

# SISTEMA DE DIAGNÓSTICO PREDICTIVO PARA AUTOMOTORES BAJO NORMA OBD II, CON CONEXIÓN INALÁMBRICA WI-FI Y CONTROLADORES TÁCTILES DE LA MARCA APPLE.



Ing. John Jairo Ceballos  
Ing. Germán Erazo L.  
Ing. Alexandra Corral  
Departamento de Ciencias de la Energía y Mecánica  
Quijano y Ordoñez y Marqués de Maéza s/n  
Latacunga - Ecuador  
Email : jjceballos@espe.edu.ec  
macorral@rspe.edu.ec  
wgerazo@espe.edu.ec



## Resumen.

El incremento en el número de componentes del vehículo controlados electrónicamente mediante ECU es continuo. Si bien, la cooperación entre esas y el acceso a la información que circula por ellas, o entre ellas, el conjunto de sensores y actuadores, sigue quedando restringido a los fabricantes de vehículos y las grandes empresas fabricantes de componentes, es necesario la creación de nuevas interfaces intuitivas y de rápido acceso de comunicación con estos componentes, para la utilización de ingenieros y aficionados en el campo automotriz, en las áreas correspondientes al diagnóstico y el confort.

El presente artículo relaciona el análisis e interpretación de los diferentes sistemas eléctricos, electrónicos y mecánicos de los diferentes tipos y marcas de automóviles, que mantengan un sistema de comunicación OBD II.

## I. INTRODUCCIÓN.

La utilización de sistemas de comunicación con redes de tipo inalámbricas, es el tipo de enlace apto para el intercambio de información, tomando en cuenta los tiempos de respuesta, que en este caso son muy altos, y los medios en los cuales van a ser instalados.

La introducción de sistemas táctiles de la marca Apple, ha contribuido a la mejora de los sistemas inalámbricos, con aplicaciones desarrolladas para actividades específicas de industrias especiales.

En este caso, en la industria automotriz, permite mantener un sistema de información organizada y universal, así como permite establecer una base de datos determinada para cada unidad y marca, la misma que será utilizada en casos de revisiones, o de diagnósticos de tipo preventivo, cuando el usuario así lo quiera.

Toda la estructura se basa en el manejo del protocolo OBDII respectivo (ISO 9141-2, SAE J1850 VPW, SAE J1850 PWM, ISO 15765 (CAN), etc.), por lo que es requerimiento indispensable establecer un lenguaje informático optimizado para su enlace, en este caso inalámbrico.

La utilización del lenguaje Objective-C es la herramienta funcional adecuada debido a sus características de alto desempeño en conexiones wireless, y su tasa rápida de respuesta tanto en conexión de subida como de bajada.

Físicamente, el esquema de conexión, está compuesto por una interfaz de lectura de códigos de tipo OBD II con generador de señal de emisión de datos Wi-Fi, un software de aplicación desarrollado en una interfaz única de usuario (NUI) y compilado en Objective-C, un dispositivo táctil de la marca Apple (iPhone, Ipad), y una red establecida de registro de información periódica e individual para cada modelo y tipo de automotor analizado.

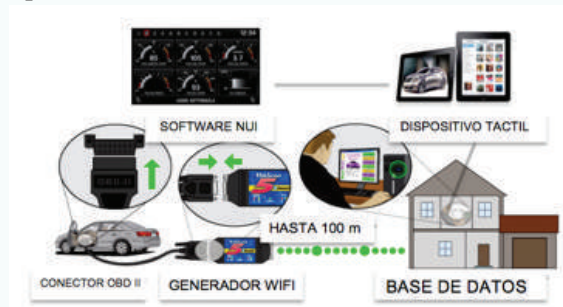


Figura 1. Diagrama de Bloques de un Sistema de Comunicación Inalámbrica

Utilizando el diseño del que fue hecho el dispositivo táctil, podemos aplicar herramientas tecnológicas de medición de avanzada, como son los acelerómetros micrométricos, los inclinómetros y giroscopios digitales, de gran importancia en el análisis tanto de vibraciones y oscilaciones causadas por el movimiento y la velocidad, las fuerzas G que soporta tanto en aceleración y frenado, en el caso del acelerómetro; y el nivel de sujeción al suelo, y viraje del automóvil

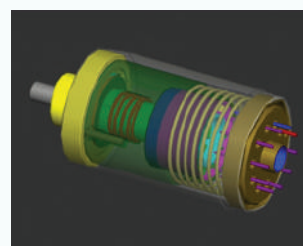


Figura 2. Acelerómetro digital micrométrico



La utilización del diseño realizado nos permite recibir información de los siguientes parámetros controlados por la ECU, dependiendo del nivel de tecnología del que este compuesto:

- Velocidad del motor.
- Compresión y vacío.
- Aceleración, poder y torque.
- Consumo de combustible.
- Avance al encendido.
- Temperatura del refrigerante.
- MAP
- MAF
- Ajuste del combustible.
- Presión del combustible.
- Relación aire/combustible, etc.

## II. DISEÑO DE LA RED DE COMUNICACIÓN

Nuestra red de enlace, entre dispositivo táctil e interfaz de generación de señal, se ve optimizada, tanto en comunicación de envío como de recepción, con la utilización de la interconexión de tipo WI-FI ya que, en comparación del resto de tipos de tecnologías de comunicación, nos da las siguientes características:

- Alcance hasta 100 mts en lugares abiertos, dependiendo de estado meteorológicos y otros factores de recepción, como alcance del generador de señal, alcance del receptor, etc.
- Fiabilidad Alta, con la utilización de una máscara de subnet y una dirección IP estática propia, y un puerto de comunicación dedicado para el envío y recepción de bits de información.
- Ancho de Banda de 3 Mbps nominal, más que suficiente para cadenas de datos que van desde 120kb hasta 1Mb como tope de información.
- Obstáculos e interferencias que esta red puede sobrepasar, los mismos que son mínimos, ya que el automóvil no está diseñado como un espacio totalmente cerrado para la comunicación de radiofrecuencias.
- Seguridad de transmisión mediana para ser un medio inalámbrico.
- Programación compatible con la empleada en el dispositivo, que es de tipo Objective-C, nativa de aplicaciones UNIX para dispositivos táctiles.
- Protocolos de comunicación IEEE 802.11b, óptimos con el receptor y el generador de señal, que utilizan protocolos de tipo estándar 802.11a/b/g. Protocolos de comunicación.

Determinado el ambiente en el que se va a desarrollar el enlace, para la conexión entre interfaz y módulo de control, primero se debe inicializar el OBDNet o red informática encriptada OBDII. Para realizar este proceso en Objective-C se llama a la función OBDNetInitialize, la misma que analiza y

informática se esta utilizando, para enviar información que sea de lectura para nuestro respectivo protocolo.

Después de haber inicializado OBDNet, se requiere crear una nueva función con el nombre OBDNET\_HANDLE para identificar nuestra conexión, en este caso inalámbrica WI-FI. Con ésta nosotros podemos extender nuestra red punto a punto a una red en malla, para no solamente tener un dispositivo táctil al alcance, sino varios tablets recibiendo y enviando información al mismo tiempo.

Para finalizar el proceso, la IPA llama las siguientes funciones complementarias:

- OBDNetCreate para crear una red inicial con valores en tiempo real fundamental para que, en el manejo de varios vehículos de estudio, no permita la conjunción de datos entre unidades analizadas.
- OBDNetWorker para designar parte de la programación a cada campo de análisis que nuestra interfaz requiera, por ejemplo el estudio de PIDs, GLCDs, medidores de tiempo real, y parte de diagnóstico predictivo.

## III. DESENCRIPTACIÓN DE INFORMACIÓN EN OBJECTIVE-C

Inicializado el enlace, la programación se debe conectar al respectivo servidor de envío de información, creado entre el generador de señal y la salida del puerto OBDII del automóvil. Este servidor de envío es diferente de acuerdo a la complejidad de la ECU, y a la cantidad de dispositivos electrónicos que esta maneja.

La función OBDNetConnection permite entrar en este servidor, cuyo único requisito es que tengamos OBDNET\_HANDLE activo como parte de la red.

Como paso inicial de la obtención de datos, la ECU siempre envía primero una lista completa de PIDs activos, si este fuera el caso, como primer parámetro de análisis. La función OBDNetGetPidList realiza este estudio, con información procedente de datos enviados de cada PID por la función OBDNET\_CALLBACK\_PID\_LIST\_ITEM.

Como ejemplo de la descriptación de información, tomaremos un valor tipo de gran utilidad, en este caso el OBDNET\_PID\_VALUE

### Uso

Establece el valor real de lectura informática en bits del parámetro PID que se requiera. Se puede obtener mediante la utilización de la función `OBDNetParameterGetCurrentValue` o mediante el callback de definición `#define OBDNET_CALLBACK_STREAMING_PID_VALUE`.

### Sintaxis

```
typedef struct
{
    unsigned long    PidValueType;
    unsigned long    TimeStamp;
    double           EnglishValue;
    double           MetricValue;
    const char       * EnglishValueString;
    const char       * MetricValueString;
    const char       * UniqueId;
} OBDNET_PID_VALUE;
```

### Parámetros

- `PidValueType` es el tipo de dato que la estructura contiene. Puede ser `OBDNET_PID_VALUETYPE_NUMBER`, en caso de valores numéricos únicos, `POBDII_PID_VALUETYPE_STRING` en caso de una cadena de valores específicos OBDII. Además indica si este valor es válido para `EnglishValue - EnglishValueString` en el sistema inglés de medición, o `MetricValue - MetricValueString` en el sistema métrico.
- `TimeStamp` establece el número de milisegundos desde que la petición de información.
- `EnglishValue`, el valor en el Sistema Inglés, si el `PidValueType` es `POBDII_PID_VALUETYPE_NUMBER`.
- `MetricValue`, el valor en el Sistema Métrico, si el `PidValueType` es `POBDII_PID_VALUETYPE_NUMBER`.
- `EnglishValueString`, una cadena de valores en el sistema inglés, si el `PidValueType` es `POBDII_PID_VALUETYPE_STRING`.
- `MetricValueString`, una cadena de valores en el sistema métrico, si el `PidValueType` es `POBDII_PID_VALUETYPE_STRING`.
- `UniqueId` identifica el PID específico. Por ejemplo "SAE.RPM", etc.

Así se procede con las demás funciones incluidas en la programación, dependiendo de la lectura que se quiera obtener, del PID que se desea analizar y del instrumento que se aplique en ese momento, teniendo en ocho grandes grupos estas funciones:

- Función de inicialización.
- Función de creación.
- Función de estado.
- Función de corrección.
- Función de conexión.
- Funciones complementarias.
- Funciones de parámetros.
- Funciones de enlace.

Definidas las funciones, la información enviada sigue un esquema de análisis en etapas, como se lo demuestra en este esquema:

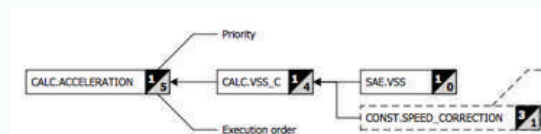


Figura 3. Diagrama de procesos OBDII

Se sigue una serie de procesos, antes de llegar a la información requerida por la IPA, que en el caso del diagrama anterior, es el cálculo de la aceleración en tiempo real, que depende en primera instancia de la corrección en la constante de la velocidad, y de los cálculos de las variables `SAE.VSS` y `CALC.VSS_C`, cuyos parámetros provienen directamente de la OBDNet.

## IV. PERSONALIZACIÓN DE LA INTERFAZ GRÁFICA DE USUARIO (NUI)

Para proveer al usuario un sistema amigable de uso y de interpretación, parte del diseño final comprende la personalización de una interfaz gráfica de usuario, la misma que por lenguaje de programación de objetos complementario Visual Basic SDK, hace uso del dispositivo táctil y del resto de complementos tecnológicos correspondientes a cada estudio.

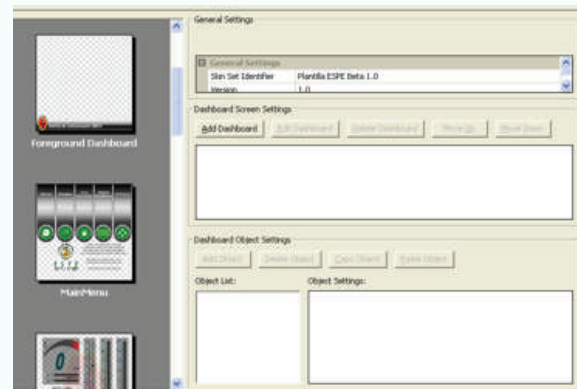


Figura 4. Programador – Compilador de objetos Visual Basic SDK

Como cualquier proyecto basado en lenguaje Visual Basic, este programador compilador tiene la peculiaridad que al mismo momento que realiza la edición de características y funcionalidad, desarrolla toda la secuencia informática



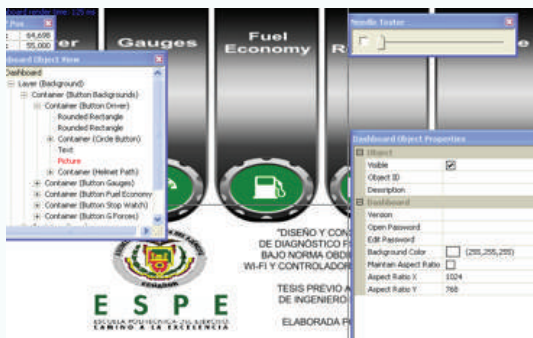
requerida por cada acción ya en Objetivo-C, que a diferencia de los lenguajes C y C++, optimiza la utilización de memoria, y el uso de recurso tanto en las llamadas de variables, como en las peticiones de retorno o Callback.

El software divide a nuestra plantilla en varios grupos de acción, medidores, gauges digitales, GLCDs dinámicos de alta velocidad de respuesta, visualizadores de inclinómetros, lectores de acelerómetros y valores alfanuméricos correspondientes, los mismos que pueden ser interpretados tanto en el sistema inglés como en el sistema métrico de medición.



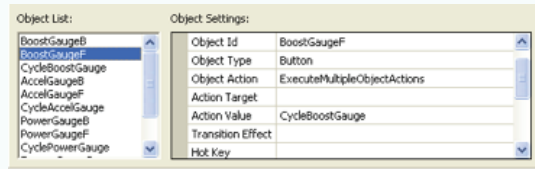
**Figura 5. Editor de GLCDs dinámicos y gauges electrónicos**

En lo relacionado a las características de la acción de cada objeto, se despliegan dos grupos que modifican, en primera instancia, los detalles gráficos del botón correspondiente (color, tamaño, fondo, imagen, etc), y los parámetros intuitivos en tiempo real (cambios de tamaño, color, valores, ubicación, etc), todo automatizadamente, sin que intervenga la acción del usuario a no ser que sea estrictamente necesario.



**Figura 6. Edición de acción y reacción de botones y medidores.**

Para finalizar en el proceso de personalización de la herramienta de diagnóstico y medición, se deben establecer los valores de recepción y envío de la información tomada desde el servidor OBDNet, de acuerdo al respectivo protocolo OBDII conforme al fabricante, en el apartado de edición de funciones y callbacks SDK como se lo señala en el siguiente gráfico:



**Figura 7. Funciones y callbacks SDK de la IPA.**

Terminado el proceso de personalización, solo nos queda introducir nuestra interfaz gráfica en la IPA de comunicación dentro del dispositivo táctil Apple escogido (IPAD, IPHONE, etc.).

### V. CONEXIÓN, PRUEBAS Y RESULTADOS.

Para comprobar el excelente funcionamiento de nuestro diseño ya introducido en la IPA, vamos a conectarnos inalámbricamente con los servidores de datos OBDNet de dos diferentes vehículos, un Mazda 3 1.6 full, y un Chevrolet Optra 1.8 estándar. Cada uno con listas de PIDs diferentes, en cuanto a componentes y a sensores, actuadores que intervienen en el sistema mecánico - electrónico.



**Figura 8. Lectura de datos mediante GLCDs.**

El enlace físico entre el generador de señal WI-FI y el dispositivo táctil con la respectiva IPA diseñada, es establecer nuestra dirección IP y pronunciar la identificación DHCP de la Tablet.

La tecnología WI-FI tiene varios parámetros (Dirección IP, Máscara de Subnet y dirección de router), los mismos que establecen una identidad virtual que ubica a cada dispositivo como receptor o generador dentro de la comunicación.

En la aplicación se utiliza los siguientes identificadores:

**Dirección IP:** 192.168.0.11

**Máscara de Subnet:** 255.255.255.0

Se utiliza un dirección IP con unidad 11 para evitar interferencias y que los envíos no se multipliquen en el ambiente, ya que por lo general, las redes de tipo infraestructura usan unidades que varían desde el 1 hasta el 10.

Conectado al servidor, nuestra IPA lee toda la lista de PIDs disponibles para el automóvil analizado, pudiendo nosotros adicionar PIDs de lectura adicionales si el caso así fuera necesario.

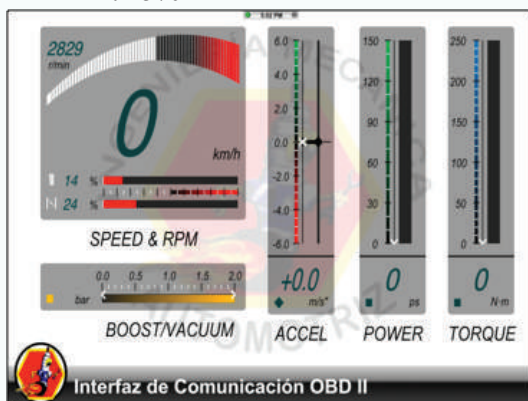
Data Logging	
Forward Acceleration	0.000m/s <sup>2</sup>
Lateral Acceleration	0.000m/s <sup>2</sup>
Vehicle Roll	0.0°
Vehicle Pitch	0.0°
Max engine torque for this P	NA
SAE_EMI_SLIP	NO DATA
Air/fuel ratio calculated from the actual lambda value from a wideband oxygen sensor	14.08:1
SAE_BAT_PWR	NO DATA
AUX_FLUX_TEMP	NO DATA
Time to empty	NA
AUX_FLUX_SPEED	NO DATA
Ignition Timing Advance for #1 Cylinder	88.0°
Calculated Load Value	33.9%
Wideband Absolute Pressure	0.0 atm 29.92 inHg
Air Flow Rate from Mass Air Flow Sensor	0.00 g/min 0.00 g/h

**Figura 9. Lista de PIDs estándar de lectura para el Mazda 3 1.6 full.**

Definida la lista de PIDs que deseamos que el servidor nos informe, es momento de utilizar el diseño personalizado que contiene los medidores, GLCDs y gauges digitales para comprobar la tasa de envío de bits.

El servidor OBDNet del Mazda 3 1.6 full, en gauges digitales, dio valores correspondientes a las RPM, el TPS o sensor de posición de la mariposa de aceleración, el APP o sensor de posición del pedal de aceleración respectivo, cuyos valores son:

- RPM: 3211
- TPS: 27%
- APP: 15%



**Figura 10. Lectura de gauges digitales.**

Así vamos comprobando los PIDs que va recibiendo por medio de la interfaz, que en el caso del Mazda 3 nos da valores de MAP, MAF y Boost que demuestran el buen estado de estos componentes, parte fundamental de un buen diagnóstico preventivo y predictivo.



**Figura 11 Lectura de flujo de aire, poder y economía de combustible.**

Como device adicional al comportamiento del automotor, y utilizando el acelerómetro como el giroscopio incorporado en el dispositivo táctil de la marca Apple, se desarrolló un analizador de viraje e inclinación del carro con respecto al nivel del suelo total tierra.



**Figura 12. Lista de PIDs estándar de lectura para el Mazda 3 1.6 full.**

Del análisis de ambos automotores, podemos sacar una tabla de composición de componentes, con lo que hay que señalar en primera instancia el nivel de tecnología del que está compuesto cada sistema de cada automotor, la complejidad en la totalidad de PIDs disponibles por cada ECU, la compatibilidad universal con diferentes tipos y marcas, y el estado de los componentes que son nuestra meta a cumplir en el campo del diagnóstico predictivo.

Tabla 1. Valores medidos de los Gauges y GLCDs.

PID	VEHICULO	Mazda 3 1.6	Optra 1.8
Comprensión		170 PSI average	160 PSI
VSS		SI 0	SI 0
RPM		SI 2829rpm	Si 2910rpm
TPS		SI 24%	SI 27%
APP		SI 14%	NA

En la tabla 1 se observan los valores obtenidos por medio de los gauges personalizados previamente, comprenden a los valores de compresión promedio, que en ambos casos están en el rango normal de acuerdo al fabricante, al vss o sensor de velocidad del vehículo que como se encuentra inmóvil en el momento del estudio da el valor 0, las RPMs en las que se hizo la medición, el nivel de apertura de la mariposa de aceleración TPS en esas revoluciones y el nivel de presión aplicada sobre el pedal del acelerador como en el caso único del Mazda.

Tabla 2. Valores medidos mediante el data logging y sensores actuadores.

PID	VEHICULO	Mazda 3 1.6	Optra 1.8
COOLANT TEMP		Si 80°C	Si 75°C
LOAD		23.9%	13.7%
MAF		Si 1.1	Si 0.27
MAP		Si 5.9 inHg	Si 7.1 inHg

En esta tabla se visualizan los valores de temperatura del refrigerante, la carga que soporta el motor el momento del estudio, el estado del MAF y la medición en pulgadas de mercurio inHg del MAP.

## **VI. CONCLUSIONES**

- El estudio de la OBDNet, no es de uso exclusivo de fabricantes de automóviles, scanners exclusivos, ni de programadores privados automotrices.
- La utilización de IPAs programadas en lenguaje Objective-C son óptimas para el análisis y el estudio del campo informático de sistemas OBDII.
- La personalización de plantillas con objetos activos del tipo Visual Basic, permite al usuario crear una guía propia con las mediciones que en verdad se crean necesarias en cada caso, y la forma como se presentan en la interfaz gráfica de usuario.
- El uso de un dispositivo móvil de la marca Apple, facilita en gran cantidad los procesos de identificación de ordenes de software a hardware, ayudando en gran parte a la reducción de conflicto en el ancho de banda de la red inalámbrica WI-FI, con el consiguiente beneficio en la mejora de la tasa de envío y recepción de bits.
- El uso de nuestro diseño, es potencialmente aplicable a las áreas de diagnóstico predictivo y preventivo de automotores, telemetría, competición y de docencia aprendizaje en el área de la informática automotriz muy poco explorada actualmente.
- Nuestro diseño es compatible para la mayoría de marcas disponibles en el mercado actualmente, siempre y cuando tengan una infraestructura tecnológica informática de tipo OBDII, con un protocolo compatible de los anteriormente señalados, y un puerto que sea adaptable a nuestro generador de señal WI-FI.

## **VII. BIBLIOGRAFÍA.**

Altenberg, A (2008). Become an Xcoder. Estados Unidos: Cocalab Edit

Chevrolet (2000). Manual de Servicio Optra Chevy. Brasil.

Hernández, L (2008). Lenguaje Objective-C. El Principio. Madrid: MacProgramadores.