Prototipo de sistema de captura y monitoreo de datos OBD-II de vehículos

Claudio Aciti[1,2], Mauricio Urraco[1], and Elías Todorovich[1,3]

[1] Universidad Nacional del Centro de la Pcia. de Buenos Aires, Tandil, Argentina, caciti@exa.unicen.edu.ar
[2] Universidad Nacional de Tres de Febrero, Departamento de Ciencia y Tecnología, Caseros, Argentina
[3] Universidad FASTA, Facultad de Ingeniería, Mar del Plata, Argentina

Abstract. En este trabajo se diseñó y construyó un prototipo de un sistema de captura de parámetros de vehículos según tecnología OBD-II. El prototipo es capaz de leer distintas magnitudes generadas por un automóvil a través de una interfaz ELM-327 USB que permite la comunicación con la computadora de a bordo del vehículo en conjunto con los datos obtenidos mediante un GPS USB. Al prototipo desarrollado se le añadió un módem 3G y un adaptador Wifi USB. El sistema se implementó en una plataforma Raspberry Pi B con sistema operativo Raspbian. El módem 3G se utilizó para permitir el envío de la información recolectada a través de Internet a un equipo o central remota, y el adaptador WiFi USB permitió configurar el Raspberry Pi para poder ser utilizado como un Access Point de manera que fuese posible conectarse con un dispositivo móvil que soporte el sistema operativo Android para visualizar los datos leídos. Las pruebas fueron realizadas sobre un automóvil Chevrolet Corsa Wagon Life Gls 1.4 4p modelo 2009.

Keywords: OBD-II, Monitor, Raspberry Pi

1 Introducción

El sistema de diagnóstico a bordo (OBD) en vehículos es un sistema integrado que, mediante el uso de sensores, brinda al conductor y al técnico información relacionada al motor del vehículo y a su funcionamiento [8]. Con el avance en las tecnologías electrónicas incorporadas por los fabricantes en vehículos motorizados, han surgido una serie de beneficios asociados al mejor desempeño de los motores. Numerosos vehículos han utilizado sistemas de control electrónico para aumentar la eficiencia, tanto de los sistemas de alimentación e inyección de combustible como del encendido. A la par, se han desarrollado diferentes formas para diagnosticar los problemas asociados a estos nuevos dispositivos electrónicos y es así como en la actualidad una computadora a bordo controla sensores y actuadores que mantienen al motor funcionando bajo condiciones favorables. Al conjunto de actuadores, sensores y software de diagnóstico se le denomina Sistema OBD o Sistema de Diagnóstico a Bordo [5].

La estandarización de este sistema comenzó en los años 60, en EEUU, y se desarrolló entre fabricantes, gobierno y entidades norteamericanas preocupadas por el medio ambiente. A partir de 1970 se dió inicio a una serie de normas y requisitos de emisiones graduales para el mantenimiento de los vehículos por periodos prolongados de tiempo. Para cumplir con estos estándares, los fabricantes recurrieron a controles electrónicos de la alimentación de combustible y los sistemas de ignición. Los sensores miden el rendimiento del motor y ajustan los sistemas para proporcionar un funcionamiento óptimo. Además se puede acceder a estos sensores para obtener un diagnóstico temprano del vehículo [9]. En 1988, se establece un conector estándar y un conjunto de señales de prueba de diagnóstico. Esto dio lugar a la primer generación de requerimientos de sistema de diagnóstico a bordo, llamada OBD-I. La segunda versión del sistema de diagnóstico a bordo, llamada OBD-II, es una mejora tanto en capacidad como en normalización respecto a lo que fue la regulación inicial de OBD. El estándar OBD-II especifica el tipo de conector de diagnóstico y sus pines, los protocolos de señalización eléctrica disponible, y el formato de los mensajes [11,13].

Para poder llevar a cabo este monitoreo se utiliza un scanner, el cual se conecta a la interfaz de diagnóstico y permite el acceso a la información del vehículo. Existen scanners básicos y avanzados. Los básicos muestran desde códigos de error hasta información almacenada por el vehículo y los más avanzados poseen una interfaz que permite ver el estado de los sensores en tiempo real, gráficas y datos almacenados, los cuales facilitan enormemente el diagnóstico por parte del usuario. Estos últimos scanners tienen un alto costo, lo que dificulta su acceso por parte de mecánicos y empresas.

Hoy en día, muchas empresas disponen de flotas de vehículos que se utilizan para uso interno o para dar servicios de transporte. La utilización de un sistema recolector de información les permite controlar el uso de los vehículos para disminuir gastos en mantenimiento e inversión, prevenir fallas y accidentes, detectar malos hábitos de manejo de sus empleados, reducir gastos operativos de combustible, de neumáticos, etc. A nivel nacional existen algunas empresas que brindan servicios basados en métricas vehiculares. Por ejemplo, Sitrack, ofrece un sistema de control de flotas que permite el monitoreo del uso de los vehículos y el accionar de los conductores enviando la información de lo ocurrido en tiempo real a un equipo remoto [3]. Otra empresa argentina, Ful-Mar, provee una interfaz de comunicación con la computadora de a bordo que permite la lectura y registro en linea de la información capturada del vehículo en funcionamiento [1]. A nivel mundial, entre las más destacadas se encuentra Plug-N-Track, una empresa norteamericana que comercializa un dispositivo para servicios de seguimiento que provee geo-localización satelital y alerta sobre hábitos de manejo indeseados [2]. Si bien es un mercado que se encuentra en constante crecimiento, todavía hay mucho que avanzar con el uso de este tipo de tecnologías.

El objetivo principal de este trabajo es desarrollar un prototipo de un sistema recolector de datos para automóviles que permita registrar in situ lo que ocurre en el sistema de a bordo y enviar la información capturada a una central remota.

Además, desarrollar una aplicación en Android para que un usuario visualice los datos capturados a través de un dispositivo móvil.

Se definen entonces como objetivos desarrollar un prototipo para recolectar datos de automóviles capaz de:

1. Monitorear el sistema de a bordo en vehículos que cumplan con el estándar OBD-II para capturar parámetros del motor tales como velocidad, RPM, posición del acelerador, temperatura del refrigerante, torque, etc.
2. Obtener la geo-localización del vehículo mediante monitoreo satelital.
3. Conectarse a Internet haciendo uso de redes telefónicas y enviar los datos remotamente cuando la conexión esté disponible; en caso contrario almacenar los datos en la memoria interna hasta que se disponga de conexión.
4. Diseñar e implementar una aplicación para Android que permita al usuario, de manera opcional, conectarse al sistema de captura de datos para visualizar los datos que están siendo recolectados.

1.1 Limitaciones

El GPS presenta una limitación debido a que en algunas zonas puede no funcionar, o funcionar de manera incorrecta presentando mucha demora para obtener las coordenadas. De manera similar sucede con el módem 3G, debido a que en Argentina aún no se cuenta con buena señal de cobertura, es posible que en ciertas zonas no sea posible establecer la conexión a Internet. Por otra parte, existe el problema de que los dispositivos conectados al Raspberry Pi consumen mayor cantidad de corriente de la que puede entregar dicho dispositivo a través de sus puertos USB. Por ejemplo, el módem 3G cuando se encuentra intentando establecer la conexión puede llegar a consumir picos de hasta 800mA, pudiendo ocasionar problemas eléctricos que como consecuencia terminan provocando una falla permanente en el Raspberry Pi. Una solución a este problema es el uso de un HUB USB con alimentación independiente.

2 Trabajos previos

Existe una gran cantidad de referencias sobre aplicaciones de OBD-II. Entre las más relevantes, [4] presenta un sistema de adquisición de datos vehiculares para el manejo automático de flotas de automóviles empleando interfaces OBD, GPS, y tecnologías WiFi. Ellos se centran en el desarrollo de algoritmos para la toma de decisiones y sugieren interesantes aplicaciones para el control de tráfico. Además, los métodos donde explican el diagnóstico remoto del estado de los vehículos resultaron útiles para el desarrollo de este trabajo.

El método de posicionamiento presentado en [10] corrige y suplanta la velocidad inercial provista por un sistema de posicionamiento como GPS mediante el valor de velocidad del sistema OBD-II del vehículo en cuestión. Ese trabajo está enfocado en tener posicionamiento disponible en múltiples condiciones dentro de una ciudad. Si bien el trabajo que se presenta en este artículo también relaciona

medidas provenientes de un GPS y un sistema OBD-II, el foco aquí está en el monitoreo.

Los resultados de un sistema como el aquí propuesto, podrían servir de insumo en trabajos como [6]. Ahí se propone un método para calcular un recorrido con los objetivos de ahorrar combustible y reducir las emisiones de un vehículo. El método toma en cuenta el estilo de conducción del chofer, las características y componentes de las posibles rutas, e información de tráfico en tiempo real. Para evaluar el sistema se usaron conjuntos de datos que incluían datos OBD-II de taxis con posiciones GPS asociadas. [14] también se centra en el ahorro de combustible tomando en cuenta estilos de manejo de automóviles. El consumo e impacto ambiental se obtiene a partir de parámetros OBD-II. Los autores desarrollan un sistema que finalmente ofrece sugerencias a los choferes para ahorrar combustible. En la línea que apunta a reducir la emisión de gases, en [15] ya habían realizado un resumen de los factores que influyen en la emisión de contaminantes mediante estudios experimentales utilizando OBD.

En [12] el objetivo es el seguimiento y análisis de flotas de vehículos terrestres. Ese sistema mide de velocidad, distancia y consumo de combustible integrando datos OBD-II y GPS. Luego transmite la información mediante Wifi a un servidor remoto.

3 Desarrollo del sistema

El objetivo del presente trabajo es desarrollar un sistema ad-hoc capaz de comunicarse con la computadora a bordo OBD-II de un vehículo para poder obtener distintas métricas relacionadas con el funcionamiento del mismo. Dichos datos recolectados del motor del vehículo son complementados con la posición geográfica brindada por un GPS-USB y pre-procesados para luego poder ser enviados a un equipo remoto. La conectividad a Internet es lograda mediante el uso y configuración de un módem 3G. Los requerimientos funcionales para este prototipo de sistema de captura y monitoreo de datos OBD-II de vehículos, son:

- Establecer la conexión con el sistema OBD-II a bordo.
- Enviar mensajes y procesar las respuestas obtenidas.
- Leer una serie de PIDs y decodificar las respuestas de acuerdo a la fórmula correspondiente a cada magnitud.
- Obtener las coordenadas de la posición del vehículo a partir del GPS.
- Establecer y mantener la conexión a Internet mediante redes móviles.
- Almacenar la información recolectada para ser enviada por redes móviles a un equipo remoto cuando se disponga de Internet. En caso contrario, se debe almacenar la información hasta que pueda ser enviada.
- Proporcionar una interfaz gráfica a través de un dispositivo móvil (tablet o smarthphone) para visualizar los datos que están siendo recolectados.

Se decidió utilizar un Raspberry Pi ya que es una computadora reducida en algunas capacidades de hardware, lo que le permite principalmente producirse y comercializarse a precios bajos. Su tamaño y consumo de energía también

son reducidos. Sobre la plataforma corre una versión de Linux Debian, llamada Raspbian, aunque admite otros SO. Todas esas características son favorables para proyectos como el aquí presentado. En este caso se usó una Raspberry Pi B, que tiene dos puertos USB y 512 MB de RAM.

En la Fig. 1 hay una vista general del sistema que muestra como interactua el Raspberry Pi con los dispositivos USB para capturar las magnitudes del motor del vehículo y los datos del GPS, establecer y mantener la conexión a Internet y utilizar la interfaz WiFi con el fin de convertirse en un Acces Point. Además se puede observar a grandes rasgos que la micro-computadora invoca a un servicio web para enviar los datos capturados a un equipo o central remota. Por otra parte un usuario puede mediante un dispositivo móvil con Android visualizar los datos a medida que éstos son leídos.

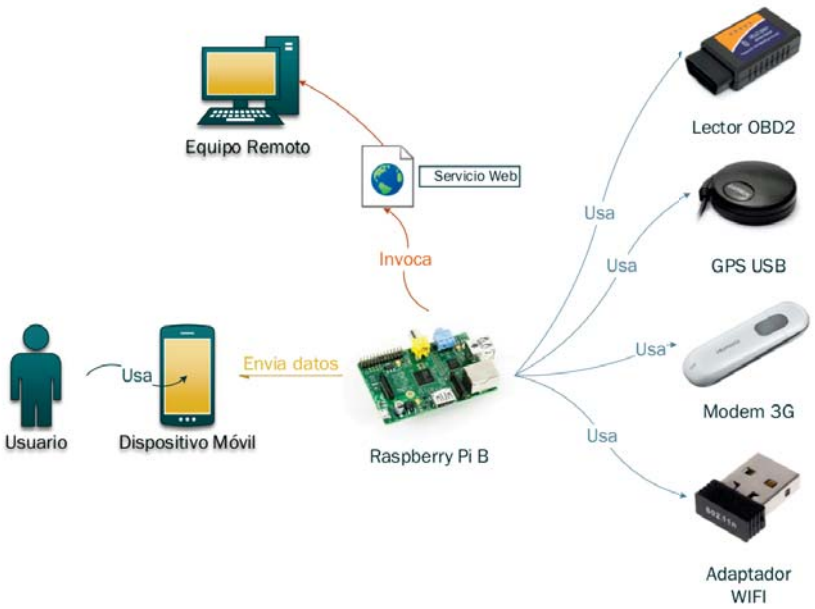

Fig. 1. Vista general del sistema.

Inicialmente se conectan todos los dispositivos a la Raspberry Pi: la interfaz ELM-327, el GPS USB, el adaptador WiFi y el módem 3G. Una vez que el vehículo se encuentra en marcha o encendido, se conecta el cable de alimentación desde el Raspberry Pi a la fuente conectada al vehículo. Cuando el micro-ordenador termina de iniciar el sistema operativo, el módulo recolector de datos, que compone el hilo principal del sistema, se inicia automáticamente. Este módulo intenta establecer comunicación con la interfaz ELM-327 mencionada previamente que sirve para leer la computadora de a bordo. Si ocurre un error

durante el establecimiento de la comunicación con la interfaz o si ésta fallara entonces todo el sistema fallaría, dando lugar a que no inicie. Por otra parte, si la conexión se establece con éxito, entonces el módulo recolector de datos inicia un bucle infinito donde se realizan las siguientes tareas:

1. Se invoca al módulo lector de GPS, implementado como un script en Python, y obtiene los datos leídos por este. Si no se reciben datos entonces el sistema continua su ejecución normalmente pero no reporta ningún valor para la latitud, la longitud y el tiempo.
2. Se leen las métricas de la computadora de a bordo OBD-II. En caso de que alguna de las lecturas falle entonces no se reportan valores para esa magnitud particular.
3. Los datos recogidos del vehículo y del GPS se escriben en paralelo a un archivo de datos y a un puerto TCP específico.
4. Cada 60 segundos se invoca al módulo de envío de datos, el cual es responsable de leer los datos escritos en el archivo, enviarlos a un equipo remoto, y vaciar el archivo leído.

Simultaneamente con este proceso recolector de datos, funciona el módulo disponibilizador de datos, el cual lee los datos recolectados a través del puerto TCP donde escribe el proceso anterior y se encarga de disponibilzarlos en forma de página web a través de otro puerto TCP. Una persona que disponga de un smartphone o tablet con sistema operativo Android y capacidad WiFi puede conectarse al Raspberry Pi como si éste fuera un Access Point y abrir la aplicación móvil desarrollada en este trabajo para visualizar los datos.

Entonces, el sistema se divide en módulos:

1. Módulo recolector de datos del vehículo
2. Módulo de envío de datos
3. Módulo disponibilizador de datos al dispositivo móvil
4. Módulo lector de GPS
5. Aplicación para el dispositivo móvil

Estos módulos se implementan de manera tal que cada uno es independiente del resto, para facilitar el testeo, y poder integrados de forma gradual.

3.1 Módulo recolector de datos

Este es el módulo central del sistema. Por un lado establece, la comunicación con la computadora de a bordo del vehículo (OBD-II) para solicitar los diferentes parámetros que se quieren monitorear. Para ello, inicializa el sistema embebido ELM-327 mediante comandos AT y luego solicita e interpreta las distintas respuestas mediante el Modo 1. Por otro lado, invoca a los módulos lector de GPS y de envío de datos a un equipo remoto.

El sistema embebido ELM-327 detecta e interpreta 9 protocolos distintos de de bajo nivel OBD. Traduce y devuelve la información en un formato único para

cualquiera de ellos, permitiendo la independencia con el protocolo interno que utiliza el vehículo. Para configurar el sistema ELM-327 se utilizan comandos AT [7]. Entre otras opciones, es posible desactivar el eco de los mensajes, desactivar la cabecera en la respuesta, o forzar el protocolo de comunicación a interpretar. A continuación se muestran algunos comandos AT a modo de ejemplo:

- AT Z → Reinicia el dispositivo y muestra su nombre.
  Envío: atz
  Respuesta: ELM327 v1.5
- AT DP → Muestra el protocolo actual que se está interpretando.
  Envío: atdp
  Respuesta: AUTO, ISO 14230-4 (KWP FAST)

Si bien el sistema OBD-II cuenta con varios modos de funcionamiento, en este trabajo se hace hincapié en el modo 1 o de flujo de datos, el cual permite acceder mediante un PID (Parameter Identification Data) a datos analógicos o digitales de salidas y entradas a la Engine Control Unit (ECU). Existen dos conjuntos de PIDs: los estándar y los propietarios. La mayoría de los PIDs OBD-II en uso no son estándar, pero para los vehículos más modernos, hay muchas más funciones compatibles con la interfaz OBD-II que están cubiertos por los PID estándar, y hay relativamente menor superposición entre fabricantes de vehículos para PIDs propietarios. Si bien existen mas de treinta PIDs estándar, mediante los cuales es posible leer diferentes métricas del vehículo, en este trabajo se consideran solo cuatro para mostrar el funcionamiento del sistema y el método de lectura de parámetros.

Velocidad: El PID correspondiente a la solicitud de la velocidad es “0d” y la respuesta ocupa un byte. El rango válido va de 0 a 255 km/h y la fórmula para decodificar la velocidad es A, siendo A el valor de respuesta.

RPM (Revoluciones por minuto): El PID correspondiente a la solicitud de las revoluciones por minuto del motor es “0c” y la respuesta ocupa dos bytes. El rango válido va de 0 a 0 16.383,75 rpm y la fórmula para decodificar las RPM es $((A * 256) + B)/4$, siendo A y B el primer y segundo byte de la respuesta.

Posición del acelerador: El PID correspondiente a la solicitud de la posición del acelerador es “11” y la respuesta ocupa un byte. El rango válido va de 0 a 100% y la fórmula para decodificar esta posición es $A * 100/255$, siendo A el valor de respuesta.

Flujo de masa de aire: El PID correspondiente a la solicitud del flujo de masa de aire o MAF es “10” y la respuesta ocupa dos bytes. El rango válido va de 0 a 655,35 gramos/segundo y la fórmula para decodicicar MAF es $((A * 256) + B)/100$, siendo A y B el primer y segundo byte de la respuesta.

3.2 Aplicación movil

El módulo disponibilizador de datos se encarga de que los datos leídos de la computadora de a bordo OBD-II y del GPS estén disponibles para cualquier usuario en el vehículo con un dispositivo móvil Android con capacidad WiFi. La idea es

que el usuario se conecte a través de WiFi al Raspberry Pi como si este fuera un AP (Access Point) y se comunique a la dirección ip del Raspberry mediante un determinado puerto desde el browser del teléfono o desde una aplicación con un browser embebido.

La aplicación Android es b;asica. Tiene un browser embebido que se direcciona automáticamente a un puerto tpc establecido en la ip del Raspberry Pi conectado. Para el desarrollo de esta aplicación se utiliza un framework llamado Ionic-Framework el cual permite implementar aplicaciones para Android y iOS a partir de código HTML5 y código JavasScript, y que además soporta el uso de AngularJS, lo cual facilita el desarrollo. La aplicación solo cuenta con una pantalla en la que muestra los valores de los datos capturados: velocidad, rpm, posición del acelerador, flujo de masa de aire, latitud, longitud y tiempo, en ese orden.

4 Caso de prueba

El sistema desarrollado fue testeado en un Chevrolet Corsa Wagon Life Gls 1.4 4p 2009. Inicialmente se probaron los módulos de manera independiente y luego se procedió con las pruebas del sistema completo. Se debieron enfrentar diversos problemas de índole eléctrica. En primer lugar, debido a que el Raspberry Pi B sólo puede entregar una cantidad limitada de hasta 500mA a través de los puertos USB y que el módem 3G sólo ya produce picos de consumo de hasta 800mA, entonces es imposible alimentar el módem mediante el Raspberry Pi. Por lo tanto existen dos soluciones: la primera es utilizar un HUB USB alimentado el cual necesita una fuente adicional y la segunda es utilizar el nuevo modelo, Raspberry Pi 2B, el cual puede configurarse para entregar hasta 1.2A entre todos sus puertos USB. Por lo tanto, con una fuente de 2A o más es posible que el módem 3G y los demás dispositivos conectados funcionen correctamente sin necesidad de un HUB USB alimentado. En este trabajo se optó por utilizar una fuente adicional.

La captura que se ve en la Fig. 2 muestra los valores de los distintos parametros mientras el vehículo está detenido y cuando se conduce a baja velocidad dentro de la ciudad. El motor se encuentra funcionando a casi 2200 RPM mientras que la velocidad es de alrededor de 30 km/h. La posición del acelerador reporta un valor mayor que cero, porque durante la conducción se presiona el acelerador.

5 Conclusiones y Trabajos futuros

Se logró desarrollar e implementar un prototipo de un sistema que recolecta datos y los envia a un equipo remoto a través de Internet. Este prototipo, tanto en hardware como en software puede ser extendido.

El prototipo permite monitorear el sistema de a bordo en vehículos que cumplan con el estándar OBD-II para capturar diferentes métricas. Además permite obtener la geo-localización del vehículo mediante monitoreo satelital utilizando un GPS.

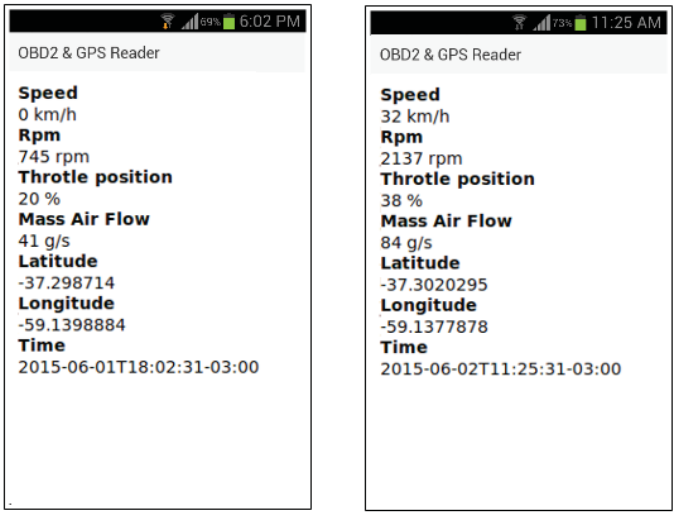

Fig. 2. Datos del vehículo encendido a velocidad cero y en marcha a baja velocidad.

Para enviar los datos, el sistema se conecta a Internet haciendo uso de redes telefónicas cuando la conexión este disponible; en caso de no tener conexión el sistema almacena los datos en la memoria interna hasta que pueda enviarlos.

Se desarrolló una aplicación movil que permite al usuario conectare al sistema de captura de datos para visualizar los datos que están siendo recolectados.

Particularmente, en este trabajo, el prototipo debe ir instalado adentro de un vehículo en funcionamiento, por lo cual encontrar y adaptar una fuente de alimentación confiable y suficiente fue complicado. Como se explicó anteriormente, en este trabajo se optó por utilizar una fuente adicional.

Finalmente, y en base a lo apreciado durante las pruebas, se obtuvo una idea general del nivel de impacto que se puede esperar del uso de este prototipo.

Si bien este trabajo se centra en desarrollar un prototipo, a partir de esta base se prevee que es posible realizar extensiones como: Mejorar la presentación de los datos en la aplicación movil; Capturar una mayor cantidad de métricas del motor del vehículo, de manera exhaustiva se pueden considerar todas las métricas estandarizadas para cualquier vehículo que soporte el protocolo OBD-II sin entrar en métricas específicas de cada marca; Se podría utilizar una batería independiente para alimentar el prototipo y así reemplazar la alimentación proveniente el automóvil. De esta forma se evita que el prototipo se apague si se detiene el vehículo o se corta el suministro electríco; Desarrollar un software que mediante diferentes algoritmos analice la información capturada e infiera datos más complejos tales como el desgaste del auto, el consumo promedio, reducción de las emisiones de gases, etc.

References

1. Ful-mar sa. http://www.ful-mar.com.ar/, accessed: 2017-07-21
2. Plugntrackgps. https://www.plugntrackgps.com/, accessed: 2017-07-21
3. Sitrack. https://www.sitrack.com, accessed: 2017-07-21
4. Aljaafreh, A., Khalel, M., Al-Fraheed, I., Almarahleh, K., Al-Shwaabkeh, R., Al-Etawi, S., Shaqareen, W.: Vehicular data acquisition system for fleet management automation. In: Vehicular Electronics and Safety (IC-VES), IEEE International Conference. pp. 130–133 (20011)
5. Corvalán, R., Osses, M., Urrutia, C., Barrientos, V.: Control de emisiones de fuentes móviles, informe final para la comisión nacional del medio ambiente. Tech. rep., Gobierno de Chile, Santiago de Chile (diciembre 2000)
6. Ding, Y., Chen, C., Zhang, S., Guo, B., Yu, Z., Wang, Y.: Greenplanner: Planning personalized fuel-efficient driving routes using multi-sourced urban data. In: 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 207–216 (March 2017)
7. Elm Electronics Inc.: Elm327 datasheet. Tech. rep., Elm Electronics Inc. (2008)
8. Henderson, B., Haynes, J.: OBD-II & Electronic Engine Management Systems. Haynes Repair Manuals, Haynes Manuals N. America, Inc. (2006)
9. Landin, C.A.M., Jimenez, U.Y.F.V.: Scanner Automotriz Interfaz PC. Master's thesis, Instituto Politécnico Nacional, Mexico D.F. (2010)
10. Lim, J., Choi, K.H., Kim, L., Lee, H.K.: Land vehicle positioning in urban area by integrated gps/beidou/obd-ii/mems imu. In: 2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE). pp. 176–180 (Aug 2016)
11. Lopes, J.C.O.: Diseño de un escáner automotriz OBDII Multi-protocolo. Master's thesis, Universidad de San Carlos de Guatemala (2014)
12. Malekian, R., Moloisane, N.R., Nair, L., Maharaj, B.T., Chude-Okonkwo, U.A.K.: Design and implementation of a wireless obd ii fleet management system. IEEE Sensors Journal 17(4), 1154–1164 (Feb 2017)
13. McCord, K.: Automotive Diagnostic Systems: Understanding OBD-I & OBD-II. S-A Design Workbench Series, CarTech (2011)
14. Meseguer, J.E., Toh, C.K., Calafate, C.T., Cano, J.C., Manzoni, P.: Drivingstyles: a mobile platform for driving styles and fuel consumption characterization. Journal of Communications and Networks 19(2), 162–168 (April 2017)
15. Ortenzi, F., Campbell, F., Zuccari, F., Ragona, R.: Experimental measurement of the environmental impact of a euro iv vehicle in its urban use. In: SAE Technical Paper 2007-01-0966 (2007)