el servidor y se colocó el servidor a la espera de una conexión TCP por el puerto 5050 (el número del puerto TCP puede ser cualquiera de uso libre y que, además, no se conozca por ser utilizado por virus o programas espías, ya que la red puede bloquearlos).

El computador mediante el cual se enviaron los comandos al módem (a través del puerto serial) utilizó la aplicación Docklight (también se puede usar *Hyperterminal*). Con ésta, se procedió a enviar todos los comandos AT para establecer la conexión a través del módem, especificando la dirección IP del computador remoto y el puerto TCP 5050. Este proceso fue realizado un gran número de veces durante varios días y los resultados eran inconsistentes; la conexión a veces se establecía, pero la mayor cantidad de veces no.

En vista a éste problema, se realizaron numerosas pruebas con diferentes computadoras y diferentes puertos TCP y los resultados seguían siendo impredecibles. Se utilizó el analizador de protocolos *Ethereal versión 0.10.12* en el servidor remoto para observar todos los paquetes TCP entrantes y descartar que algún “firewall” estuviese bloqueando la solicitud de conexión. Mediante el analizador de

protocolos se pudo observar que los mensajes de solicitud de conexión a veces llegaban y a veces no, por lo cual se descartó que el problema estuviese en el servidor.

Con los resultados arrojados por el analizador de protocolos se recurrió a consultar al personal técnico de la empresa Movilnet (la cual proporcionó el módem y la línea de prueba) para descartar que la red estuviese bloqueando o descartando los paquetes IP provenientes del módem. El personal técnico consultó el sistema y se aseguró que la línea estuviese completamente habilitada y sin ningún tipo de restricciones. Movilnet aseguró el problema no se encontraba en la red, pero también descartó que la falla estuviese en el módem, ya que el mismo no presentaba problemas en conectarse a Internet si se instalaba como módem genérico en Windows (en una computadora), que era la aplicación más común que la empresa le había dado a este equipo hasta el momento.

A pesar de la respuesta del personal técnico de la empresa, el funcionamiento del módem seguía siendo cuestionable, debido a que el modo de operación como módem genérico de Windows y como módem de la Unidad Remota era completamente diferente. En el primer caso, Windows es el encargado de manejar la pila de protocolos TCP/IP y el módem prácticamente transmite los paquetes que recibe; en el segundo caso, la Unidad Remota utiliza la pila de protocolos TCP/IP del propio módem, el cual se encarga del ensamble de los paquetes.

Finalmente se consultó a la empresa Microtel, la cual es contratista de las empresas con tecnología CDMA en Venezuela y que utilizan el módem interno del EMII-800 (DTSS-800) en aplicaciones de telefonía fija inalámbrica. Se realizaron numerosas pruebas similares a las ya realizadas y se obtuvieron los mismos resultados, por lo cual se decidió reprogramar el módem con una versión más nueva del software interno. Después de la actualización se realizaron alrededor de 5 pruebas en el sitio y los resultados fueron satisfactorios. A partir de ese momento se

realizaron numerosas pruebas del módem durante varios días, obteniendo los resultados esperados.

Como resultado de las pruebas con el módem, se pudo observar lo siguiente:

- En el proceso de conexión, se reciben mensajes de notificación y caracteres especiales ,como CR y LF, que no aparecen documentados en el manual. En este sentido, la Unidad Remota se diseño de tal manera que pudiese soportar hasta 4 comandos errados antes del esperado. Además, los caracteres LF son descartados y no se almacenan en el “Buffer” de recepción.
- La aplicación *Hyperterminal*, utilizada para simular el servidor TCP, tiene la propiedad de retransmitir todos los paquetes que recibe, de manera que se refleja como un “eco” de todos los mensajes transmitidos desde el módem. Por lo tanto, esta aplicación no es ideal para simular el servidor, aunque fue utilizada en un comienzo mientras se desarrollaba la primera versión de éste.

Apéndice 6. Pruebas con vehículo simulado

A continuación se presentan las pruebas realizadas a la Unidad Remota con la interfaz OBDII simulada, lo que se logró con la ayuda de un computador y un hardware externo de adaptación de voltajes. El envío de los comandos se realizó mediante una computadora conectada de manera local a través del puerto serial. Los comentarios se encuentran resaltados en color amarillo. Los saltos para poder observar lo que ocurre en cada interfaz para cada momento se encuentran resaltados en verde.

PRUEBA1. Prueba de error en la inicialización de ISO 9141

- **Hyperterminal (computador local)**

SMRI_UCAB> MONITOR

00 01 02 03 04 05 06 07 08 09 /Muestra los PID anotados en la lista de pedidos

SMRI_UCAB> CONTINUO 01 /Solicita lectura de la lista de pedidos 1 vez.

Véase lo que ocurrió en la interfaz OBDII, punto 1

*2

ECU no inicializa /mensaje de notificación del error por parte de la Unidad Remota

SMRI_UCAB> /La unidad queda lista para recibir nuevos comandos

- **Docklight (Vehículo)(Interfaz OBDII ISO9141)**

*1

56 49 56 4F /Estos números son una mala interpretación de la secuencia de inicialización a 5bps, debido a que la computadora está configurada a 9.6 Kbps

09/08/2006 09:56:21.26 [TX] - 55 /Byte de sincronismo.

09/08/2006 09:56:21.28 [RX] - 55

09/08/2006 09:56:23.18 [TX] - 08 /Primera palabra clave

09/08/2006 09:56:23.20 [RX] - 08

/Nótese que todo lo que se envía, se recibe, debido a que la línea K es en dos direcciones (*bidireccional*). Los números en azul son los transmitidos y los rojos los recibidos.

Después de enviar estos datos, se detuvo la secuencia de inicialización para observar el comportamiento de la Unidad Remota ante este error.

Véase punto 2

PRUEBA2. Correcta inicialización de la interfaz ISO 9141 y lectura del ECU.

Prueba de “Tiempo fuera”

- **Hyperterminal (computador local o remoto)**

```
SMRI_UCAB> MONITOR
```

```
00 01 02 03 04 05 06 07 08 09 /muestra lista de pedidos
```

```
SMRI_UCAB> CONTINUO 01 Véase punto 1
```

```
*2
```

```
DATA 000FFFFFF Véase punto 3
```

```
DATA 012FA1337E
```

```
DATA 07FF
```

```
DATA 000FFFFFF
```

```
DATA 012FA1337E
```

```
DATA 07FF
```

```
DATA 012FA1337E
```

```
DATA 000FFFFFF
```

```
DATA 000FFFFFF
```

```
DATA 000FFFFFF
```

```
DATA 000FFFFFF
```

```
DATA 000FFFFFF
```

```
LEER
```

```
Ok!
```

```
SMRI_UCAB> DATA 012FA1337E
```

```
DATA 07FF
```

```
/En este punto no se respondió al mensaje de solicitud en la interfaz ISO 9141. Por lo tanto, la Unidad Remota envió un mensaje de notificación del error y quedó lista para recibir nuevos comandos
```

```
ECU timeout
```

```
SMRI_UCAB>
```

- **Docklight (Vehículo)(Interfaz OBDII ISO9141)**

```
56 49 56 4F
```

```

09/08/2006 10:02:27.70 [TX] - 55
09/08/2006 10:02:27.72 [RX] - 55
09/08/2006 10:02:28.79 [TX] - 08
09/08/2006 10:02:28.81 [RX] - 08
09/08/2006 10:02:29.75 [TX] - 08
09/08/2006 10:02:29.78 [RX] - 08 F7
09/08/2006 10:02:30.98 [TX] - CC /Último byte de la secuencia de inicialización

09/08/2006 10:02:31.01 [RX] - CC 68 6A F1 01 00 C4 /Primer mensaje de solicitud (a
partir del byte 0x68)

09/08/2006 10:02:32.37 [TX] - 48 6B 01 00 0F FF FF C1 /Mensaje de respuesta
Salte a punto 2

*3

/Segundo mensaje de solicitud (a partir del byte 0x68)

09/08/2006 10:02:32.40 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 01 C5
09/08/2006 10:02:36.15 [TX] - 48 6B 01 01 2F A1 33 7E 36 /Mensaje de respuesta
/Luego, los datos de estas respuestas son enviados por la interfaz serial. Así ocurre
con dada una de las respuestas.

09/08/2006 10:02:36.18 [RX] - 48 6B 01 01 2F A1 33 7E 36 68 6A F1 01 02 C6
09/08/2006 10:02:39.82 [TX] - 48 6B 01 07 FF BA
09/08/2006 10:02:39.86 [RX] - 48 6B 01 07 FF BA 68 6A F1 01 03 C7
09/08/2006 10:02:43.62 [TX] - 48 6B 01 00 0F FF FF C1
09/08/2006 10:02:43.65 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 04 C8
09/08/2006 10:02:47.34 [TX] - 48 6B 01 01 2F A1 33 7E 36
09/08/2006 10:02:47.37 [RX] - 48 6B 01 01 2F A1 33 7E 36 68 6A F1 01 05 C9
09/08/2006 10:02:51.03 [TX] - 48 6B 01 07 FF BA
09/08/2006 10:02:51.06 [RX] - 48 6B 01 07 FF BA
09/08/2006 10:02:51.59 [TX] - 48 6B 01 01 2F A1 33 7E 36
09/08/2006 10:02:51.62 [RX] - 48 6B 01 01 2F A1 33 7E 36
09/08/2006 10:02:52.20 [TX] - 48 6B 01 00 0F FF FF C1
09/08/2006 10:02:52.23 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 06 CA
09/08/2006 10:02:56.81 [TX] - 48 6B 01 00 0F FF FF C1
09/08/2006 10:02:56.84 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 07 CB
09/08/2006 10:03:00.51 [TX] - 48 6B 01 00 0F FF FF C1
09/08/2006 10:03:00.54 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 08 CC
09/08/2006 10:03:04.42 [TX] - 48 6B 01 00 0F FF FF C1
09/08/2006 10:03:04.45 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 09 CD
09/08/2006 10:03:08.25 [TX] - 48 6B 01 00 0F FF FF C1
09/08/2006 10:03:08.28 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 00 C4
09/08/2006 10:03:15.61 [TX] - 48 6B 01 01 2F A1 33 7E 36
09/08/2006 10:03:15.64 [RX] - 48 6B 01 01 2F A1 33 7E 36 68 6A F1 01 01 C5
09/08/2006 10:03:19.53 [TX] - 48 6B 01 07 FF BA
09/08/2006 10:03:19.56 [RX] - 48 6B 01 07 FF BA 68 6A F1 01 02 C6

```

PRUEBA3. Verificación de errores

- **Hyperterminal (computador local o remoto)**

```
SMRI_UCAB> MONITOR
```

```
00 01 02 03 04 05 06 07 08 09
```

```
SMRI_UCAB> CONTINUO
```

```
Ok!
```

SMRI_UCAB> DATA 000FFFFF

DATA 012FA1337E

DATA 07FF

ECU ChckS error ;Se recibió un error en el byte de "Checksum". Se detiene la monitorización y la Unidad Remota queda lista para recibir nuevos comandos.

SMRI_UCAB>

- **Docklight (Vehículo)(Interfaz OBDII ISO9141)**

```
56 49 56 4F
09/08/2006 10:28:12.72 [TX] - 55
09/08/2006 10:28:12.73 [RX] - 55
09/08/2006 10:28:13.34 [TX] - 08
09/08/2006 10:28:13.36 [RX] - 08
09/08/2006 10:28:13.57 [TX] - 08
09/08/2006 10:28:13.59 [RX] - 08 F7
09/08/2006 10:28:14.89 [TX] - CC
09/08/2006 10:28:14.92 [RX] - CC 68 6A F1 01 00 C4
09/08/2006 10:28:16.53 [TX] - 48 6B 01 00 0F FF FF C1
09/08/2006 10:28:16.56 [RX] - 48 6B 01 00 0F FF FF C1 68 6A F1 01 01 C5
09/08/2006 10:28:21.53 [TX] - 48 6B 01 01 2F A1 33 7E 36
09/08/2006 10:28:21.56 [RX] - 48 6B 01 01 2F A1 33 7E 36 68 6A F1 01 02 C6
09/08/2006 10:28:25.50 [TX] - 48 6B 01 07 FF BA
09/08/2006 10:28:25.53 [RX] - 48 6B 01 07 FF BA 68 6A F1 01 03 C7
09/08/2006 10:28:29.09 [TX] - 48 6B 01 07 FF BB
09/08/2006 10:28:29.12 [RX] - 48 6B 01 07 FF BB
```

Apéndice 7. Pruebas del protocolo OBDII con sistema espía.

Luego de haber realizado pruebas de la Unidad Remota con vehículo y módem simulado, se procedió a probar el sistema en condiciones reales. Al observar que la Unidad Remota fallaba en el proceso de inicialización de la interfaz ISO 9141, se decidió crear un “sistema espía” para monitorizar la comunicación entre un equipo de diagnóstico profesional y el vehículo, con el objetivo de determinar diferencias entre el protocolo usado por el equipo profesional y el implementado en la Unidad Remota, con el fin de descartar diferencias que pudiesen existir entre el sistema de diagnóstico del vehículo y los estándares ISO 9141 y SAE J1979. El sistema espía utiliza un computador con puerto serial y un hardware externo de acoplamiento óptico hacia la interfaz ISO 9141. Obsérvese la figura 1.

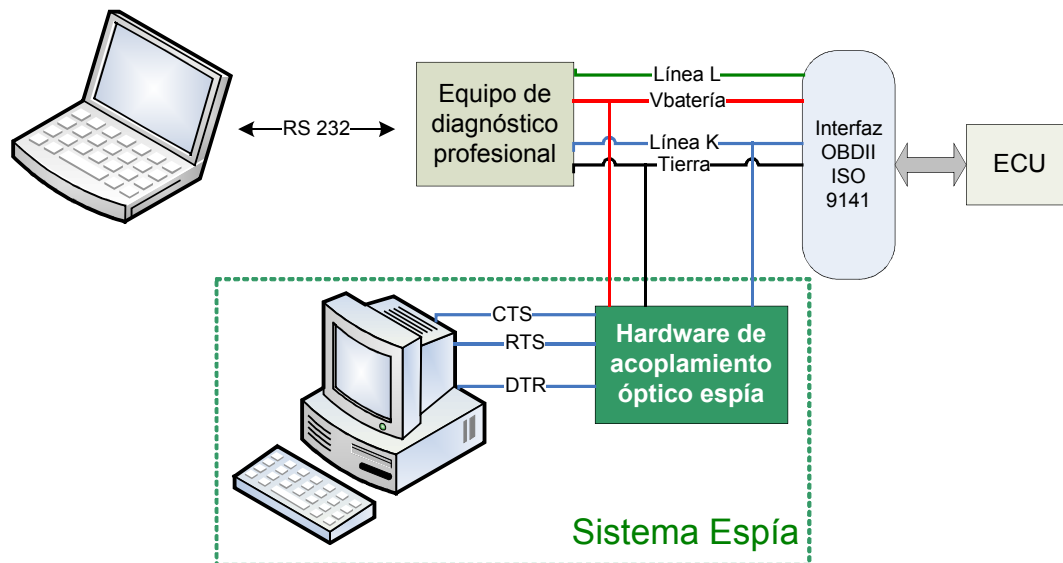


Figura 1. Sistema espía.

El acoplador óptico tiene el mismo diseño del circuito utilizado en la Unidad Remota para la “escucha” de la línea K, con la diferencia que, en lugar de polarizar el fototransistor con 5V respecto a tierra (para obtener una señal con niveles TTL), se polariza con 12V en la resistencia del colector (obtenidos de la línea RTS con salida en “1” lógico) y se coloca una señal de -12V en el emisor (obtenidos de la línea DTR

en estado “0” lógico) para obtener, de esta manera, los niveles RS-232 necesarios para la correcta detección de los bits, mediante la línea CTS. El valor de la resistencia de colector del fototransistor se modificó 480 ohmios. En la siguiente figura se muestra el hardware de acoplamiento óptico diseñado.

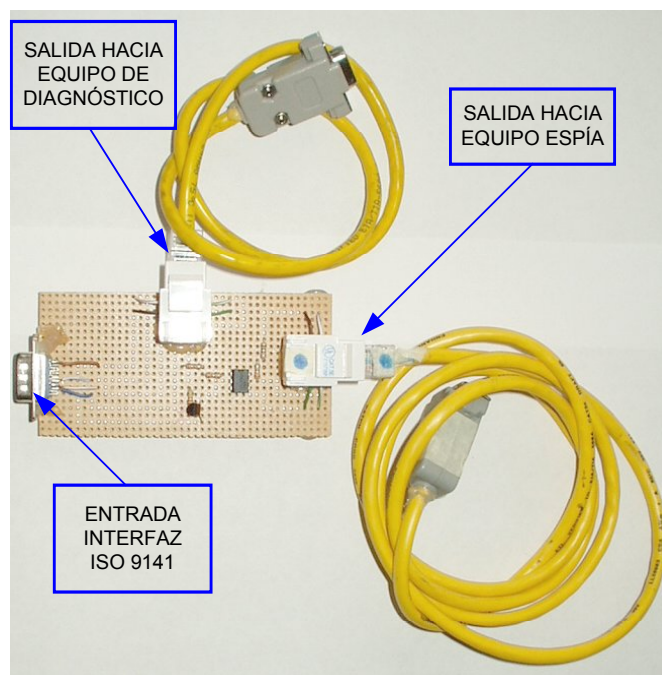


Figura 2. Hardware de acoplamiento óptico del sistema espía.

El sistema espía funciona como un osciloscopio con memoria, el cual detecta solo dos niveles de voltaje (alto y bajo), con una frecuencia de muestreo de 10 microsegundos. La información es almacenada de manera codificada (guarda el tiempo entre cambios de estado), con el objetivo de comprimir el tamaño del archivo resultante. El programa espía fue diseñado en lenguaje ensamblador *MASM* y el computador utilizado fue un Pentium I, aunque el código solo empleó instrucciones de procesador 286.

A continuación se muestra una tabla con los datos extraídos de una monitorización realizada con el software espía:

Tiempo sin transiciones (tiempo/10us) (HEX)	Tiempo sin transiciones (tiempo/10us) (Decimal)	Descifrado e interpretación	Byte enviado (binario)	Comentarios
367934	3569972			
00004D91	19857	START		
00009B86	39814	11		
00009B1F	39711	00	00110011	inicialización a 5bps
00009B89	39817	11		
00009B1E	39710	00		
0000891A	35098	STOP		
0000000A	10	START		
0000000A	10	1		
9	9	0		
0000000A	10	1		
9	9	0	01010101	Byte de sincronismo. Duración aproximada del bit = 100 microsegundos (Velocidad. aprox 10.4kbps)
0000000A	10	1		
9	9	0		
0000000A	10	1		
0000000A	10	0		
0000042D	1069	STOP		
26	38	Start + 000		
0000000A	10	1	00001000	Palabra clave 0x08 (1 solo ECU)
25	37	0000		
000003B2	946	Stop		
27	39	Start + 000		
0000000A	10	1	00001000	Palabra clave 0x08 (1 solo ECU)
24	36	0000		
00000B86	2950	Stop		
9	9	Start		
0000001D	29	111	11110111	Inverso de la última palabra clave
0000000B	11	0		
00000DCB	3531	1111+stop		
0000001E	30	start+00		
11	17	11	11001100	Inverso del byte de inicialización
14	20	00		
0000173E	5950	11+stop		
27	39	start+000		
0000000A	10	1		
0000000A	10	0	01101000	68
13	19	11		Tipo de mensaje: diagnóstico, solicitud
9	9	0		
267	615	stop		
11	17	start+0		

0000000A	10	1		
0000000A	10	0		
0000000A	10	1		
0000000A	10	0	01101010	6A
13	19	11		Destino: A los ECU de diagnóstico
9	9	0		
266	614	stop		
0000000A	10	start		
0000000A	10	1	11110001	F1
0000001D	29	000		Origen: Dirección física del dispositivo de diagnóstico
0000028A	650	1111+stop		
0000000A	10	start		
9	9	1		
44	68	0000000	00000001	1
266	614	stop		Servicio: 01 (data actual)
0000000A	10	start		
9	9	1		
0000001C	28	000		
0000000A	10	1	00010001	11
0000001D	29	000		PID: "Absolute Throttle Position"
266	614	stop		
0000000A	10	start		
0000000A	10	1		
0000000A	10	0		
7	7	1		
0000000C	12	0	11010101	D5
8	8	1		"Checksum"
0000000A	10	0		
00000BD0	3024	11+stop		
27	39	start+000		
9	9	1		
14	20	00	01001000	48
9	9	1		Tipo de mensaje: diagnóstico, respuesta
0000000A	10	0		
8	8	stop		
0000000C	12	start		
13	19	11		
9	9	0		
8	8	1	01101011	6B
0000000A	10	0		Destino: A todos los equipos de diagnóstico
13	19	11		
0000000A	10	0		
9	9	stop		
30	48	start+0000		
0000000B	11	1	00010000	10
0000001D	29	000		Origen: Dirección física del ECU

7	7	stop		
0000000C	12	start		
7	7	1		
31	49	00000	01000001	41
9	9	1		Servicio: Respuesta a servicio 01
0000000A	10	0		
0000000A	10	stop		
0000000A	10	start		
9	9	1		
0000001D	29	000	00010001	11
0000000A	10	1		PID: "Absolute Throttle Position"
0000001C	28	000		
0000000A	10	stop		
0000000A	10	start		
7	7	1		
14	20	00	00011001	19
13	19	11		Valor medido (se interpreta como "25%")
0000001E	30	000		
0000000A	10	stop		
12	18	start+0		
0000001D	29	111		
0000000A	10	0	001011110	5E
7	7	1		Checksum
14	20	00		
7501	29953	stop		
28	40	start+000		
8	8	1		
9	9	0	01101000	68
14	20	11		Tipo de mensaje: diagnóstico, solicitud
9	9	0		
267	615	stop		
13	19	start + 0		
0000000A	10	1		
9	9	0		
0000000B	11	1	01101010	6A
7	7	0		Destino: A los ECU de diagnóstico
13	19	11		
0000000B	11	0		
265	613	stop		
0000000A	10	start		
9	9	1	11110001	F1
0000001D	29	000		Origen: Dirección física del dispositivo de diagnóstico
0000028C	652	1111 +stop		
0000000B	11	start		
9	9	1	00000001	1
44	68	0000000		Servicio: 01 (data actual)

265	613	stop		
0000001D	29	start+00		
13	19	11	00001100	C
25	37	0000		PID: RPM del motor
266	614	stop		
30	48	start + 0000		
0000000A	10	1	11010000	D0
0000000A	10	0		Checksum
00000BCB	3019	11+stop		
27	39	start+000		
0000000A	10	1		
13	19	00	01001000	48
0000000A	10	1		Tipo de mensaje: diagnóstico, respuesta
0000000A	10	0		
9	9	stop		
0000000B	11	start		
10	16	11		
0000000A	10	0		
0000000A	10	1	01101011	6B
0000000A	10	0		Destino: A todos los equipos de diagnóstico
13	19	11		
0000000A	10	0		
9	9	stop		
30	48	start+0000		
0000000A	10	1	00010000	10
0000001C	28	000		Origen: Dirección física del ECU
0000000A	10	stop		
8	8	start		
0000000B	11	1		
31	49	00000	01000001	41
7	7	1		Servicio: Respuesta a servicio 01
0000000A	10	0		
0000000A	10	stop		
0000001C	28	start+00		
14	20	11	00001100	C
26	38	0000		PID: RPM del motor
0000000A	10	stop		
13	19	start+0		
0000000A	10	1		
0000000A	10	0	00001010	A
8	8	1		Primer Byte de data
26	38	0000		
9	9	stop		
0000000C	12	start		
9	9	1		
8	8	0	10111101	BD

28	40	1111		Segundo Byte de data
9	9	0		En conjunto, ambos bytes dan el valor de 2749/4 = 687,25 RPM
13	19	1 + stop		
9	9	start		
0000001D	29	111		
0000000A	10	0	11010111	D7
0000000A	10	1		Checksum
9	9	0		
0000764C	30284	11 + stop		

Tabla 1. Datos extraídos de monitorización con sistema espía.

De los datos mostrados en la tabla anterior se puede notar lo siguiente:

- La capa de enlace cumple con lo descrito en ISO 9141 y el formato de los mensajes de diagnóstico es de SAE J1979.
- El proceso de inicialización de la interfaz es el descrito en ISO 9141.
- La velocidad de transmisión de la línea K es cercana a los 10.4 Kbps, por lo cual se supone que el vehículo es compatible con ISO 9141-2. Esto se nota debido a que los tiempos de bits están cercanos a los 100 microsegundos, (los tiempos de la tabla deben ser multiplicados por 10 microsegundos para obtener la duración real). La forma más sencilla de determinar la velocidad aproximada es observando el byte de sincronismo, donde la duración de cada bit está alrededor de los 10*10 microsegundos.
- El equipo de diagnóstico utilizado en la prueba no aprovecha al máximo la velocidad de la interfaz, debido a que, en primer lugar, los tiempos de parada entre cada byte son mayores al tiempo de un bit y, en segundo lugar, el tiempo que transcurre entre la recepción de una respuesta y el envío de una nueva solicitud es de aproximadamente 300ms, pudiendo hacerla en 55ms.
- El ECU utiliza un solo bit de parada entre los bytes, aprovechando al máximo la velocidad de la interfaz; además, responde a los mensajes de solicitud en un tiempo aproximado de 30ms, cumpliendo con lo establecido en SAE J1979 (inferior a 50ms).

El análisis con el sistema espía comprobó que el vehículo de prueba era compatible con ISO 9141-2 (tiempos de bit alrededor de 100 microsegundos) y que el

protocolo empleado por el equipo profesional era idéntico al empleado por la Unidad Remota (ISO 9141 y SAE J1979). Por lo tanto, se descartó que el error presentado en la Unidad Remota se debiese al protocolo OBDII del vehículo o a una diferencia entre los estándares y la implantación real de los mismos.