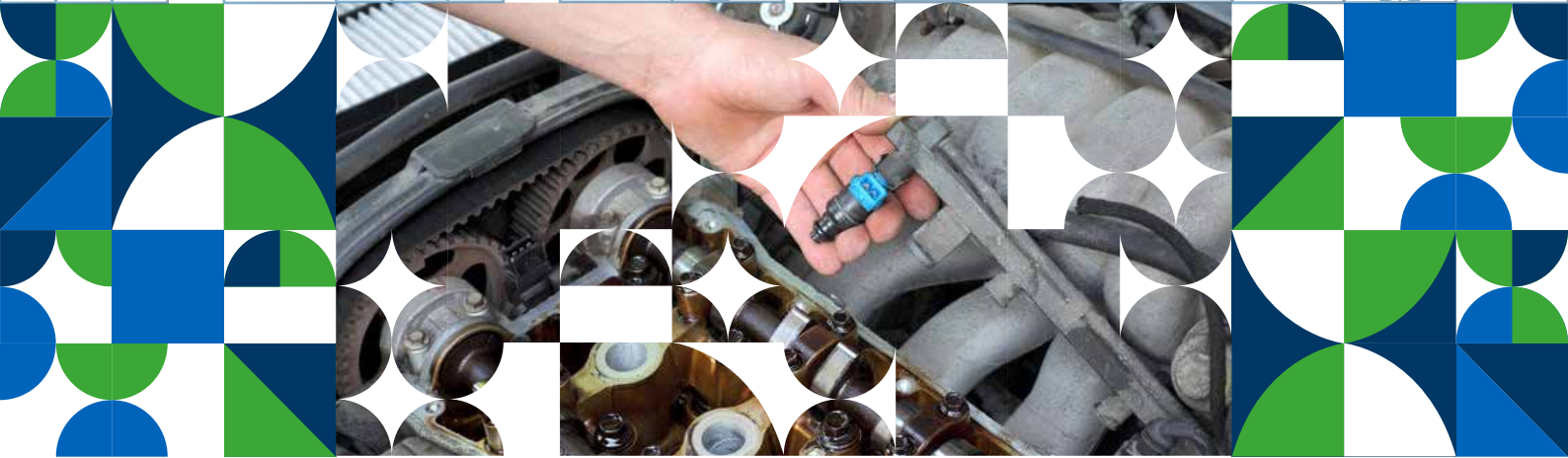
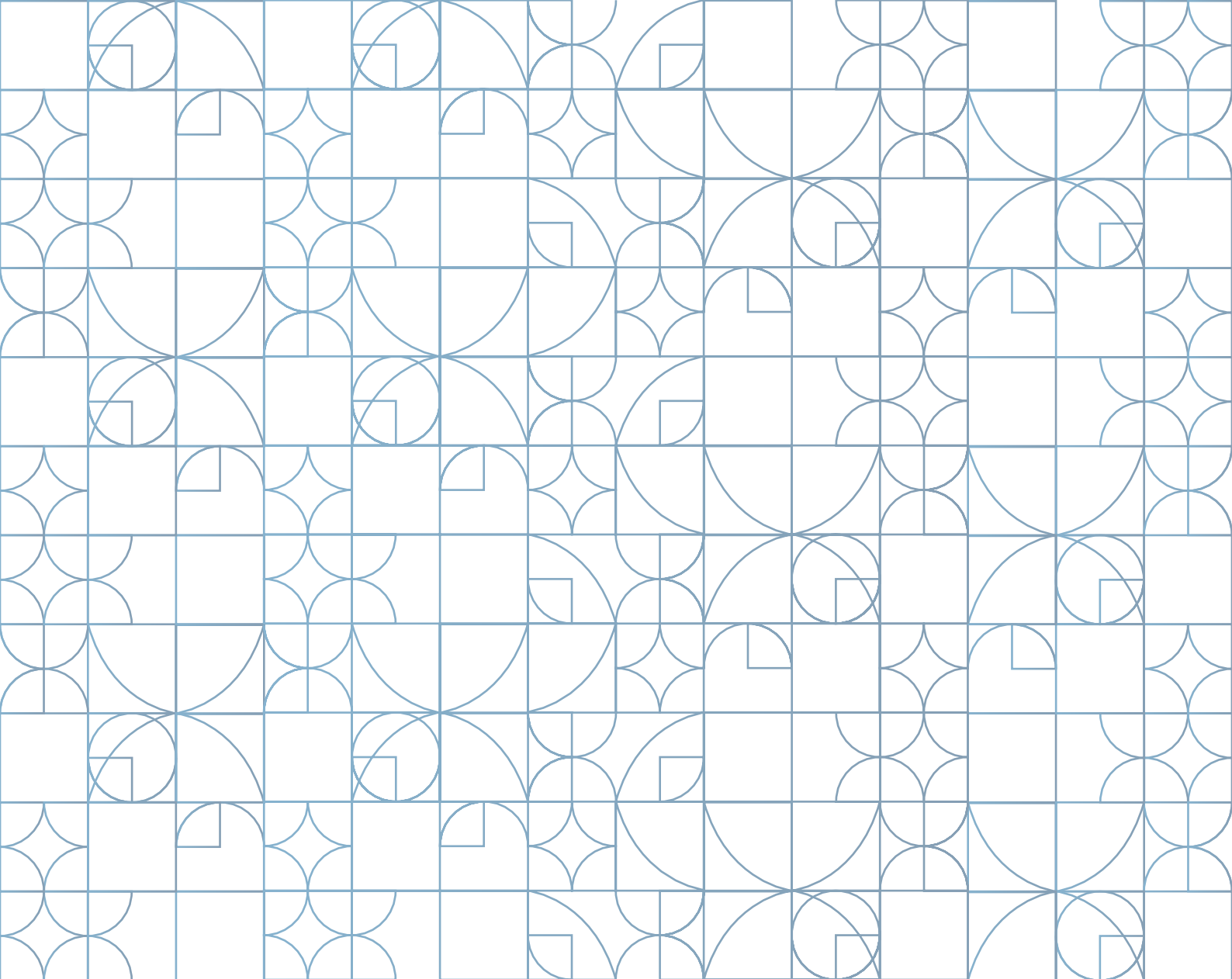




**METODOLOGÍA APLICADA PARA LECTURA, MODIFICACIÓN
Y ESCRITURA DE MAPAS DE INYECCIÓN EN MÓDULOS DE
CONTROL AUTOMOTRICES DE GESTIÓN DE MOTOR**

**APPLIED METHODOLOGY FOR READING, MODIFYING AND
WRITING INJECTION MAPS IN AUTOMOTIVE ENGINE
MANAGEMENT CONTROL MODULES**



METODOLOGÍA APLICADA PARA LECTURA, MODIFICACIÓN Y ESCRITURA DE MAPAS DE INYECCIÓN EN MÓDULOS DE CONTROL AUTOMOTRICES DE GESTIÓN DE MOTOR

APPLIED METHODOLOGY FOR READING, MODIFYING AND WRITING INJECTION MAPS IN AUTOMOTIVE ENGINE MANAGEMENT CONTROL MODULES

Álvaro Daniel Rodríguez Cuenca ¹,
Claudio Hernán Ortiz Molina ²,
Rómulo Fernando Astudillo Bravo ³,

¹ Instituto Superior Tecnológico del Austro, Ecuador, alvaro.rodriguez@instecclrg.edu.ec

² Instituto Superior Tecnológico del Austro, Ecuador, claudio.ortiz@instecclrg.edu.ec

³ Departamento de Investigación en Electrónica Automotriz, Grupo Automotriz en Video, Costa Rica, romulo@automotrizenvideo.com

RESUMEN

El presente trabajo aborda el tema de los sistemas de control electrónico en vehículos modernos, enfocándose en la gestión de los procesos, cálculo, y gobierno a través de Módulos de Control Electrónico (ECU). Se destaca que los sistemas de control utilizan una variedad de sensores y actuadores para interactuar con la ECU, la cual ejecuta programas de control almacenados en memorias especiales. El objetivo principal es proponer una metodología para extraer, modificar y escribir el software de los módulos de control electrónico de vehículos, permitiendo optimizar su funcionamiento según necesidades específicas del usuario.

El método se divide en varias etapas, que incluyen la selección del módulo de control por marca/modelo del vehículo y fabricante del microcontrolador, la determinación de la lectura por diferentes métodos como K-TAG, VVDI-Prog, etc., el respaldo y tratamiento de la información extraída, la modificación de archivos y consideraciones del checksum, y el proceso de escritura de los registros modificados en el módulo de control. La aplicación del método demuestra su efectividad para corregir problemas específicos de emisiones, utilizando herramientas de lectura y escritura de registros, así como estrategias de comparación y modificación de datos. El documento subraya la importancia crítica de gestionar eficazmente el software en los módulos de control electrónico de vehículos modernos, así como la necesidad de desarrollar y aplicar metodologías prácticas y efectivas para abordar estos desafíos en el contexto de la reparación y el mantenimiento automotriz.

Palabras claves: Modulo de control electrónico, microcontrolador, checksum, software, DAMOS.

ABSTRACT

This work addresses the topic of electronic control systems in modern vehicles, focusing on process management, calculation, and governance through Electronic Control Modules (ECU). It is highlighted that control systems use a variety of sensors and actuators to interact with the ECU, which executes control programs stored in special memories. The main objective is to propose a methodology to extract, modify and write the software of vehicle electronic control modules, allowing their operation to be optimized according to specific user needs.

The method is divided into several stages, including selection of control module by vehicle make/model and microcontroller manufacturer, determination of reading by different methods such as K-TAG, VVDI-Prog, etc.,



backup and treatment of the extracted information, file modification and checksum considerations, and the process of writing the modified records to the control module.

The application of the method demonstrates its effectiveness to correct specific emissions problems, using log reading and writing tools, as well as data comparison and modification strategies. backup and treatment of the extracted information, file modification and checksum considerations, and the process of writing the modified records to the control module.

The application of the method demonstrates its effectiveness to correct specific emissions problems, using log reading and writing tools, as well as data comparison and modification strategies. The paper highlights the critical importance of effectively managing software in modern vehicle electronic control modules, as well as the need to develop and apply practical and effective methodologies to address these challenges in the context of automotive repair and maintenance.

Keywords: Electronic control module, microcontroller, checksum, software, DAMOS.

Recibido: Agosto 2024
Received: August 2024

Aceptado: Diciembre 2024
Accepted: December 2024

1. INTRODUCCIÓN

En los vehículos distribuidos en la actualidad los fabricantes han implementado, cada vez más, sistemas que generan asistencia, control, monitoreo, gestión mediante un Módulo de Control Electrónico (*Electronic Control Unit - ECU*) que trabaja en conjunto con transductores (sensores) y actuadores. Los sensores y actuadores forman como periféricos la interfaz entre el vehículo y la ECU como unidad de procesamiento. Además, existen buses de datos de comunicación bidireccionales (p. ej., bus CAN), a través de los cuales se intercambia información con otras ECU o sensores [1]. Principalmente, el objetivo de dichos sistemas es el incremento de la eficiencia energética, disminución de la complejidad en la distribución eléctrica (número de líneas de cable y conexiones), reducción del peso y la adaptación del vehículo a distintas condiciones topográficas, operativas, ambientales relacionadas con su circulación [2]. Los requisitos impuestos a las ECU están estandarizados en ISO 16750 [3]. También existen normas empresariales para todos los fabricantes de automóviles.

Para la gestión de los procesos, cálculo, gobierno, entre otros las ECUs utilizan una amplia gama de sistemas de control tanto de lazo abierto como de lazo cerrado, estos con base a parámetros que describen las condiciones de funcionamiento del mecanismo de interés, se procesan mediante uno o varios microcontroladores ejecutando el programa de control (*software*) que permite mediante una etapa de potencia, el control de los actuadores. El programa de control se almacena en una memoria especial del sistema y el hardware de la ECU lo ejecuta, procesando los algoritmos necesarios que permiten el correcto funcionamiento del sistema [4]. Generalmente el software de la ECU se almacena ya sea en memorias internas o externas al microcontrolador, dichas memorias son del tipo no volátil (ROM, PROM, EPROM, Flash EEPROM y EEPROM) y en dependencia de su tipo la información almacenada en éstas se puede o no borrar, además un módulo de control requiere para las operaciones de cálculo y guardado de resultados de memorias volátiles (SRAM y DRAM) las cuales no tienen una capacidad permanente de retención de su información, pues, ésta se elimina al retirar su alimentación eléctrica (ver Figura 1).

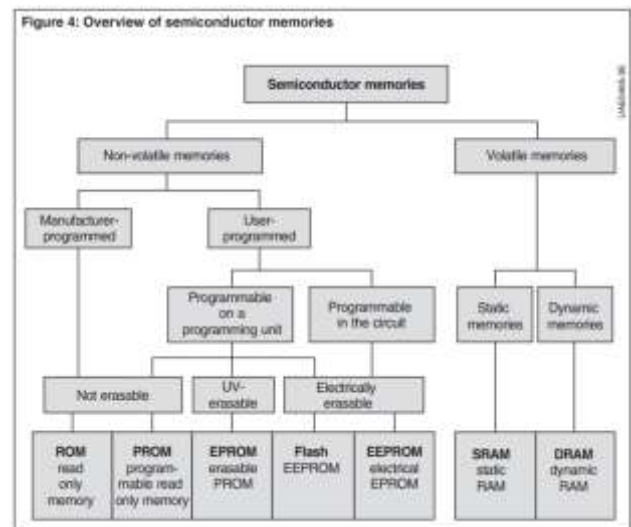


Figura 1. Tipos de memorias semiconductoras [5]

El software puede ser revisado por el fabricante y someterse a procesos de actualización para mejorar su desempeño, corregir averías, agregar funciones, entre otros. Además, en función de las prestaciones que incorpora el vehículo se pueden habilitar o desactivar características del mismo, como por ejemplo cuando la versión de un automóvil incorpora transmisión manual o automática, esta se puede gestionar por una PCM (*Powertrain Control Module*) incorporando el hardware necesario para la gestión del motor y la transmisión, pero, en el caso de una transmisión manual se deshabilita en su software el gobierno de la transmisión mediante la deshabilitación en la EEPROM, modificando su programación EoL (*End of Line*).

En muchos módulos de control, en el software almacenado, se pueden incorporar parámetros operativos relacionados con características propias y únicas de un vehículo (generalmente guardados en una memoria EEPROM y también en la Flash EEPROM), por citar algunas se pueden mencionar su VIN (*Vehicle Identification Number*), algoritmos del sistema antirrobo (IDs de transpondedores, *Crypto key*, *PIN Code*), características operativas del sistema como el número de arranques, SOH y SOC de la batería para el sistema de carga inteligente y el *start /stop*, equipamiento del vehículo según su programación EoL y muchas funciones más que variarán en número y complejidad en dependencia del fabricante y el modelo del automóvil.

La información registrada en las memorias ROM, PROM, EPROM, FLASH EEPROM, EEPROM de un módulo de control se almacena en formato

hexadecimal en una estructura tipo matricial con ordenamientos de filas y columnas; cada carácter hexadecimal contiene un nibble de información (4 bits) y los registros se pueden agrupar “palabras” en formatos de 8 bits, 16 bits y 32 bits en dependencia de la forma en la que trate la información el fabricante. En la Figura 2 se aprecia la información de la memoria EEPROM del microcontrolador de una ECM Denso Z601 con un formato de 8 bits utilizando 2 caracteres hexadecimales por cada posición de fila y columna (1 Byte).

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	FF	5A	27	7A	2D	7C	84	DC	02	46	FF	01	00	00	E7	40
00000010	FF	33	00	00	FF	00	03	D7	2C	A5	A3	01	A6	5A	FF	28
00000020	FF	26	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00000030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00000040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00000050	30	30	32	30	31	30	FF	30	FF	FF	2A	FF	FF	FF	FF	FF
00000060	3D	03	FF	FF	FF	FF	41	FF	FF	FF	FF	FF	FF	FF	FF	FF
00000070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00000080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00000090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000000A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000000B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000000C0	FF	FF	9F	0E	30	FF	FF	34	5A	FF	56	FF	35	35	30	
000000D0	FF	FF	34	5A	34	20	34	35	33	39	FF	34	FF	41	FF	30
000000E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000000F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Figura 2. Información contenida en una la memoria EEPROM del MCU de una ECM Denso Z601

Generalmente, cuando los fabricantes realizan una revisión / actualización de la cartografía con la que opera un módulo de control, la información de las memorias EPROM, FLASH EEPROM, EEPROM se modifica, a través de una reprogramación Flash (*Flash reprograming*) y la nueva versión de software se la conoce generalmente como una calibración. Su implementación se sustenta en un estudio de operatividad del sistema que se gestiona, mejorando su desempeño o corrigiendo averías, cuando esto se ha realizado se genera un Boletín Técnico de Servicio (TSB) donde se describen las optimizaciones realizadas y el proceso para la carga de dicha calibración en un módulo de control.

Con relación a lo mencionado con anterioridad, la dificultad que se presenta en el ámbito de la reparación, mantenimiento y diagnóstico de los sistemas automotrices con pertinencia a la temática citada radica principalmente en la falta de accesibilidad a las herramientas que tratan el software de los módulos de control automotrices, el alto costo de estas e incluso los bloqueos regionales para su compra y acceso, haciendo que el servicio de actualización sea casi exclusivo del centro autorizado por el fabricante, generando inconvenientes cuando se necesita cambiar un

módulo dañado, cargar su software, actualizarlo y programar su configuración EOL.

Además, en el Ecuador, el poder adquisitivo de la población es limitado si se compara con los países desarrollados, donde los repuestos y vehículos tienen un costo inferior y el ingreso promedio por persona es mayor, la necesidad del usuario que demanda de los técnicos la reparación de módulos de control (a pesar de que los fabricantes de automóviles no lo recomienden) es evidente y en algunos casos dichos procesos se complementan con la modificación de su software, justificando ante esto la necesidad generar una metodología para tratar dicha información.

Inicialmente, se plantea una metodología que permita tratar la información de los módulos de control de motor, resaltando que al poder modificar el mismo se puede optimizar el funcionamiento de un vehículo adaptándolo a una necesidad específica del usuario, ya sea ésta en mejoramiento de las prestaciones asociadas a la potencia o la economía de combustible. Además de poder cargar revisiones de software sin herramienta oficial del fabricante.

La aplicación de dicha metodología se realizará sobre la PCM de un vehículo JAC S5 2020 con problemas de emisiones para la mejora del desempeño, dicha reparación en la actualidad incluye averías relacionadas con el software de las ECUs, donde es relevante poder extraer el mismo de las memorias / microcontroladores, respaldarlo, modificarlo (en el caso de ser necesario) y escribirlo nuevamente utilizando equipos dedicados para este uso (programadores).

2. MATERIALES Y MÉTODOS

En el mercado se ofertan equipos que vulneran la seguridad de los componentes de interés (memorias y microcontroladores) permitiendo su lectura y escritura, entre los ofertados para la extracción de información de ECUs de motor (ECM), tren motriz (PCM) y caja de cambios (TCM) se pueden citar: K-Tag, Kess y Powergate de Alientech, New Genius y New Trasdata de Dimsport, Flex de MagicMotorsport, VVDI-Prog (oferta limitada de modelos de ECM) de Xhorse. Siendo relevantes para esta investigación los equipos K-Tag, Kess y VVDI-Prog.

A su vez la información leída puede modificarse mediante software de uso específico para esta

aplicación, los programas más comunes en el mercado con este fin son WinOLS de EVC Electronic, ECM Titanium de Alientech, Race de Dimsport, siendo relevante para esta investigación la modificación mediante WinOLS.

Respecto a la generación de una metodología que permita la extracción de la información, su modificación y posterior escritura en el módulo de control se aporta por parte de los investigadores el desarrollo de una secuencia ordenada de procedimientos que dictaminan si es o no factible dicho proceso, el cual se describe en las siguientes subsecciones de este apartado.

A. Selección del módulo por marca / modelo del vehículo, familia del módulo de control y fabricante del microcontrolador

Las herramientas citadas con anterioridad permiten una selección del módulo en función de diversos parámetros, en términos generales se categorizan las ECUs en función del fabricante del vehículo, su modelo y familia de ECM, por citar un ejemplo práctico con el software K-Suite utilizado por K-Tag y Kess con referencia en la Figura 3 se aprecia que para el vehículo del fabricante Peugeot cuyo modelo es 207 del año 2009 se puede seleccionar una ECM de la marca Bosch cuya familia es MED 17.4.



Figura 3. Selección de ECM perteneciente a un vehículo Peugeot 207 2009 en el software K-Suite [6]

También se puede seleccionar el módulo en función del fabricante del Microcontrolador que equipa, por citar algunos de los que más se encuentran en ECMs y PCMs del campo automotriz están: ST, Renesas, Infineon, Mitsubishi, NEC, Motorola, entre otros.

B. Determinación de lectura por puertos JTAG, BDM, Boot Mode, K-Line, CAN Bus, línea

dedicada, extracción del componente o desde el DLC

Los fabricantes de los vehículos pueden permitir o no la extracción y escritura de la información de una ECM, PCM o TCM directamente desde el DLC (puerto OBDII o SAE J1962), por lo que no llega a ser necesario retirar la ECU del vehículo, generalmente, los procesos citados se realizan por K-Line en módulos antiguos y CAN Bus en ECUs actuales. Entre las herramientas que pueden realizar una lectura y escritura de esta forma se destacan Kess, Flex y New Genius, pero, es importante resaltar que en el proceso se debe respaldar la alimentación de la batería mediante un equipo dedicado (estabilizador de tensión) como se muestra en la Figura 4.



Figura 4. Extracción de información de ECM Transtron 8982107100 de Isuzu DMax 3.0 2016 mediante Kess, respaldo de batería con estabilizador de tensión Schumacher INC-100

Los módulos pueden incorporar en su PCB (*Printed Circuit Board*) puertos de uso específico para la lectura y escritura de la información presente en sus memorias / microcontrolador, estas terminales se conocen como puertos JTAG (*Joint Test Action Group*) o BDM (*Background Debug Mode*) como se aprecian en la Figura 5, para la lectura/escritura se puede soldar líneas a los terminales aportados por la herramienta o utilizar un adaptador específico en conjunto con una mesa BDM (Figura 6).



Figura 5. Puerto BDM en PCB de ECM Bosch MED 17.4 de Peugeot 207.



Figura 6. Conexión a ECM Bosch MED 17.4 de Peugeot 207 mediante puerto BDM con adaptador 14P600KT04 utilizado en K-Tag [7].

Ciertos microcontroladores / memorias requieren de una conexión externa para permitir la lectura y escritura de su información, la cual se suele indicar en el software de la herramienta utilizada (Figura 7), a través de lo citado el componente estaría en Boot Mode, es importante considerar que la lectura / escritura se hace mediante conexiones directas a sus terminales (microcontrolador / memoria), uso de un puerto BDM / JTAG o conexión a la bornera del módulo de control mediante K-Line, CAN Bus o línea dedicada (Figura 8), por lo que no resulta suficiente la conexión única del terminal del Boot Mode (Figura 9).



Figura 7. Puntos de conexión a MCU Infineon TriCore TC1766 de ECM Delphi MT-60 para lectura por CAN Bus utilizando VVDI-Prog [8].



Figura 8. Lectura de ECM Hitachi A56-Y32 de Nissan X-Trail 2010 soldando líneas dedicadas utilizando K-Tag.



Figura 9. Conexión para Boot Mode de MCU Infineon TriCore TC1766 de ECM Delphi MT-60 correspondiente a Chevrolet Sail 2013, lectura posterior por CAN Bus mediante VVDI-Prog.

La conexión de elementos directamente en la PCB no siempre es una opción viable, en ocasiones resulta necesario la extracción del elemento de interés (Figura 10) para colocarlo en un sócalo / adaptador específico (Figura 11), es importante recalcar que en estos procesos el usuario debe evaluar la dificultad en el soldado / desoldado del componente, pues la funcionalidad del módulo de control puede estar comprometida si el usuario no posee la pericia para la ejecución del proceso, además se debe considerar la limpieza de la superficie de contacto y el aislamiento térmico de los elementos en la periferia de la memoria / microcontrolador cuando se desmonta y monta este utilizando calor.



Figura 10. Desoldado de memoria EEPROM ST9501K de ECM Híbrida Delphi HV-340 de Chevrolet Corsa Evolution 2006 para posterior lectura con VVDI-Prog



Figura 11. Lectura de memoria EEPROM 93C66 extraída de ECM Hitachi A56-Y32 de Nissan X-Trail 2010 mediante VVDI-Prog.

C. Respaldo y tratamiento previo de la información

Es importante destacar que se debe asegurar la integridad de la información extraída de un módulo de control se requieren realizar varias lecturas del microcontrolador / memoria en cuestión, se utiliza software específico para comparar la información de la lectura número 1 y la lectura número 2 asegurando su integridad si estas son iguales, entre los programas que pueden cumplir la actividad mencionada se citan: UltraCompare, UltraEdit, Batronix Prog-Express Express, Software de VVDI-Prog, entre otros. La información debe guardarse con parámetros de interés que identifiquen al módulo leído como pueden ser placa del vehículo, año modelo/ serie, VIN, marca y familia de ECM, defecto previo, defecto corregido, procesos realizados o cualquier parámetro que permita identificar los registros (Figura 12).

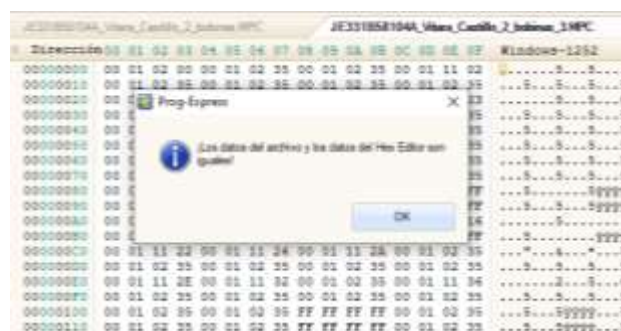


Figura 12. Comparación de registros de dos lecturas realizadas en módulo Mitsubishi E6T13 de un Suzuki Gran Vítara 2007.

D. Modificación de archivos y consideraciones del checksum

Los registros de los módulos de control pueden estar protegidos ante procesos de modificación por parte del fabricante, en el campo de la telecomunicación, transmisión de datos y programación una de las estrategias más comunes de uso para verificar la integridad de la información es el cálculo de la suma de comprobación o *checksum* de su data [9], dicho proceso consiste en aplicar un algoritmo que utilice total o parcialmente la información de la memoria o el microcontrolador aportando un resultado esperado, el cual, se somete a comparación con el registro de *checksum* grabado en la memoria y asegura que los registros no han sido modificados; cabe destacar que en la data pueden existir algunas zonas de comprobación de *checksum* de la información.

Las herramientas de lectura y escritura pueden identificar dentro de su base de datos (presente en su software) las zonas de corrección de *checksum* y ajustarlas para mantener la coherencia en estos registros, los programadores K-Tag, Kess de Alientech y New Trasdata de Dimsport tienen la facultad de realizar este proceso, pero, también hay herramientas que no permiten dicha corrección cuando se escriben archivos modificados en la memoria / microcontrolador como lo son Flex de Magicmotorsport, VVDI-Prog de X-Horse y New Genius de Dimsport, por lo cual, es imperativo que previo a la escritura se realicen correcciones de *checksum* con software especializado para este uso, por citar un caso mediante WinOLS se pueden comprar *checksum points* aplicables a un grupo de familias de módulos de control que permiten la corrección previa de dichos registros antes de su escritura.

Es importante destacar también que la información extraída debe ser segmentada y delimitada, pues, dentro de esta se almacenan registros asociados a cartografía, DTCs, programación EoL, registros particulares del vehículo (VIN, kilometraje / millaje, aprendizajes, por mencionar algunos). En WinOLS se puede cargar sobre los registros de lectura de una memoria / microcontrolador un registro DAMOS (*Database for Microcontroller Oriented Systems*) [4], el cual delimita y segmenta los registros, siendo de carácter relevante para su uso considerar la versión de hardware y software para la que es aplicable el DAMOS en cuestión (Ver Figura 13), generalmente los archivos mencionados tienen un costo y se adquieren de proveedores externos.

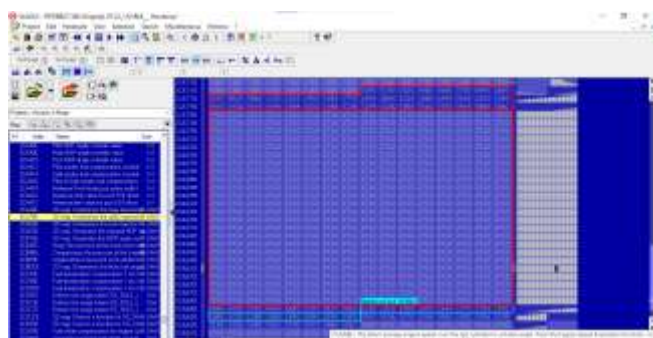


Figura 13. DAMOS aplicado a registros de tractocamión Peterbilt 386 en WinOLS, enmarcado en rojo se delimita un mapa de cartografía y a la izquierda se lista.

Los registros modificados se pueden exportar como

un archivo independiente y su extensión varía en función de la herramienta que se utiliza para leer y escribir la información en el módulo de control, las extensiones de uso común en K-Tag y Kess son: MPC para registros del microcontrolador, EPR para registros de la memoria EEPROM, FLS para los registros de la Flash EEPROM, MAPAS para registros de cartografía que permiten su trabajo en WionOLS (solo en Kess) y registros *Backup* que son respaldos comprimidos de toda la información del elemento leído (Figura 14).

Nombre	Tipo	Tamaño
SID904 Backup	Archivo	603 KB
SID904.EPR	Archivo EPR	4 KB
SID904.FLS	Archivo FLS	2,048 KB
SID904.MAPAS	Archivo MAPAS	2,560 KB
SID904.MPC	Musepack Audio ...	513 KB

Figura 14. Extensiones de registros leídos de una ECM Siemens SID904 utilizando Kess [10].

E. Proceso de escritura de registros

La escritura de la información en un módulo de control debe considerar la secuencia ordenada de pasos aportada en el proceso de lectura, es relevante definir la acción a realizar con los archivos pues en el caso de una clonación de módulos de control resulta suficiente la escritura de registros sin modificar, o inclusive solamente de los que tengan relación con la programación EoL.

En el caso de registros modificados se debe necesariamente evaluar los aportes con relación a corrección de *checksum* y extensiones pertinentes al elemento de escritura, además de comprobar el tamaño del registro, que debe ser igual al original.

3. RESULTADOS

El presente método se aplica en una reparación de un vehículo marca JAC, modelo S5 del año 2020 que incorpora una ECM marca Delphi MT-80, con un sistema de inyección indirecta a gasolina que gobierna un motor Turbo Green Jet 2.0.

Con referencia en el informe técnico de la avería anexo en su apartado "Etapa 1: Verificar el Reclamo o Problema", se detalla que el vehículo genera altas emisiones de Monóxido de Carbono (CO) en prueba de ralentí acelerado en RTV (2500

rpm) con base a NTE INEN 2204, después de la revisión correspondiente de sistemas asociados a la avería se identifica que el sistema de inyección no considera la señal del S_{1B_1} de O_2 para regular el Ajuste de Combustible al Corto Plazo (*Short Therm Fuel Trim* - STFT) ni el Ajuste de Combustible al Largo Plazo (*Long Therm Fuel Trim* - LTFT) a pesar de que se cumple la condición para entrar en condición operativa de Lazo Cerrado (*Closed Loop*) según PIDs aportados en la línea de datos de la ECM verificado con equipo de diagnóstico (Figura 15).



Figura 15. Registros de PIDs de ECM Delphi MT-80 de vehículo JAC S5 2020 asociados a STFT, LTFT, $S_{1B_1}O_2$ y condición operativa en Lazo cerrado.

Con base en el apartado “Etapa 5: Reparación del problema” del informe técnico anexo se evidencia que cuando las revoluciones se incrementan se puede ver que el $S_{1B_1}O_2$ retroalimenta a la ECM sobre una dosificación con riqueza en la mezcla aire combustible, su tensión llega aproximadamente a 0,93V, lo cual consecuentemente según el comportamiento típico de los motores de combustión la gasolina generará un incremento de las emisiones de CO, HC; además, de esto se repercutirá en la reducción del coeficiente Lambda (<1), la cantidad de O_2 y CO_2 (la reducción de este gas indica una combustión deficiente, pero, que dicha condición disfuncional se corrige en el vehículo al accionar el aire acondicionado pues se modifican las condiciones de carga sobre el motor y se varía el STFT según el aporte operativo mencionado en [11] por Martin (ver Figura 16).

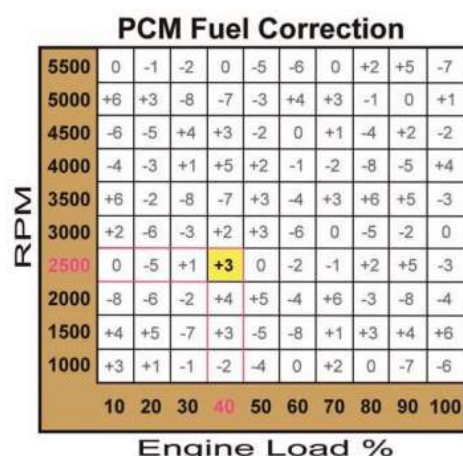


Figura 16. Mapa general de valores adaptativos de ajuste de combustible STFT en función del régimen de giro del motor y su carga [11].

Según información aportada por el centro de servicio autorizado de la marca, en una investigación previa con referencia a la avería, un lote de los vehículos en cuestión no gestiona correctamente el ajuste de combustible a corto plazo al régimen de giro y condiciones de carga de la prueba en ralentí acelerado según RTV bajo NTE INEN 2204 (2500 rpm, sin carga con vehículo estacionario), siendo necesaria una actualización de los registros de la Flash EEPROM, pero, la carga de la nueva calibración no es factible debido a la indisponibilidad del equipo necesario para el proceso.

Cabe destacar que en una campaña de corrección de la avería en cuestión en el pasado el centro de servicio autorizado actualizó parte de la flota con el problema mediante personal autorizado de fábrica, y que de dicho proceso se posee una ECM que almacena la calibración mencionada.

En primera instancia se lee la información de la ECM mencionada con dos equipos, K-Tag y VVDI-Prog, la información de la Flash EEPROM del microcontrolador Infineon Tricore TC1766 se segmenta en dos, PFlash (*Program Flash*) y DFlash (*Data Flash*), sus pesos según información técnica del fabricante son de 1504KB y 32KB respectivamente [12], para el proceso de lectura se debe realizar una conexión para el *Boot Mode* la cual se aprecia en la Figura 17 con referencia al software VVDI-Prog, la lectura se da mediante CAN Bus.



Figura 17. Conexión de módulo Delphi MT-80 V3 para lectura y escritura por CAN Bus y Boot Mode con referencia al so VVDI-Prog [8].

No se dispone de un DAMOS para el vehículo en cuestión que permita la segmentación de la información de la DFlash en partes y mapas para migrar los correspondientes, por lo que mediante estrategias de comparación entre registros se modifica la información distinta entre archivos resaltando que, no se puede migrar toda la información principalmente porque en ella existen registros asociados al sistema antirrobo del vehículo lo cual inhabilita su arranque inmovilizándolo, la comparación de registros se observa en la Figura 18 y para la misma se utiliza el software de VVDI-Prog.

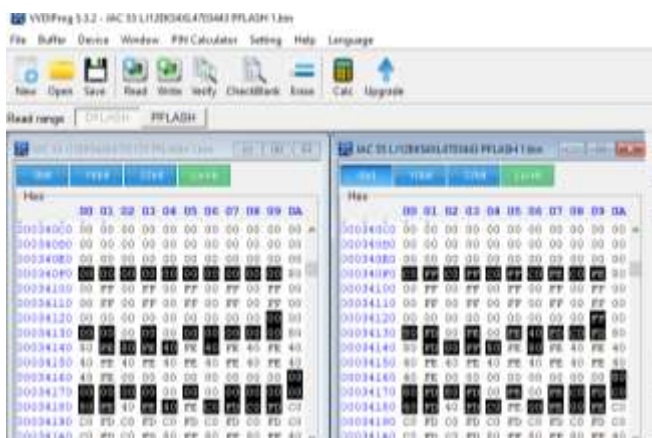


Figura 18. Comparación de registros entre versiones de ECMs, diferencias marcadas en negro

Mediante la migración progresiva de datos se escriben los registros en el módulo utilizando K-Tag, ya que al no tener la facultad de corrección de *Checksum* VVDI-Prog el vehículo inhabilita el arranque debido a incoherencias en dicho parámetro, la modificación por partes del registro de la DFlash de la ECM mencionada corrige la avería con referencia a la emisión de CO y HC en prueba estática en ralentí acelerado según NTE INEN 2204 detallado en el informe técnico anexo.

4. CONCLUSIÓN

El aumento en la complejidad de los sistemas de vehículos contemporáneos, junto con la implementación de sistemas de control electrónico, destaca la importancia de gestionar eficazmente el software de los módulos de control electrónico para garantizar un rendimiento óptimo y cumplir con los estándares de emisiones.

La dificultad para acceder y modificar el software de los módulos de control electrónico, debido a barreras como el alto costo de las herramientas especializadas y los bloqueos regionales, subraya la necesidad de desarrollar metodologías accesibles y eficaces para la reparación y el mantenimiento de vehículos.

El proceso de extracción, modificación y escritura de software en los módulos de control electrónico presenta una serie de desafíos técnicos, como la selección adecuada de parámetros, la manipulación física de componentes electrónicos y la corrección de errores de checksum. Estos desafíos resaltan la importancia de contar con personal capacitado y herramientas adecuadas.

La aplicación del método propuesto en la reparación de un vehículo específico demuestra su efectividad para corregir problemas específicos de emisiones, lo que resalta la relevancia práctica de la metodología desarrollada y su capacidad para abordar problemas reales en la industria automotriz.

El documento subraya la importancia crítica de gestionar eficazmente el software en los módulos de control electrónico de vehículos modernos, así como la necesidad de desarrollar y aplicar metodologías prácticas y efectivas para abordar estos desafíos en

el contexto de la reparación y el mantenimiento automotriz.

5. REFERENCIAS

- [1] H. Randoll, Elektronische Steuerung von Dieselmotoren, Wiesbaden, 2016.
- [2] . F. Landhäußer, A. Michalske, M. L. Susaeta, . M. Grosser, M. Heinzelmann, J. Feger, L.-M. Fink, W. Gerwing, G. Klaus , B. Illg, J. Kurz, R. Mayer, D. Ottenbacher, A. Werner, J. Wiesner y M. Walther, Electronic Diesel Control (EDC), Wiesbaden : Springer , 2014.
- [3] ISO, «ISO 16750: Road vehicles – Environmental conditions and testing for electrical and electronic equipment.,» vol. 3, p. 13, 2018.
- [4] F. Landhäußer, A. Michalske, M. L. Susaeta, M. Grosser, M. Heinzelmann, J. Feger, L.-M. Fink, W. Gerwing, G. Klaus, B. Illg, J. Kurz, R. Mayer y D. Ottenbacher, Electronic Control Unit (ECU), Wiesbaden: Springer, 2014.
- [5] K.-H. Dietsche y . K. Reif , Automotive Handbook, Karlsruhe: Robert Bosch GmbH, 2022.
- [6] Alientech Srl, «Selección de ECM en K-Suite,» Alientech Srl, Trino, 2012.
- [7] Alientech Srl, «Adaptador 14P600KT04,» Alientech Srl, Trino, 2012.
- [8] Shenzhen Xhorse Electronics Co., Ltd., «Diagrama de conexión para lectura de ECM Delphi MT-60,» Shenzhen Xhorse Electronics Co., Ltd., Shenzhen , 2024.
- [9] M. Barr, Programming Embedded Systems in C and C++, California: O'Really & Associates, Inc., 1999.
- [10] R. Pintado, «Curso de Reprogramación en Vehículos Automotores con WinOLS,» Instituto de Tecnología Automotriz Avanzada ITAA, Montevideo, 2020.
- [11] T. Martin , How to use Automotive Diagnostic Scanners, Minneapolis: Motorbooks, 2015.
- [12] Infineon Technologies AG, Data Sheet, V1.0 TC1766 32-Bit Single-Chip Microcontroller TriCore, Múnich: Infineon Technologies AG, 2008.