



# **UNIVERSIDAD TÉCNICA DEL NORTE**

## **FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ**

### **TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MANTENIMIENTO AUTOMOTRIZ**

#### **TEMA:**

**REPROGRAMACIÓN DE FUNCIONES ESPECIALES EN VEHÍCULOS  
MODERNOS APLICANDO INGENIERÍA INVERSA A REGISTROS DE  
MEMORIAS Y MICRO CONTROLADORES DE MÓDULOS AUTOMOTRICES**

**AUTORES: FABRICIO RAFAEL ARCINIEGA GARCÍA  
JIMY ALEXANDER FUELTAN FUELTAN**

**DIRECTOR: ING. RAMIRO ROSERO, MSc.**

**IBARRA, OCTUBRE 2020**

## CERTIFICADO

### ACEPTACIÓN DEL DIRECTOR

En mi calidad de director del plan de trabajo de grado, previo a la obtención del título de ingeniería en Mantenimiento Automotriz, nombrado por el Honorable Consejo Directivo de la Facultad de Ingeniería en Ciencias Aplicadas.

### CERTIFICO:

Que una vez analizado el plan de grado cuyo título es “Reprogramación de funciones especiales en vehículos modernos aplicando ingeniería inversa a registros de memorias y micro controladores de módulos automotrices” presentado por los señores: Fabricio Rafael Arciniega García con número de cédula 040196463-0 y Jimmy Alexander Fultán Fultán con número de cédula 050375022-6 doy fe que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte de los señores integrantes del jurado examinador que se designe.

En la ciudad de Ibarra, a los 02 días del mes de Octubre del 2020

Atentamente



Ing. Ramiro Andrés Rosero Añazco MSc.

**DIRECTRO DEL TRABAJO DE GRADO**



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN**  
**A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

**1. IDENTIFICACIÓN DE LA OBRA**

En cumplimiento del Art. 144 de la ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CEDULA DE IDENTIDAD:</b>	050375022-6		
<b>APELLIDOS Y NOMBRES:</b>	Fueltan Fueltan Jimy Alexander		
<b>DIRECCIÓN:</b>	Eusebio Borrero y Antonio José de Sucre – Ibarra		
<b>EMAIL:</b>	jafueltanf@utn.edu.ec		
<b>TELÉFONO FIJO</b>		<b>TELÉFONO MÓVIL:</b>	0980839024
DATOS DE CONTACTO			
<b>CEDULA DE IDENTIDAD:</b>	040196463-0		
<b>APELLIDOS Y NOMBRES:</b>	Arciniega García Fabricio Rafael		
<b>DIRECCIÓN:</b>	Hernán González de Saa y Sánchez y Cifuentes – Ibarra		
<b>EMAIL:</b>	fbarciniegag@utn.edu.ec		
<b>TELÉFONO FIJO</b>		<b>TELÉFONO MÓVIL:</b>	0962664184
DATOS DE LA OBRA			
<b>TÍTULO:</b>	REPROGRAMACIÓN DE FUNCIONES ESPECIALES EN VEHÍCULOS MODERNOS APLICANDO INGENIERÍA INVERSA A REGISTROS DE MEMORIAS Y MICRO CONTROLADORES DE MÓDULOS AUTOMOTRICES.		
<b>AUTOR (ES):</b>	Fabricio Rafael Arciniega García, Jimy Alexander Fueltan Fueltan		
<b>FECHA:</b>	02 de octubre del 2020		
SOLO PARA TRABAJOS DE GRADO			
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/>	<b>PREGRADO</b>	<input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA</b>	INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ		
<b>ASESOR/DIRECTOR</b>	Ing. Ramiro Andrés Rosero Añazco MSc.		

## 2. CONSTANCIA

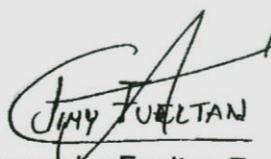
Los autores manifiestan que la obra objeto de la presente autorizacion es original y se desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asumen la responsabilidad sobre el contenido de la misma y saldra de defensa de la Universidad en caso de reclamacion por parte de terceros.

Ibarra, a los 02 dias del mes de Octubre del 2020

### AUTORES:



Fabricio Rafael Arciniega García  
040196463-0



Jimy Alexander Fueltan Fueltan  
050375022-6

## DEDICATORIA

El presente trabajo de titulación se lo dedico en primer lugar a Dios por darme la fuerza e iluminar mi camino día tras día, por darme el valor de lucha constante y permitirme el haber llegado a este punto de mi formación profesional.

A mis padres José y Carmen, quienes me han brindado su apoyo incondicional en todo el proceso de mi carrera universitaria, por forjarme valores de respeto, responsabilidad y disciplina, por ser el amor más puro que una persona puede encontrar en un ser humano y demostrarme siempre su cariño.

A mis hermanos Mauricio, Carolina y Patricio, por todo su cariño y apoyo para lograr mis objetivos, por haberme dado unos hermosos sobrinos quienes son mi fuente de inspiración para luchar todos los días.

A mis amigos, pero sobre todo de manera especial a Stalin y Milena por creer en mí, por su apoyo constante y por su motivación a seguir adelante.

*“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado, un esfuerzo total es una victoria completada”*

**JIMY FUELTAN**

## DEDICATORIA

El presente trabajo de grado es dedicado con mucho cariño y orgullo a mis padres Rafael Arciniega y Gladys García, ya que gracias al apoyo incondicional de ellos he logrado, cumplir esta meta que me propuse, quienes con tanto esfuerzo han hecho lo imposible para que nunca me falte nada, es un orgullo ser su hijo.

A mis hermanos, Antony por su apoyo incondicional, por ser un hermano ejemplar y siempre estar pendiente de la familia. Con mucho cariño y nostalgia a mi hermano Damián, en tu partida dejaste un vacío irreparable en nosotros, sé que desde cualquier lugar que te encuentres nos estas cuidando, y a mi hermano Jhan, ya que eres mi motivación para ser cada día una mejor persona y un ejemplo para ti.

A toda mi hermosa familia, en especial a mi tía Margot por apoyarme y estar ahí como una segunda madre y a mi abuelita Inelda por ser un ejemplo en mi vida, por su amor y apoyo en todo momento. Y como no agradecer a mi novia Rosalie, por su amor incondicional y sus palabras de aliento que han logrado que esta meta se cumpla.

**FABRICIO ARCINIEGA**

## AGRADECIMIENTO

Agradezco a Dios por guiar mi camino y mantenerme con salud en cada paso transcurrido en mi formación como profesional, a mi familia por el apoyo incondicional, por los recursos necesarios para conseguir mis objetivos y por los consejos brindados para ser una buena persona ante la sociedad.

Además, una enorme gratitud a la Universidad Técnica del Norte, en especial a la carrera de Ingeniería en Mantenimiento Automotriz, por darme la oportunidad de ser parte de esta familia, forjarme valores de ética y formarme como persona y profesional, a los docentes por aportarme un sin fin de conocimientos que sin duda alguna me servirán en todas las áreas que desempeñare a futuro.

Un enorme agradecimiento al Ing. Andrés Cevallos MSc., por brindarnos su apoyo, su conocimiento y todos los equipos necesarios para cumplir con nuestro trabajo sin ningún contratiempo, de igual manera, agradecer a José Luis Guerrero un buen amigo y compañero por su apoyo en este trabajo de titulación.

También, quiero agradecer de manera especial a nuestro director de tesis al Ing. Ramiro Rosero MSc., por su paciencia, sus conocimientos y sobre todo su buena predisposición para ayudarnos en la realización de este proyecto de grado, y a los asesores, Ing. Fausto Tapia MSc., e Ing. Jorge Melo MSc., por contribuir y ser partícipes en todo este trabajo.

*“Los retos son lo que hacen la vida interesante, y superarlos es lo que hace que la vida tenga un significado”*

**JIMY FUELTAN**

## **AGRADECIMIENTO**

Agradezco infinitamente a mis padres quienes han sido el eje principal de superación en toda mi vida, por haberme brindado la oportunidad de estudiar una carrera universitaria, por su esfuerzo, dedicación, y entera confianza. A mi novia por darme palabras de aliento y no dejar que me rinda. Y a toda mi familia por brindarme su apoyo moral e incondicional.

Mi agradecimiento al Ingeniero Andrés Cevallos por brindarnos todo su apoyo, conocimientos y equipos para poder realizar la investigación, a mi primo Bryan Cazares y a José Luis Guerrero por facilitarnos sus vehículos para realizar las pruebas pertinentes.

Mi agradecimiento sincero a la Universidad Técnica del Norte, la Facultad de Ingeniería en Ciencias Aplicadas, y en especial a mi querida Carrera de Ingeniería en Mantenimiento Automotriz, quien me brindo sólidos conocimientos para mi formación profesional.

Agradezco categóricamente a mis docentes por transmitirme todos sus conocimientos y sabidurías, de manera especial al Ing. Ramiro Rosero Msc, tutor del presente trabajo, y a mis dos asesores, Ing. Fausto Tapia Msc, e Ing. Jorge Melo Msc, por contribuir y ser partícipes en esta tesis.

**FABRICIO ARCINIEGA**

## ÍNDICE DE CONTENIDOS

	<b>PÁGINA</b>
<b>RESUMEN</b>	<b>XIX</b>
<b>ABSTRAC</b>	<b>XX</b>
<b>INTRODUCCIÓN</b>	<b>XXI</b>
<b>CAPITULO I</b>	<b>1</b>
<b>1. REVISIÓN BIBLIOGRÁFICA</b>	<b>1</b>
1.1 Objetivos	1
1.1.1 Objetivo general	1
1.1.2 Objetivo específicos	1
1.2 Justificación	2
1.3 Alcance	3
1.4 Antecedentes	4
1.5 Unidades de control electrónico (ECU)	5
1.5.1 Componentes de una ECU	6
1.5.2 Mapa de ECU	7
1.6 Módulos automotrices	9
1.6.1 Módulo de control electrónico del motor ECU-ECM	9
1.6.2 Módulo de control de la carrocería BCM	11
1.6.3 Módulo de control del Inmovilizador INMO BOX	13
1.6.4 Módulo del panel de instrumentos electrónico	17
1.7 Memorias y Microcontroladores	18
1.7.1 Microcontrolador	18
1.7.2 Memoria	24
1.8 Tipos de programación	31
1.8.1 Reprogramación flash	32
1.8.2 Reprogramación ECU estándar	33
1.8.3 Reprogramación ECU programable	33

1.9 Protocolos de Comunicación	33
1.9.1 CAN (Controller Area Network)	34
<b>CAPITULO II</b>	<b>42</b>
<b>2. MATERIALES Y MÉTODOS</b>	<b>42</b>
2.1 Metodología de investigación	42
2.1.1 Enfoque Investigativo	42
2.1.2 Tipo de investigación	43
2.2 Procesos metodológicos	44
2.3 Materiales y equipos	46
2.3.1 X100 pad 2 pro	46
2.3.2 Upa-usb	47
2.3.3 K-tag	48
2.3.4 Can bus analyzer	48
2.3.5 Mini vvdi key tool	49
2.4 Proceso de identificación de funciones especiales	49
2.4.1 Flujograma para la identificación de funciones especiales	50
2.4.2 Identificación de funciones especiales x100 pad 2 pro	50
2.5 Proceso para obtener el archivo de volcado	52
2.5.1 Flujograma para la obtención de archivo de volcado	53
2.5.2 Archivo de volcado ECU	53
2.5.2 Archivo de volcado BCM y panel de instrumentos	56
2.6 Proceso de obtención de pin code de transponder de llaves	59
2.6.1 Flujograma para la obtención de obtención de pin code	60
2.6.2 Obtención código transponder de llave	61
2.7 Proceso de evaluación archivo de volcado	62
2.7.1 Flujograma para la evaluación de archivo de volcado	62
2.8 Proceso de infiltración en la red can	65
2.8.1 Flujograma para la infiltración en la red can	65

<b>CAPITULO III</b>	<b>68</b>
<b>3. RESULTADOS Y DISCUCIÓN</b>	<b>68</b>
3.1 Funciones especiales en modulos automotrices	68
3.1.1 Identificación funciones especiales x100 pad 2 pro	68
3.1.2 Reseteo de funciones en vehículo seleccionado	78
3.2 Obtención archivos de volcado	85
3.2.1 Modulo de control electrónico ECU	86
3.2.2 Módulo de control de la carrocería BCM	94
3.2.3 Panel de instrumentos electrónico	98
3.3 Obtención pin code y transponder de llaves	105
3.3.1 Obtención código de seguridad pin code	105
3.3.2 Obtención código de transponder de llaves	107
3.3.2 Programación de llave	109
3.4 Evaluación de archivos de volcado	115
3.4.1 Análisis archivo de volcado ecu	116
3.4.2 Comparación de archivos de volcado de la ECU antes y después de un reset	120
3.4.3 Análisis archivo de volcado panel de instrumentos	136
3.4.4 Análisis archivo de volcado bsi	144
3.4.4 Comparación de archivos de volcado BSI	148
3.5 Infiltración en la red can	154
3.5.1 Obtención códigos red can	154
3.5.2 Reprogramación del ángulo del acelerador - kia sportage R	157
3.5.3 Análisis de códigos de programación	161
3.5.4 Validación códigos de reprogramación	164
<b>CAPITULO IV</b>	<b>166</b>
<b>4. CONCLUSIONES Y RECOMENDACIONES</b>	<b>166</b>
4.1 Conclusiones	166
4.2 Recomendaciones	168
<b>BIBLIOGRAFÍA</b>	<b>170</b>

**ANEXOS**

## ÍNDICE DE TABLAS

TABLA NÚM.	DESCRIPCIÓN	PÁGINA
1.1	Ventajas y desventajas Lenguaje Basic	21
1.2	Ventajas y desventajas Lenguaje C	22
1.3	Ventajas y desventajas Lenguaje ensamblador	23
1.4	Tipos de protocolos de comunicación	34
1.5	Tabla de variantes con dos simples bits en el protocolo CAN	40
1.6	Tabla de variantes con múltiples combinaciones de bits en el protocolo CAN	41
2.1	Especificaciones técnicas de vehículos para caso práctico.	46
3.1	Origen y marca de vehículos - Función Reset luz de servicio	70
3.2	Origen y marca de vehículos - Función ajuste sensor ángulo de dirección	71
3.3	Origen y marca de vehículos - Función Reset presión del neumático	73
3.4	Origen y marca de vehículos - Función de ajuste de sensor de acelerador	74
3.5	Origen y marca de vehículos - Función BMS Reset	75
3.6	Origen y marca de vehículos - Función freno eléctrico de parking	76
3.7	Origen y marca de vehículos - Función freno eléctrico de parking	78
3.8	Características generales ECU	86
3.9	Configuración de códigos colores cable 14P600KT02	90
3.10	Resumen conexiones cable 14P600KT02	91
3.11	Transformación de numero PIN a valor hexadecimal	116
3.12	Transformación Pin Code de alfanumérico a hexadecimales	117
3.13	Análisis bytes memoria eeprom ST95160 antes y después Reset luz de servicio	123
3.14	Análisis bytes memoria eeprom ST95160 antes y después del ajuste del sensor de aceleración	130
3.15	Registros cíclicos	135
3.16	Datos del kilometraje en valores hexadecimales	141
3.17	Datos de kilometraje en valores decimales	141
3.18	Ejemplo de inicio de kilometraje	143
3.19	Algoritmo del cálculo de kilometraje	144
3.20	Número VIN de alfanumérico a hexadecimales	146
3.21	Representación del kilometraje en la BSI Valeo	150
3.22	Bytes de antes y después de la reprogramación de llaves	152
3.23	Códigos emitidos de la RED CAN	163

## ÍNDICE DE FIGURAS

<b>FIGURA NÚM.</b>	<b>DESCRIPCIÓN</b>	<b>PÁGINA</b>
1.1	División de bloques de una ECU	6
1.2	Mapa de la Unidad Electrónica de Control	8
1.3	Vista de valores hexadecimal de un mapa de Inyección	8
1.4	Modulo de control electrónico ECU	9
1.5	ECUs que intercambian información con BCM	12
1.6	Esquema de un sistema Inmovilizador	14
1.7	Antena conectada al Módulo del Inmovilizador	15
1.8	Componentes de un microcontrolador	19
1.9	Arquitectura Van Neumann	20
1.10	Arquitectura Harvard	21
1.11	Memoria eeprom	26
1.12	Memoria eprom	27
1.13	Memoria tipo DIL	27
1.14	Memoria tipo PLCC	28
1.15	Memoria tipo SOP	29
1.16	Nomenclatura de memorias por colores	30
1.17	Conexión sin CAN BUS	35
1.18	Conexión con CAN BUS	35
1.19	Interconexión de múltiples unidades de control en el vehículo	36
1.20	Simulación trama de bits con bombillas apagadas (0) y encendidas (1)	39
1.21	Bit con valor uno	39
1.22	Bit con valor cero	40
2.1	Proceso metodológico de obtención de datos y proceso de reprogramación	45
2.2	X100 pad 2 pro - Version Full	47
2.3	Programador Upa-usb	47
2.4	Programador K-tag	48
2.5	Programador Can bus analyzer Microchip	48
2.6	Mini vvddi key tool	49
2.7	Proceso metodológico para la identificación de funciones especiales	50
2.8	Simbología funciones especiales X100 pad 2 pro	51
2.9	Conexión X100 pad 2 pro con el vehículo	51
2.10	Proceso metodológico para la obtención de archivos de volcados	53
2.11	K-tag Version master 7.020	54
2.12	Conexión al circuito impreso de la centralita	54
2.13	Conectores ECU	55
2.14	Carcasa ECU	55
2.15	Conexión K-tag mediante puerto USB	55
2.16	Menú principal módulo K-tag	56

<b>2.17</b>	Módulo UPA versión V 1.3	57
<b>2.18</b>	Adaptadores programador UPA	57
<b>2.19</b>	Ubicación módulo de control BCM	58
<b>2.20</b>	Panel de instrumentos electrónico	58
<b>2.21</b>	Menú principal equipo UPA	59
<b>2.22</b>	Proceso metodológico para la obtención de pin code	60
<b>2.23</b>	Equipo Mini vvdi key	61
<b>2.24</b>	Interface Mini vvdi	61
<b>2.25</b>	Proceso metodológico para la evaluación de archivos de volcado	62
<b>2.26</b>	DUMP de archivo de volcado	63
<b>2.27</b>	Programa “fairdell hex cmp”	64
<b>2.28</b>	Menú principal HexCmp	64
<b>2.29</b>	Proceso metodológico para la infiltración en la red can	65
<b>2.30</b>	Herramienta Can bus analyzer	66
<b>2.31</b>	Conexiones para el proceso de obtención de códigos	66
<b>2.32</b>	Ventana de trabajo Can bus Analyzer	67
<b>3.1</b>	Señal - Reset luz de servicio	69
<b>3.2</b>	Señal - Ángulo de dirección	70
<b>3.3</b>	Señal - Reset presión neumático	72
<b>3.4</b>	Señal - Ajuste sensor de aceleración	73
<b>3.5</b>	Señal - BMS Reset	75
<b>3.6</b>	Señal - Freno eléctrico de parking	76
<b>3.7</b>	Señal - DPF	77
<b>3.8</b>	Menú funciones especiales – Reset luz de servicio	79
<b>3.9</b>	Menú selección origen del vehículo - Reset luz de servicio	79
<b>3.10</b>	Menú selección marca del vehículo - Reset luz de servicio	80
<b>3.11</b>	Menú reinicio manual del aceite del motor	80
<b>3.12</b>	Menú selección modelo del vehículo - Reset luz de servicio	81
<b>3.13</b>	Instrucciones reseteo luz de servicio	81
<b>3.14</b>	Menú funciones especiales – Ajuste sensor de acelerador	82
<b>3.15</b>	Menú selección origen del vehículo – Ajuste sensor de acelerador	83
<b>3.16</b>	Menú selección marca del vehículo - Ajuste sensor de acelerador	83
<b>3.17</b>	Menú selección marca del vehículo - Ajuste sensor de acelerador	84
<b>3.18</b>	Selección sensor TPS	84
<b>3.19</b>	Vehículo Peugeot 307 año 2003	86
<b>3.20</b>	Centralita ECU	87
<b>3.21</b>	Menú K-tag - Opción protocolos	88
<b>3.22</b>	Menú protocolos - Selección de ECU	88
<b>3.23</b>	Puntos de conexión	89
<b>3.24</b>	Conexión punto BOOT ECU	89
<b>3.25</b>	Cable 14P600KT02	90
<b>3.26</b>	Puntos de conexión cable 14P600KT02	91
<b>3.27</b>	Conexión K-TAG con la centralita	91

<b>3.28</b>	Opciones de Lectura y escritura	92
<b>3.29</b>	Hardware y Software ECU	92
<b>3.30</b>	Proceso de lectura	93
<b>3.31</b>	Archivos extraídos ECU	93
<b>3.32</b>	Microcontrolador MC68HC912DG128	94
<b>3.33</b>	Centralita electrónica BSI	95
<b>3.34</b>	Identificación de color de pines - Puerto BD9	95
<b>3.35</b>	Posición de soldadura de los cables	96
<b>3.36</b>	Conexión UPA y BSI	96
<b>3.37</b>	Selección tipo de microcontrolador	97
<b>3.38</b>	Lectura microcontrolador	97
<b>3.39</b>	Archivos extraídos BSI	98
<b>3.40</b>	Posición de agujas panel de instrumentos	99
<b>3.41</b>	Centralita panel de instrumentos	99
<b>3.42</b>	Memoria EEPROM 25C020	100
<b>3.43</b>	Aplicación de estaño en centralita del panel de instrumentos	100
<b>3.44</b>	Extracción de memoria eeprom del panel de instrumentos	101
<b>3.45</b>	Adaptador SOIC a DIP	101
<b>3.46</b>	Montaje adaptador SOIC a DIP al adaptador UPA	102
<b>3.47</b>	Conexión memoria eeprom y programador UPA	102
<b>3.48</b>	Menú UPA opción abrir hoja nueva	103
<b>3.49</b>	Selección memoria EEPROM 25C020	103
<b>3.50</b>	Menú UPA selección READ	104
<b>3.51</b>	Archivos extraídos panel de instrumentos	104
<b>3.52</b>	Menú inmovilizador x100 pad 2 pro	105
<b>3.53</b>	Selección read security code	106
<b>3.54</b>	Pin code vehículo Peugeot 307	106
<b>3.55</b>	Llave insertada en el equipo Mini vvdI	107
<b>3.56</b>	Interface Mini vvdI – Opción transponder, reconoce y clona	108
<b>3.57</b>	Código de transponder	108
<b>3.58</b>	Interface Mini vvdI – Opción generar transponder	109
<b>3.59</b>	Selección código ID46 PCF7936	110
<b>3.60</b>	Selección marca del vehículo - Mini vvdI	110
<b>3.61</b>	Colocación chip virgen en equipo VVDI	111
<b>3.62</b>	Generar transponder	111
<b>3.63</b>	Selección marca del vehículo – Menú inmovilizador	112
<b>3.64</b>	Selección Program keys – Menú inmovilizador	112
<b>3.65</b>	Selección modelo del vehículo – Menú inmovilizador	113
<b>3.66</b>	Selección modo de comunicación	113
<b>3.67</b>	Opción program keys	114
<b>3.68</b>	Opción ingresar PIN CODE	114
<b>3.69</b>	Ubicación Pin Code en memoria EEPROM ST95160	117
<b>3.70</b>	Tamaño archivo de volcado de la memoria EEPROM ST95160	118

<b>3.71</b>	Tamaño archivo de volcado de la memoria Flash AM29F400BB Mapas	119
<b>3.72</b>	Tamaño archivo de volcado de la memoria Micro ST10C167	120
<b>3.73</b>	Tamaño del archivo Memoria eeprom ST95160 antes y después del Reset luz de servicio	121
<b>3.74</b>	Comparación de datos o bytes de la memoria eeprom ST95160 antes y después del Reset luz de servicio	122
<b>3.75</b>	Tamaño del archivo Memoria Flash AM29F400BB Mapas antes y después del Reset luz de servicio	126
<b>3.76</b>	Comparación archivo de volcado de la memoria Micro ST10C167 antes y después del Reset luz de servicio	127
<b>3.77</b>	Tamaño del archivo Memoria eeprom ST95160 antes y después del ajuste del sensor de aceleración	128
<b>3.78</b>	Comparación de datos o bytes de la memoria eeprom ST95160 antes y después del ajuste del sensor de aceleración	129
<b>3.79</b>	Comparación archivo de volcado de la memoria Flash AM29F400BB Mapas antes y después del ajuste del sensor de aceleración	133
<b>3.80</b>	Comparación archivo de volcado de la memoria Micro ST10C167 antes y después del ajuste del sensor de aceleración	134
<b>3.81</b>	Tamaño archivo de volcado panel de instrumentos	137
<b>3.82</b>	Lecturas realizadas memoria eeprom 25C020	137
<b>3.83</b>	Lectura de archivos de 274 746 km y 274 825 km	138
<b>3.84</b>	Comparación entre archivos	139
<b>3.85</b>	Representación del kilometraje	140
<b>3.86</b>	Ubicación Pin Code-archivo de volcado BSI	145
<b>3.87</b>	Ubicación del número VIN en el archivo de volcado BSI	146
<b>3.88</b>	Tamaño del archivo de volcado BSI	147
<b>3.89</b>	Lecturas realizadas módulo BSI	148
<b>3.90</b>	Ubicación Kilometraje BSI lectura 1	149
<b>3.91</b>	Archivo de volcado de antes y después de la programación de llaves	151
<b>3.92</b>	Vehículo Kia Sportage R	155
<b>3.93</b>	Conexión Can bus analyzer y x100 pad 2 pro	155
<b>3.94</b>	Software Can bus Analyzer	156
<b>3.95</b>	Ventana Fixed trace	156
<b>3.96</b>	Comunicación con valores hexadecimales	157
<b>3.97</b>	Menú funciones especiales – Reset ajuste del ángulo del acelerador	158
<b>3.98</b>	Menú selección origen del vehículo	158
<b>3.99</b>	Menú selección marca del vehículo	159
<b>3.100</b>	Menú selección TPS (CLEAR LEARNING VALUE)	159
<b>3.101</b>	Menú selección modelo vehículo	160
<b>3.102</b>	Menú selección año del vehículo	160
<b>3.103</b>	Menú selección tipo de vehículo	161
<b>3.104</b>	Códigos hexadecimales de programación.	162
<b>3.105</b>	Códigos de reprogramación	165

## ÍNDICE DE ANEXOS

<b>ANEXO NÚM.</b>	<b>PÁGINA</b>
1. Ficha técnica equipos de programación	179
2. Archivos de volcado ECU	184
3. Archivo de volcado BSI	187
4. Archivo de volcado panel de instrumentos	188
5. Códigos emitidos en la RED CAN	189

## RESUMEN

En el presente trabajo de investigación se realizó la reprogramación de funciones especiales de los vehículos Peugeot 307 y Kia Sportage R, en donde, se examinó los registros internos de las memorias y microcontroladores de la ECU, BSI y panel de instrumentos con la implementación de equipos de programación como K-TAG, UPA-USB y X100 PAD 2 PRO. Para el desarrollo de este proyecto hemos dividido nuestro trabajo en tres capítulos los cuales se detalla a continuación:

El capítulo I presenta los objetivos trazados, el planteamiento del problema, la justificación y antecedentes de la investigación. Además, se abarca los principios y fundamentos teóricos de módulo automotrices, memorias, microcontroladores, tipos de programación y todo lo relacionado a las redes de comunicación CAN. El capítulo II trata sobre los materiales y métodos utilizados en el estudio. Se da a conocer los tipos de investigación, procesos metodológicos que indican los pasos a seguir para cumplir con los objetivos planteados. El capítulo III, enmarca los resultados obtenidos del proceso práctico realizado, del primer vehículo mencionado se determinó que contaba con funciones como Reset luz de servicio y ajuste del ángulo del sensor de aceleración. Una vez realizado el reseteo de cada función se extrajo el archivo de volcado de cada módulo para examinar los datos hexadecimales y verificar la ubicación en la que se encuentran. También, se extrajo el PIN CODE del vehículo para verificar que las lecturas realizadas a los diferentes módulos son correctas y necesario para realizar reprogramación de llaves proceso que se realizó para determinar los valores que se modifican al momento de generar una nueva llave. Posteriormente en el segundo vehículo con ayuda del CAN BUS Analyzer y X100 PAD 2 PRO, se realizó la infiltración a la red CAN para interpretar en tiempo real la interacción de la ECU con equipos externos. De este proceso se logró determinar la comunicación que realiza la computadora con equipos externos, y se obtuvo los códigos de reprogramación para realizar un ajuste del ángulo del sensor del acelerador del vehículo. Finalmente, del estudio realizado se pudo obtener dos métodos para realizar reprogramación a funciones especiales, el primer método es mediante la obtención de códigos hexadecimales, escribirlos en el archivo de volcado de la memoria y reprogramar, y un segundo método consiste en detectar los códigos de reprogramación emitidos en la red CAN, escribir y enviar los códigos con una herramienta desarrolladora y depuradora. Todos los datos y valores encontrados fueron determinantes para brindar fiabilidad al estudio realizado.

## **ABSTRACT**

The present investigation studies the reprogramming of special functions of the vehicles Peugeot 307 and Kia Sportage r, where the internal registries of the memories and microcontrollers of the ECU, BSI and instrument panel were examined by implementing the programming equipments K-TAG, UPA-USB and X100 PAD 2 PRO. For the development of this project the content is divided in three chapters which are detailed below.

Chapter I presents the objectives, the problem statement, the justification and background of the research. In addition, it covers the principles and theoretical foundations of automotive modules, memories, microcontrollers, programming types and everything related to CAN communication networks. Chapter II deals with the materials and methods used in the study. The types of research, methodological processes that indicate the steps to be followed in order to achieve the proposed objectives are presented. Chapter III, frames the results obtained from the practices. The first vehicle mentioned resulted to have functions such as Reset service light and adjustment of the angle of the acceleration sensor. Once the reset of each function was performed, the dump file of each module was extracted to examine the hexadecimal data and verify the location of the data. Also, the vehicle's PIN CODE was extracted to verify that the readings made to the different modules are correct and necessary to perform key reprogramming to determine the values that are modified at the time of generating a new key. Later, in the second vehicle with the help of the CAN BUS Analyzer and X100 PAD 2 PRO, the infiltration to the CAN network was performed to interpret in real time the interaction of the ECU with external equipment. This process made it possible to determine the communication between the computer and external equipment, and to obtain the reprogramming codes to adjust the angle of the vehicle's accelerator sensor. Finally, the study revealed two methods to perform reprogramming of special functions. The first method is by obtaining hexadecimal codes, writing them in the memory dump file and reprogramming them. The second method is to detect the reprogramming codes issued in the CAN network and to write and send the codes with a developer and debugger tool. All data and values found were decisive to provide the reliability of the study.

## INTRODUCCIÓN

La industria automotriz está pasando por una revolución histórica con la implementación de nuevos productos en automóviles que ha llevado a un avance tecnológico al campo de la ingeniería dejando en el pasado el uso de componentes mecánicos gracias a la aplicación de la electrónica, autotrónica, software y programación. En base a este contexto se considera necesaria la investigación referente a reprogramación de funciones especiales en módulos automotrices como pilar fundamental al perfeccionamiento de las unidades de control. En los últimos años los vehículos traen integrado distintas funciones especiales como: inmovilizadores, códigos de motor, frenos inteligentes, centrales electrónicas, bolsas de aire, luz de servicio de aceite, luz de cinturón de seguridad entre otros. Para procesos de reprogramación del trabajo de estas funciones no se puede realizar mecánicamente o con equipos de diagnóstico básico por lo que es necesario de nuevas herramientas tecnológicas. En este sentido los equipos de reprogramación profesional como X100 Pad 2 Pro, UPA-USB, K-TAG, CAN BUS ANALYZER, entre otros; permiten indagar en los registros de memorias y microcontroladores de los diferentes módulos de control como la ECU, BCM, INMO BOX, y modulo del panel de instrumentos para la obtención y análisis de valores hexadecimales que permiten realizar las funciones antes mencionadas. El presente proyecto investigativo tiene como idea principal la obtención de datos del archivo de volcado de las memorias automotrices para romper la barrera de restricción entre la información que pertenece a los fabricantes de automóviles y equipos de programación profesionales, permitiendo que en estudios futuros se pueda diseñar un dispositivo que realice el mismo trabajo y que tenga un valor económico más accesible a los ya existentes en el mercado.

# CAPITULO I

## 1. REVISIÓN BIBLIOGRÁFICA

### 1.1 OBJETIVOS

#### 1.1.1 OBJETIVO GENERAL

Reprogramar funciones especiales en vehículos modernos aplicando ingeniería inversa a registros de memorias y micro controladores de módulos automotrices.

#### 1.1.2 OBJETIVO ESPECÍFICOS

- Identificar las funciones especiales de vehículos modernos, características, efectos y procesos de reprogramación con la implementación de equipos de programación profesional.
- Obtener el archivo de volcado de los registros de memorias y micro controladores de la unidad de control electrónica (ECU), módulo de control de la carrocería (BCM) y panel de instrumentos para inspeccionar los valores hexadecimales modificados al realizar funciones especiales.
- Explorar las memorias de la Inmo Box para extraer el PIN CODE, el número VIN del vehículo y códigos crypto de transponder de llaves.
- Evaluar los registros del archivo de volcado de la ECU, BCM y panel de instrumentos antes y después de la reprogramación aplicando ingeniería inversa a los procesos relacionados a las funciones especiales.
- Identificar los códigos emitidos en la red CAN del vehículo que realizan las funciones especiales del vehículo.

## 1.2 JUSTIFICACIÓN

El presente proyecto está orientado a la aplicación de ingeniería inversa a registros de memorias y micro controladores de módulos automotrices en vehículos modernos con la utilización de equipos de programación profesionales.

El campo de la ingeniería automotriz ha evolucionado ampliamente, precedentemente se tomaba en cuenta solo la parte mecánica, posteriormente paso a la aplicación de la electrónica y actualmente debido a grandes avances tecnológicos se está implementado diferentes software y programas de control automotriz. Cabe recalcar que dicho software es reprogramable, entonces, la reprogramación de funciones especiales a módulos automotrices es un eje para corregir defectos de programación original o introducir perfeccionamientos a las unidades de control.

Los equipos de programación profesional son instrumentos de escaneo de datos, basados en nuevas tecnologías de difícil acceso por su costo elevado y la información sobre el funcionamiento de los equipos es escasa o se encuentra restringida, lo que propone con este estudio es un método para llegar a descubrir que funciones cumple en las memorias y microcontroladores de automóviles e identificar los datos, códigos y valores hexadecimales que permiten realizar estas funciones en los vehículos.

La Encyclopædia Británica (2019), dice que la tecnología es un concepto que se asocia recurrentemente a la innovación, definida como la “aplicación del conocimiento científico a los objetivos prácticos de la vida humana, o como a veces se expresa, al cambio y la manipulación del ambiente humano”.

El actual trabajo de investigación será un avance para fabricar equipos de programación profesionales en Ecuador y que puedan cumplir con funciones especiales, el proyecto se sustenta en lo acordado en el objetivo 5 del Plan Nacional de Desarrollo (TODA UNA VIDA) que hace referencia a: “Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria” (Senplades, 2017, pág. 80). Y a la política 5,6; que toma en cuenta como punto de partida a la matriz productiva del país, incentivando a las universidades y al sector público a formar profesionales que sean capaces de realizar investigaciones de carácter tecnológico, para aportar al cambio en la producción y economía, generando desarrollo, bienestar y progreso.

El enfoque del proyecto está encaminado a la transferencia tecnológica y al desarrollo, la innovación y el emprendimiento ya que se está realizando una investigación en el campo de la electrónica y software automotriz con la reprogramación de memorias del automóvil con el uso de herramientas tecnológicas. La idea de este estudio es romper la barrera de restricción de información que solo pertenece a los fabricantes de automóviles y equipos de programación profesionales, es decir, su análisis va a estar apoyado en un software de ingeniería que permitirá obtener códigos y datos reales que sean modificables en los archivos de volcado de memorias y microcontroladores, aspectos que son determinantes al momento de fabricar equipos de diagnóstico automotriz que sean de un costo económico y accesible.

### **1.3 ALCANCE**

En el presente proyecto se realizará la reprogramación de funciones especiales en vehículos modernos aplicando ingeniería inversa a registros de memorias y micro controladores de módulos automotrices con la implementación de equipos de programación profesional. Para lo cual, se buscará la información necesaria de los vehículos seleccionados, identificando las funciones especiales, características y efectos.

Para el análisis de datos se utilizara los equipos de reprogramación UPA, CARPROG, IPROG y KTAG, con el cual se obtendrá el archivo de volcado de los registros de memorias y micro controladores de la ECU y BCM para examinar los valores hexadecimales que sean modificables, de igual manera se va a inspeccionar la memoria del panel de instrumentos para analizar sus registros reales y las memorias de la Inmo Box para extraer el PIN CODE, el número VIN y códigos crypto de transponder de llaves y finalmente se identificara los códigos emitidos en la red CAN con la utilización de CAN BUS ANALYZER y MICROCHIP.

Con los datos obtenidos se realizará comparaciones antes y después de la reprogramación aplicando ingeniería inversa a los procesos relacionados a las funciones especiales y así se pueda diseñar en estudios futuros de investigación un dispositivo que ejecute un trabajo similar y que tenga un costo accesible, debido a que los equipos actuales son de un valor elevado.

## 1.4 ANTECEDENTES

Las transformaciones profundas en la naturaleza de los vehículos tienen que ver con los sistemas y componentes electrónicos; “Un vehículo convencional hasta la década de 1960 solo lo conformaban unidades mecánicas y a partir de 1970 la inserción de partes electrónicas inicia su ascenso” (Lara Rivero, 2014, pág. 12).

A partir de 2010 las prestaciones están enfocadas hacia los vehículos inteligentes y vehículos autónomos. Junto con el crecimiento de la tecnología de la información y la electrónica la industria automotriz se relaciona con la aplicación de nuevos métodos como la del internet móvil, incluyendo la digitalización y el desarrollo de los vehículos inteligentes (Noroña & Gómez, 2019, pág. 118).

Las computadoras a bordo del automóvil emiten información de las condiciones de funcionamiento a todos los demás sistemas. Esto permite mayor control del vehículo, por ejemplo: condiciones de frenado, selección de cambio de velocidades, monitoreo de sistema de carga, sistema de luces, entre otros (Molina K. R., 2015, pág. 41).

Los automóviles modernos están bien equipados con más de cientos de ECUs y sensores basados en microcontroladores que se comunican a través de múltiples redes y protocolos de comunicación como la Red de área del controlador (CAN), la Red de interconexión local (LIN), el Transporte de sistemas orientado a los medios (MOST) y FlexRay (Sharma & Möller, 2018, pág. 0465).

Entre otros sistemas electrónicos tenemos el inmovilizador que ha evolucionado significativamente desde 1990 que salieron al mercado, tampoco contaban con luces de servicio, y no hacía falta una reprogramación de sistemas electrónicos.

Hoy en día los nuevos carros con sus sistemas modernos cada vez tienen más inserción de computadoras, por lo tanto, es necesario tener un equipo adecuado para poder realizar mediciones. Los ingenieros automotrices deben de estar a la par de los vehículos actuales, así como los automóviles tienen grandes avances en tecnología, los equipos externos de medición también deben de ser sofisticados y tener un crecimiento a la par.

En el cambio de la naturaleza de los vehículos automotores con lo referente en componentes y sistemas electrónicos se conoce que; “El 90% de la innovación que se ha producido en la

industria automotriz está relacionada con los componentes electrónicos y de los cuales el 80% se relaciona con el software” (Pérez Darquea, 2018, pág. 4).

Durante los últimos años gran parte de las innovaciones incorporadas a los vehículos automotores están relacionadas con la electrónica y la tendencia es que estas proyecciones continúen ya que la tecnología electrónica provee el medio más eficiente y, en algunos casos, el único disponible para el logro de las mejoras funcionales que se planea incorporar a los vehículos en el futuro cercano (Noroña & Gómez, 2019, pág. 119).

## **1.5 UNIDADES DE CONTROL ELECTRÓNICO (ECU)**

Los automóviles actuales cuentan con diversos módulos electrónicos de control o ECUs, que vienen programadas desde su matriz y cumplen diferentes funciones específicas que se logran mediante la ayuda de sensores, los cuales proveen la información necesaria para que el vehículo tenga un rendimiento óptimo.

“La Computadora Automotriz toma sus decisiones mediante la información llegada de los sensores y la información que tiene su memoria interna y envía sus señales de gestión a los diferentes actuadores” (Sánchez & Taipe, 2018, pág. 16).

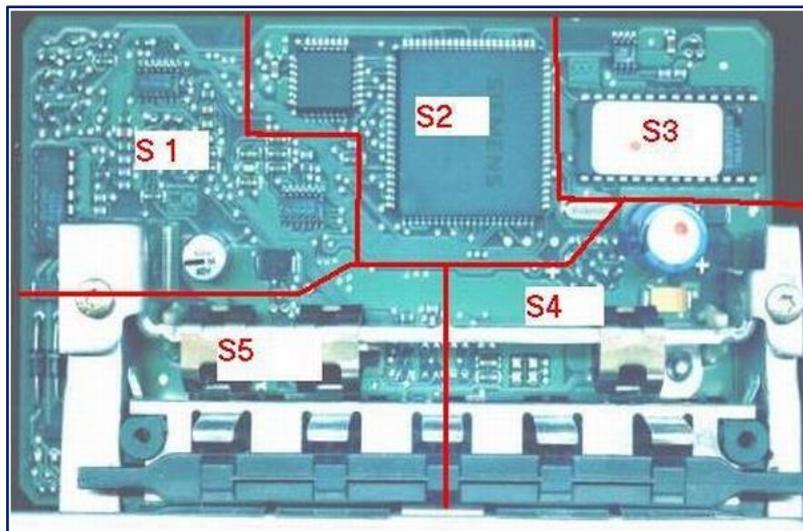
Una ECU está hecho básicamente de hardware y software (firmware). El hardware está hecho de varios componentes electrónicos en una placa de circuito impreso (PCB).

El componente más importante es un chip microcontrolador junto con un EPROM o un chip de memoria Flash. El software (firmware) es un juego de códigos de menor nivel que se ejecuta en el microcontrolador. (Rodríguez Rodríguez, 2019, pág. 46).

Una unidad electrónica de control consiste en un procesador de alto rendimiento, el microcontrolador recibe datos que están almacenados internamente en una memoria programable y los ejecuta.

Por ejemplo; en motores a gasolina la función que cumple es determinar la cantidad de combustible inyectado.

### 1.5.1 COMPONENTES DE UNA ECU



**Figura 1.1** División de bloques de una ECU  
(CISE electrónica, 2011)

En la Figura 1.1, se puede observar los diferentes bloques de trabajo de una computadora automotriz mediante los cuales realiza su funcionamiento. entre estos bloques se puede encontrar los siguientes:

- Bloque de alimentación o soporte
- Bloque de control y procesamiento
- Bloque de entrada
- Bloque de salida

#### 1.5.1.1 Bloque de Entrada (S1)

Está constituido por todos los circuitos que operan como receptores de las diferentes señales de los sensores que van a ingresar al microprocesador.

Se denomina bloque de entrada a todos los circuitos que se encuentran como receptores de las diferentes señales que van a ingresar a la UEC y antes de que lleguen al microprocesador. Encontramos en este sentido: filtros, amplificadores, conversores análogos a digital, comparadores, recortadores, entre otros (Sánchez & Taípe, 2018, pág. 19).

### **1.5.1.2 Bloque de control y procesamiento (S2, S3)**

Un bloque de control es un circuito que cumple las funciones de programación de la ECU y está formado por: procesador, memorias y varios circuitos que intervienen en el trabajo del software.

### **1.5.1.3 Bloque de alimentación o soporte (S4)**

Cumple la función de alimentar a todos los circuitos de entrada, control y salida, es decir; esta parte es la fuente de alimentación de la computadora del vehículo.

Este circuito está diseñado para proteger el módulo y mantener un nivel de tensión estable al interior de este, en los elementos de procesamiento no se permite cambios en los niveles de tensión recordemos que en un automóvil el sistema de carga se caracteriza por los cambios de voltaje y constituyen la fuente de alimentación de la ECU (Sánchez & Taipe, 2018, pág. 18).

### **1.5.1.4 Bloque de salida (S5)**

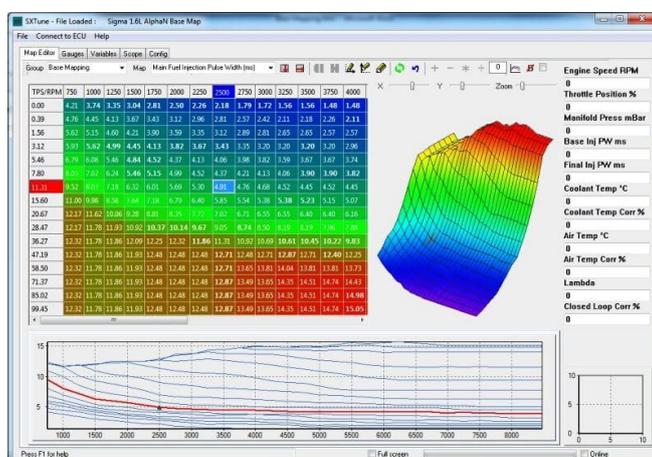
En una computadora automotriz, existen circuitos de entrada y salida para el procesamiento de señales en el microprocesador.

De acuerdo como las señales de los elementos electrónicos son tratadas al ingresar, antes de llegar al microprocesador por circuitos previos que se han denominado bloque de entrada, existen luego circuitos que se encuentran entre las salidas del microprocesador y los diferentes elementos que van a ser los denominados actuadores (Sánchez & Taipe, 2018, pág. 22).

## **1.5.2 MAPA DE ECU**

El mapa de la ECU es el software que viene configurado de fábrica y este viene grabado previamente en la memoria. Estos datos vienen en cuadrículas de 64 Datos es decir 8 por 8

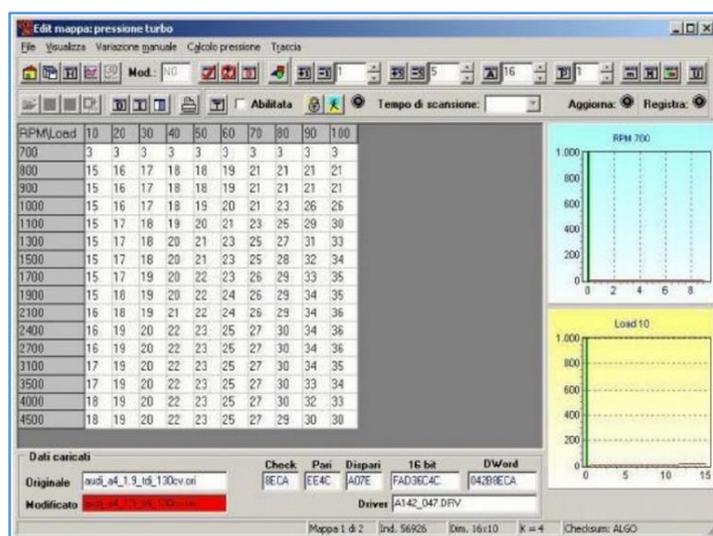
y utiliza un sistema que se llama interpolación para calcular los valores que se encuentran entre esas cuadrículas (Auto Avance, 2019).



**Figura 1.2** Mapa de la Unidad Electrónica de Control (Auto Avance, 2019)

En la Figura 1.2, se puede apreciar un mapa tridimensional de inyección de la ECU, si el mapa tiene un número mayor de filas y columnas (X Y) la precisión para configurar los parámetros será mayor.

En este mapa se pueden hacer las modificaciones necesarias y oportunas mediante los atajos de teclado que están indicados en el menú de “Variación manual”, estas variaciones se hacen en incrementos o decrementos del 10% del valor máximo de la celda, pero también se puede ingresar directamente el valor numérico que corresponda según el criterio del programador (Tinizaray Romero, 2015, pág. 22).



**Figura 1.3** Vista de valores hexadecimal de un mapa de Inyección (Tinizaray Romero, 2015)

En la Figura 1.3, se muestra las celdas modificadas manualmente las cuales se marcarán en color rojo.

## 1.6 MÓDULOS AUTOMOTRICES

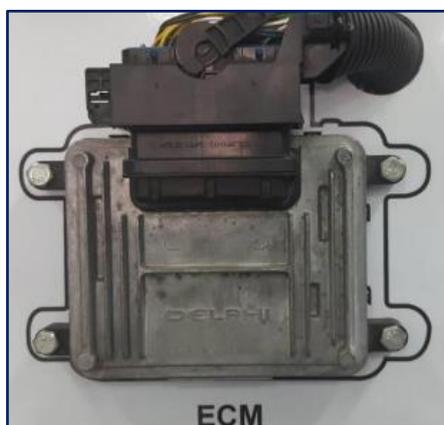
Hoy en día, el diseño de un vehículo incluye varias funciones electrónicas, tales como navegación, información de tráfico, control adaptativo, control de tracción, control de estabilización, y sistemas de seguridad. Estas funciones son implementadas por unidades electrónicas de control embebidas en los vehículos modernos (Caldera, 2016, pág. 8).

Los principales módulos automotrices de control electrónico que podemos encontrar en vehículos modernos son los siguientes:

- ECU-ECM: Módulo de control electrónico del motor
- BCM: Módulo de control de la carrocería
- InmoBox: Módulo de control del Inmovilizador
- Panel de instrumentos electrónico

### 1.6.1 MÓDULO DE CONTROL ELECTRÓNICO DEL MOTOR ECU-ECM

“La unidad de control electrónico o (ECU) sigla en inglés (engine control unit) es una unidad que administra sistemas que interactúan para la operación y/o combustión del motor” (Mamani Yujra, 2015, pág. 15).



**Figura 1.4** Modulo de control electrónico ECU  
(Cevallos C. M., 2016, pág. 35)

Como se puede apreciar en la Figura 1.4, una ECU, es una pequeña computadora que contiene toda la lógica necesaria para controlar tanto la entrada de información enviada por los sensores, como las diferentes señales de salida o activación, además determina la presión y cantidad de combustible que se necesita, controla todos los elementos de ajuste e inyectores y el momento del encendido (Crespo & León, 2017, pág. 8).

#### **1.6.1.1 Funciones módulo de control electrónico del motor ECU**

Las unidades de control electrónicas de los motores actuales cumplen varias funciones específicas como: inyección de combustible, ignición, distribución de válvulas y arranque.

**Control de la inyección de combustible:** En sistemas con acelerador electrónico la ECU controla la inyección del motor de vehículos de combustible, la unidad central determina la cantidad de combustible; si presionamos el acelerador, el ECU permitirá una mayor entrada de aire al motor e inyectará más combustible dependiendo de la cantidad de aire.

**Control del tiempo de ignición:** Para un motor de ignición hace falta una chispa para iniciar la combustión, una ECU ayuda a controlar el tiempo exacto de la chispa o tiempo de ignición para una mejor potencia y disminuir el consumo de combustible.

**Control de distribución de válvulas:** Hay algunos vehículos que poseen distribución de válvulas, la Unidad de control aquí controla el tiempo en el ciclo de motor en el que las válvulas tienen que abrirse, de esta manera se optimiza el flujo del aire que entra en el cilindro, ganado en potencia y evitando una combustión mala (BinaryElectronic & Sistem, 2015).

#### **1.6.1.2 Entradas y Salidas de una ECU**

En el módulo electrónico del motor la señal de entrada es enviada desde un sensor o interruptor, una vez que el módulo realice las funciones indicadas permitirá que las ordenes sean dirigidas hacia las señales de salida o actuadores.

**Sensores:** Tienen la capacidad de transformar una magnitud física en corriente eléctrica y se encargan de recolectar la información del funcionamiento del motor y enviarla hacia la

ECU (señales de entrada), estos elementos electrónicos se encuentran instalados en varios puntos del motor (Romo , Cajisaca , & Pinos , 2016, pág. 14).

Los sensores con los que la ECU realiza el cálculo de carga e inyección son: sensor de masa de aire (MAF), sensor de posición del acelerador (TPS), sensor de temperatura del refrigerante (ECT), sensor de presión absoluta (MAP), sensor de posición del cigüeñal (CKP), entre otros.

**Actuadores:** Los actuadores son componentes electrónicos que constituyen la interfaz entre el procesamiento de la información generada por los sensores y el proceso mecánico, además; reciben señales eléctricas desde el módulo de control según su programación para corregir los parámetros de funcionamiento del motor (Romo , Cajisaca , & Pinos , 2016, pág. 22).

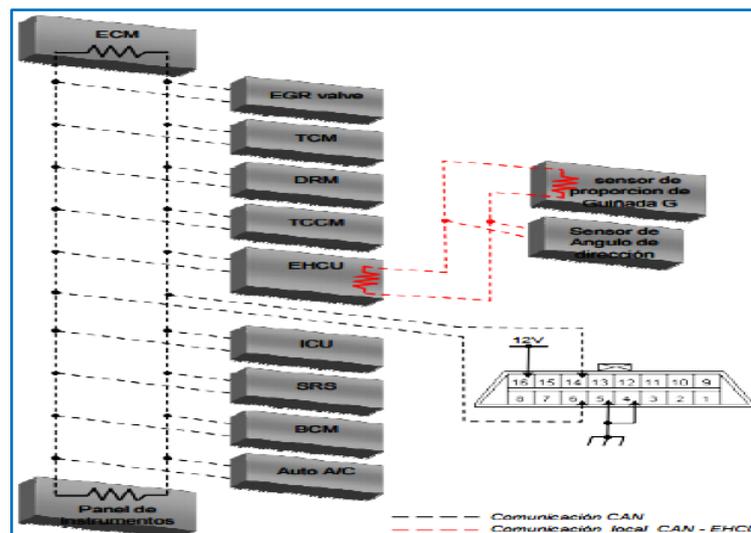
Los actuadores que intervienen en el sistema de inyección electrónica de combustible y cuyo control depende de las señales que son enviadas por los sensores del sistema hacia la ECU son: Inyectores de combustible, bobina de encendido, relés eléctricos y electroválvulas en general (Andino & Corrales , 2017, pág. 26)

## **1.6.2 MODULO DE CONTROL DE LA CARROCERÍA BCM**

En los primeros vehículos de presencia electrónica, los sistemas de iluminación y control de puertas estaban controlados de manera individual.

El módulo conocido como BCM ha reunido la función de todos estos dispositivos independientes para facilitar el control y mejorar el rendimiento del sistema eléctrico del automóvil, centralizando el control de las funciones del cuerpo, la comodidad y la asistencia al usuario (Alves Da Silva, 2017, pág. 20).

En la electrónica de automoción, el BCM (módulo de control de carrocería) es un término genérico para una unidad electrónica de control responsable de supervisar y controlar diversos accesorios electrónicos en el cuerpo de un vehículo.



**Figura 1.5** ECU que intercambian información con BCM  
(Alves Da Silva, 2017, pág. 20)

Típicamente, en un coche el BCM controla el accionamiento de las ventanas, espejos eléctricos, aire acondicionado, inmovilizador del sistema, cierre centralizado, etc. El BCM se comunica con diferentes módulos de control electrónico a través del protocolo de comunicación para transferir información entre sí, como se puede observar en la Figura 1.5 (Chinga, 2013, pág. 54).

### 1.6.2.1 Funciones módulo de control de la carrocería

El módulo de control de la carrocería es una computadora que controla algunas funciones de los automóviles actuales como: la función de bloqueo de puertas, función de antirrobo, función de iluminación entre otras.

**Sistema de iluminación del Automóvil:** El sistema de iluminación, clave en la seguridad activa de un vehículo proporciona la cantidad adecuada de luz para la correcta conducción del automotor en diferentes situaciones adversas de visibilidad; a través de dispositivos lumínicos tales como luces incandescentes, leds, faros, neón, halógenos, etc., que en conjunto y en una distribución adecuada sirven de localización, información y advertencia de acciones ejecutadas por el conductor.

**Sistema de alarma de seguridad:** Un sistema de alarma es un elemento de seguridad activa que permite advertir al conductor de situaciones anormales en el vehículo, y ejecutar una

acción rápida de acuerdo al problema presente. En la actualidad la mayoría de las alarmas modernas están constituidas por distintos tipos de sensores, interruptores, sirenas, receptor de radio para control inalámbrico, batería auxiliar de alimentación del sistema de alarma y una unidad central de monitoreo (Torres , 2015, pág. 13).

Dependiendo de la tecnología del vehículo utilizada, podemos encontrar sistemas de antirrobo como el inmovilizador que es un método de alarma que es controlada electrónicamente a través de una centralita que activara todos los dispositivos sonoros y mecanismos que cortan la corriente que bloquean el volante y puertas del automóvil.

**Sistemas auxiliares primarios del vehículo:** Los sistemas primarios del automóvil están relacionados con sistemas eléctricos que permiten el control de apertura y cierre de ventanas, conocido como elevallas, elevavidrios entre otros nombres mediante un movimiento mecánico a través de un motor.

Otro sistema auxiliar del vehículo que podemos encontrar es el limpiaparabrisas. Según (Torres , 2015, pág. 15), “su función consiste en la limpieza de los parabrisas de sustancias que puedan impedir la adecuada visualización al conductor, como puede ser lluvia, nieve, polvo, etc.”

Además de las funciones ya mencionadas, el BCM controla la ventilación y calefacción de los automóviles que intervienen en las características ambientales del habitáculo para hacerlo más confortable, actuando sobre la temperatura del aire. El sistema de ventilación facilita la renovación del aire en el interior del habitáculo del vehículo y sistemas más complejos combinan el sistema de calefacción y el de aire acondicionado (Torres , 2015, pág. 15).

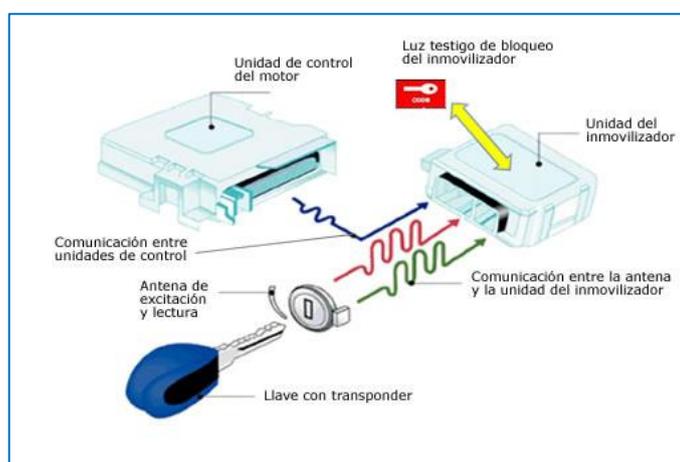
### **1.6.3 MÓDULO DE CONTROL DEL INMOVILIZADOR INMO BOX**

Los sistemas inmovilizadores han surgido de la necesidad de disminuir el robo de vehículos, y es necesario mencionar que esta problemática siempre ha existido desde la aparición del automóvil en el mercado en general: y por esta razón se han implementado varios sistemas de seguridad en los automóviles teniendo como primer dispositivo de seguridad la utilización de la llave, ya que está evitaba la duplicidad, para evitar que otra persona ajena pudiera subir al automóvil (Molina L. F., 2014, pág. 8).

Este sistema en sus inicios fue implementado como un sistema de seguridad pasiva en vehículos de la marca FORD.

Los inmovilizadores son dispositivos que bloquean el paso de corriente al sistema de encendido del motor, a la bomba de combustible, o quitan la corriente al ECM del vehículo. Esta medida de seguridad impide que el motor se encienda y de esta manera evitar que los ladrones hagan cambios en el sistema de switch para encender el vehículo (Báez & Cabrera , 2010, pág. 17).

### 1.6.3.1 Partes del sistema inmovilizador



**Figura 1.6** Esquema de un sistema Inmovilizador (Infotaller , 2016)

En la Figura 1.6, se muestra un esquema de un sistema inmovilizador el cual indica las cuatro partes principales y que se mencionan a continuación:

- La llave de encendido con transponder
- La antena
- El módulo inmovilizador
- Unidad electrónica de control del motor

La llave en su interior tiene un chip, el cual tiene un código grabado único. Este chip no usa batería y recibe la energía cuando se acerca a la chapa de encendido a través de la antena receptora. En algunas marcas se emplea además una llave maestra o llave de programación que por lo general es de distinto color (Molina L. F., 2014, pág. 10).



**Figura 1.7** Antena conectada al Módulo del Inmovilizador  
(Chinga, 2013, pág. 52)

Como se indica en la Figura 1.7, la antena es un circuito eléctrico especial, realizado con el fin de que emita ondas de radio al espacio o reciba energía electromagnética, está ubicada en la moldura de la chapa de encendido y está conectada al módulo del inmovilizador (Chinga, 2013, pág. 52).

Por otro lado, el módulo Inmovilizador “es el encargado de comparar los códigos de la llave con el grabado en su memoria, también de comunicarse con el computador del motor y autorizar el encendido y la inyección en caso de que los códigos sean iguales” (Molina L. F., 2014, pág. 11).

La unidad de mando del módulo del inmovilizador se encuentra ubicada normalmente bajo el tablero cerca de la columna de dirección, la cual es controlada directamente desde la unidad de mando del motor que realiza funciones como: la inyección de combustible y del encendido del motor. Sin la autorización del inmovilizador, el computador del motor no inyectará gasolina y el vehículo quedará bloqueado” (Molina L. F., 2014, pág. 12).

### **1.6.3.2 Inmovilizador con transponder**

El inmovilizador es un sistema de anti arranque codificado donde en la llave de ignición se almacena un código electrónico por medio de un dispositivo llamado Transponder incorporado a la llave, que permiten al conductor poder arrancar el vehículo, solo si cuenta con esta llave y si el código de ésta es reconocido por la computadora del motor (Chinga, 2013, pág. 24) .

Un transponder es un dispositivo que permite ser identificado mediante señales de radio frecuencia y es utilizado en las llaves es muy pequeño en otras palabras es una pastilla electrónica miniaturizada que contiene una memoria no volátil (no requiere de energía constante para la retención de la información) (Molina L. F., 2014, pág. 21).

Hoy en día el campo automotriz cuenta con transponder fijo, crypto y Rolling code. El transponder fijo es aquel que en cada ocasión que es “interrogado” por el sistema de seguridad (Inmovilizador) del vehículo, responde con el mismo código. En este caso se puede duplicar sin ningún problema en máquinas de escritorio siempre y cuando se tenga como mínimo una copia (Báez & Cabrera , 2010, pág. 29).

Un transponder crypto tiene la misma funcionalidad que un fijo, con la diferencia de que el código de seguridad va “tapado” o enmascarado por un programa. En estos casos solo podrán activarse o programarse transponder iguales a los de la llave original mediante equipos especiales; estos chips se encuentran protegidos por el fabricante (Molina L. F., 2014, pág. 22).

Un transponder Rolling code es un sistema más avanzado en comparación con los anteriores ya que responden con un código diferente que cambia en base a un algoritmo de evolución y son usados por varias marcas de vehículos en la actualidad. Es un sistema seguro, admite hasta 18 millones de billones de combinaciones y hace que los códigos cambien cada vez que se utiliza la llave (Molina L. F., 2014, pág. 23).

Báez & Cabrera (2010) afirman que lo complicado es que la llave guarda una parte de la información y el computador del carro guarda la otra, denominado “información por bloques”, de esta manera, podemos duplicar la información de la llave, pero por ningún motivo podemos duplicar la información del computador del carro. Esta información solo la puede saber el fabricante. Al día de hoy este tipo de sistemas no pueden ser copiados y tampoco pueden ser activados en el vehículo.

### **1.6.3.3 Funcionamiento módulo Inmovilizador**

En el sistema de inmovilizador con transponder, como se mencionó anteriormente, la llave incorpora un pequeño chip insertado en el mango de la misma y que emite un código por

radiofrecuencia en el momento en que se acciona el contacto, este código es captado por una antena o unida lectora, normalmente ubicada rodeando el conmutador de arranque (Tapia , 2007, pág. 7).

El sistema funciona de la siguiente manera: cuando el conductor introduce la llave en el switch de ignición, se genera una señal que es transmitida a un módulo encargado de digitalizarla, luego la envía a la computadora que controla el sistema de inyección y encendido del auto donde compara el código que recibió con uno que el fabricante grabó en ella. Si ambos son iguales, permite que se dé el arranque; en caso contrario, elimina las señales hacia los inyectores impidiendo la puesta en marcha del carro (Chinga, 2013, pág. 24).

En el caso de que uno de los dos códigos no coincida, la unidad de mando del motor pierde la autorización de arranque y el motor se para aproximadamente a los dos segundos de haberse iniciado el arranque, estos dos segundos en los que el motor arranca son necesarios porque la autorización y el código circulan por un único cable entre las unidades de mando del inmovilizador y del motor (Molina L. F., 2014, pág. 16).

El Pin Code es un código de seguridad que tiene cada automóvil y es entregado por los fabricantes a los diferentes distribuidores, importadores o concesionarios del mundo, este código de seguridad es el acceso al computador del auto; sin este es imposible ingresar a generar nuevas llaves. Sin embargo, en el mercado se encuentran software para la lectura de los PIN CODE, hay que tener en cuenta la marca y el modelo para ver si el software tiene funcionalidad y no aplica para el 100% de los autos (Báez & Cabrera , 2010, pág. 33).

#### **1.6.4 MÓDULO DEL PANEL DE INSTRUMENTOS ELECTRÓNICO**

Se conoce como modulo del panel de instrumentos electrónico o tablero de instrumentos del vehículo al conjunto de indicadores en forma de relojes analógicos o digitales, o una combinación de ambos.

Además de los relojes, están una serie de testigos luminosos de simbología normalizada en donde se puede apreciar cuando el automóvil tiene una falla o avería en el sistema.

Los indicadores en sus inicios contaban de una aguja que hacía su desplazamiento por medios electromecánicos, y actualmente por un motor paso a paso y el desplazamiento es generado por un circuito integrado. Los cuadros de instrumentos de vehículos modernos de última generación en su interior llevan incorporados microprocesadores capaces de mostrar valores reales por medio de gráficos en pantallas LCD. Por otro lado, los testigos son lámparas integradas en el módulo del panel de instrumentos de una manera ordenada y visibles con el propósito de llamar la atención del conductor, para evitar problemas de seguridad o mecánicos graves.

El módulo del panel de instrumentos cumple la función de enviar la información del estado y la capacidad de trabajo de los sistemas más trascendentales de gestión y operación del automóvil, tales como; la velocidad del vehículo, tacómetro o cuentarrevoluciones, la temperatura del refrigerante, indicador de combustible restante kilometraje recorrido, las luces de advertencia, testigo de presión de aceite del refrigerante, carga de la batería entre otros.

Este sistema de control, como los diferentes módulos automotrices, transfieren información o se conectan mediante el protocolo de comunicación CAN, a través de esta red todos los sistemas desde los más complejos se comunican entre sí. De esta forma se tiene un acceso rápido a toda la información necesaria para mantener actualizado el panel de instrucciones.

## **1.7 MEMORIAS Y MICROCONTROLADORES**

Son circuitos integrados programables, capaces de ejecutar las órdenes grabadas en su memoria. Se emplean para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suelen ir incorporados en el propio dispositivo al que los gobierna (Angulo Usategui & Angulo Martínez, 2003, pág. 1).

### **1.7.1 MICROCONTROLADOR**

El microcontrolador es un computador dedicado. En su memoria sólo mora un programa consignado a gobernar una aplicación determinada; los actuadores y sensores del dispositivo a controlar se encuentran conexiados mediante líneas de entrada/salida, y los

requerimientos son atendidos por los recursos complementarios disponibles. El microcontrolador cuando se encuentra programado y configurado solamente puede gobernar la tarea asignada.

Los microcontroladores más utilizados que podemos encontrar en el mercado automotriz son: AVR, ARM, MSP 430 Y PIC.

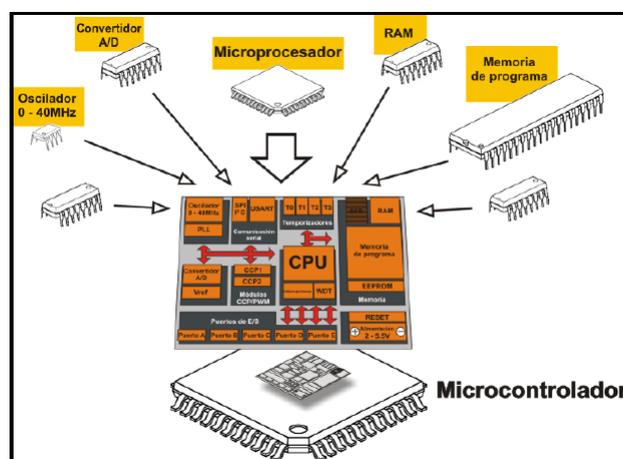
Un microcontrolador en su interior tiene incluidas las tres funcionalidades de una computadora:

- Unidad central de procesamiento (CPU).
- Memoria.
- Periféricos de entrada/salida.

### 1.7.1.1 Componentes especiales que lo conforman

En algunas ocasiones, para controlar algunos procesos mediante los pines de entrada y salida no son suficientes, por lo que se requiere algo más, las posibilidades son amplias como:

- Conversor de análogo a digital (A/D).
- Entender procesos en tiempo real (técnica de “interrupciones”).
- Salidas de modulación por ancho de bandas (PWM).
- Establecer comunicación con otro microcontrolador.
- Medir periodos de tiempo entre sucesos.



**Figura 1.8** Componentes de un microcontrolador (Verle, 2009)

En la Figura 1.8, se puede observar los componentes principales de un microcontrolador los cuales se los describe a continuación:

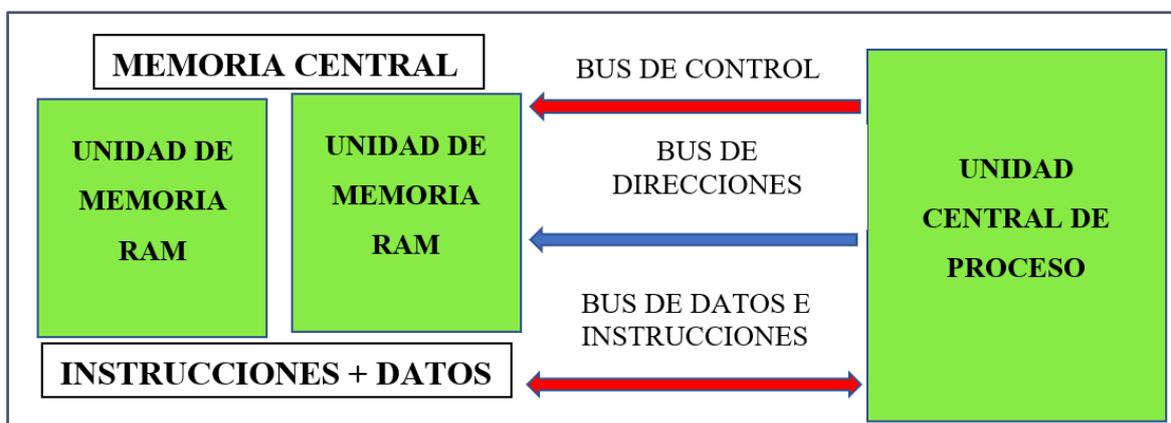
- Memoria RAM para entender los datos
- Memorias para el programa (ROM/EPROM/EEPROM/Flash).
- Procesador o unidad central de proceso (CPU).
- Para comunicarse con el exterior líneas de E/S.
- Algunos módulos para el control de periféricos.
- Generador de impulsos de reloj que sincronizan el trabajo de todo el sistema.

### 1.7.1.2 Características del microcontrolador

Al estar todos los microcontroladores integrados en un chip su estructura y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales: procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones que se destinan preferentemente (Barra Zapata & Barra Zapata, 2015, pág. 26).

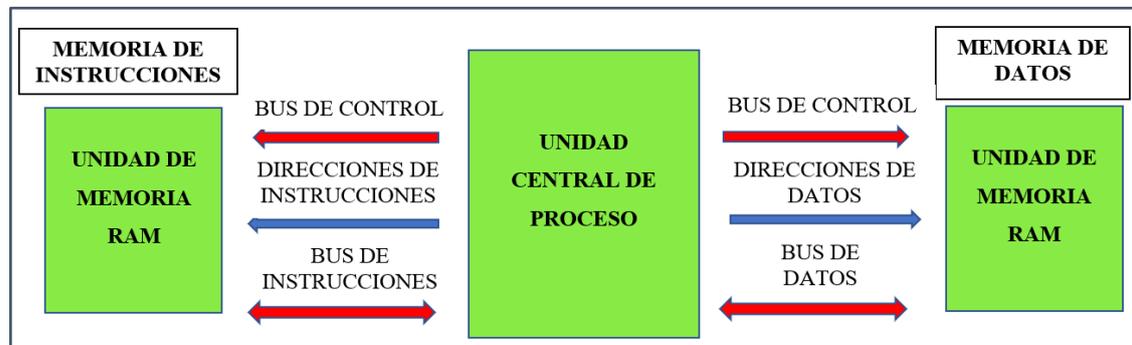
### 1.7.1.3 Arquitectura básica

Los microcontroladores inicialmente adoptan la arquitectura clásica de Van Neuman, actualmente se impone la arquitectura Harvard.



**Figura 1.9** Arquitectura Van Neumann

En la Figura 1.9, se puede observar la arquitectura de Van Neumann que posee una sola memoria principal en la cual almacena instrucciones y datos de forma indistinta. Para acceder a esta memoria se hace mediante un sistema de direcciones, datos y control.



**Figura 1.10** Arquitectura Harvard

En la Figura 1.10, se aprecia la arquitectura Harvard que dispone de dos memorias independientes una que contiene solo instrucciones; y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias, mediante esto la velocidad del sistema aumenta (Barra Zapata & Barra Zapata, 2015, pág. 27).

#### 1.7.1.4 Lenguajes de programación de Microcontroladores

- **Lenguaje Basic**

En la Tabla 1.1, se detalla las ventajas y desventajas más relevantes del lenguaje de programación Basic.

**Tabla 1.1** Ventajas y desventajas Lenguaje Basic

<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
- Es un lenguaje muy simple y con instrucciones fácilmente legibles, incluso por no expertos.	- Nunca va a tener el control del programa en cuanto tiempos de ejecución y control de registros bit a bit.
	- Es muy complicado el manejo de interrupciones simultáneas en este lenguaje.
	- La mayoría de compiladores para este lenguaje pueden utilizarse únicamente bajo ambiente Windows.

**Fuente:** (Arteaga, 2007)

- **Lenguaje C**

El lenguaje de programación en C, es un lenguaje conocido como de alto nivel y una de las características del lenguaje de programación en C, es que es un lenguaje estructurado, lo que permite generar código claro y sencillo, ya que está basado en el modularidad (Cevallos & Reina , 2013, pág. 42).

Las principales ventajas y desventajas del lenguaje de programación C se las detallas en Tabla 1.2.

**Tabla 1.2** Ventajas y desventajas Lenguaje C

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> <li>- Lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas.</li> <li>- A pesar de su bajo nivel es el lenguaje más eficiente en existencia, habiendo compiladores para casi todos los sistemas conocidos.</li> <li>- Proporciona facilidades para realizar programas modulares y/o utilizar código o bibliotecas existentes.</li> <li>- Por su flexibilidad y ser un lenguaje de alto nivel, es empleado por muchos programadores</li> <li>- Es un lenguaje de alto nivel más cercano a la máquina.</li> <li>- Es aceptado por la empresa fabricante Microchip, incluso ellos tienen algunos compiladores C.</li> </ul>	<ul style="list-style-type: none"> <li>- Los programas al compilarlos pueden resultar un poco extensos y pesados por ello debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar.</li> <li>- Los programas al compilarlos pueden resultar un poco extensos y pesados por ello debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar.</li> </ul>

**Fuente:** (Cevallos & Reina , 2013, pág. 42)

- **Lenguaje ensamblador**

(Cevallos & Reina , 2013, pág. 42) afirman que: Si se comienza con la programación de PICs se debe primero dominar el lenguaje ensamblador, no se aconseja el lenguaje BASIC. Una vez que se haya dominado el lenguaje ensamblador para gama baja y media puede iniciarse con el lenguaje C, debido a que el lenguaje C es un lenguaje de alto nivel y puede

resultar útil combinándolo con el lenguaje ensamblador ya que ahorra tiempo de programación, depuración y simulación sobre todo en la gama alta y en los dsPIC.

Las ventajas y desventajas del lenguaje ensamblador se las indica de una manera detallada en la Tabla 1.3.

**Tabla 1.3** Ventajas y desventajas Lenguaje ensamblador

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> <li>- Es el lenguaje de bajo nivel natural de la línea PIC tanto para gama baja, media o alta.</li> <li>- Con él se tiene un aprovechamiento eficiente de los recursos del PIC.</li> <li>- Se pueden crear macros con este lenguaje, para después simplificar el código en diferentes desarrollos.</li> <li>- Con él se pueden controlar los tiempos y los registros bit a bit.</li> <li>- Excelente para manejar interrupciones simultáneas.</li> </ul>	<ul style="list-style-type: none"> <li>- En realidad, el lenguaje ensamblador no presenta ningún inconveniente, excepto cuando no se tiene experiencia en programación puede tardarse el desarrollo de alguna rutina en comparación con los otros lenguajes.</li> </ul>

**Fuente:** (Arteaga, 2007)

- **Lenguaje de maquina**

Es el sistema de códigos interpretable directamente por un circuito micro programable, como el microprocesador de una computadora.

Este lenguaje se compone de un conjunto de instrucciones que determinan acciones que serán realizadas por la máquina y un programa de computadora consiste en una cadena de estas instrucciones de lenguaje de máquina, normalmente estas instrucciones son ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos.

“El lenguaje máquina es específico de cada máquina o arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre ellas” (UNAM, s.f.).

La Universidad Internacional de Valencia (2018) dice que la combinación de unos y ceros hace referencia al alfabeto binario, que está formada por estos caracteres y que acaban formándose en distintos parámetros de voltaje para representar los procesos.

Cuando esta combinación de números termina de prepararse en una instrucción que llegará al microprocesador, se tendrá que dar final a la operación por parte de un proceso denominado como compilación.

### **1.7.2 MEMORIA**

Una memoria es un dispositivo de almacenamiento de datos binarios. Las memorias almacenan bloques de datos generalmente de 8 bits que corresponden a un byte.

Un byte está compuesto por dos bloques de 4 bits que reciben el nombre de nibble. Una unidad de información se denomina palabra y está compuesta por uno o más bytes (Flóres Fernández, 2010, pág. 153).

El número de unidades que puede almacenar de datos es la capacidad de la memoria. Para colocar los datos en una posición de memoria específica se realiza a través de la operación de escritura y la operación de lectura. Actualmente se cuenta con memorias volátiles y memorias no volátiles que se diferencian en la capacidad de mantener la información almacenada al momento de un corte de energía.

Las memorias volátiles al desconectar la unidad de control por el interruptor de encendido, esta memoria pierde todos los datos almacenados y los valores de adaptación (valores aprendidos sobre estados del motor y de servicio) tienen que determinarse de nuevo en este caso al conectar otra vez la unidad de control (Paredes, 2011, pág. 27).

Una memoria no volátil es la memoria encargada de almacenar el programa con su forma de ejecución, además es una memoria no volátil ya que el programa instalado no se borra al quedarse sin energía.

Los tres procesadores principales son el RAM (memoria temporal), el ROM (programa básico de computadora) y el PROM (programa de sintonía fina), estos tres microprocesadores son el corazón de la CPU (OOCITIES.ORG, 2009).

### 1.7.2.1 Tipos de memoria

En el campo automotriz existen diferentes procesadores de control, los cuales se detallan a continuación:

- **Memoria RAM (solo escritura)**

En una memoria RAM los datos se pueden leer y escribir de forma aleatoria, es decir, que se puede leer y escribir de una posición de memoria deseada. Además, los tiempos de respuesta en una memoria RAM son bajos, esto indica que es una memoria supremamente eficiente. Al escribir un dato en una posición de la memoria, los datos anteriores son reemplazados por el nuevo dato.

- **Memoria ROM**

Una memoria ROM almacena y conserva datos forma permanente, pueden ser leídos, pero no modificados, es una memoria no volátil ya que el programa instalado no se borra al quedarse sin energía, ya que cuando el chip es fabricado se graba su contenido. Este es muy utilizado cuando se realiza una producción en serie.

- **Memoria SAM**

Es una memoria de acceso Secuencial, se emplea en memorias secundarias. Es el tipo de acceso que se utiliza en sistemas de memoria de tipo registro de desplazamiento. Un registro de desplazamiento puede ser de dos tipos: de tipo FIFO (el primero en entrar es el primero en salir) o LIFO (el último en entrar es el primero en salir) (García & Rubio, 2017, pág. 41).

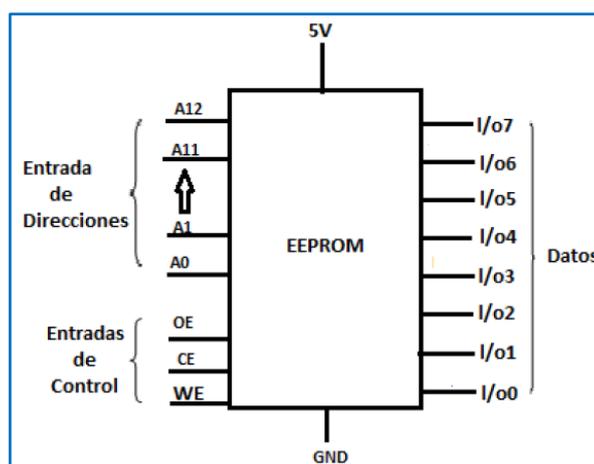
- **Memoria PROM (Programable ROM)**

Memoria ROM programable, se suministra sin programación para que el usuario programe su contenido en función del trabajo que le interese desarrollar en su equipo. Una vez grabada

se convierte en ROM. Se usa mucho para grabar constante que dependen de cada usuario particular pero que son totalmente permanentes una vez definidos sus valores, tal como la programación de manejo de una máquina una secuencia de luces de texto en un aviso, etc. (Mejía Mesa, 2005, pág. 65).

- **Memoria EEPROM**

Las PROM Borrables eléctricamente, se pueden borrar y programar mediante impulsos eléctricos. Los dos tipos de EEPROM son los MOS de puerta flotante y la de silicio de óxido nitroso (NMOS, Metal Oxide Silicon). La aplicación de una tensión en la puerta de control de la estructura de puerta flotante permite la eliminación y el almacenamiento de la carga en la puerta flotante (Flóres Fernández, 2010, pág. 157).



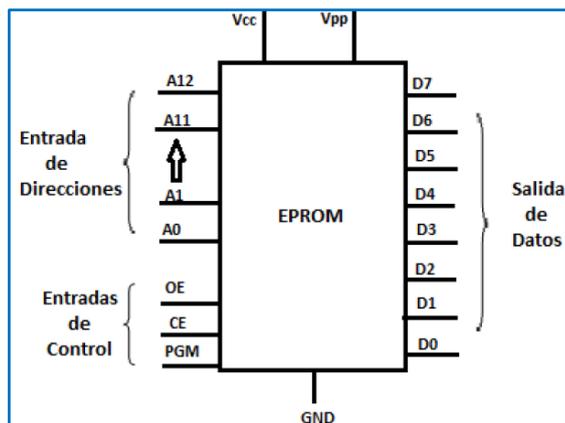
**Figura 1.11** Memoria EEPROM

(Tocci , Widmer , & Moss, 2007, pág. 806)

Las EEPROM como se puede observar en la Figura 1.11, se pueden reprogramar dentro del propio circuito, lo que permite reconfigurar cualquier sistema.

- **Memorias EPROM (Erasable PROM)**

Es un tipo de memoria programable y borrrable, que se puede programar y borrar mediante una lampara de luz ultravioleta, y es una memoria no volátil. Se puede reconfigurar cualquier sistema ya que se puede reprogramar dentro del propio circuito.



**Figura 1.12** Memoria EPROM  
(Tocci , Widmer , & Moss, 2007, pág. 806)

En la Figura 1.12 se aprecia, una EPROM que es una PROM borrable. Una EPROM puede ser reprogramada, si antes se borra los datos que contenga la memoria. Utiliza matrices NMOSFET con estructura de puerta aislada.

La puerta del transistor aislada no tiene conexión eléctrica y puede almacenar una carga eléctrica durante un período de tiempo indefinido. Los bits en esta matriz se representan mediante la presencia o ausencia de una carga almacenada en la puerta. El borrado consiste en la eliminación de la carga de la puerta (Flóres Fernández, 2010, pág. 157).

- **Memoria tipo DIL**

Este encapsulado fue unos de los primeros usados para los autos equipados con inyección electrónica desde el año 1990.



**Figura 1.13** Memoria tipo DIL  
(Auto Avance , 2013)

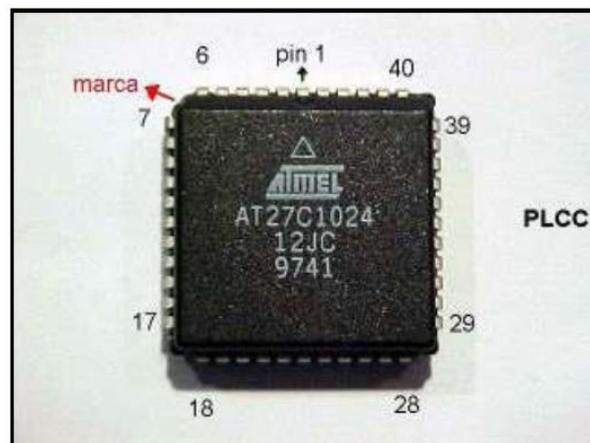
Como se aprecia en la Figura 1.13, este posee 2 líneas con pines de acceso, estas pueden ser de 28 o 32 pines.

Una marca o muesca se puede observar en su encapsulado, esta marca la podemos tomar como una guía para orientarnos con respecto a su numeración, esta guía nos indicara la ubicación del pin 1 de la memoria.

- **Memoria tipo PLCC**

Este encapsulado fue el segundo utilizado por las terminales automotrices, él mismo cuenta con un tamaño reducido y la configuración de sus pines envuelve los 4 lados a diferencia de su antecesor (DIL). Lo particular de esta configuración es que los pines se encuentran hacia adentro y su montaje es superficial a la placa madre.

Esto simplifica el tamaño que ocupa en la placa madre y también aporta mayor capacidad en su interior. Este tipo de memorias se borran eléctricamente. Sobre un lateral posee una marca que nos indica la posición pin 1. Pueden ser encontradas montadas directamente sobre la placa como también sobre un zócalo (Auto Avance , 2013).



**Figura 1.14** Memoria tipo PLCC  
(Díaz & Sopa, 2013, pág. 10)

La memoria tipo PLCC cuenta con una cantidad de 32, 44, 48 pines como se muestra en la Figura 1.14.

- **Memoria tipo SOP**

Este encapsulado es uno de los más utilizados en la actualidad, posee una capacidad de almacenamiento de 2 hasta 32 megas en el uso automotriz, este encapsulado logra tanto en su costo, como en capacidad y espacio un compromiso adoptado por los fabricantes de electrónica automotriz. Esta Memoria puede borrarse eléctricamente (Auto Avance , 2013).



**Figura 1.15** Memoria tipo SOP  
(Auto Avance , 2013)

En la Figura 1.15, se indica la cantidad de pines utilizados que son de 44 o 48 terminales, también posee una marca el cual nos indica la posición del pin.

- **Memoria SOIC**

Este encapsulado es un circuito pequeño siendo uno de los más utilizados en el campo automotriz, está constituido de dos bloques paralelos, el número de pines depende de cada circuito que se esté utilizando, se puede encontrar de 8 y 32 pines, un punto o una muesca que se puede apreciar en su encapsulado nos indica el pin número 1 de la memoria. Luego de identificar el pin 1 de la memoria se cuenta hacia la derecha para identificar los demás pines, el pin de mayor número, corresponde al pin de la parte superior izquierda del encapsulado.

### 1.7.2.2 Lenguajes de interpretación y sus equivalentes

Los datos de las memorias se interpretan en sistemas numéricos, existes diferentes sistemas numéricos pero el mejor para interpretar es el hexadecimal.

- **Sistema decimal**

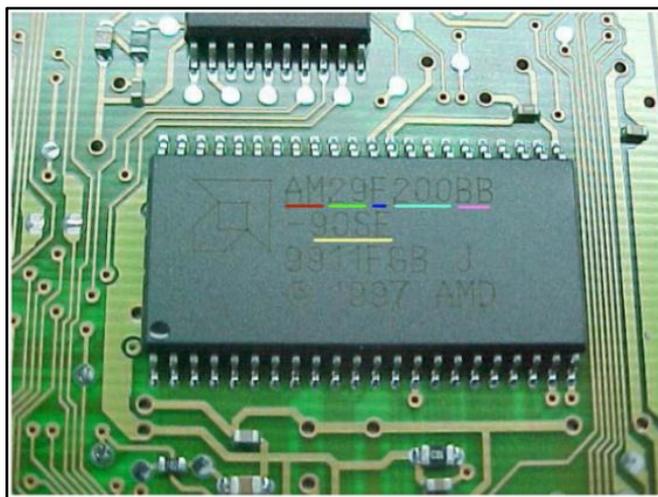
Es un sistema empleado por las computadoras, es un sistema base 10, ya que tiene 10 símbolos que van de 0 a 9, cada símbolo se lo denomina dígito. Para trabajar con números que se encuentran en diferentes sistemas se debe colocar un subíndice que lo identifique. Ejemplo, para representar el número 20 en decimal se debe colocar  $20_{10}$ .

- **Sistema hexadecimal**

El sistema numérico hexadecimal es un sistema base 16 debido a que tiene 16 símbolos los cuales son los números del 0-9 y de la letra A-F. Es necesario identificar la base del número colocándola como subíndice. Si se desea representar el número 95B en binario, se coloca 95B y la cantidad de combinaciones que se pueden obtener en hexadecimal corresponde a 16 donde n es el número de símbolos (Flóres Fernández, 2010, pág. 22).

### 1.7.2.3 Interpretación de su nomenclatura

En cada una de las memorias programables que poseen los diferentes tipos de unidad de control electrónica en el campo automotriz, se pueden describir sus características por medio de los colores que representan sus prestaciones” (Sánchez & Taípe, 2018, pág. 38).



**Figura 1.16** Nomenclatura de memorias por colores  
(Augeri, 2015)

En la Figura 1.16, se puede observar la nomenclatura de memorias automotrices y que según (Grijalva, 2012, pág. 7) siempre traen una numeración la cual indica la marca del fabricante, su capacidad entre otros datos y sus características se identifican por los diferentes colores que se mencionan a continuación:

Rojo (AM): Marca del fabricante

Verde (29): Familia a la cual pertenece. Para memorias colocadas en vehículos en su mayoría se encuentran en 2 tipos de familias No. 27 y No. 29, cada familia estructuralmente cumple condiciones; como la velocidad de acceso, condiciones de circuitos internos; como compuertas, estructuras, etc.

Azul (F): flash" se refiere a que su borrado puede realizarse eléctricamente, se realiza mediante un programador de memorias EPROM, su ventaja es desarrollar las tensiones de las memorias y beneficios del flash y borrado en memorias DIL.

Celeste (200): son la capacidad que posee la memoria, lo que puede almacenar internamente y pueden atribuir a las memorias DIL, PLCC Y SOP.

Amarillo (-90): El tiempo de acceso es el que se necesita para localizar y leer una información almacenada, el tiempo de acceso es una característica importante para determinar la velocidad de resolución de un sistema, conociendo el tiempo de acceso se puede predecir el tiempo necesario para procesar un trabajo.

## 1.8 TIPOS DE PROGRAMACIÓN

En los inicios de la informática, los programas se escribían directamente en código máquina y requerían un conocimiento profundo del ordenador. Un científico que quisiera resolver un problema matemático ayudándose del ordenador tenía entonces dos problemas más: cómo generar un algoritmo que resolviera el problema y cómo traducir ese algoritmo a código máquina (Noguera Otero & Terrén, 2010, pág. 127).

Aunque es un proceso muy sencillo, reprogramar la unidad de control del motor puede acabar en un fallo catastrófico. Si, la comunicación se corta por algún motivo durante el proceso, la centralita quedará completamente inutilizada.

Para evitar (en lo posible) que pase este inconveniente, hay que entender cómo se debe proceder de forma correcta con cada uno de los diferentes tipos de reprogramación de unidades de mando. Por supuesto, se deberá seguir a rajatabla las instrucciones de fabricante (Pumarica & Tonato, 2018, pág. 31).

Por la finalidad y el objetivo que se desea cumplir, de acuerdo a la reprogramación y el tipo de centralitas automotrices, se han creado diferentes tipos de reprogramación las cuales permitirán que el usuario las pueda emplear teniendo en cuenta que muchas de las centralitas no podrán ser reprogramadas directamente sin contar con diferentes procedimientos y/o elementos que permitirán el mejor empleo y análisis de los datos internos de las mismas (Semante, 2016, pág. 46).

Basándose en lo expuesto anteriormente se puede mencionar tres métodos de reprogramación:

- Flash
- ECU estándar
- ECU programable

### **1.8.1 REPROGRAMACIÓN FLASH**

Cada módulo es programado en fábrica con software que le permiten realizar ciertas funciones. Como por ejemplo dentro de la ECM hay un software que contienen cientos o incluso miles de parámetros para controlar el salto de chispa, la cantidad de combustible, velocidad de cruce, control de las emisiones, facilidad de conducción y el rendimiento (Díaz & Sopa, 2013, pág. 14).

La práctica de la actualización de software en estos módulos es comúnmente conocida como flash reprogramación. En los concesionarios de automóviles nuevos el flash reprogramación es relativamente sencillo, porque los técnicos de servicio están conectados online a la fábrica de automóviles y poseen las herramientas especializadas de servicio del distribuidor dedicado a la reprogramación (Díaz & Sopa, 2013, pág. 14).

### **1.8.2 REPROGRAMACIÓN ECU ESTÁNDAR**

“Este tipo de reprogramación, se realiza directamente sobre los mapas característicos de inyección e ignición que son codificados y programados desde su fabricación” (Semante, 2016, pág. 47)

Para este tipo de reprogramación se deben tomar en cuenta pasos como la lectura del archivo de la ECU, que consiste en obtener un archivo que contiene los parámetros modificables expresados en valores hexadecimales. Después de realizar una correcta lectura se modifican los mapas, para esto existen varios programas y el uso de cada uno depende del modelo y la marca del vehículo. Finalmente, en una reprogramación ECU estándar se graba la información una vez realizado las modificaciones deseadas.

### **1.8.3 REPROGRAMACIÓN ECU PROGRAMABLE**

La ECU programable es una categoría especial de las Unidades de Control, ya que su comportamiento no está prefijado, es decir, sus parámetros deben ser configurados por el usuario (Díaz & Sopa, 2013, pág. 27).

Estas unidades más modernas utilizan UEC con sistemas OBD-II, capaces así de poder ser programadas mediante puertos OBD de manera externa, pudiendo ser modificadas mediante el uso de un computador portátil conectado al vehículo, en el cual podrán visualizarse todas las características de funcionamiento del mismo y podrá modificarse, por ejemplo, la cantidad de combustible que se debe inyectar en el motor, la mezcla correcta de oxígeno y combustible o distintos parámetros claves necesarios en el vehículo (Semante, 2016, pág. 52).

## **1.9 PROTOCOLOS DE COMUNICACIÓN**

La evolución del automóvil en el ámbito de la electrónica ha sido muy radical, de poseer una sola unidad controladora, ha pasado a tener varias unidades de control que cumplen funciones propias dentro del automóvil, como los componentes y sistemas del motor, confort, frenos, suspensión y seguridad.

Este progreso de la industria automotriz conlleva a que aumente la cantidad de sistemas en el automóvil controlados electrónicamente, por esta razón el uso de los protocolos de comunicación desempeña un papel primordial en el intercambio de información de los ordenadores, con la finalidad de disminuir la cantidad de cables que conectan a los diferentes elementos y así optimizan los recursos (Carpio , 2013, pág. 2).

Existen diferentes tipos de protocolos que difieren unos de otros dependiendo de la marca y modelo del vehículo y se los detalla en la Tabla 1.4.

**Tabla 1.4** Tipos de protocolos de comunicación

<b>PROTOCOLO</b>	<b>APLICACIÓN</b>
CAN (Controller Area Network)	Utilizado por el grupo VAG (Audi, Volkswagen, Seat, Skoda), grupo PSA (Citroën, Peugeot), Renault, Mercedes, BMW, Volvo.
J1850	Utilizado por Chrysler, GM, Ford.
VAN (Vehicle Area Network)	Utilizado en Citroën, Peugeot.
BEAN (Body Electronic Area Network)	Utilizado por Toyota.
MOST (Media Oriented Systems Transport Corporation)	Se aplica a sistemas de infotainment
FLEXRAY	Este consorcio está formado por Volkswagen, Daimler AG, General Motors, NXP, Bosch GmbH, Motorola, BMW, etc.

“En el campo de la informática automovilística, cuando se habla de protocolo se hace referencia al conjunto de reglas establecidas para comunicar diversas centralitas entre ellas a través de un BUS” (Cano Martínez, 2012, pág. 76).

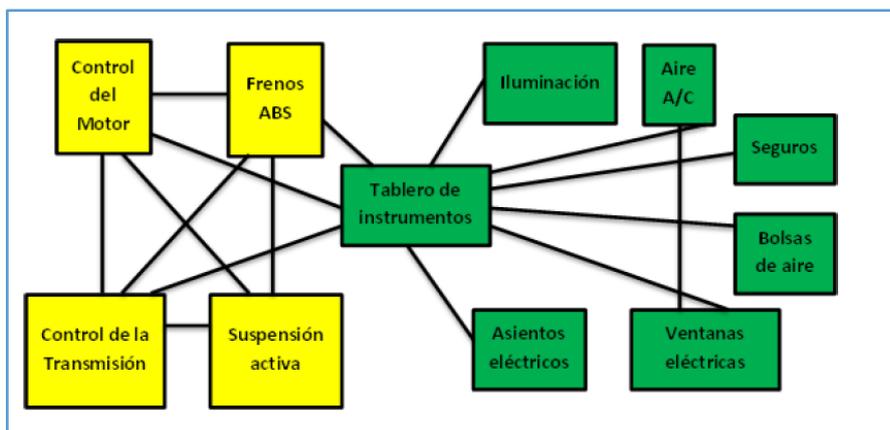
### **1.9.1 CAN (CONTROLLER AREA NETWORK)**

CAN-BUS es un protocolo o lenguaje de comunicación en serie desarrollado por Bosch para el intercambio de información entre unidades de control electrónicas del automóvil, tales como sistemas de gestión del motor, control de luces, aire acondicionado, bloqueo central entre otros, orientados a proporcionar confort y seguridad al conductor (Buitrago , 2017, pág. 6).

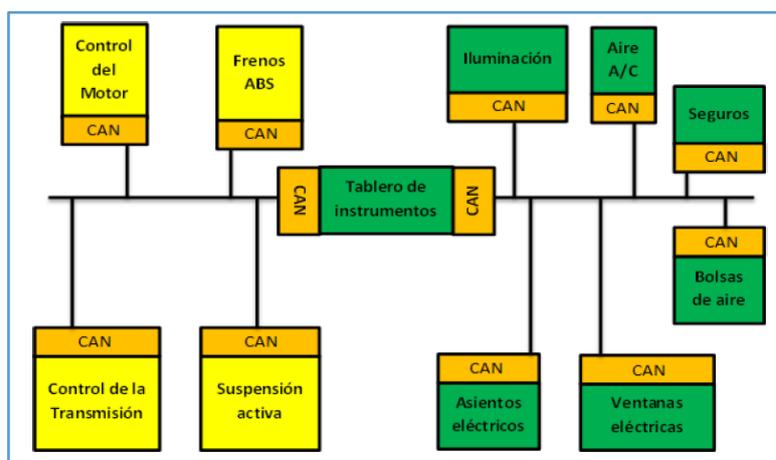
El establecimiento de una red CAN para interconectar los dispositivos electrónicos internos de un vehículo tiene la finalidad de sustituir o eliminar el cableado. Las ECUs, sensores,

sistemas antideslizantes, etc. se conectan mediante una red CAN a velocidades de transferencia de datos de hasta 1 Mbps (Niza Yanchapaxi, 2009).

A continuación, se presenta ejemplos de conexión sin y con CAN BUS, ha sido implementado por diferentes fabricantes de vehículos desde 2008.



**Figura 1.17** Conexión sin CAN BUS  
(Ibáñez Galindo, 2015, pág. 10)



**Figura 1.18** Conexión con CAN BUS  
(Ibáñez Galindo, 2015, pág. 10)

En las Figuras 1.17 y 1.18, se puede apreciar las diferentes formas de conexión CAN BUS y la importancia del uso del protocolo.

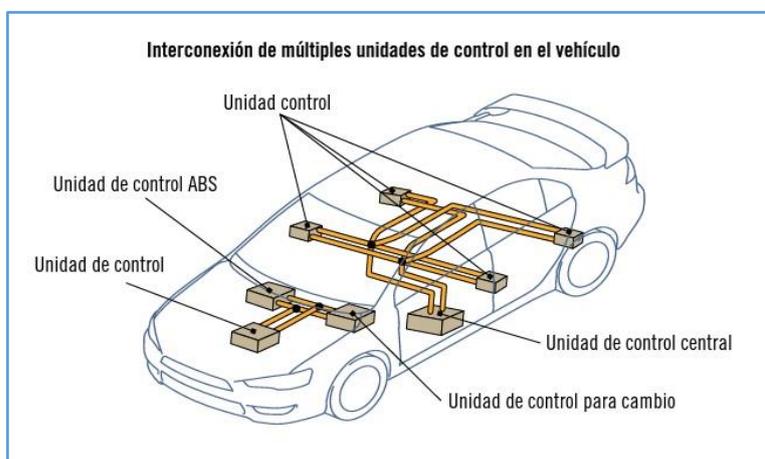
Existen dos formas diferentes para realizar una transmisión de datos:

- Transmitir cada información individualmente por un cable propio, para cada información es necesaria un cable propio, esto genera un aumento en el número de cables que se precisan proporcionalmente a las necesidades de transmisión.

- A través de dos únicos cables (BUS) transmite toda la información de forma colectiva, esta se ha vuelto la solución a la transmisión de datos, debido a que la primera posibilidad hoy en día se ha vuelto innecesaria, por la gran cantidad de datos que se deben de transmitir. CAN-BUS por dos cables bidireccionales se transmite toda la información, transmitiendo en ambos cables la misma información.

### 1.9.1.1 Principios de interconexión

“El principio de interconexión es un sistema básico que se compone de varias unidades de control que se encuentran conectadas en paralelo al cable del BUS de datos a través de los transceptores” (Cano Martínez, 2012, pág. 65).



**Figura 1.19** Interconexión de múltiples unidades de control en el vehículo  
(Cano Martínez, 2012, pág. 64)

En la Figura 1.19, se puede apreciar de manera detallada el principio de interconexión de múltiples unidades de control como la unidad del sistema de frenos ABS y la unidad de control del conjunto de cambios que son controladas por una memoria central del vehículo.

El protocolo CAN para la transmisión de datos cuenta con un controlador y un transceptor que son módulos independientes de los nodos, lo que permite que estos no tengan que destinar recursos en la gestión de las comunicaciones, acceso al medio o colisiones entre otros. A pesar de que algunos microcontroladores poseen módulos CAN en un único encapsulado, internamente son circuitos independientes en la mayoría de los casos (Martínez Requena, 2017, pág. 5).

### **1.9.1.2 Beneficios del protocolo CAN BUS**

Uno de los puntos fuertes de esta tecnología, y que le ha hecho perdurar en el tiempo a pesar de la aparición de otros protocolos de comunicación con mayor velocidad o capaces de transmitir a mayor distancia, son las garantías de comunicación que ofrece, las cuales son muy importantes a la hora de desarrollar sistemas con características de tiempo real o de alta integridad (Martínez Requena, 2017, pág. 4).

Los principales beneficios que ofrece el protocolo de comunicación CAN BUS son:

- Es normalizado, la red común o bus de diferentes fabricantes en comunicación de subsistemas se simplifica y economiza.
- El procesador anfitrión (host) dispone de tiempo para ejecutar tareas propias, debido a que la carga en comunicaciones es autorizada por el procesador anfitrión.
- Tiene una gran reducción en cableado conexiones punto a punto gracias a ser una red multiplexado.

### **1.9.1.3 Características del protocolo CAN BUS**

Principales características del protocolo de comunicación CAN BUS son:

El paradigma de comunicación de datos que hace referencia al modelo productor consumidor, es la relación entre productor y diferentes consumidores. La información que recibe CAN se descompone en mensajes, a los que se les destina un identificador, y son encapsulados en tramas para su transmisión.

Los nodos tienen el trabajo de decidir en aceptar o no aceptar dicho mensaje, basándose en el identificador único dentro de la red que tiene cada mensaje.

Alta confiabilidad, disponibilidad y robustez para trabajar en sistemas de comunicación segura en tiempo real operando en ambientes hostiles; además, su excelente detección de errores y capacidades de confinamiento del sistema incrementa su confiabilidad en ambientes con alto ruido, finalmente, la factible tasa de transferencia de datos de hasta 1 Mbps hace al sistema CAN recomendable para aplicaciones de relativa alta velocidad y control en tiempo real (Vlacic, Parent, & Harashima, 2001, pág. 19).

La red CAN actualmente es la más utilizada en vehículos, los impulsos electrónicos en forma de señal cuadrada, son los encargados de realizar una transmisión de datos con la ayuda de un bus de datos, que se encuentra constituido por dos cables entrelazados.

Los impulsos eléctricos son convertidos en mensaje de código binario con un transceptor que tienen las unidades de control, para así ser comprendidos por el controlador y el microcontrolador.

Una vez llegado el tren de bits al transceptor. El tren es convertido en valores de tensión apropiados por el transceptor y luego son transmitidos consecutivamente por el cable BUS de datos.

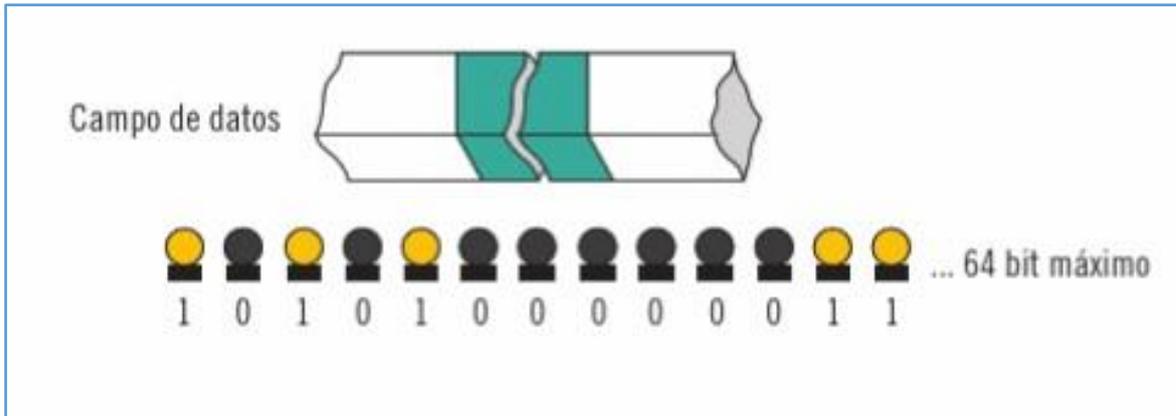
El mensaje que circula por el cable BUS llega a otro transceptor de recepción, los valores de tensión son convertidos nuevamente en un tren de bits, que son transmitidos a las unidades de control apropiados a través de un cable RX (cable de recepción). Los valores seriales de bits son transformados en mensaje por las unidades de control. El vehículo tiene diferentes unidades de control CAN que son las siguientes:

- CAN transacción o tren motriz
- CAN confort
- CAN entretenimiento
- CAN seguridad
- CAN cuadro de instrumentos

#### **1.9.1.4 Funcionamiento del protocolo CAN**

Los protocolos dentro del campo de datos contienen la información que se desea transmitir en ese preciso momento.

Los demás campos también contienen información, pero a modo de seguridad de transmisión. (Cano Martinez, 2012, pág. 81) afirma: “El protocolo de datos consta de varios bits enlazados, cada bit puede adoptar cada vez un solo estado o bien los valores cero (0) o uno (1)”.

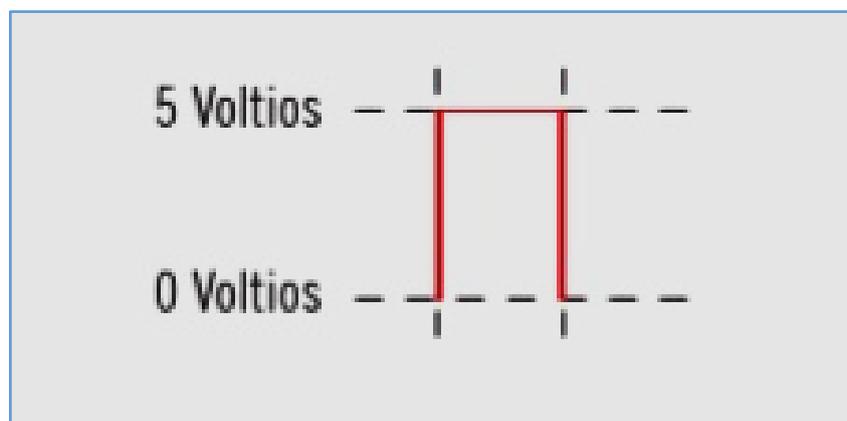


**Figura 1.20** Simulación trama de bits con bombillas apagadas (0) y encendidas (1)  
(Cano Martinez, 2012, pág. 81)

En la Figura 1.20, se aprecia un ejemplo con bombillas que explica el sistema binario, en donde se añade una variante más a la codificación según las necesidades y es comandada por el interruptor, este apagará marcando un (0) y encenderá marcando un (1).

El transceptor CAN, que funciona como interruptor, tiene conexión con los cables de BUS.

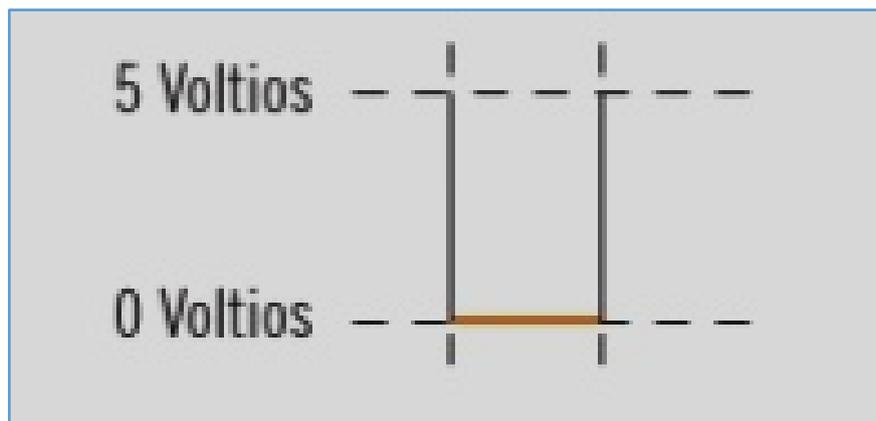
**Estado de bit con valor (1).** El transceptor abierto emite un bit con valor uno.



**Figura 1.21** Bit con valor uno  
(Cano Martinez, 2012, pág. 82)

Este al abrir conecta 5 voltios (según el área de trabajo puede ser un valor de 2.5 voltios), como se muestra en la Figura 1.21.

**Estado de bit con valor (0).** El transceptor cerrado emite un bit con valor cero.



**Figura 1.22** Bit con valor cero  
(Cano Martinez, 2012, pág. 82)

Al cerrar el transceptor conecta a masa (0) quedando el cable con una tensión aproximadamente de 0 voltios, esto se puede observar en la Figura 1.22.

La forma de transmitir la información por medio de dos bits enlazados: para un mayor número de transmisiones tiene que haber un mayor número de bits. En la Tabla 1.5, se puede observar cuatro posibles variantes con la combinación de voltajes 0 y 5 voltios, indicando la temperatura del líquido refrigerante y el estado del elevallunas.

**Tabla 1.5** Tabla de variantes con dos simples bits en el protocolo CAN

TABLA DE VARIANTES CON DOS SIMPLES BITS EN EL PROTOCOLO CAN					
Posible variante	Segundo BIT	Primer BIT	Representación grafica	Información estado del Evaluna	Información Temperatura Refrigerante
Uno	0 voltios	0 voltios		En movimiento	10 °C
Dos	0 voltios	5 voltios		En reposo	20 °C
Tres	5 voltios	0 voltios		En zona de inicio de parada	30 °C
Cuatro	5 voltios	5 voltios		En detección de bloqueo superior	40 °C

**Fuente:** (Cano Martinez, 2012, pág. 83)

Para obtener mayor número de variantes posibles, la transmisión de datos de bits debe ser mayor. En la Tabla 1.6, se puede observar posibles variantes con múltiples combinaciones de bits de voltajes 0 y 5 voltios, indicando la temperatura del líquido refrigerante.

**Tabla 1.6** Tabla de variantes con múltiples combinaciones de bits en el protocolo CAN

<b>TABLA DE VARIANTES CON MÚLTIPLES COMBINACIONES DE BITS EN EL PROTOCOLO CAN</b>					
<b>Variantes con 1 bit</b>	<b>Posible información</b>	<b>Variantes con 2 bits</b>	<b>Posible información</b>	<b>Variantes con 3 bits</b>	<b>Posible información</b>
0V	10 <sup>0</sup> C	0V-0V	10 <sup>0</sup> C	0V-0V-0V	10 <sup>0</sup> C
5V	20 <sup>0</sup> C	0V-5V	20 <sup>0</sup> C	0V-0V-5V	20 <sup>0</sup> C
		5V-0V	30 <sup>0</sup> C	0V-5V-0V	30 <sup>0</sup> C
		5V-5V	40 <sup>0</sup> C	0V-5V-5V	40 <sup>0</sup> C
				5V-0V-0V	50 <sup>0</sup> C
				5V-0V-5V	60 <sup>0</sup> C
				5V-5V-0V	70 <sup>0</sup> C
				5V-5V-5V	80 <sup>0</sup> C

**Fuente:** (Cano Martinez, 2012, pág. 83)

## **CAPITULO II**

### **2. MATERIALES Y MÉTODOS**

#### **2.1 METODOLOGÍA DE INVESTIGACIÓN**

El presente capítulo está enfocado en dar a conocer los procesos y métodos concisos en la aplicación de ingeniería inversa en reprogramación de memorias y microcontroladores de vehículos modernos referente a funciones especiales, seguidamente se procede a identificar las características funciones especiales y procesos de reprogramación en vehículos seleccionados, posteriormente se obtiene y evalúa el archivo de volcado de la ECU, BCM, Inmo Box y módulo de panel de instrumentos electrónico, inspeccionando valores hexadecimales modificables con el uso de equipos de programación, también se procede a extraer el PIN CODE, el número VIN y códigos crypto de transponder de llaves de la Inmo Box, para finalmente indagar en la red CAN del vehículo e identificar los códigos que modifican las funciones especiales. En el presente proyecto se ha empleado al método documental, método analítico y método experimental como pilar de la investigación.

##### **2.1.1 ENFOQUE INVESTIGATIVO**

La importancia de realizar los procesos de obtención de datos y análisis de los resultados obtenidos de la reprogramación de la ECU, BCM, Inmo Box y módulo de panel de instrumentos es comprender como se da la interacción de modificación de códigos a memorias y microcontroladores mediante la utilización de equipos de programación profesional.

De esta forma se pretende hacer un diagnóstico y dar soluciones a todas las eventualidades que se presentan en la electrónica del automóvil relacionada con las funciones especiales, además se quiere dar un enfoque más profundo en la red de comunicación CAN del vehículo y del equipo de programación para analizar los códigos hexadecimales emitidos y posteriormente en investigaciones futuras se pueda realizar un herramienta que cumpla funciones similares y sea más accesible para la comunidad automotriz.

## **2.1.2 TIPO DE INVESTIGACIÓN**

### **2.1.2.1 Método documental**

Para el desarrollo del presente trabajo se utiliza un método documental o bibliográfico para la recopilación de diferentes fuentes de investigación tales como artículos científicos, libros, tesis, revistas técnicas entre otros, de manera que la información sea confiable y aporte para el desarrollo del tema establecido.

El método documental es una serie de métodos y técnicas de búsqueda, procesamiento y almacenamiento de la información contenida en los documentos, en primera instancia, y la presentación sistemática, coherente y suficientemente argumentada de nueva información en un documento científico (Tancara, 1993, pág. 93).

### **2.1.2.2 Método analítico**

“El método analítico es un camino para llegar a un resultado mediante la descomposición de un fenómeno en sus elementos constitutivos” (Lopera, Ramírez, Zuluaga, & Ortiz, 2010, pág. 17).

Es un método analítico porque se analizarán los resultados de los datos obtenidos de registros de memorias de la ECU, BCM, panel de instrumentos y Inmo Box antes y después del proceso de reprogramación.

### **2.1.2.3 Método Experimental**

El método experimental es un método científico que se basa en la observación de los hechos, mediante el razonamiento inductivo, es el estudio que puede ser susceptible de observación, a partir de premisas particulares se obtiene conclusiones generales.

En el trabajo investigativo se pretende alcanzar resultados mediante la realización de pruebas y toma de datos de valores hexadecimales en el caso práctico de reprogramación de memorias en módulos automotrices para determinar que códigos son configurables y los efectos que produce en el vehículo.

## 2.2 PROCESOS METODOLÓGICOS

En la Figura 2.1 se esquematizo un flujograma que ayuda a la elaboración y el desarrollo del presente trabajo de investigación, resumiendo el proceso a seguir en la obtención y análisis de datos.

En el mismo se detalla la selección del vehículo para identificar las características, funciones especiales y procesos de reprogramación que se puede aplicar, para posteriormente de una manera ordenada realizar el desmontaje del módulo de control del motor (ECU), módulo de control de la carrocería (BCM), modulo del panel de instrumentos y sistema inmovilizador para la obtención del archivo de volcado e inspección de valores hexadecimales.

Después de obtener el archivo de volcado de cada módulo de control se realiza la evaluación de datos que contienen, pero de una manera individual.

Primero, se analiza los datos de la ECU y BCM para identificar que códigos son programables y se efectúa el caso práctico de reprogramación para comparar los valores obtenidos antes y después de este proceso e identificar qué efectos se producen en el vehículo.

El proceso también consiste en el análisis y evaluación de datos del archivo de volcado de la memoria del panel de instrumentos, de igual manera se identifica que códigos hexadecimales son programables para realizar el proceso de reprogramación e identificar qué efectos producen.

Además, se realiza la evaluación de los registros de la memoria de la Inmo Box o sistema inmovilizador, para realizar un análisis de los valores hexadecimales, se extrae el Pin Code que es un código de seguridad que tiene cada automóvil, el número VIN y código cripto de transponder de llaves con la finalidad de la reprogramación de códigos y mejoramiento del sistema.

Por último, se realiza la infiltración en la Red CAN para indagar e identificar los códigos de programación realizados.

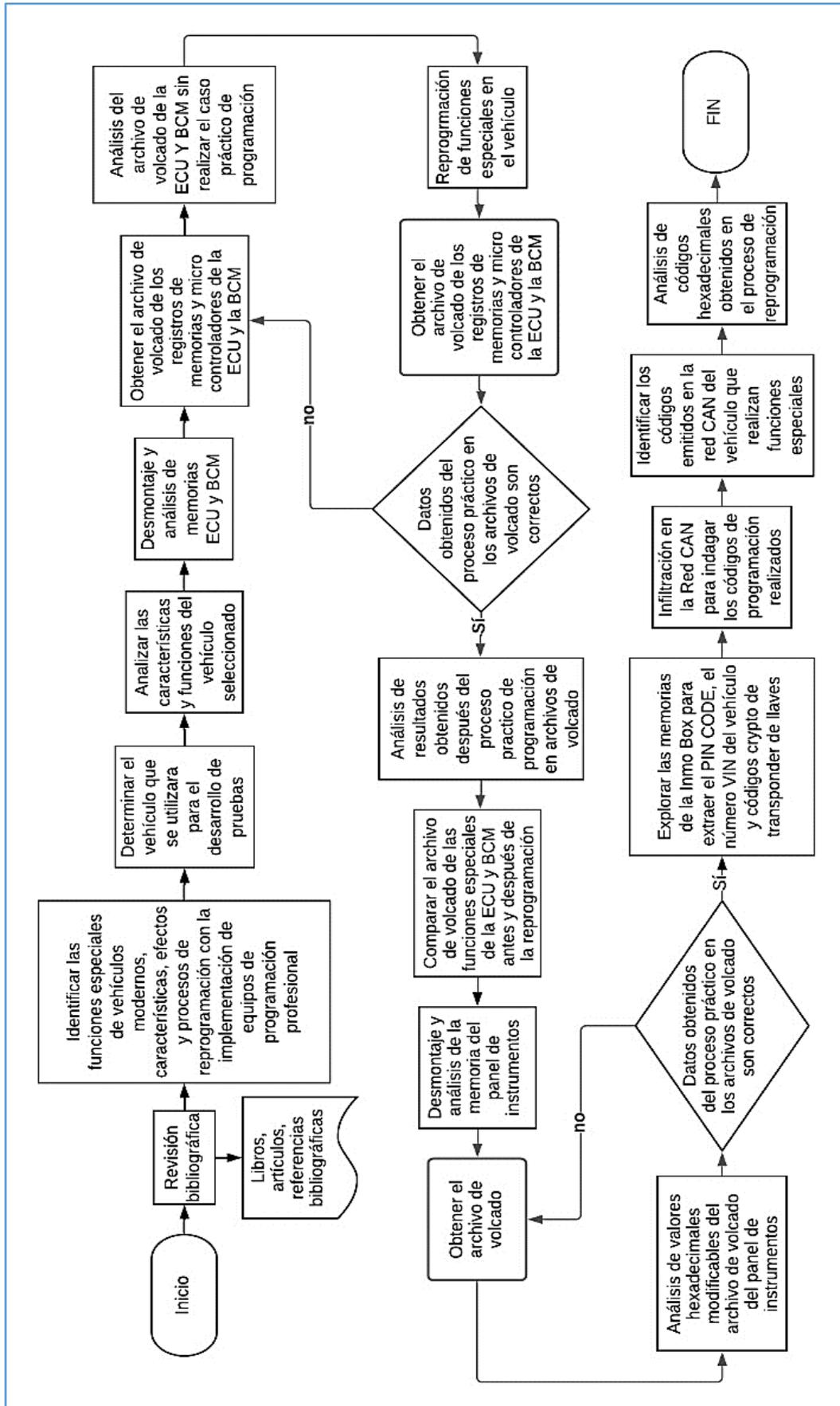


Figura 2.1 Proceso metodológico de obtención de datos y proceso de reprogramación

## 2.3 MATERIALES Y EQUIPOS

Para la obtención de datos del trabajo de investigación se utilizaron dos vehículos previamente seleccionados previamente cumpliendo con características técnicas y tecnológicas necesarias para el desarrollo del mismo.

Las especificaciones generales de los vehículos se dan a conocer de una manera más detallada en la Tabla 2.1

**Tabla 2.1** Especificaciones técnicas de vehículos para caso práctico.

<b>Marca</b>	<b>Peugeot</b>	<b>Kia</b>
<b>Clase</b>	Automóvil	Automóvil
<b>Modelo</b>	307 XS 1.6 16V 5P	Sportage r
<b>Año</b>	2003	2012
<b>Potencia (Hp/rpm)</b>	108/5800	152/6200
<b>Torque (Nm/rpm)</b>	147/4000	191/4700
<b>Caja de cambios</b>	De 5 velocidades	De 5 velocidades
<b>Tipo de combustible</b>	Gasolina	Gasolina

En lo relacionado a los equipos de trabajo para el desarrollo del presente tema de estudio, debido a que se indagara en los registros de memorias y microcontroladores de los módulos electrónicos de control de los vehículos mencionados anteriormente para la obtención de códigos hexadecimales, se utilizara diferentes equipos de programación profesional los cuales se describen a continuación:

### 2.3.1 X100 PAD 2 PRO

X100 PAD2 Pro es un programador de teclas para tabletas. No solo proporciona a los clientes una programación profesional clave, sino también las funciones especiales más necesarias para el taller. X-100 PAD2 es una tableta con las últimas tecnologías para realizar programación clave, ajuste de kilometraje, restablecimiento de luz de servicio de aceite, restablecimiento de luz de servicio de correa de distribución, restablecimiento de luz de advertencia de presión de neumáticos, restablecimiento de EPB, restablecimiento del cuerpo del acelerador, restablecimiento de DPF, restablecimiento de batería, dirección calibración del ángulo y diagnóstico del motor OBD-II, y se agrega especialmente con VW 4th y 5th IMMO (XTOOL, n.d.).



**Figura 2.2** X100 PAD 2 PRO - Version Full  
(AutoExacto, 2019)

En la Figura 2.2 se puede apreciar la nueva versión del XTOOL, X100 PAD2 PRO, con funciones especiales, es una Tablet programador, proporciona una programación de clave profesional.

### 2.3.2 UPA-USB

Las aplicaciones principales que posee este equipo son las siguientes: Extracción de PIN-CODE, Chip Tuning, Reseteo de módulos de Airbag, desmovilización, clonado de módulos (Ecu, BSI, UCH, etc.), Programación de Inmovilizadores, Ajuste de kilometraje, etc.



**Figura 2.3** Programador UPA-USB

En la Figura 2.3 se puede observar el UPA-USB que es un programador muy conocido en la electrónica automotriz. Soporta la totalidad de los dispositivos programables que se utiliza en tableros electrónicos, fusileras, Ecus de inyección, estéreos, módulos de airbag, etc.

### 2.3.3 K-TAG

El equipo de programación se comunica con todas las unidades de control, incluido las de generación actual mediante una comunicación directa con la micro. Permittedole leer y escribir de una forma directa el microprocesador de la centralita, EEPROM y memoria flash.



**Figura 2.4** Programador K-TAG  
(ALIENTECH, 2018)

K-TAG como se indica en la Figura 2.4 es un programador de banco para transmisión automática y centralitas de motor. Es un programador profesional que se utiliza para reescribir directamente a banco un archivo original o para modificar la de un vehículo.

### 2.3.4 CAN BUS ANALYZER

El analizador de bus CAN se conecta a un PC mediante el cable USB suministrado. Necesitará un PC con sistema operativo Microsoft Windows y un puerto USB. También hay un conector DB9 para el bus CAN.



**Figura 2.5** Programador CAN BUS ANALYZER Microchip

Microchip APGDT002 CAN Bus Analyzer Tool como se aprecia en la Figura 2.5 es una herramienta de desarrollo que es un monitor de bus CAN que se utiliza para desarrollar y depurar redes CAN de alta velocidad.

### 2.3.5 MINI VVDI KEY TOOL

La herramienta VVDI puede realizar funciones como: clonar transponder, editar transponder original, preparar o generar transponder, y renovar chip transponder.



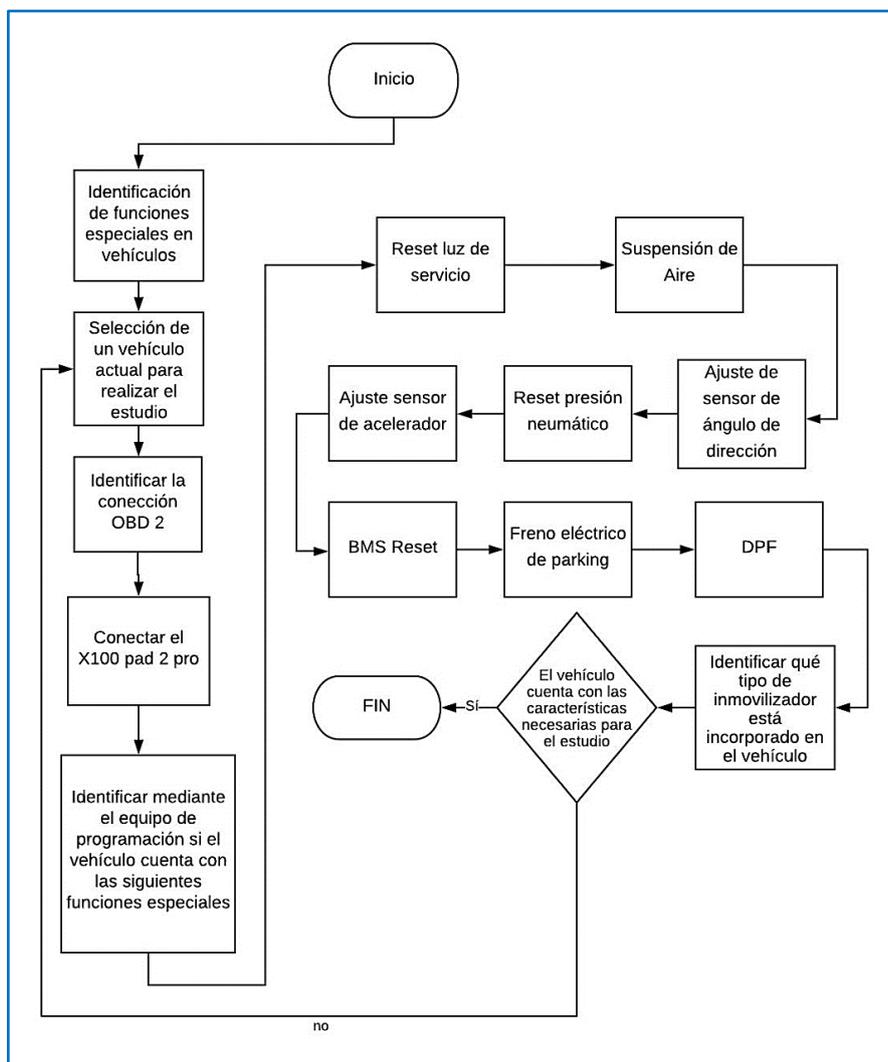
**Figura 2.6** MINI VVDDI KEY TOOL

En la figura 2.6 se indica la máquina de clonación y programación, Xhorse MINI VVDI Key Tool que funciona en iPhone y Android se puede comprobar el software en la pantalla de un teléfono celular o Tablet; es fiable y fácil de usar.

## 2.4 PROCESO DE IDENTIFICACIÓN DE FUNCIONES ESPECIALES

Los vehículos que hoy en día se encuentran en el mercado traen incorporado diferentes opciones mecánicas y digitales controladas mediante una computadora automotriz o ECU que brindan seguridad y confort a los ocupantes. Así, los fabricantes de automóviles siguen desarrollando vehículos con nuevas tecnologías, y de esta manera responder a las necesidades de los operarios.

## 2.4.1 FLUJOGRAMA PARA LA IDENTIFICACIÓN DE FUNCIONES ESPECIALES

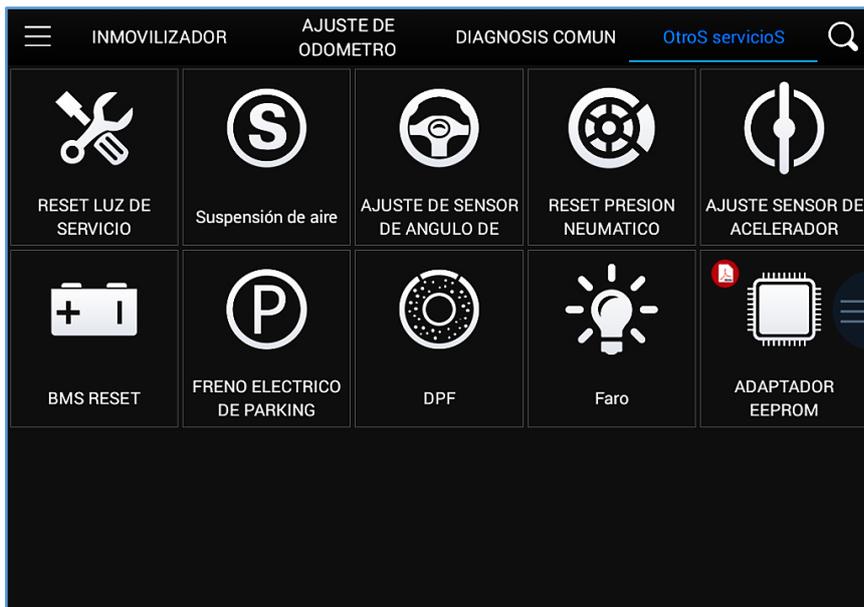


**Figura 2.7** Proceso metodológico para la identificación de funciones especiales

En la Figura 2.7 se esquematiza un diagrama de flujo para resumir de manera concisa el proceso para determinar las funciones especiales que podemos encontrar actualmente en vehículos modernos.

## 2.4.2 IDENTIFICACIÓN DE FUNCIONES ESPECIALES X100 PAD 2 PRO

En este tópico se realizó una investigación bibliográfica de todas las funciones especiales que tienen los vehículos, cada una de ellas se las encontrara en los diferentes módulos de control como la ECU, BCM, INMO BOX y panel de instrumentos electrónico.



**Figura 2.8** Simbología funciones especiales X100 PAD 2 PRO  
(Captura obtenida del equipo X100 PAD 2 PRO)

El proceso de identificación de funciones especiales se realizó mediante el equipo X100 PAD2 PRO, con el cual se reconoció funciones como: Reset luz de servicio, suspensión de aire, ajuste de sensor de ángulo de dirección, reset presión neumático, ajuste sensor de acelerador, sistema de reinicio de la batería o BMS Reset, freno eléctrico de parking y el sistema de filtro de partículas o DPF de vehículos diésel. Estas señales se las representa mediante testigos luminosos o símbolos los cuales se puede observar de una manera más detallada en la Figura 2.8.



**Figura 2.9** Conexión X100 PAD 2 PRO con el vehículo

Después del reconocimiento de las funciones existentes en la actualidad mediante un equipo de programación se procedió a la identificación de funciones en el vehículo, en primera instancia, como se observa en la Figura 2.9 se lo realizó acoplado el equipo al vehículo a través del conector OBD II para indagar los datos internos y verificar las diferentes funciones con las que cuenta.

Una vez identificadas las funciones del vehículo se procede a resetear cada función de manera individual y obtener un archivo de volcado original y un archivo de volcado después de este proceso.

## **2.5 PROCESO PARA OBTENER EL ARCHIVO DE VOLCADO**

La información del vehículo está contenida en la memoria de los diferentes módulos de control, al realizar la lectura de esta se obtendrá un archivo el cual se modifica más adelante.

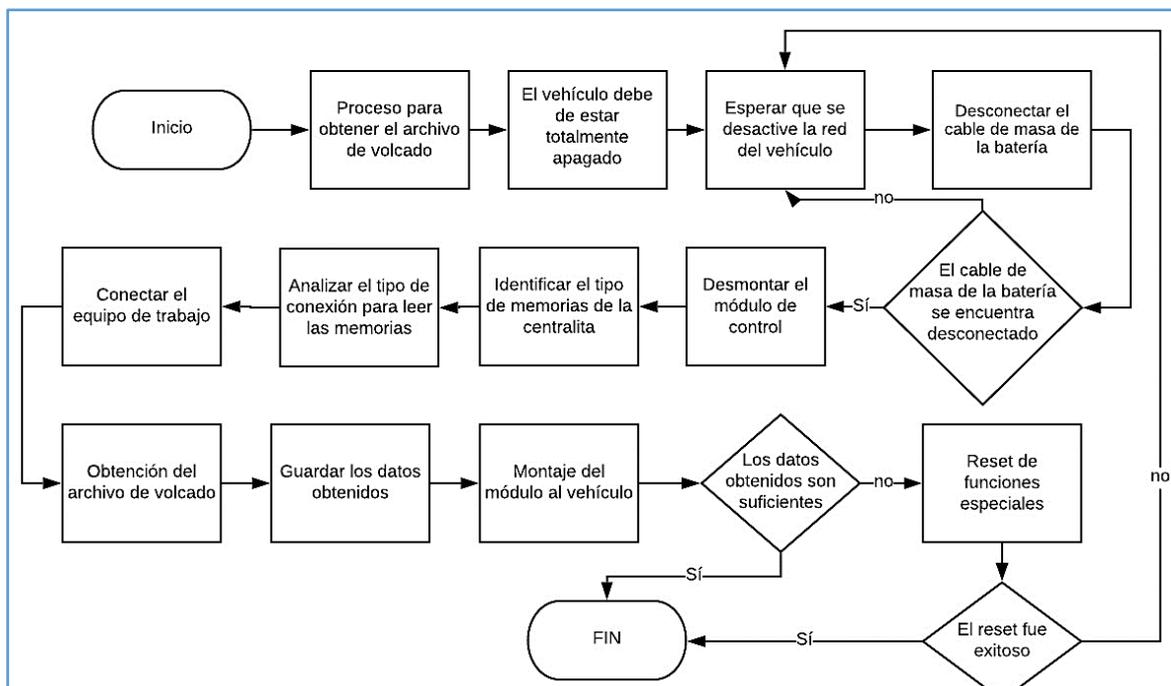
La obtención del archivo de volcado es una parte fundamental para realizar procesos de reprogramación en módulos automotrices, ya que guarda toda la información del vehículo mediante valores hexadecimales, por lo tanto, estos datos ayudan a identificar las funciones específicas que se realizan tanto al interior y exterior del automóvil.

La lectura de la memoria de la ECU, BSI y panel de instrumentos depende del modelo y año del vehículo, según estos parámetros podemos encontrar casos prácticos como desoldar y desmontar la memoria (Díaz & Sopa, 2013, pág. 21).

En el caso de desoldar la memoria en vehículos antiguos específicamente para que la información pueda ser leída por medio de un lector de memorias se toma en consideración desoldar la misma, este proceso a su vez requiere de experiencia para la manipulación ya que por lo contrario podría presentar daños en la información interna o físicamente (Semante, 2016, pág. 47).

En el caso de desmontar la memoria existen unidades de control que no necesitan realizar el desmontaje de las memorias desoldándolas ya que vienen incrustadas de fábrica en un sócalo, el cual permite su desmontaje con gran facilidad, y de la misma manera poder leer la información por medio de un programador de memorias (Semante, 2016, pág. 47).

### 2.5.1 FLUJOGRAMA PARA LA OBTENCIÓN DE ARCHIVO DE VOLCADO



**Figura 2.10** Proceso metodológico para la obtención de archivos de volcados

En la figura 2.10 se esquematiza un diagrama de flujo donde se resume el proceso a seguir para obtener archivos de volcado de los módulos de control del automóvil. Cabe mencionar que para la obtención de datos se puso en práctica los casos antes mencionados.

### 2.5.2 ARCHIVO DE VOLCADO ECU

Para la identificación de las características generales de la ECU y realizar la lectura de la memoria fue necesario contar con equipos de programación profesionales. El equipo K-TAG es una herramienta utilizada en el campo de la electrónica y software automotriz el cual permite leer y escribir ECUs automotrices.

La empresa Alientech proporciona el módulo K-TAG en dos versiones.

- Master: La versión master del módulo trae incorporado todos los protocolos y pasos a seguir para trabajar en diversas centralitas y además realiza la autocorrección del CheckSum, que es la suma de verificación matemática de todos los valores almacenados en una memoria para proteger la integridad de los mismos, proceso de

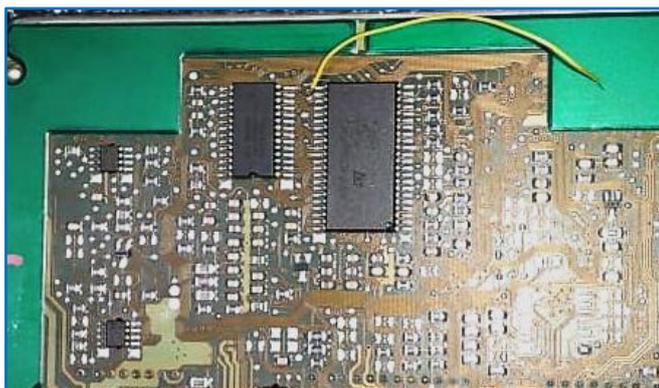
gran relevancia al momento que se realiza una modificación de los registros internos de la ECU.

- Slave: La versión Slave está enfocada directamente para personas que comienzan en el campo de la programación automotriz y chip tuning, cuenta con opciones limitadas y se necesita adquirir los protocolos de lectura. Además, los archivos leídos se deben enviar a técnicos especialistas de la empresa proveedora para la revisión y autocorrección del CheckSum.



**Figura 2.11** K-TAG Version master 7.020

La Figura 2.11 indica el equipo K-TAG Versión master 7.020 que cuenta con un software 2.23, utilizado en el desarrollo de la obtención del archivo de volcado en la ECU.



**Figura 2.12** Conexión al circuito impreso de la centralita

El módulo K-TAG facilita el acceso a cada uno de los componentes como microcontrolador, EEPROM y memoria flash mediante previa conexión de un puerto BDM, que es un conector para el equipo de programación. Esto se lo realiza soldando un cable directamente a la placa del circuito impreso de la centralita, como se puede apreciar en la Figura 2.12.



**Figura 2.13** Conectores ECU



**Figura 2.14** Carcasa ECU

Para este proceso es necesario el desmontaje del módulo de control o ECU, retirando los conectores del vehículo con la memoria y pernos de fijación de la carcasa como se indica en las Figuras 2.13 y 2.14.



**Figura 2.15** Conexión KTAG mediante puerto USB

La conexión del módulo KTAG es posible con la ayuda de una computadora portátil en donde se puede guardar los archivos leídos y se conecta mediante un cable USB al equipo de programación como se muestra en la Figura 2.15



**Figura 2.16** Menú principal módulo KTAG  
(Captura obtenida del equipo K-TAG)

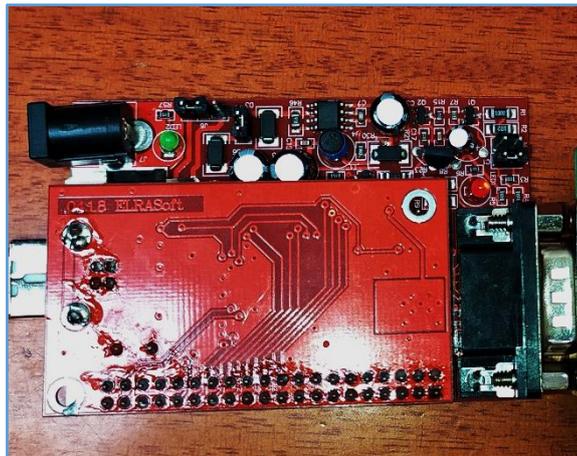
Una vez que el módulo KTAG se conecta a la computadora portátil, se prosigue a la lectura y respaldo de los archivos de la misma.

La Figura 2.16 indica el menú principal del equipo en donde se puede seleccionar opciones como: la forma de ingreso que puede ser mediante selección de vehículo o protocolo de comunicación, además se puede encontrar configuración del idioma en el que se desea trabajar (este puede ser inglés o alemán) y finalmente se dirige al acceso de la selección de la ECU, en donde aparecen recomendaciones para seguir un proceso ordenado para la obtención del archivo.

## 2.5.2 ARCHIVO DE VOLCADO BCM Y PANEL DE INSTRUMENTOS

Para la identificación y lectura del módulo de la carrocería BCM y panel de instrumentos electrónico se realizó con la implementación del equipo de programación UPA.

El módulo UPA es un equipo electrónico que cumple funciones de lectura y escritura de microcontroladores que se enlaza por medio de una computadora portátil o escritorio.



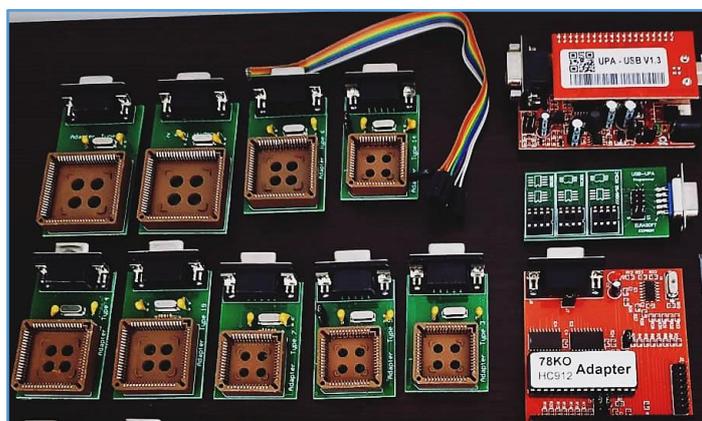
**Figura 2.17** Módulo UPA versión V 1.3

En la Figura 2.17 se puede observar el módulo de programación UPA versión V 1.3 con el que se realizó la obtención de datos en este apartado.

El módulo UPA cuenta con características de funcionamiento como:

- UPA se utiliza simplemente para lectura y escrituras de memorias EEPROM y microcontroladores
- No maneja años, modelo y marcas de vehículos.
- UPA no tiene funciones integradas para modificar la información de los dispositivos de memoria como cambio de kilometraje, Reset de bolsas de aire, Reset luz de servicio entre otros. Esto se lo realiza de manera manual.

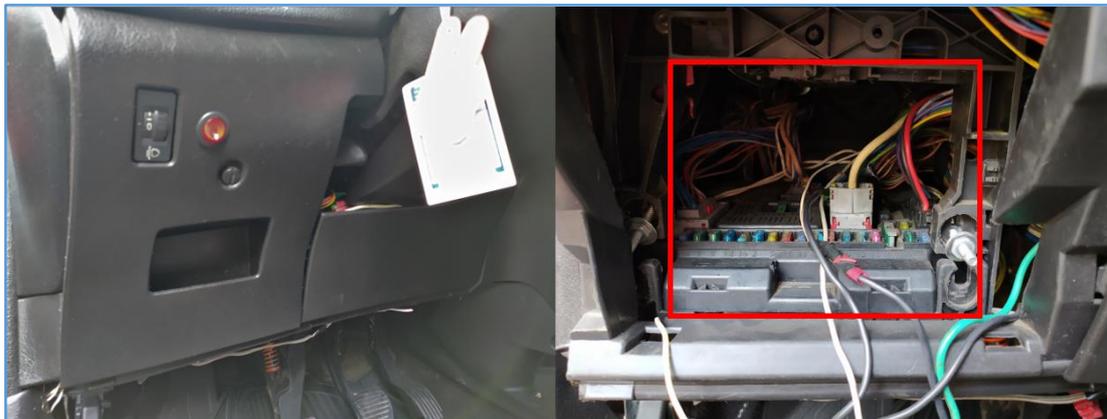
El programador UPA dispone de adaptadores que facilitan la lectura de memorias de los diferentes módulos de control permitiendo un trabajo cómodo y seguro.



**Figura 2.18** Adaptadores programador UPA

En la Figura 2.18 se puede observar los diferentes adaptadores del equipo UPA.

Para este proceso es importante identificar la ubicación de los módulos de control BCM y panel de instrumentos en el vehículo para realizar el desmontaje de la centralita de cada unidad.



**Figura 2.19** Ubicación módulo de control BCM

Por lo general el módulo de la carrocería se encuentra ubicado junto al control de apertura del capo, junto a la caja de fusibles como se observa en la Figura 2.19.

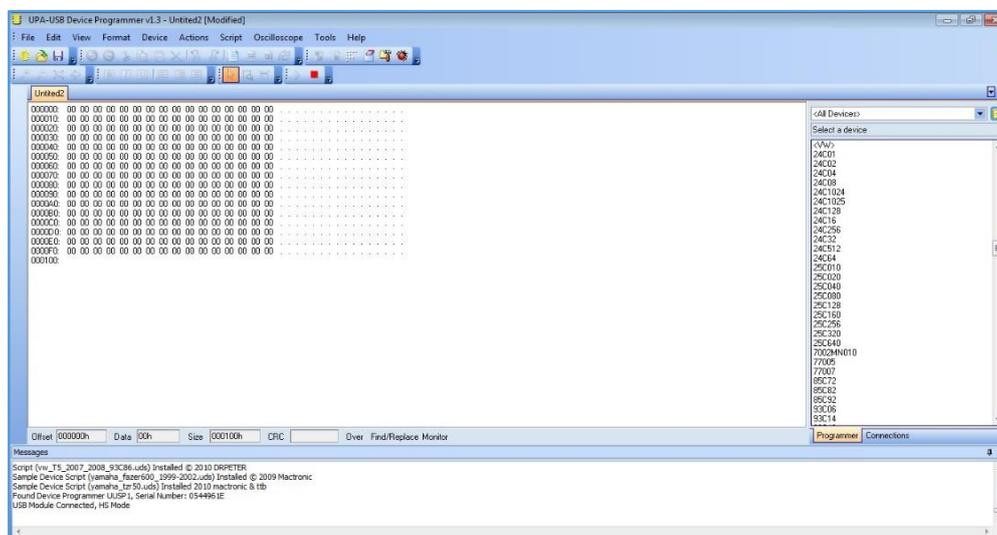
Por otra parte, el módulo del grupo de instrumentos es visible para el usuario, pero en este caso es de gran relevancia identificar la posición de las agujas del indicador de temperatura del refrigerante, indicador de combustible, tacómetro o cuentarrevoluciones y del indicador de velocidad, estas precauciones ayudaran a un correcto montaje del módulo.



**Figura 2.20** Panel de instrumentos electrónico

En la Figura 2.20 se puede apreciar el tablero del panel de instrumentos del vehículo seleccionado.

Una vez realizado el desmontaje del módulo BCM y panel de instrumentos se procede retirar la carcasa para acceder al circuito impreso y analizar el tipo de conexión para acceder al archivo de volcado. El equipo de programación UPA se vincula al computador mediante un cable DLC tipo USB al computador donde aparece el menú principal del software del módulo de programación.



**Figura 2.21** Menú principal equipo UPA

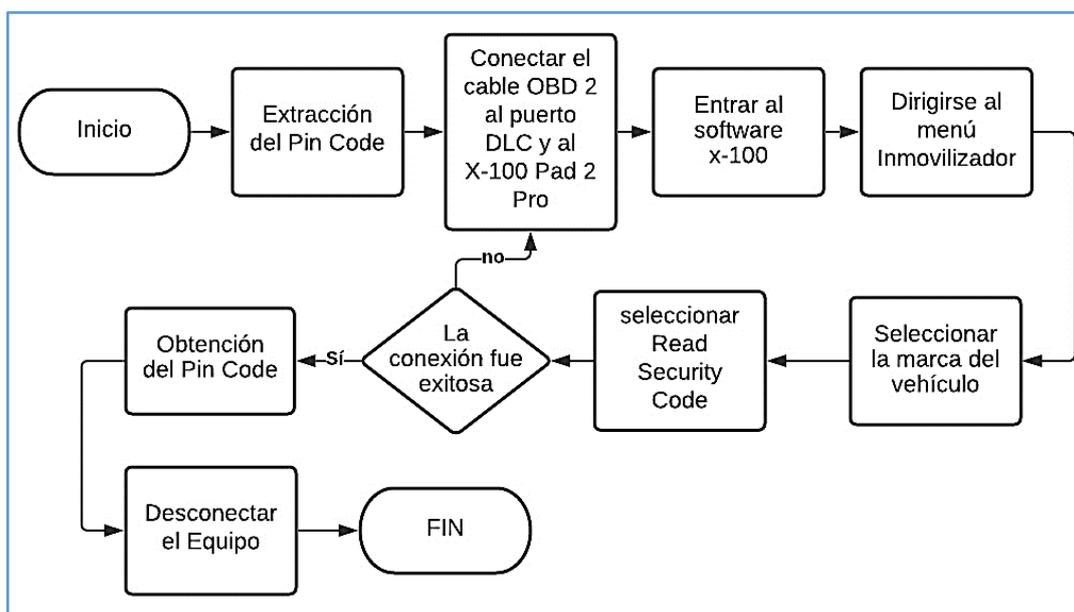
En la Figura 2.21, se muestra el menú principal del módulo UPA, en esta pantalla se selecciona el tipo de memoria de los diferentes módulos de control y se da lectura al archivo original.

## 2.6 PROCESO DE OBTENCIÓN DE PIN CODE DE TRANSPONDER DE LLAVES

En el Ecuador los vehículos actuales traen incorporados sistemas de seguridad antirrobo, la información de este sistema se la encuentra en las unidades de control electrónica ECU o BCM, en los registros internos se encuentra códigos los cuales mediante valores hexadecimales expresan la información del código de seguridad o PIN CODE del vehículo, tipo de motor y código del transponder.

El PIN CODE se encuentra encriptado en el sistema inmovilizador del vehículo y es entregado por los fabricantes o diferentes concesionarios del país. Además, es uno de los puntos de mayor relevancia al momento de programar porque permite el acceso directo a la computadora del auto; sin este código de seguridad es imposible generar la clonación de una nueva llave.

### 2.6.1 FLUJOGRAMA PARA LA OBTENCIÓN DE OBTENCIÓN DE PIN CODE



**Figura 2.22** Proceso metodológico para la obtención de PIN CODE

En la Figura 2.22 se observa un proceso metodológico de la obtención del pin code de transponder de llaves, en la cual comprende como conectar el equipo de programación al vehículo permitiendo tener información de los datos internos.

Existen varios métodos para obtener el código de seguridad de la llave del vehículo siendo el más sencillo y práctico de realizar la comunicación mediante escáner. Este método consiste en la conexión del equipo mediante el puerto OBD II, selección de la marca, el modelo y obtener el PIN CODE. Para este proceso fue necesario la implementación del equipo X100 PADO 2 PRO ya que cumple con las características antes mencionadas.

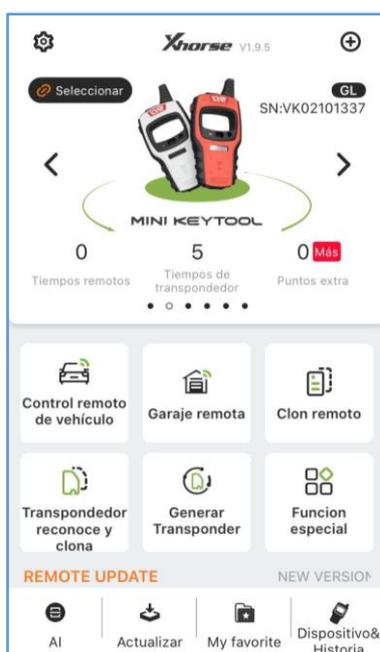
## 2.6.2 OBTENCIÓN CÓDIGO TRANSPONDER DE LLAVE

Para la programación de una nueva llave del vehículo seleccionado para el caso práctico fue necesario la verificación del tipo de microchip y código de transponder de la llave original. La obtención del código de transponder se lo realizo mediante el uso del equipo MINI VVDI KEY.



**Figura 2.23** Equipo MINI VVDI KEY

En la Figura 2.23 se puede apreciar el MINI VVDI KEY que es un equipo clonador de chip de transponder de llaves. Este módulo permite la lectura e identificación del código de transponder del chip insertado en el encapsulado de llave. Además, permite clonar un nuevo chip con una lectura automática o de manera manual y en la interface del equipo indica si es posible duplicar.



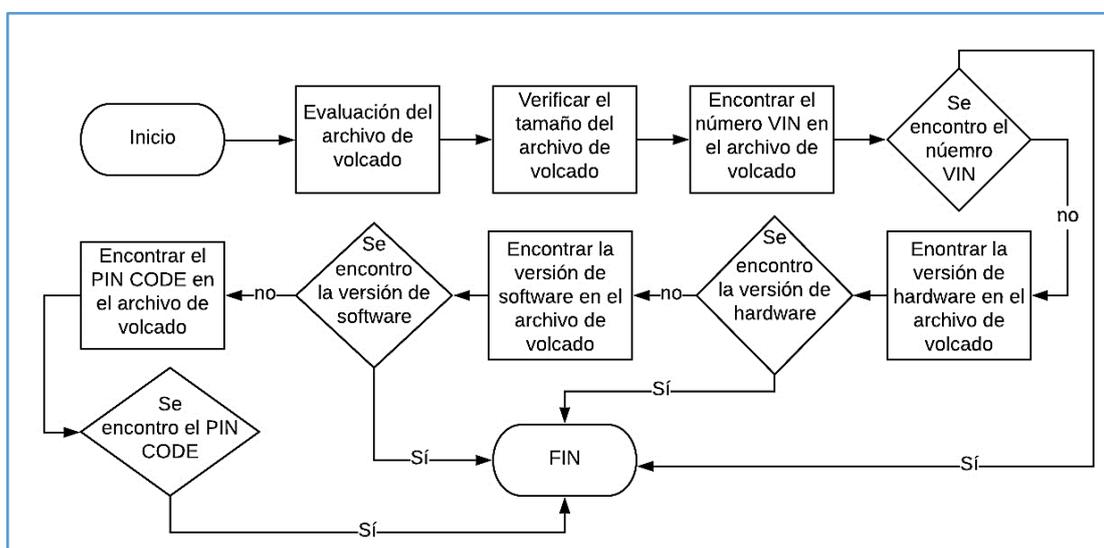
**Figura 2.24** Interface MINI VVDI

En la Figura 2.24 se aprecia la pantalla principal de la aplicación en esta se encuentra principalmente las funciones que realiza el MINI VVDI, dentro de esta interface se encuentra las opciones lectura del transponder opción utilizada para leer el chip insertado en el mango de la llave del vehículo; y generar transponder utilizada para la clonación del transponder de la llave original a un chip virgen.

## 2.7 PROCESO DE EVALUACIÓN ARCHIVO DE VOLCADO

La evaluación del archivo de volcado es un proceso para verificar si la lectura del archivo de los módulos automotrices se la ha realizado de manera correcta, debido a que en el transcurso de la lectura existen errores comunes como el mal funcionamiento del equipo de programación.

### 2.7.1 FLUJOGRAMA PARA LA EVALUACIÓN DE ARCHIVO DE VOLCADO



**Figura 2.25** Proceso metodológico para la evaluación de archivos de volcado

En la Figura 2.25 se muestra un diagrama de flujo en donde se resume los pasos a seguir para tener una evaluación de manera correcta.

En el vehículo que fue objeto de estudio se encontró tres archivos de volcados originales correspondientes a cada módulo de control.

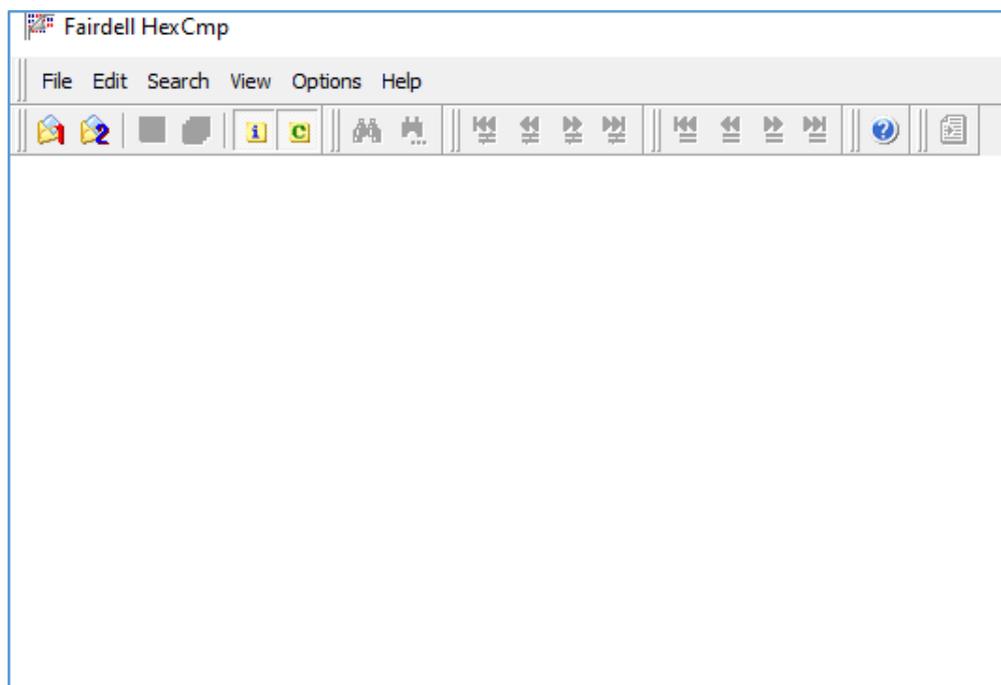




**Figura 2.27** Programa “FAIRDELL HEX CMP”

En la Figura 2.27 se puede apreciar el nombre y logotipo del programa, que es un software libre de fácil acceso e instalación.

HexCmp es un programa que cumple funciones de verificación, modificación y comparación de datos de archivos binarios y códigos hexadecimales. La comparación se la realiza con dos archivos diferentes que se abren en una pantalla dividida y los valores hexadecimales modificados se resaltan con un esquema de color diferente y visible.



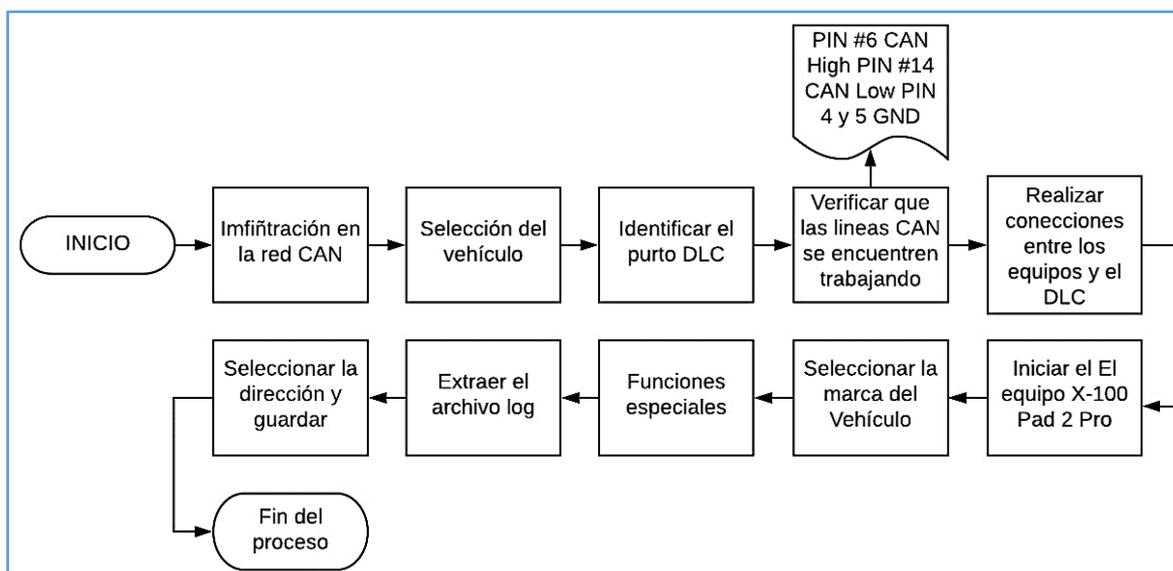
**Figura 2.28** Menú principal HexCmp

La Figura 2.28 muestra en menú principal del programa HexCmp en donde se abre los diferentes archivos que serán objeto en la comparación de datos.

## 2.8 PROCESO DE INFILTRACIÓN EN LA RED CAN

Otro método para realizar reprogramación en módulos automotrices es a través de la red CAN, el escáner emite a la computadora un código de reprogramación indicándole que modulo requiere reprogramar, la infiltración en la red CAN es un procedimiento para indagar y obtener el código de reprogramación, y con los datos obtenidos se pueda realizar una reprogramación sin necesidad de utilizar un equipo costoso.

### 2.8.1 FLUJOGRAMA PARA LA INFILTRACIÓN EN LA RED CAN



**Figura 2.29** Proceso metodológico para la infiltración en la red can

En la Figura 2.29 se indica un proceso metodológico para la infiltración e identificación de códigos emitidos para las funciones especiales en la red CAN.

Para este proceso fue necesario implementar equipos de programación profesional como el equipo X100 PAD 2 PRO y el equipo microchip CAN BUS ANALYZER que es una herramienta para desarrollar y depurar una red CAN de alta velocidad y fácil de usar, el cual refleja en la pantalla de una computadora portátil todos los procesos, las ordenes de trabajo que recibe y las acciones que ejecuta la unidad de control electrónico del vehículo en tiempo real.



**Figura 2.30** Herramienta CAN BUS ANALYZER

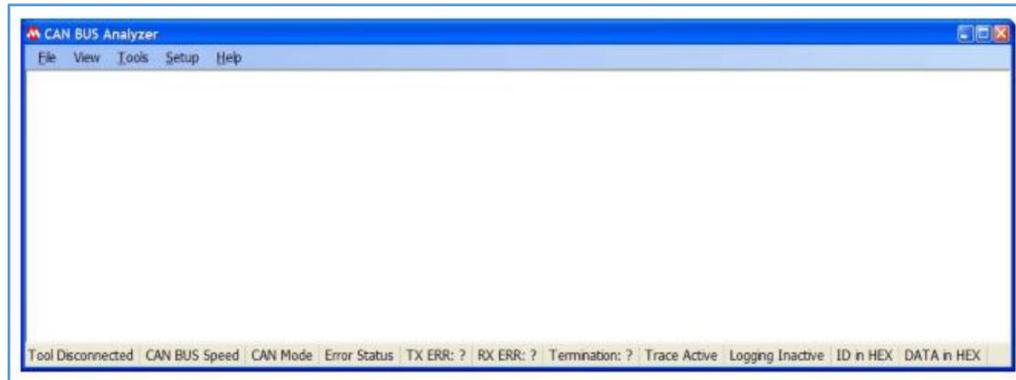
En la Figura 2.30 se puede apreciar la herramienta CAN BUS Analyzer, que es compatible con sistemas OBD II y cuenta con el protocolo ISO 11898-2 que tiene la ventaja de funcionamiento de una red CAN de alta velocidad.

La herramienta se conecta directamente a al puerto DLC del vehículo mediante un conector DB9 con terminal de tornillo ajustable por seguridad y fiabilidad en la transferencia de datos.



**Figura 2.31** Conexiones para el proceso de obtención de códigos

En la Figura 2.31 se indica las conexiones que se realizaron para este proceso, con un cable doble conector OBD II y SPLITTER de 16pines cable macho a hembra, se conecta el equipo X100 PAD 2 PRO y del otro extremo se conecta el microchip CAN BUS ANALYZER que a su vez va conectado a la laptop por cable USB.



**Figura 2.32** Ventana de trabajo CAN BUS Analyzer

Una vez realizado todas las conexiones necesarias para la comunicación de datos, en la pantalla de la computadora portátil se ejecuta el software del microchip donde aparece una ventana de la interfaz gráfica o menú principal como se muestra en la Figura 2.32.

## **CAPITULO III**

### **3. RESULTADOS Y DISCUSIÓN**

En este capítulo se describirá el proceso que conlleva la obtención de información bibliográfica para el análisis de las funciones especiales identificadas en los diferentes módulos automotrices y el reseteo de las funciones encontradas en el vehículo Peugeot 307 año 2003. Después se dará a conocer la obtención y evaluación de datos de los archivos de volcados para extraer los códigos hexadecimales de programación del mismo vehículo. Además, se realizará un análisis del proceso en la infiltración de la red CAN del vehículo KIA Sportage r año 2011 para verificar que valores hexadecimales cambian al momento de reprogramar de manera directa en la red de comunicación del vehículo.

#### **3.1 FUNCIONES ESPECIALES EN MODULOS AUTOMOTRICES**

En este apartado se da a conocer las características generales de cada función de los vehículos modernos, el módulo en el que se encuentra y el momento que entra en acción cuando se generan códigos de falla o existen errores en el automóvil. Por otro lado, se identifica los vehículos que hoy en día traen incorporado estas funciones tomando en cuenta el origen y marca del equipo X100 PAD 2 PRO. Finalmente se describe el proceso de la identificación y reseteo de funciones especiales en el vehículo Peugeot 307.

##### **3.1.1 IDENTIFICACIÓN FUNCIONES ESPECIALES X100 PAD 2 PRO**

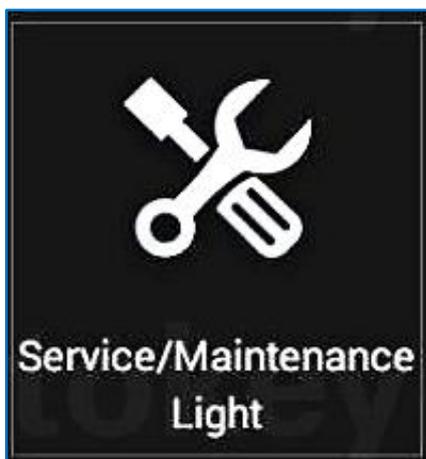
Como se mencionó en el punto anterior la identificación de funciones existentes en la actualidad en vehículos con tecnología electrónica se la realizó con la implementación del equipo de programación X100 PAD 2 PRO. Este equipo cuenta con funciones relacionadas al mantenimiento preventivo del motor, correcciones sobre el ángulo de dirección, ajuste del sensor de posición del acelerador, revisiones referentes al cambio de la batería, entre otras.

Las funciones encontradas en el equipo se dan a conocer de una manera más detallada a continuación:

### 3.1.1.1 Reset luz de servicio

La luz de advertencia de servicio se enciende de forma permanente indicando que el vehículo ha alcanzado un límite de kilómetros recorridos por lo que es necesario una inspección o un mantenimiento preventivo.

Esta señal se encuentra de forma de una llave inglesa la cual quedara de forma permanente encendida una vez que el vehículo alcance dicho kilometraje.



**Figura 3.1** Señal - Reset luz de servicio

La Figura 3.1, muestra la señal de luz de servicio, esta se encenderá durante seis segundos cada vez que se encienda el vehículo para probar la bombilla y luego se apagará. Antes de cumplir los 5000 km de recorrido se encenderá una luz intermitente para advertir un mantenimiento requerido.

La luz permanecerá encendida cuando se ha llegado al intervalo de los 5000 km de recorrido, siendo este un recordatorio final para realizar un mantenimiento.

La advertencia de la luz de servicio influye en una inspección y ajuste de todos los fluidos, también un cambio de aceite, cambio de filtro de motor, inspección multipunto, rotación de neumáticos, esto varía dependiendo la marca y el modelo.

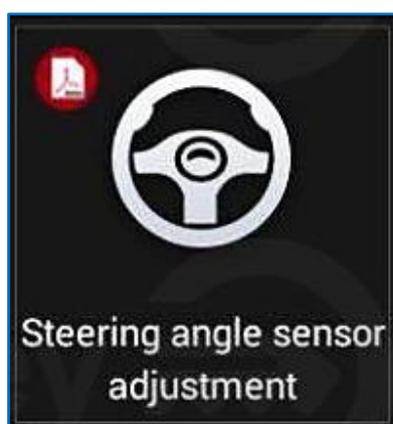
En la Tabla 3.1, se describe el origen y marca de los vehículos modernos que cuentan con la función de Reset luz de servicio.

**Tabla 3.1** Origen y marca de vehículos - Función Reset luz de servicio

AMERICA	ASIA	CHINA	EUROPA
BUICK	ACURA	CHERY	PORSHE
CADILLAC	HONDA	JMC	BMW
CHEVROLET	HYUNDAI	ROEWE	BENZ
CHEVROLET/GMC	ISUZU	MG	VOLVO
HUMMER	KIA	BAIC	LANDROVER
CHRYSLER/DODGE	MITSUBISHI	BARGWARD	SAAB
FORD/MERCURY	NISSAN	SUZUKI	PEUGEOT
OLDSMOBILE	INFINITY	FAW	CITROEN
PONTIAC	SUZUKI	FAW DAIHATSU	OPEL
SATURN	TOYOTA	GREAT WALL	JAGUAR
SCION	LEXUS	ZOTYE	SMART
	SUBARU	CHANGAN	MASERATI
	PROTON	MAXUS	FERRARI
	MAZDA	QOROS	LAMBORGHINI
		YAMAAUTO	BUGATI
		HNMAZDA	FIAT
		COWIN AUTO	LANCIA
		GEELY	ALFA ROMEO
			VW/AUDI/SKODA

### 3.1.1.2 Ajuste de sensor de ángulo de dirección

La luz de advertencia del ángulo de dirección se encuentra en el cuadro de instrumentos electrónico. Se utiliza para avisar sobre funciones anómalas o fallos en la dirección asistida electromecánica.



**Figura 3.2** Señal - Ángulo de dirección

En la Figura 3.2, se observa la señal del ajuste del sensor del ángulo de dirección, este testigo luminoso se encenderá de dos maneras dependiendo el color, si se enciende en amarillo,

indicará un aviso no relevante. Por otro lado, si se enciende en color rojo y existe una señal de aviso acústico hay que acudir de manera urgente a un centro de mantenimiento vehicular. Cabe recalcar que esta función se encuentra en la ECU del vehículo.

Es necesario tener el sensor de ángulo de dirección bien calibrado, para que algunos sistemas no funcionen incorrectamente, generando situaciones inseguras en la conducción. Uno de los inconvenientes que se puede presentar es que la dirección asistida actúa demasiado o muy poco.

Por lo tanto, es necesario realizar una calibración cuando, se ha sustituido el sensor de ángulo de dirección, se ha sustituido la unidad de control ABS, se ha sustituido la columna de la dirección, cambio de los ángulos de inclinación, cualquier trabajo en la dirección o suspensión.

En la Tabla 3.2 se describe el origen y marca de los vehículos modernos que cuentan con la función de ajuste de sensor de ángulo de dirección.

**Tabla 3.2** Origen y marca de vehículos - Función ajuste sensor ángulo de dirección

AMERICA	ASIA	CHINA	EUROPA
GM	ACURA	PO CHUN CAR	PEUGEOT
CRYSLER	HONDA	BAIC	CITROEN
FORD	HYUNDAI	FOTON	VW
DODGE	MAZDA	BISU	AUDI
JEEP	KIA	BYD	SKODA
DODGE	SUBARU	CHANG AN	SEAT
	NISSAN	GREAT WALL	LAND ROVER
	INFINITY	CHANG FE MOTORS	VOLVO
	LEXUS	CHANG HE	BENZ
	TOYOTA	DONG FENG	JAGUAR
	SSANG YONG	SOUEAST	BMW
		TRUMPCHI	MINI
		JAC	PORSHE
		JMC	RENAULT
		CHERY	DACIA
		MUSTANG	FERRARY
		COWIN AUTO	MASERATI
		GEELY	

### 3.1.1.3 Reset presión neumático

El sistema TPMS de control de presión de los neumáticos es un dispositivo que se encuentra dentro de los sistemas de seguridad, este sistema cumple la función de alertar al conductor al momento de que una de las ruedas del vehículo no tenga la presión necesaria.



**Figura 3.3** Señal - Reset presión neumático

El testigo luminoso de esta función se puede observar en el tablero de instrumentos de forma de un neumático o bien con las letras TPMS como se indica en la Figura 3.3.

Los sistemas básicos indican al conductor que un neumático ha perdido presión sin especificar el mismo, en cambio los sistemas más avanzados indican cual es el neumático que ha perdido presión, además indica la presión y la temperatura de cada uno a través del sistema a bordo.

Existen dos formas de funcionamiento, de forma directa recibe información del sensor ABS que mide el número de vueltas que da cada rueda, cuando en un neumático ha disminuido la presión tiene menos diámetro entonces tiene que dar más vueltas para recorrer la misma distancia.

La segunda forma de funcionamiento es de forma directa, que está instalado en cada neumático un sensor de presión, pueden estar pegados en la goma del neumático o puestos en la válvula. El Reset de presión de neumático se lo aplica cuando suceden las siguientes eventualidades, desprogramación de la unidad de mando, presión de los neumáticos insuficiente, falsas alarmas (el calor del sol puede hacer subir la presión), daño en la estructura de un neumático, sustitución de una rueda y cuando la batería del sensor esta desgastada.

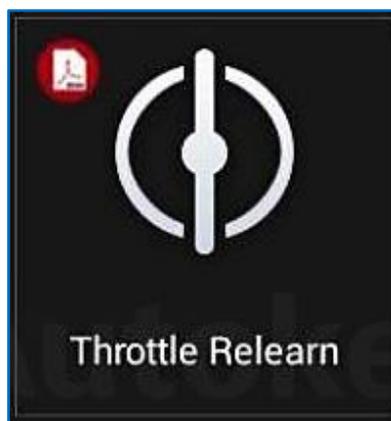
En la Tabla 3.3, se describe el origen y marca de los vehículos modernos que cuentan con la función de Reset presión del neumático.

**Tabla 3.3** Origen y marca de vehículos - Función Reset presión del neumático

AMERICA	ASIA
GM	TOYOTA
	LEXUS
	MITSUBISHI

### 3.1.1.4 Ajuste sensor de acelerador

En el momento que existe una falla en el sensor de posición del acelerador, la unidad electrónica de control o ECU, sustituye la señal errada que envía el sensor TPS por una señal artificial que se basa en el giro que realiza el motor.



**Figura 3.4** Señal - Ajuste sensor de aceleración

El sensor flojo o dañado causa señales intermitentes y se enciende la lámpara de CHECK ENGINE o luz de advertencia del motor de manera permanente, el testigo luminoso del ajuste del sensor del acelerador se puede apreciar en la Figura 3.4.

Uno de los tantos problemas que presentan los vehículos modernos, es una mala calibración del cuerpo de aceleración. Existen diversos motivos por lo que sucede. Tenemos los siguientes: si se realizó una afinación, si se cambia la computadora del auto o el cuerpo de aceleración, cuando el ralentí o la velocidad de avance de tiempo se encuentra fuera del rango de operación, y por un mantenimiento preventivo del motor. En estas cuestiones, es necesario realizar una calibración del cuerpo de aceleración. Para lo cual en diferentes casos

es necesario el uso de un scanner automotriz, para que la nueva condición de funcionamiento este programada en la computadora.

Un mal funcionamiento del sensor de aceleración nos puede dar las siguientes fallas: Luz Mil, falta de aceleración, aceleración lenta, falta de aceleración, marcha en ralentí inestable o alta, códigos de falla afines con TPS, motor de control de la mariposa o APP. Es necesario realizar una calibración para mejorar el funcionamiento del motor además el cuerpo de aceleración no solo está encargado de realizar la aceleración del vehículo, también realiza los controles de tracción, ayuda para la transmisión automática, en los sistemas de START / STOP, el parqueo automático y la aceleración automática dirigida de manera externa.

En la Tabla 3.4 se describe el origen y marca de los vehículos modernos que cuentan con la función de ajuste de sensor de acelerador.

**Tabla 3.4** Origen y marca de vehículos - Función de ajuste de sensor de acelerador

AMERICA	ASIA	CHINA	EUROPA
GM	ACURA	BRILLIANCE	AUDI
CRYSLER	HONDA	BAIC	BENZ
FORD	HYUNDAI	BYD	BMW
DODGE	DAEWOO	CHANGAN	CITROEN
JEEP	KIA	CHANGHE SUZUKI	JAGUAR
LINCOLN	TOYOTA	CHERY	OPEL
	NISSAN	DAIHATSU	LAND ROVER
	INFINITY	ENRANGER	PORSHE
	LEXUS	FAW	PEUGEOT
	MITSUBISHI	FOTON	RENAULT
		FODAY	SAAB
		GEELY	SEAT
		GREAT WALL	VOLVO
		GAC GONOW	SKODA
		JAC	VW
		LIFAN	
		LINDWIND	
		SUZUKI	

### 3.1.1.5 BMS Reset

A partir del año 2011 se incorporó en ciertos vehículos el sistema de reinicio de la batería, con el objetivo de que la operación de los módulos y el sistema eléctrico sea más eficiente. Esta función se encuentra en el módulo de control de la carrocería BCM, y se realiza cuando

la batería tenga 750 días de uso aproximadamente y se necesite un cambio, una vez realizado esto se reinicia el módulo de control.



**Figura 3.5** Señal - BMS Reset

Para el reseteo de la función se ingresa a los datos del vehículo para conocer cuantos días de trabajo tiene la batería, si excede en días se procede al cambio y reseteo. En la Figura 3.5 se muestra la señal de advertencia que indica el vehículo cuando existe un síntoma de falla.

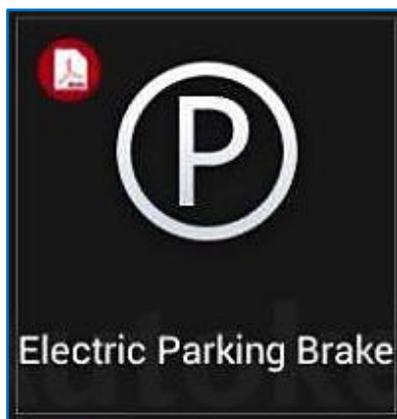
En la Tabla 3.5 se describe el origen y marca de los vehículos modernos que cuentan con la función BMS Reset.

**Tabla 3.5** Origen y marca de vehículos - Función BMS Reset

<b>AMERICA</b>	<b>ASIA</b>	<b>EUROPA</b>
FORD	MITSUBISHI	RENAULT
LINCOLN	NISSAN	PEUGEUT
	INFINITI	CITROEN

### **3.1.1.6 Freno eléctrico de parking**

El freno de estacionamiento eléctrico EPB es un dispositivo que sustituye al freno de estacionamiento accionado a través de una palanca. Esta función se logra por medio de la unidad de electrónica de control ECU.



**Figura 3.6** Señal - Freno eléctrico de parking

Al momento que surge una falla en el sistema de frenado, se activa un testigo luminoso en el panel de instrumentos esta señal se indica en la Figura 3.6, al momento de esta avería se recomienda que un técnico especializado se encargue de desbloquear el dispositivo.

Es necesario realizar una activación o una calibración, cuando se realiza un mantenimiento al sistema de frenos, se reemplace el juego de pastillas o se sustituya los discos de freno.

La adaptación eléctrica EPB se la realiza para la correcta retracción del pistón del caliper a su posición base en su función electromecánica antes de desmontar el juego de frenos. También se necesitar realizar una adaptación eléctrica después de haber armado el sistema de frenos, se debe hacer un ajuste y cierre de calipers dejando al sistema de EPB calibrado.

En la Tabla 3.6, se describe el origen y marca de los vehículos modernos que cuentan con la función freno eléctrico de parking.

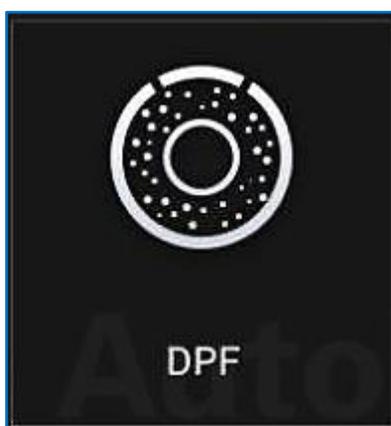
**Tabla 3.6** Origen y marca de vehículos - Función freno eléctrico de parking

AMERICA	ASIA	CHINA	EUROPA
GM	ACURA	QOROS	AUDI
CRYSLER	HONDA	BAOJUN	BENZ
FORD	HYUNDAI	BORGWARD	BENTLEY
LINCOLN	SCION	BESTURN	BUGATTI
JEEP	KIA	BEIJING	BMW
	TOYOTA	BYD	CITROEN
	NISSAN	CHANGAN	FERRARI
	SUBARU	CHERY	JAGUAR
	LEXUS	DONG FENG	LANDROVER
		SOUEAST	MASERATI
		GEELY	MINI
		LUXGEN	MAYNACH
		GREAT WALL	OPEL
		MG	PEUGEOT

### 3.1.1.7 DPF

El DPF es un filtro de partículas utilizado en vehículos diésel tiene la función de atrapar hollín y luego eliminarlo mediante un proceso de regeneración. Datos estadísticos indican que se puede eliminar hasta el 85 % del hollín y en varios casos hasta el 100 %.

Este proceso de limpieza es llevado a cabo por el ECM o Módulo de Control Electrónico de Motor y lo hace mediante sensores de temperatura interna y diferenciales de presión sobre el filtro.



**Figura 3.7** Señal - DPF

En la Figura 3.7, se observa el testigo luminoso o señal que el vehículo indica cuando existe un código de falla, esto sucede cuando el filtro de partículas diésel se encuentra con partículas de suciedad.

Para limpiar el DPF existen varios métodos, regeneración pasiva es transparente para el conductor ya que se realiza automáticamente cuando se hace viajes largos en autopistas, la única indicación que existe es una luz de advertencia de temperatura de escape en el tablero, indicando que la temperatura es más alta de lo habitual, regeneración activa ocurre cuando no existe suficiente calor en el escape, es auto accionada por el PCM, enviando un comando para elevar la temperatura del escape, adicionando pequeñas cantidades de combustible antes del DPF, y por último se tiene la regeneración estacionaria, esta última es la misma que la regeneración activa, pero se la lleva a cabo cuando el vehículo no está siendo conducido y es realizada con una herramienta de escaneo.

La regeneración estacionaria se la realiza cuando el vehículo tiene falta de potencia, el porcentaje de contenido de hollín es muy alto, saturación del filtro, por la anulación de una regeneración anterior o se inició una regeneración automática y fue interrumpida.

En la Tabla 3.7 se describe el origen y marca de los vehículos modernos que cuentan con la función de filtro de partículas diésel.

**Tabla 3.7** Origen y marca de vehículos - Función freno eléctrico de parking

<b>AMERICA</b>	<b>ASIA</b>	<b>EUROPA</b>
GM	MITSUBISHI	PEUGEOT
	INFINITI	CITROEN
	NISSAN	

### **3.1.2 RESETEO DE FUNCIONES EN VEHÍCULO SELECCIONADO**

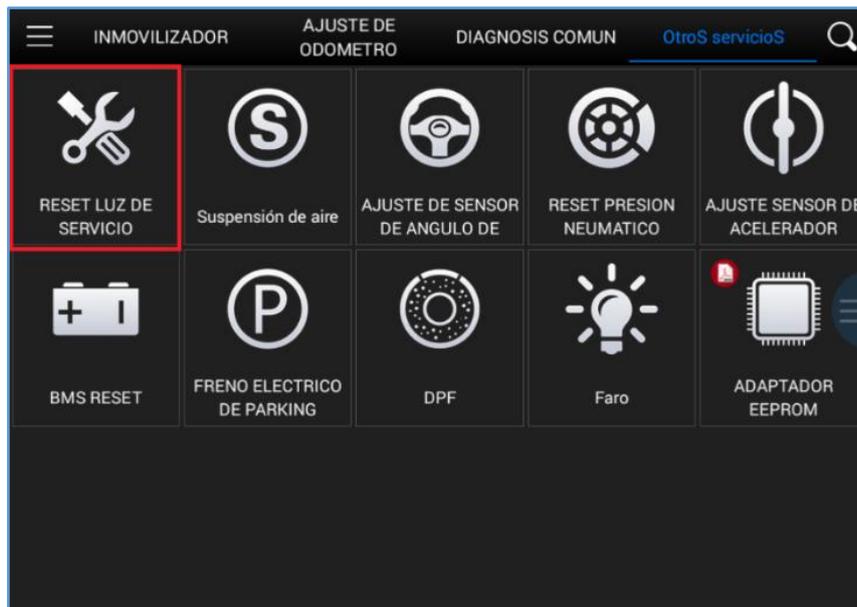
Mediante la conexión del equipo X100 PAD 2 PRO con el vehículo de la marca Peugeot modelo 307 año 2003, a través del conector OBD II como se observó en la figura 2.8 en el capítulo anterior. Se indagó los datos internos y se constató que cuenta con funciones de Reset de luz de servicio y ajuste sensor del acelerador.

Una vez identificadas las funciones del vehículo del caso práctico, se procedió al reseteo de cada función de una manera lógica para poder analizar su ubicación en los diferentes archivos de volcado de la unidad electrónica de control.

Realizado el proceso de reseteo se procederá al desmontaje de la unidad electrónica de control y se obtendrá el archivo de volcado después de este proceso, posteriormente se obtiene el archivo de volcado original para el análisis y comparación de códigos hexadecimales.

#### **3.1.2.1 Reseteo función Reset luz de servicio**

En este ítem se da a conocer el procedimiento que se siguió para un correcto reseteo de la función Reset luz de servicio, el objetivo de llevar a cabo este proceso es encontrar la ubicación exacta de códigos expresados en valores hexadecimales que cumplen esta función.



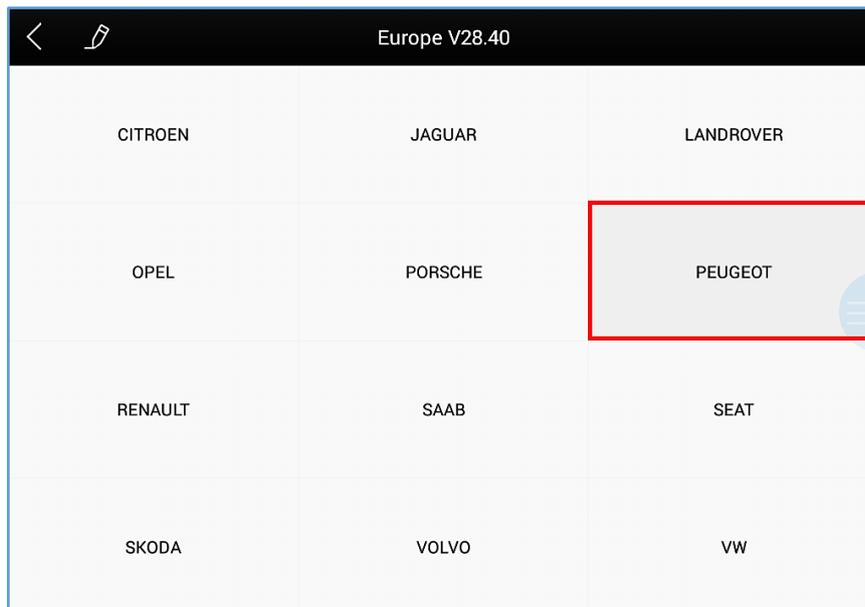
**Figura 3.8** Menú funciones especiales – Reset luz de servicio  
(Captura obtenida del equipo X100 PAD 2 PRO)

El reseteo de las funciones identificadas se realizó en el menú de otros servicios o funciones especiales del equipo, en este menú en primer lugar se seleccionó la función Reset de luz de servicio como se observa en la Figura 3.8.



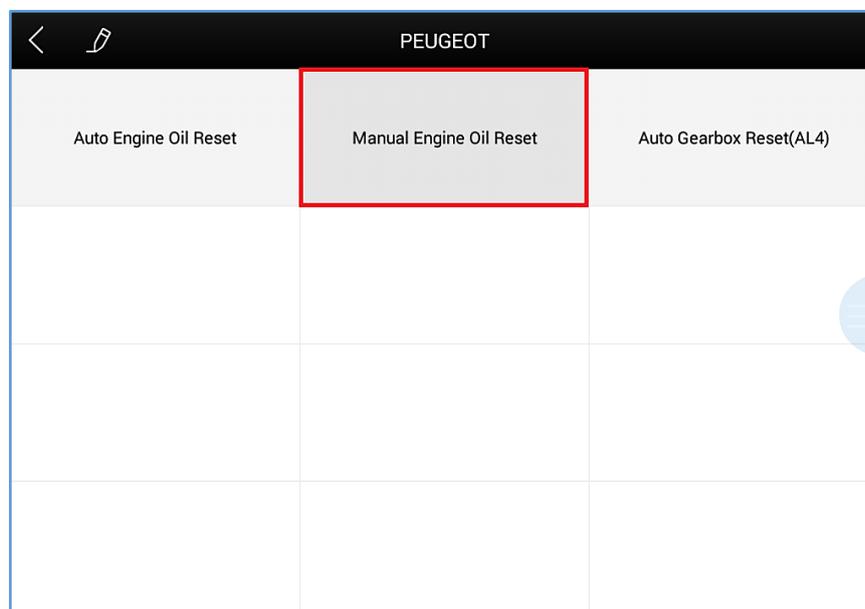
**Figura 3.9** Menú selección origen del vehículo - Reset luz de servicio  
(Captura obtenida del equipo X100 PAD 2 PRO)

A continuación, se abre una ventana con el menú para la selección del origen del vehículo, en este caso se escogió la opción EUROPE como se aprecia en la Figura 3.9, se toma esta opción debido a que el vehículo del caso práctico es de marca de automóviles francesa.



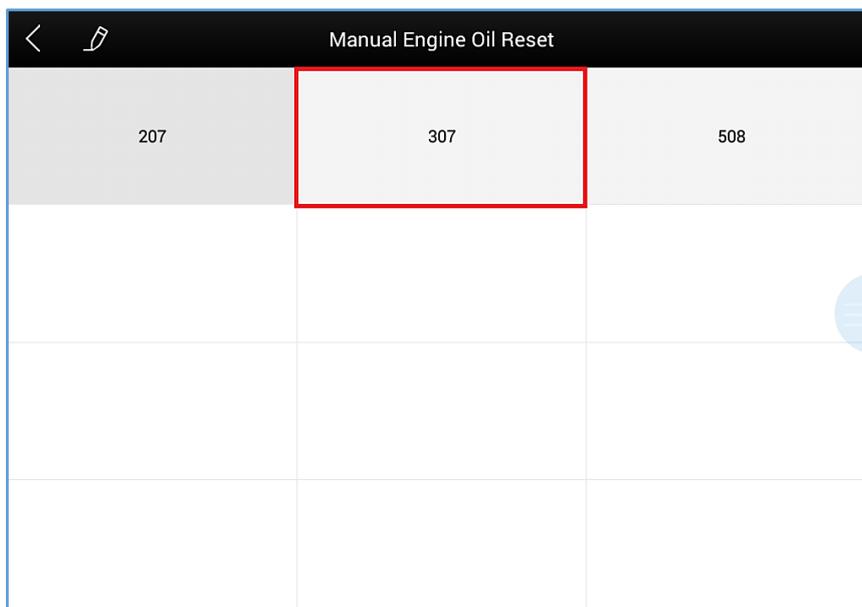
**Figura 3.10** Menú selección marca del vehículo - Reset luz de servicio  
(Captura obtenida del equipo X100 PAD 2 PRO)

Después de seleccionar el origen del vehículo se abre una nueva ventana en donde se encontró diferentes marcas europeas de automóviles, en este menú se escogió la marca Peugeot como se observa en la Figura 3.10.



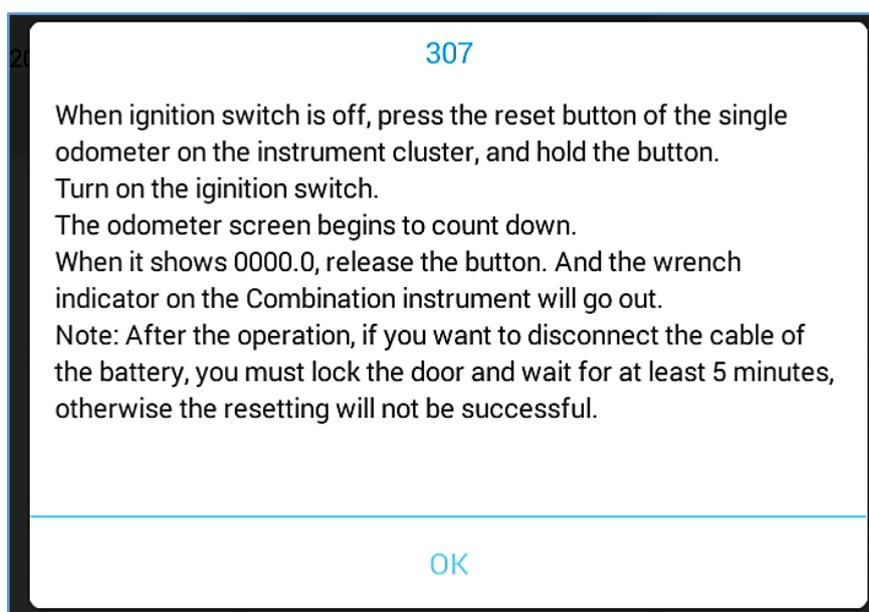
**Figura 3.11** Menú reinicio manual del aceite del motor  
(Captura obtenida del equipo X100 PAD 2 PRO)

Paso siguiente, después de seleccionar el vehículo de la marca Peugeot se abre la ventana para la selección del reseteo de la función relacionada al reseteo manual del aceite del motor esto se puede apreciar en la Figura 3.11.



**Figura 3.12** Menú selección modelo del vehículo - Reset luz de servicio  
(Captura obtenida del equipo X100 PAD 2 PRO)

Al momento de seleccionar el reseteo manual del aceite del motor, que tiene relación con la función Reset luz de servicio, se abre una nueva ventana para la selección del modelo del vehículo como se observa en la Figura 3.12, en esta ventana se escogió 307 debido a que son las características del vehículo seleccionado para el caso práctico.



**Figura 3.13** Instrucciones reseteo luz de servicio  
(Captura obtenida del equipo X100 PAD 2 PRO)

Para finalizar este proceso se siguió las instrucciones del equipo como se puede observar en la Figura 3.13, cuando el botón de encendido se encontró en la posición de estado apagado

se presionó el botón de reinicio del odómetro individual en el grupo de instrumentos y se mantuvo presionado por algunos segundos. A continuación, el botón de encendido del interruptor se presiona para tener en el estado de encendido y la pantalla del odómetro comenzó a contar hacia atrás, cuando indicó 0000.0, se dejó de presionar el botón y el indicador de llave inglesa del instrumento combinado se apagó.

Después de la operación, si se desea desconectar el cable de la batería, se debe bloquear la puerta y esperar al menos 5 minutos, de lo contrario el reinicio no será exitoso.

### 3.1.2.2 Reseteo función ajuste sensor de acelerador

En este literal se da a conocer el procedimiento que se siguió para un correcto reseteo de la función ajuste del sensor del acelerador conocido también como sensor TPS. La finalidad de llevar a cabo este proceso de igual manera que en el anterior ítem es encontrar la ubicación exacta de códigos expresados en valores hexadecimales que cumplen esta función.



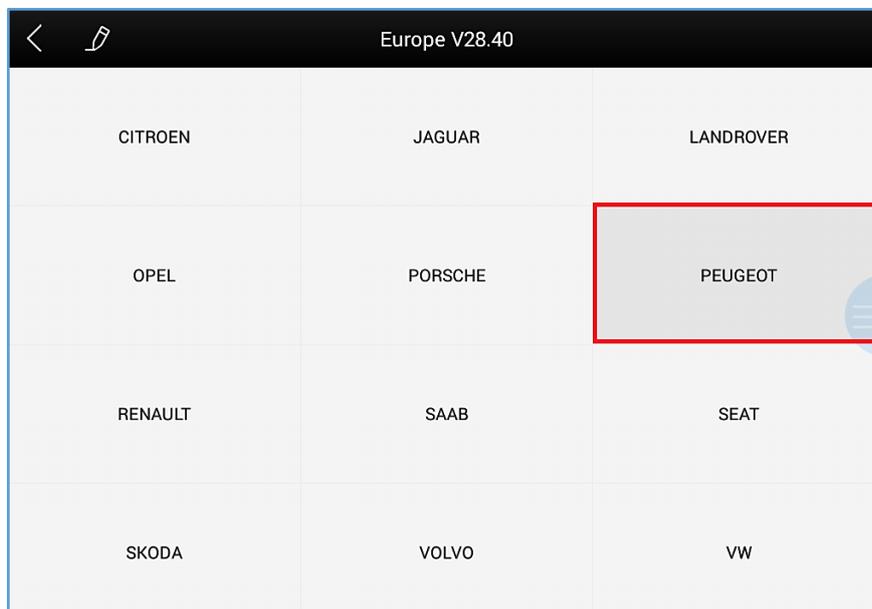
**Figura 3.14** Menú funciones especiales – Ajuste sensor de acelerador  
(Captura obtenida del equipo X100 PAD 2 PRO)

Para el inicio del reseteo de esta función se ingresa al menú de funciones especiales del equipo de programación, en este menú se escogió la opción ajuste de sensor del acelerador como se observa en la Figura 3.14.



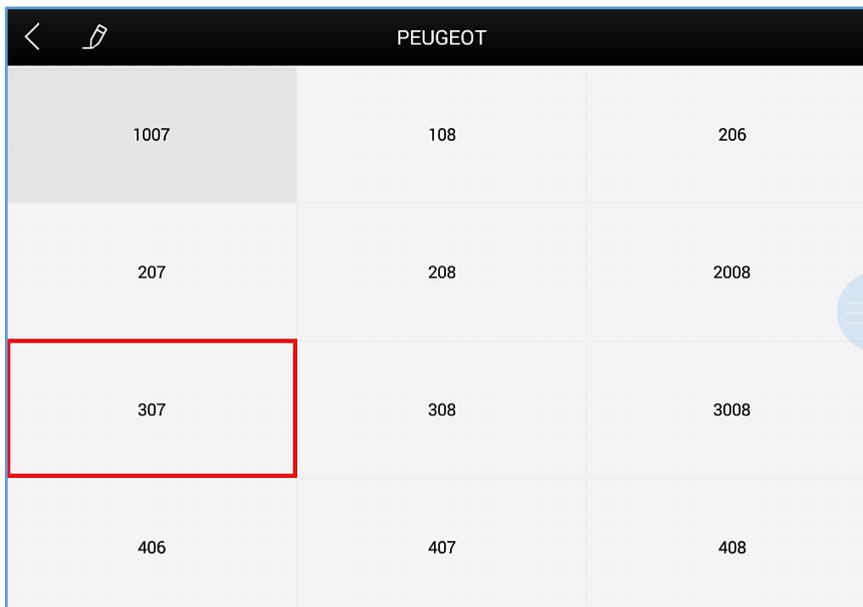
**Figura 3.15** Menú selección origen del vehículo – Ajuste sensor de acelerador  
(Captura obtenida del equipo X100 PAD 2 PRO)

Siguiente paso, después se abrió una ventana para la selección del origen del vehículo, como es el mismo vehículo que el caso anterior, se escogió la opción EUROPE como se puede observar en la Figura 3.15.



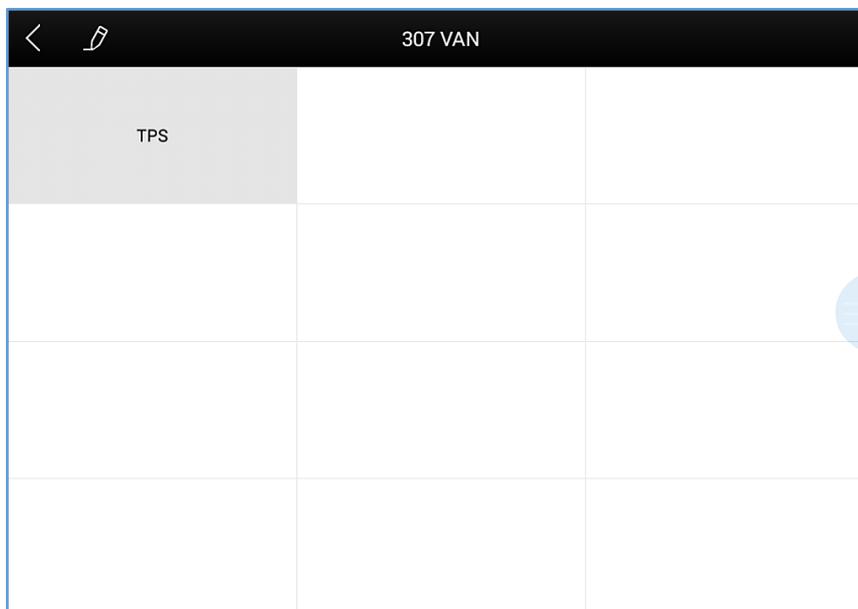
**Figura 3.16** Menú selección marca del vehículo - Ajuste sensor de acelerador  
(Captura obtenida del equipo X100 PAD 2 PRO)

Después de seleccionar el origen del vehículo se abre una nueva ventana en donde se encontró diferentes marcas europeas de automóviles de igual manera que en el apartado anterior, en este menú se escogió la marca Peugeot como se observa en la Figura 3.16.



**Figura 3.17** Menú selección marca del vehículo - Ajuste sensor de acelerador  
(Captura obtenida del equipo X100 PAD 2 PRO)

Después de seleccionar el origen y la marca Peugeot se abre una ventana nueva para la selección del modelo del vehículo, en este caso en particular se seleccionó el modelo 307 como se aprecia en la Figura 3.17.



**Figura 3.18** Selección sensor TPS  
(Captura obtenida del equipo X100 PAD 2 PRO)

Una vez elegidas todas las características técnicas del vehículo, se abre una ventana para la selección del resteo del sensor TPS como se puede ver en la Figura 3.18, con la conexión realizada en esta función se pudo verificar que la información es transmitida mediante el

protocolo de comunicación VAN que es común en los vehículos de la marca Peugeot. Para este proceso se tiene en cuenta que esta función es utilizada para cuatro puntos como:

- Reemplazar el sensor de oxígeno
- Reemplazar o limpiar el cuerpo del acelerador
- Reemplazar el sensor del pedal del acelerador
- Reemplace la ECU del motor

En este caso no se realizó ninguna de las opciones antes mencionadas, ya que el propósito del reseteo de esta función es obtener un archivo de volcado después de realizar este proceso para analizar los códigos hexadecimales inherentes a esta función. Para finalizar este proceso se siguió las instrucciones del equipo, las cuales se enlista a continuación:

- En primer lugar, se apagó el vehículo durante 60 segundos para reiniciar los valores encontrados en el interior del sistema.
- Después de esperar el tiempo establecido se encendió el vehículo y se esperó 15 segundos.
- Nuevamente se apagó el vehículo y se esperó 15 segundos.
- Luego, se encendió y se esperó 15 segundos más, se presionó ok en la pantalla del equipo y el reseteo se lo realizó de una manera exitosa.

Una vez que el reseteo esta completo se recomienda arrancar el motor y dejar que funcione en ralentí, apagar todo el equipo eléctrico, aire acondicionado, hasta que el ventilador de refrigeración se encienda. Cuando el conjunto del ventilador esté conectado, se espera hasta que se detenga, a continuación, se aumentó el régimen del motor a 5000 rpm, se soltó el pedal del acelerador y el motor regreso al estado de ralentí.

### **3.2 OBTENCIÓN ARCHIVOS DE VOLCADO**

Como ya se ha descrito en el capítulo anterior, los archivos de volcado de módulos automotrices guardan toda la información del vehículo expresada en códigos hexadecimales, por la tanto, estos códigos se descifran en el DUMP de cada archivo para identificar la ubicación de cada función del auto.

En este punto se describe el procedimiento que se siguió en cada módulo de control para obtener el archivo de volcado de manera individual. En primer lugar, se describe los pasos a seguir para conseguir el archivo de la ECU, después se da a conocer el proceso que se persiguió para el archivo de volcado de la BSI y finalmente se describe la obtención del archivo del panel de instrumentos electrónico.



**Figura 3.19** Vehículo Peugeot 307 año 2003

En la Figura 3.19 se puede observar el vehículo de la marca Peugeot modelo 307, vehículo en el cual se realizaron todos los procesos descritos anteriormente.

### 3.2.1 MODULO DE CONTROL ELECTRÓNICO ECU

En este apartado, en primer lugar, se procedió a identificar las características generales de la ECU del vehículo. Los datos de la unidad electrónica del motor se los encontró en la carcasa de la misma y se las presenta en la Tabla 3.8.

**Tabla 3.8** Características generales ECU

<b>Marca</b>	BOSCH
<b>Versión</b>	ME 7.4.4
<b>Familia o Protocolo</b>	397

Estos datos específicos también se identificaron en el software del módulo K-TAG para acceder a las instrucciones de conexión entre el equipo y la ECU.



**Figura 3.20** Centralita ECU

Con el previo desmontaje de la ECU, y retiro de los tornillos de fijación de la carcasa de la centralita, se colocó el circuito impreso en una superficie limpia, libre de líquidos y sustancias que puedan causar daños en su estructura como se puede apreciar en la Figura 3.20, misma superficie fue empleada para el desarrollo del trabajo práctico.

La lectura de los registros internos con KTAG requiere de una conexión a los pines de los conectores de la ECU y como se mencionó en el capítulo II y se pudo apreciar en la figura 2.11, es necesario realizar una conexión de un cable soldado al circuito interno de la misma.

### **3.2.1.1 Selección de ECU en el módulo K-TAG**

En este ítem se realizó la conexión del equipo K-tag a una fuente de alimentación que debe contener un regulador de voltaje para evitar errores en la lectura del archivo de volcado de la ECU en caso que exista una pérdida de energía.

Se realizó el enlace del K-TAG a la computadora portátil mediante un cable USB para ingresar al menú principal del módulo.

Al momento de abrir el software del equipo se realizó la configuración del idioma que por recomendación del fabricante se seleccionó el idioma inglés debido a que en esta configuración se encuentran detallados los procesos de conexión de la ECU.



**Figura 3.21** Menú K-TAG - Opción protocolos  
(Captura obtenida del equipo K-TAG)

En el menú principal del módulo K-TAG para la selección de la ECU se escogió la opción protocolos como se indica en la Figura 3.21, debido a que en la opción de vehículos no se encontró las especificaciones técnicas de la misma.

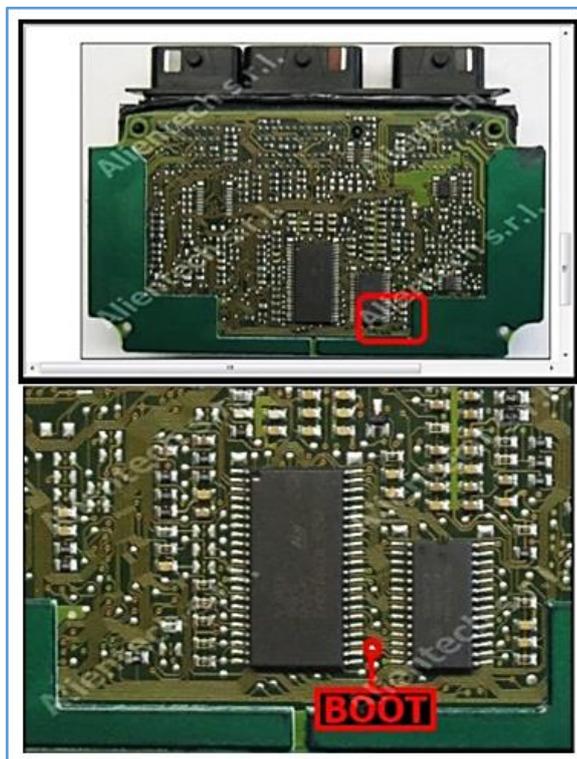
KessV2	ECU	Family
BOOTLOADER TRICORE	BOSCH M7.9.7 KIA	597
BOOTLOADER ST 10Fxxx	BOSCH M7.9.7 LADA	376
OBDII	BOSCH M798 KIA	167
<b>K-TAG</b>	BOSCH ME2.8 MERCEDES	622
BDM MPC5XX	BOSCH ME7.0 AM25F400 VOLVO	624
BDM 68K	BOSCH ME7.0 AM25F800 VOLVO	325
BDM HC12	BOSCH ME7.01 VOLVO	626
BDM HC16	BOSCH ME7.1.1 BENTLEY M85	613
BOOTLOADER MITSUBISHI	BOSCH ME7.1.1 VAG	539
BOOTLOADER ST 10Fxxx	BOSCH ME7.1.1 VAG HW I	50
BOOTLOADER TRICORE	BOSCH ME7.1.1 VAG HW II	51
BOOTLOADER M32R	BOSCH ME7.1.1 VAG HW III	52
JTAG MPC5XX	BOSCH ME7.2 BMW	604
JTAG RENESAS	BOSCH ME7.2 PORSCHE	240
NBD NEC 76F00XX	BOSCH ME7.3.2 FERRARI/MASERATI	42
	BOSCH ME7.4.4 POLARIS	57
	<b>BOSCH ME7.4.4 PSA</b>	<b>397</b>
	BOSCH ME7.4.5 PSA	25

**Figura 3.22** Menú protocolos - Selección de ECU  
(Captura obtenida del equipo K-TAG)

La ECU con la que se trabajó es una BOSCH 7.4.4 PSA perteneciente al grupo Citroën, Peugeot y Renault de la familia 397 del grupo K-TAG BOOTLOADER ST 10F, esto se puede apreciar de una manera más detallada en la Figura 3.22, en esta pantalla se presionó OK y se avanzó a las instrucciones de conexión.

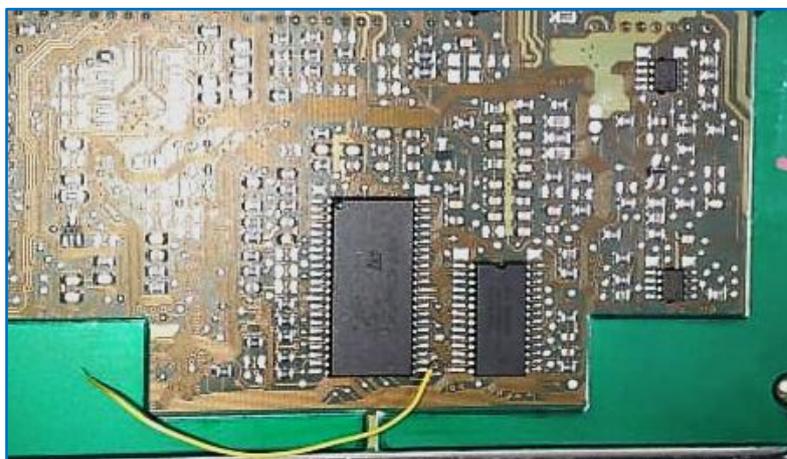
### 3.2.1.2 Conexión ECU y Módulo K-TAG

Luego de seleccionar la ECU en la pantalla aparecen varias recomendaciones y los pasos a seguir para la conexión del módulo K-TAG a la placa de la centralita.



**Figura 3.23** Puntos de conexión

En este punto se eligió la opción “Here” en la opción “connect to the ECU”, y de manera simultánea apareció imágenes del circuito interno de la ECU indicando los puntos de conexión como se observa en la Figura 3.23.



**Figura 3.24** Conexión punto BOOT ECU

Se realizo la conexión directa a la centralita de la ECU, donde fue necesario soldar un cable en el punto BOOT como se observa en la Figura 3.24, esto se lo realizo con la ayuda de estaño y un caudín tipo lápiz tomando en cuenta las precauciones necesarias para no causar daños en el circuito impreso.



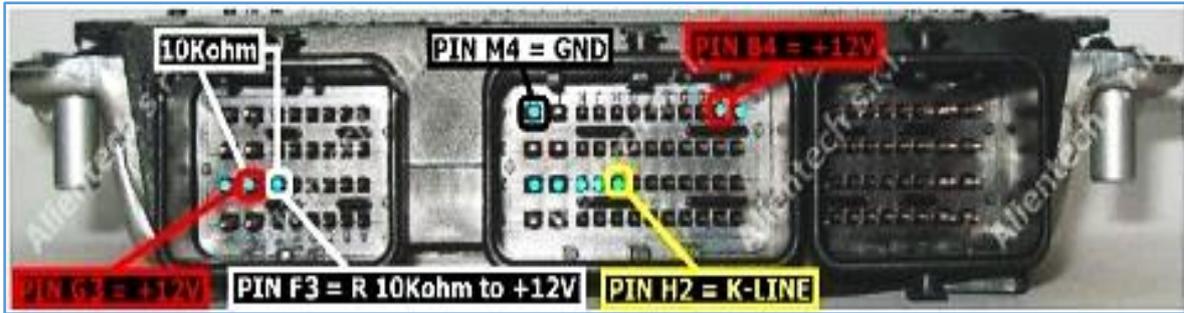
**Figura 3.25** Cable 14P600KT02

Para la conexión en el puerto del módulo K-TAG y los pines de los conectores de la ECU fue necesario una resistencia de 10K Ohm y un cable 14P600KT02 como se lo indica en la Figura 3.25, el cual cuenta con todos los cables con una configuración de códigos colores y el socket para conectarse en el puerto del equipo.

En la Tabla 3.9, se detalla la configuración de códigos de colores del cable 14P600KT02 que se utilizó para la conexión a los pines de los conectores de la ECU y al circuito impreso de la centralita. Se puede tomar con mayor relevancia a los cables de color, rojo, negro, amarillo y plomo.

**Tabla 3.9** Configuración de códigos colores cable 14P600KT02

COLOR	DESCRIPCIÓN
Red	Power +12V
Orange	Key switched Power +12V
Black	Ground GND
Yellow	K-Line
White	CAN-high
Green	CAN-Low
Purple	VPP
Gray	BOOT
Brown	Reset
Blue	CNF 1



**Figura 3.26** Puntos de conexión cable 14P600KT02

Después de identificar la conexión en el punto BOOT, el programa muestra las indicaciones para conectar la resistencia de 10K ohm y el cable 14P600KT02 a los diferentes pines de la ECU. En la Figura 3.26 se puede observar las conexiones y los pines utilizados.

En la Tabla 3.10 se detalla de manera resumida las conexiones necesarias de todos los cables utilizados.

**Tabla 3.10** Resumen conexiones cable 14P600KT02

PIN	COLOR	DESCRIPCIÓN
B4-G3	Red	+12V
M4	Black	Ground
H2	Yellow	K-line
Cable soldado	Gray	BOOT
G3-F3		Resistencia soldada



**Figura 3.27** Conexión K-TAG con la centralita

En la Figura 3.27, se puede apreciar la conexión realizada del módulo K-TAG con la placa de la centralita.

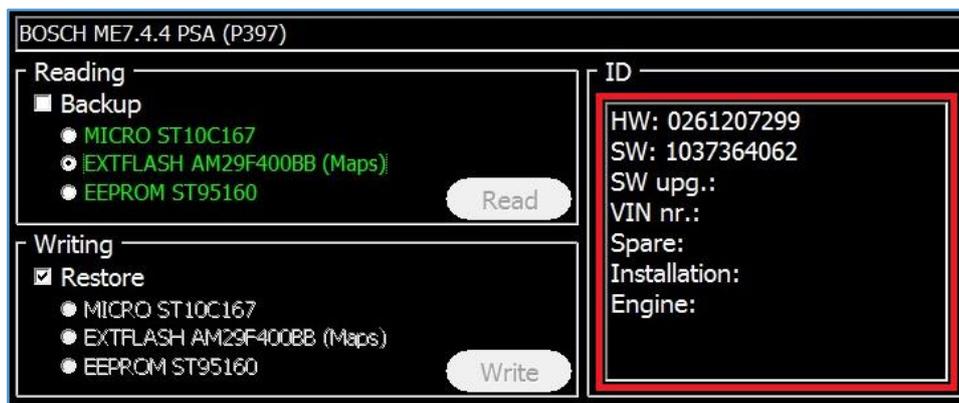
### 3.2.1.3 Lectura ECU

Realizadas las conexiones necesarias entre el módulo K-TAG y la placa de la centralita de la ECU, se procedió con la lectura y respaldos de los archivos de la misma. Al final de las indicaciones del ítem anterior se presionó OK y el software del equipo se trasladó a la pantalla de lectura y escritura donde se reconoció el tipo de ECU y las diferentes memorias que esta dispone.



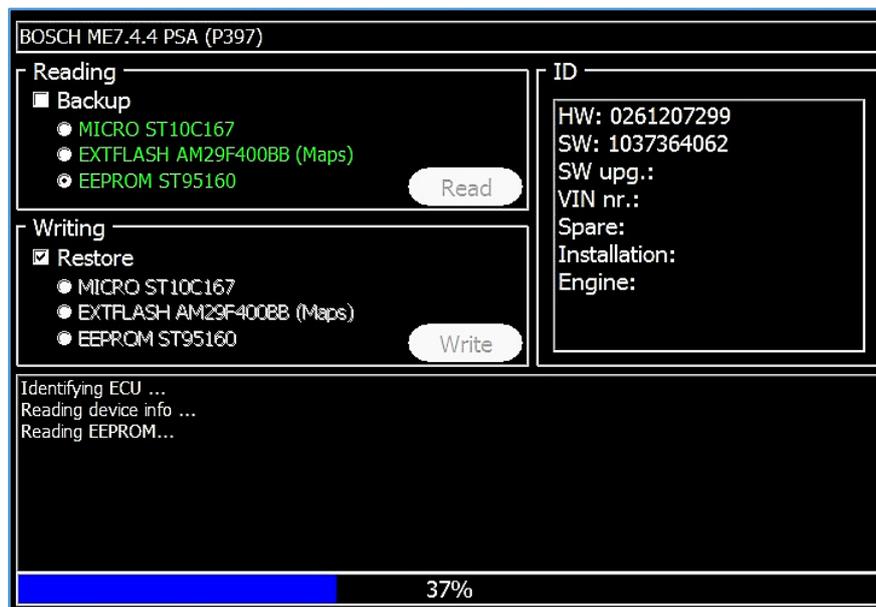
**Figura 3.28** Opciones de Lectura y escritura  
(Captura obtenida del equipo K-TAG)

En la Figura 3.28, se observa que la centralita cuenta con un microcontrolador ST10C167, una memoria flash AM29F400BB donde se localizan los mapas, y una memoria EEPROM ST95160. En el siguiente paso se realizó la lectura para la extracción de los archivos; eligiendo la opción “READ” el módulo identifica el software y hardware con los que cuenta la ECU BOSCH ME 7.4.4 PSA.



**Figura 3.29** Hardware y Software ECU  
(Captura obtenida del equipo K-TAG)

La Figura 3.29, muestra que la centralita cuenta con un hardware 0261207299 y un software 1037364062 los cuales coinciden con la etiqueta de la ECU.



**Figura 3.30** Proceso de lectura  
(Captura obtenida del equipo K-TAG)

Como se indica en la Figura 3.30, la lectura de las diferentes memorias se la realizo de manera separada y unificada en la opción BACKUP.

En total se obtuvo 4 archivos: microcontrolador, memoria flash, EEPROM y el archivo unificado.

Backup 1	267 328	267 328	2020-06-23 12:21
Lec 1 EEPROM ST95160	2 048	711	2020-06-23 12:06
Lec 1 Flash AM29F400B...	524 288	209 556	2020-06-23 12:04
Lec 1 Micro ST10C167	32 768	15 096	2020-06-23 12:07
Lec 2 EEprom ST95160	2 048	711	2020-06-23 12:15
Lec 2 Flash AM29F400BB	524 288	210 085	2020-06-23 12:14
Lec 2 Micro ST10C167	32 768	15 096	2020-06-23 12:16

**Figura 3.31** Archivos extraídos ECU

Al finalizar la lectura de manera directa se dirigió al explorador de Windows para determinar la dirección y la carpeta en la cual se va a guardar los archivos extraídos como se puede ver en la Figura 3.31, por recomendaciones del fabricante y seguridad se realizó 2 lecturas para la extracción del archivo de volcado original.

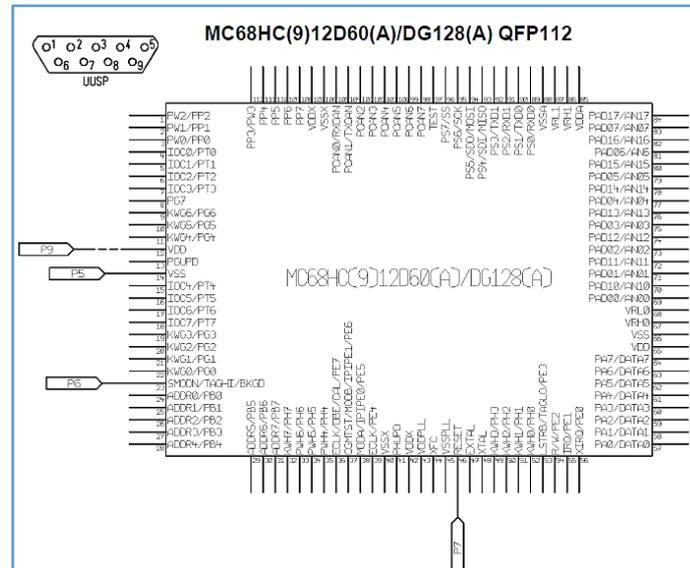
### 3.2.2 MÓDULO DE CONTROL DE LA CARROCERÍA BCM

En este punto, una vez identificado la ubicación del módulo del control de la carrocería se procedió al desmontaje del mismo para la obtención de la centralita electrónica y analizar el tipo de conexión para acceder al archivo de volcado. Como se explicó en el capítulo II.

En el vehículo Peugeot 307 se encontró un módulo de control BSI (Body System Interface) o también conocido como caja de servicios inteligentes, este sistema viene a ser lo mismo que la BCM (Body Control Module). BSI es un sistema multiplexado de los vehículos del grupo PSA (Peugeot y Citroën).

#### 3.2.2.1 Identificación microcontrolador BSI

El modelo de microcontrolador que tiene la BSI VALEO es un MC68HC912DG128, siendo el microcontrolador muy diminuto para ser desoldado o realizar una conexión de lectura directa, se procede a identificar los puntos de soldadura para este modelo.



**Figura 3.32** Microcontrolador MC68HC912DG128

Mediante el manual que proporciona UPA se puede acceder a las conexiones pertinentes para poder leer el microcontrolador y extraer el archivo de volcado, en la Figura 3.32, se puede apreciar las indicaciones de conexión mediante conector USSP, estas son VDD al Pin 9, VSS a Pin 5, BKGD a Pin 6 y RESET a Pin 7.

VDD es voltaje de alimentación positiva aplicada al drenador de los transistores, VSS es voltaje de alimentación positivo aplicado al surtidor de los transistores, BKGD pin de reprogramación, RESET señal de reinicio.

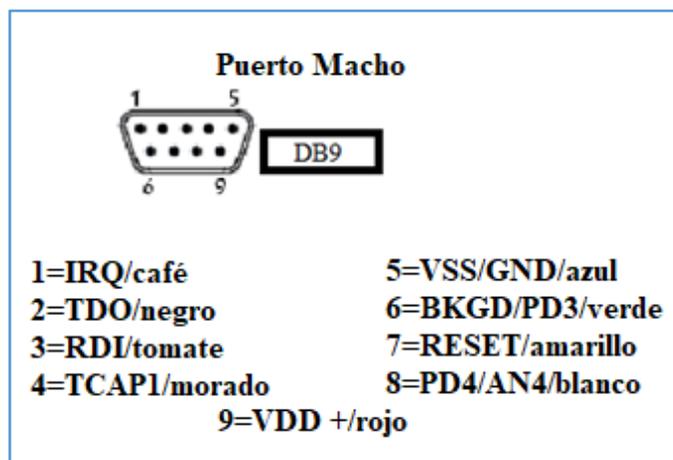


**Figura 3.33** Centralita electrónica BSI

En la Figura 3.33, se puede observar la centralita electrónica de la BSI VALEO ya desmontada del vehículo y sacada de la protección, por la dificultad de acceder al circuito impreso y a su vez al microcontrolador se optó por identificar mediante un multímetro las líneas de conexión a VDD, VSS, BKGD y RESET.

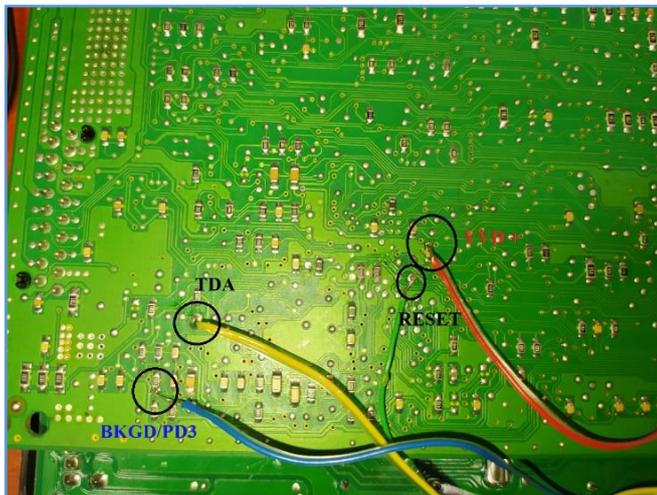
### 3.2.2.2 Conexión modulo UPA y microcontrolador

Para poder acceder al archivo de volcado de la BSI, se utiliza el software UPA, para el cual necesita conexiones especiales. Como primer paso se procesa a identificar la conexión del cable puerto DB9 al circuito impreso de la BSI.



**Figura 3.34** Identificación de color de pines - Puerto BD9

La conexión necesita de TDO, BKGD/PD3, Reset y VDD. Cada número de pin del puerto DB9 con su respectivo color se puede ver en la Figura 3.34.

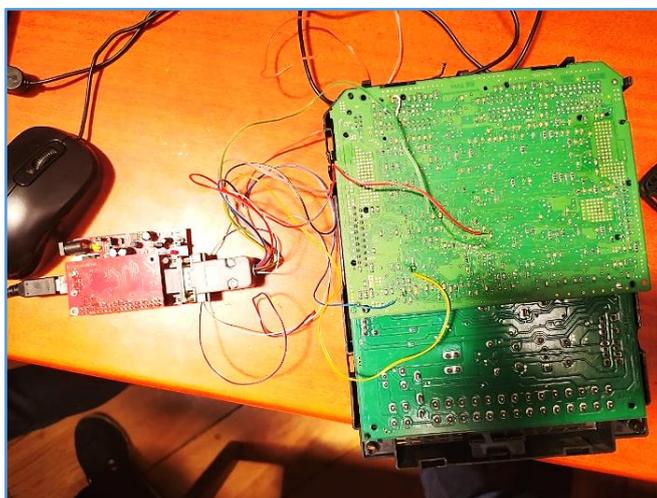


**Figura 3.35** Posición de soldadura de los cables

En la Figura 3.35, se puede apreciar la ubicación y color de cada cable que debe de ir en determinado para poder acceder a la información del microcontrolador.

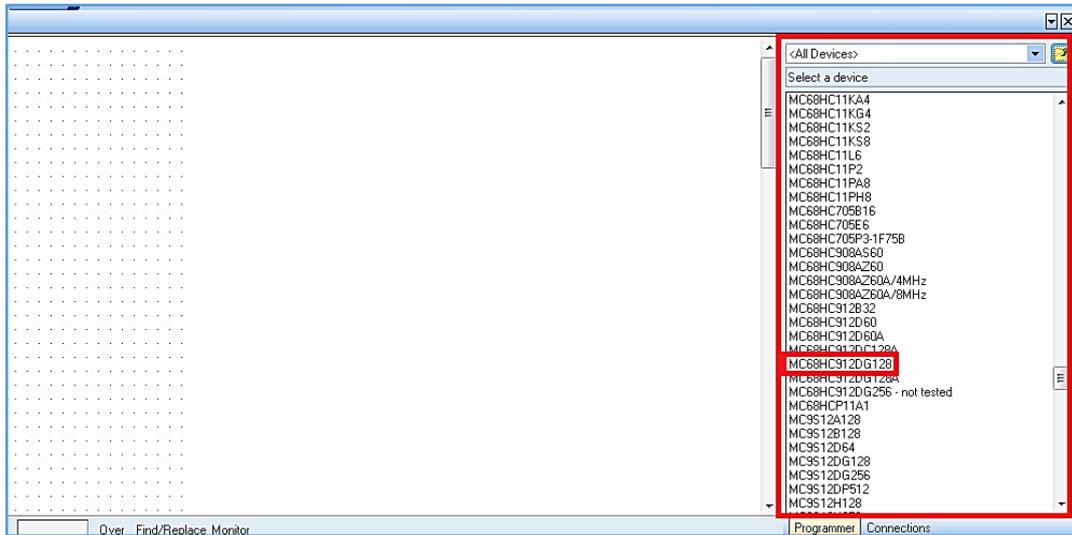
### 3.2.2.3 Lectura Archivo de volcado BSI

Una vez que se tiene soldado los cables en la parte posterior del circuito impreso, se procedió a conectar el puerto DB9 al módulo UPA, posteriormente se conectó a la laptop mediante cables USB. Una vez realizado las conexiones necesarias se inició el software UPA esperar que cargue para la lectura del módulo BSI.



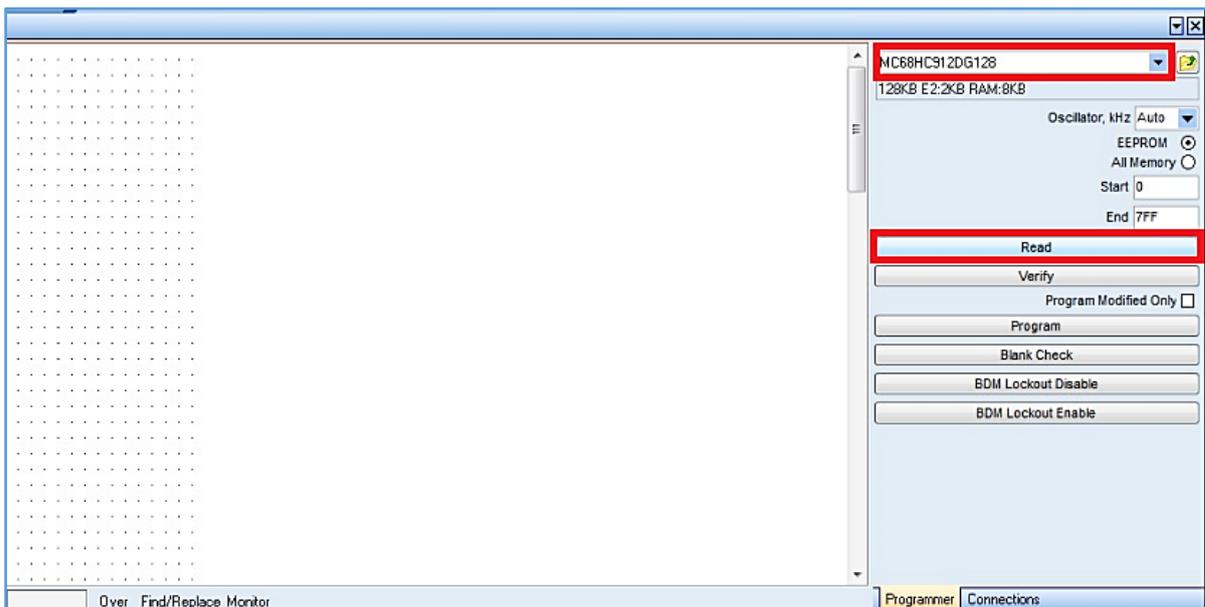
**Figura 3.36** Conexión UPA y BSI

En la Figura 3.36, se puede observar las conexiones pertinentes entre la BSI y UPA, antes de conectar a la computadora se debe de verificar que la soldadura este correcta en cada punto.



**Figura 3.37** Selección tipo de microcontrolador  
(Captura obtenida del equipo UPA-USB)

Los pasos a seguir para la lectura y respaldos del archivo de volcado de la BSI son, en el software dirigirse al apartado All Devices y posteriormente se despliegan un sinnúmero de opciones a elegir como se observa en la Figura 3.37, en este caso se seleccionó el microcontrolador MC68HC912D6128.



**Figura 3.38** Lectura microcontrolador  
(Captura obtenida del equipo UPA-USB)

Una vez seleccionado la opción MC68HC912D6128 se procede a seleccionar la opción READ, como se observa en la Figura 3.38, después de escoger esta opción se procede a la lectura del microcontrolador.

Nombre	Fecha de modificación	Tipo	Tamaño
 Lectura 1.bin	7/7/2020 16:21	Archivo BIN	2 KB
 Lectura 2.bin	7/7/2020 16:21	Archivo BIN	2 KB

**Figura 3.39** Archivos extraídos BSI

Luego de haber realizado la selección del microcontrolador en la opción “ALL DEVICES” y dar lectura, se dirige a la pestaña file y seleccionar guardar archivo como, donde apareció la dirección y la carpeta en la cual se guardó los archivos extraídos.

Por recomendaciones del fabricante y seguridad se realizó 2 lecturas para la extracción del archivo de volcado original como se muestra en la Figura 3.39.

Una vez obtenidos los datos necesarios se debe de desconectar los cables, desoldar los cables del circuito impreso de la BSI para que sea armada y colocada otra vez en el vehículo.

### 3.2.3 PANEL DE INSTRUMENTOS ELECTRÓNICO

En este punto, una vez identificado la ubicación del módulo del panel de instrumentos se procedió al desmontaje del mismo para la obtención de la centralita electrónica, como se explicó en el capítulo II.

Antes de retirar la carcasa que protege la centralita se observó la posición de las agujas de todo el conjunto que conforma el panel de instrumentos para no tener alteraciones de funcionamiento en los diferentes indicadores.

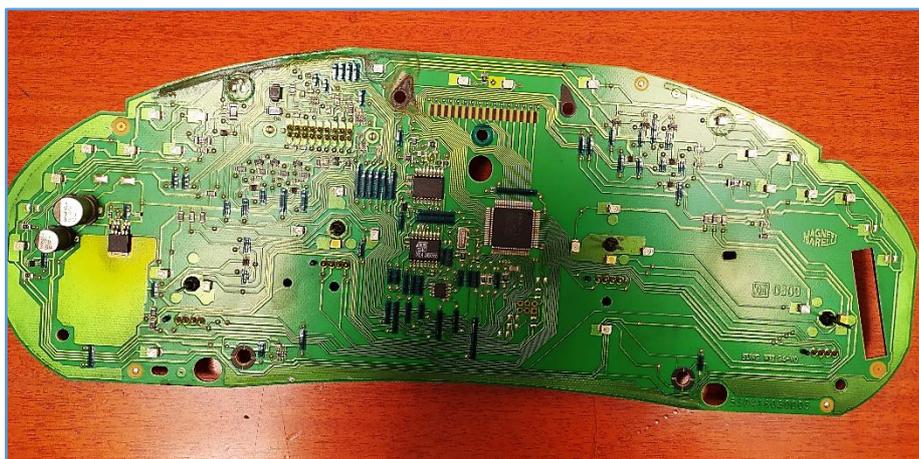


**Figura 3.40** Posición de agujas panel de instrumentos

La posición de las agujas de los diferentes indicadores del tablero de instrumentos se puede observar detalladamente en la Figura 3.40.

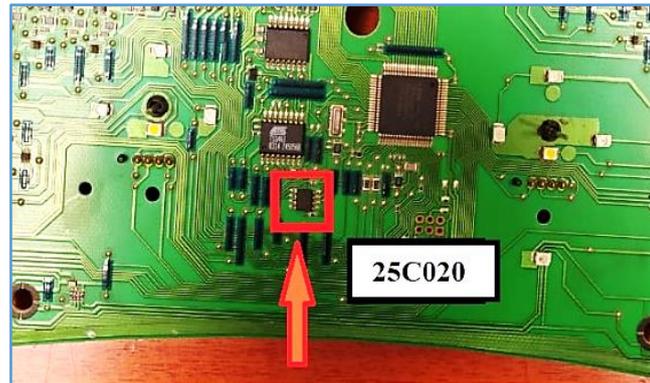
### 3.2.3.1 Identificación memoria programable

En el momento en que se retira las agujas de los indicadores del tablero, el siguiente paso que se realizó es sacar los tornillos de fijación de la carcasa que protege el circuito impreso del panel de instrumentos para identificar el tipo de memoria programable que contiene



**Figura 3.41** Centralita panel de instrumentos

La placa de la centralita de igual manera que el de la ECU se colocó en una superficie limpia para evitar daños en su entorno, esto se puede apreciar en la Figura 3.41.



**Figura 3.42** Memoria EEPROM 25C020

El paso siguiente fue identificar el tipo memoria en el circuito impreso, para este caso práctico se determinó que el módulo del panel de instrumentos cuenta con una memoria EEPROM 25C020 tipo SOIC eléctricamente borrable con 8 pines de montaje superficial como se indica en la Figura 3.42.

### 3.2.3.2 Extracción de la memoria EEPROM 25C020

Para realizar la lectura del archivo de volcado del módulo del panel de instrumentos fue necesario desoldar la memoria EEPROM del circuito impreso. La extracción de la memoria se la efectuó con la implementación de equipos de soldadura electrónica como cautín, estaño, pasta de soldar y una pinza.



**Figura 3.43** Aplicación de estaño en centralita del panel de instrumentos

Para el proceso de la extracción de la memoria EEPROM, en primer lugar, se aplicó estaño en los pines al contorno de la memoria como se aprecia en la Figura 3.43, este proceso se lo realizo con el fin de producir un cortocircuito en la memoria y poder desmontarle de la placa de la centralita de manera segura.

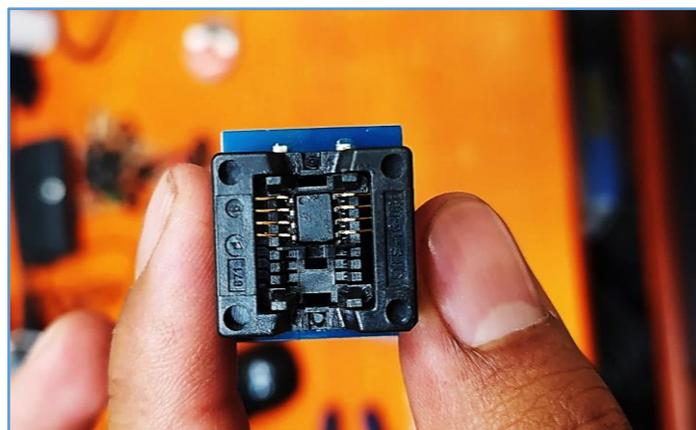


**Figura 3.44** Extracción de memoria EEPROM del panel de instrumentos

Una vez aplicado estaño en los pines se procede a la extracción de la memoria EEPROM 25C020 de la placa de la centralita del panel de instrumentos con la ayuda de un caudín precalentado y una pinza como se muestra en la Figura 3.44. Para este proceso es importante observar la posición en la que se encuentra ubicada la memoria para determinar la muesca de la memoria y el pin número 1. Por otro lado, es recomendable no calentar más de 3 a 5 segundos la soldadura, ya que podría dañar el circuito integrado.

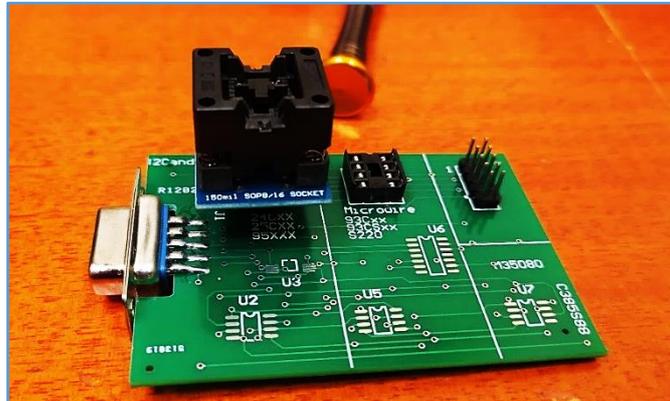
### 3.2.3.3 Conexión memoria EEPROM y modulo UPA

En este apartado se procedió al montaje de la memoria EEPROM 25C020 al adaptador UPA, para esto fue necesario la implementación de un segundo adaptador de tipo SOIC a DIP de un tamaño más pequeño.



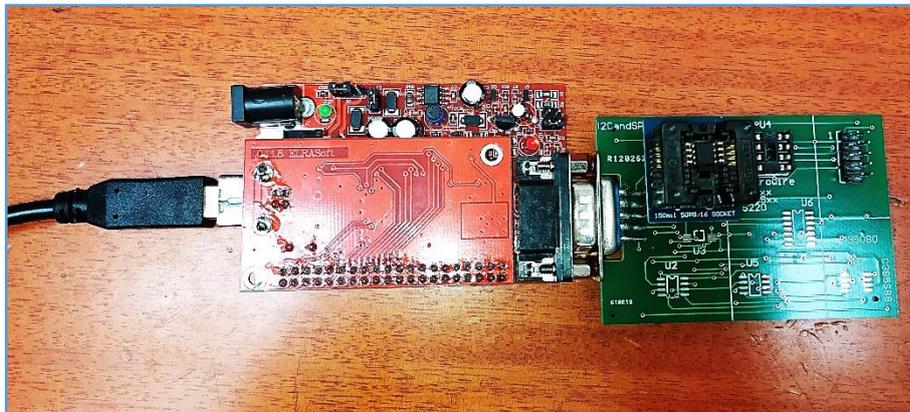
**Figura 3.45** Adaptador SOIC a DIP

En la Figura 3.45, se muestra el adaptador de tipo SOIC a DIP, se ocupó este adaptador debido a que la memoria tipo DIP es más robusta y más fácil de leer que la memoria tipo SOIC.



**Figura 3.46** Montaje adaptador SOIC a DIP al adaptador UPA

Paso siguiente se procedió a montar el adaptador tipo SOIC a DIP al adaptador UPA como se puede apreciar en la Figura 3.46.

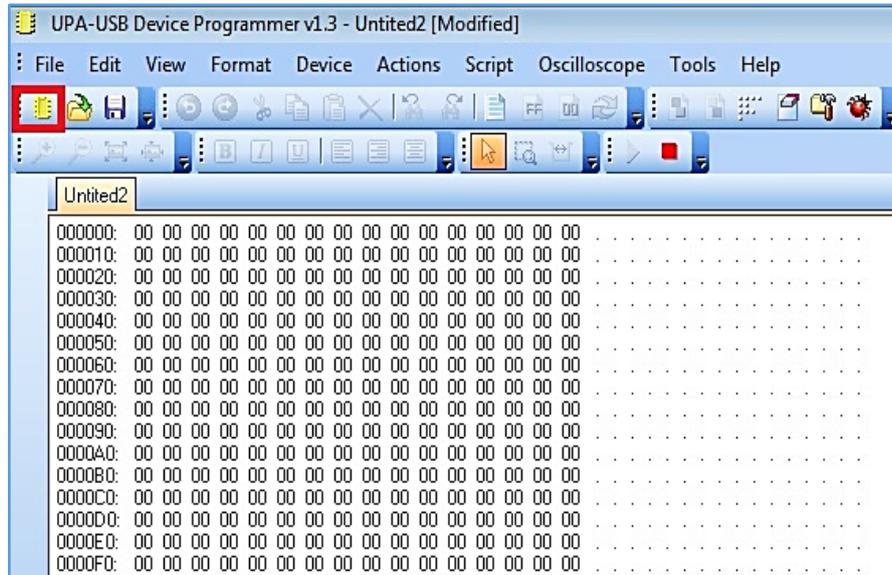


**Figura 3.47** Conexión memoria EEPROM y programador UPA

Finalmente, en este apartado se realizó el montaje de la memoria 25C020 al adaptador tipo SOIC a DIP y conectar el programador UPA mediante un cable tipo USB al adaptador y a la computadora portátil como se puede apreciar en la Figura 3.47.

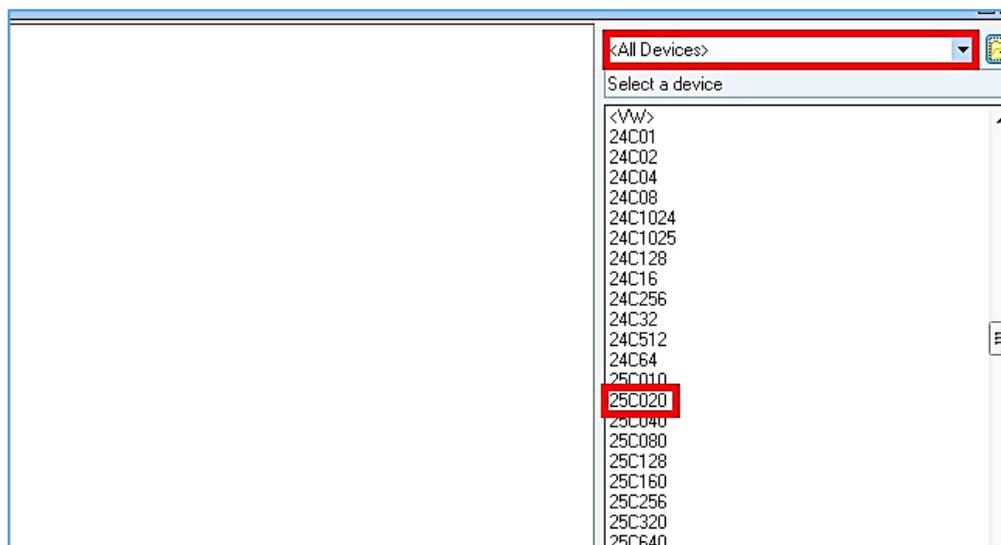
#### 3.2.3.4 Lectura memoria EEPROM 25C020

Ejecutadas las conexiones necesarias entre la memoria EEPROM 95020, el adaptador y el módulo UPA, se procedió con la lectura y respaldos del archivo de volcado original del módulo del panel de instrumentos electrónico.



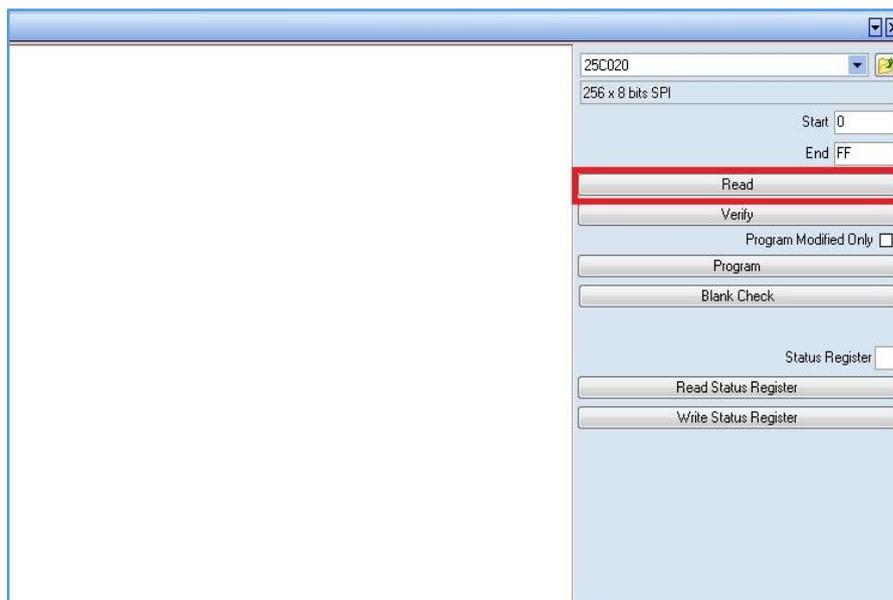
**Figura 3.48** Menú UPA opción abrir hoja nueva  
(Captura obtenida del equipo UPA-USB)

El primer paso para la lectura del archivo es abrir el software del módulo UPA en la computadora portátil, esperar que carguen los datos, y se abrirá el menú principal del módulo UPA en donde se escoge la opción abrir una hoja de trabajo nueva en el icono seleccionado como se indica en la Figura 3.48.



**Figura 3.49** Selección memoria EEPROM 25C020  
(Captura obtenida del equipo UPA-USB)

En este caso práctico se escogió la memoria 25C020, a continuación, se abre una ventana con la opción “READ” como se indica en la Figura 3.49, en este punto se da lectura al archivo de volcado original del módulo automotriz.



**Figura 3.50** Menú UPA selección READ  
(Captura obtenida del equipo UPA-USB)

En el momento que se seleccionó abrir una hoja de trabajo nueva en la pantalla aparece la opción “ALL DEVICES” en donde se escogió el tipo de memoria con el que se desea trabajar. Esto se puede observar en la Figura 3.50.

Luego de haber realizado la selección de la memoria en la opción “ALL DEVICES” y dar lectura de la memoria EEPROM 250020, se dirige a la pestaña file y seleccionar guardar archivo como, donde apareció la dirección y la carpeta en la cual se guardó los archivos extraídos.

Nombre	Fecha de modificación	Tipo	Tamaño
 Lectura 1.bin	2/7/2020 11:16	Archivo BIN	1 KB
 Lectura 2.bin	2/7/2020 11:16	Archivo BIN	1 KB

**Figura 3.51** Archivos extraídos panel de instrumentos

Por recomendaciones del fabricante y seguridad se realizó 2 lecturas para la extracción del archivo de volcado original como se muestra en la Figura 3.51.

### 3.3 OBTENCIÓN PIN CODE Y TRANSPONDER DE LLAVES

En este apartado se da a conocer en primer lugar, los pasos a seguir para la obtención del PIN CODE, necesario para verificar que la lectura del archivo de volcado de la unidad electrónica de control ECU es correcta.

Además, se detalla el procedimiento de la obtención del código del chip de transponder de llaves que es de gran relevancia para la programación o clonación de una nueva llave, proceso que se realizó como método de diagnóstico para determinar que valores hexadecimales cambian y determinar la ubicación en el archivo de la unidad electrónica de control de la carrocería BSI.

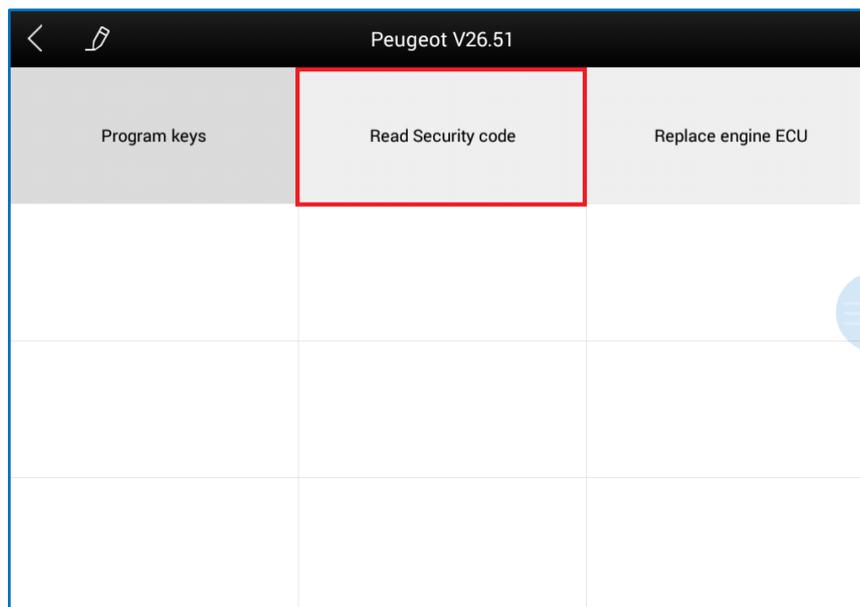
#### 3.3.1 OBTENCIÓN CODIGO DE SEGURIDAD PIN CODE

Para iniciar este proceso se conectó el equipo X100 PAD 2 PRO mediante el puerto OBD II al vehículo, esta conexión se puede visualizar en el capítulo II en la figura 2.8, esto se realizó en el menú inmovilizador del equipo.



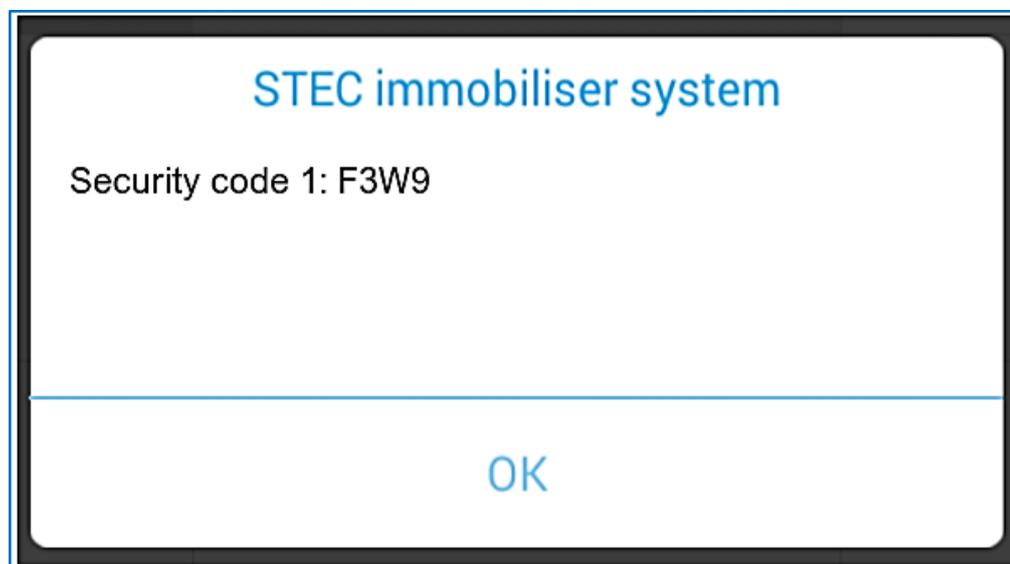
**Figura 3.52** Menú inmovilizador X100 PAD 2 PRO  
(Captura obtenida del equipo X100 PAD 2 PRO)

En la Figura 3.52, se puede observar el menú inmovilizador del equipo donde se seleccionó la marca del vehículo, este caso se escogió la opción Peugeot.



**Figura 3.53** Selección READ SECURITY CODE  
(Captura obtenida del equipo X100 PAD 2 PRO)

Después de la selección de la marca del vehículo, se abre una nueva ventana en la pantalla del equipo en donde se pueden visualizar tres opciones como se observa en la figura 3.53, en donde se escogió la opción lectura del código de seguridad.



**Figura 3.54** PIN CODE vehículo Peugeot 307  
(Captura obtenida del equipo X100 PAD 2 PRO)

Finalmente, después de seleccionar la opción de lectura del código de seguridad se abre una nueva ventana en donde se pudo obtener el PIN CODE del vehículo como se observa en la Figura 3.51, este código es expresado en un valor alfa numérico.

### 3.3.2 OBTENCIÓN CÓDIGO DE TRANSPONDER DE LLAVES

En este ítem se realizó la clonación de una llave original del vehículo Peugeot 307 en otra llave con transponder con un chip en blanco, el proceso de programación se realizó con ayuda del equipo X100 PAD 2 PRO.

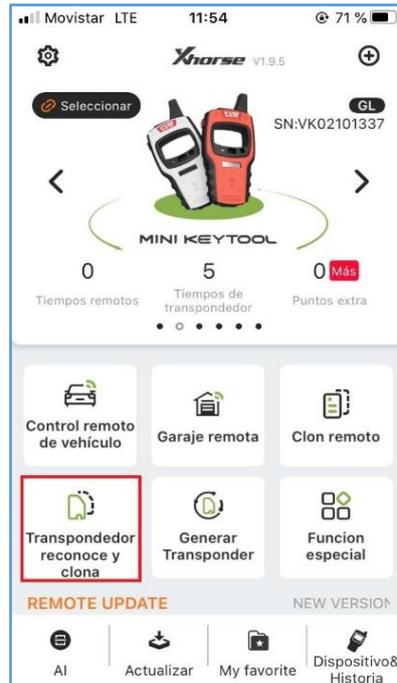
Antes de realizar este proceso fue necesario identificar el código de transponder de que utiliza la llave de encendido para aplicar los pasos a seguir de una manera adecuada.



**Figura 3.55** Llave insertada en el equipo MINI VVDI

Para realizar la lectura del transponder de la llave original se insertó la llave en el equipo MINI VDDI como se observa en la Figura 3.55.

La llave tiene en el mango una protección de material plástico y en su interior viene incrustado el chip del transponder. En este método es necesario verificar que el transponder este ubicado de manera correcta a la altura del equipo para obtener los datos en la interface.



**Figura 3.56** Interface MINI VVDI – Opcion transponder, reconoce y clona  
(Captura obtenida del equipo MINI-VVDI)

En la Figura 3.56, se puede observar la interface del equipo MINI VVDI, la conexión del equipo con la aplicación se la realizo en un teléfono celular mediante vía bluetooth, en el menú principal se seleccionó la opción transponder, reconoce y clona para la lectura del chip original.



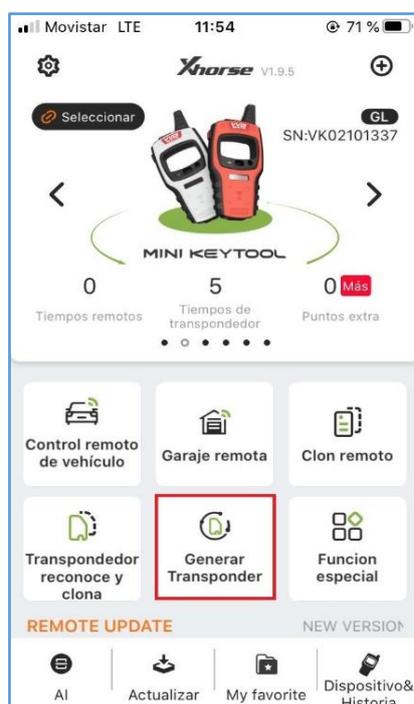
**Figura 3.57** Código de transponder  
(Captura obtenida del equipo MINI-VVDI)

En la Figura 3.57, se puede observar el código de transponder PCF7936, obtenido después de realizar el procedimiento de una manera adecuada, este código fue útil para la programación de una nueva llave del vehículo seleccionado para el caso práctico.

### 3.3.2 PROGRAMACIÓN DE LLAVE

En este punto se detalla el procedimiento que se desarrolló para la reprogramación de dos llaves del vehículo Peugeot 307, para la llave número uno se realiza el reseteo de la llave original y se la genero nuevamente, la segunda llave es la copia programada con un chip virgen y uso del código de transponder.

En primer lugar, en un chip en blanco se generó el código obtenido anteriormente, este proceso se realizó de igual manera en el menú principal de la interface del equipo MINI VVDI.



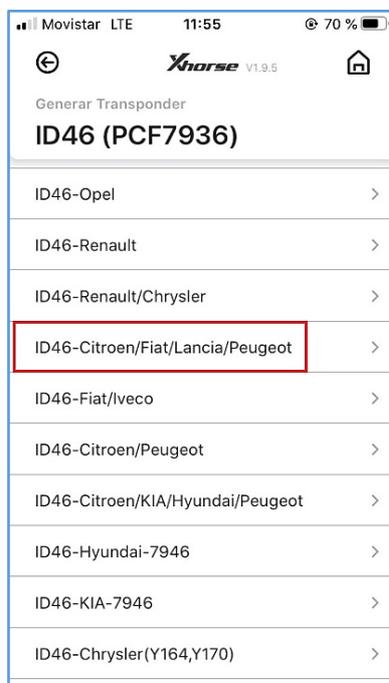
**Figura 3.58** Interface MINI VVDI – Opción generar transponder  
(Captura obtenida del equipo MINI-VVDI)

Para la generación del nuevo chip, como se puede apreciar en la Figura 3.58, se seleccionó la opción generar transponder.



**Figura 3.59** Selección código ID46 PCF7936  
(Captura obtenida del equipo MINI-VVDI)

Después la elección de generar transponder se abrió una nueva ventana en donde se escogió el código ID46 (PCF7936) obtenido anteriormente, esto se puede ver de una manera más detallada en la Figura 3.59.



**Figura 3.60** Selección marca del vehículo - MINI VVDI  
(Captura obtenida del equipo MINI-VVDI)

Una vez seleccionado el código de transponder, la interface mostro una nueva ventana en la cual se tomó la opción de la marca del vehículo como se muestra en la Figura 3.60, en este menú se escogió al grupo Citroën /Fiat/Lancia/Peugeot debido a que todas estas marcas cuentan con el código de chip del transponder obtenido.



**Figura 3.61** Colocación chip virgen en equipo VVDI

Después de seleccionar y realizar los pasos de una manera ordenada se colocó el chip en blanco de transponder en la posición de la bobina del equipo como se muestra Figura 3.61.



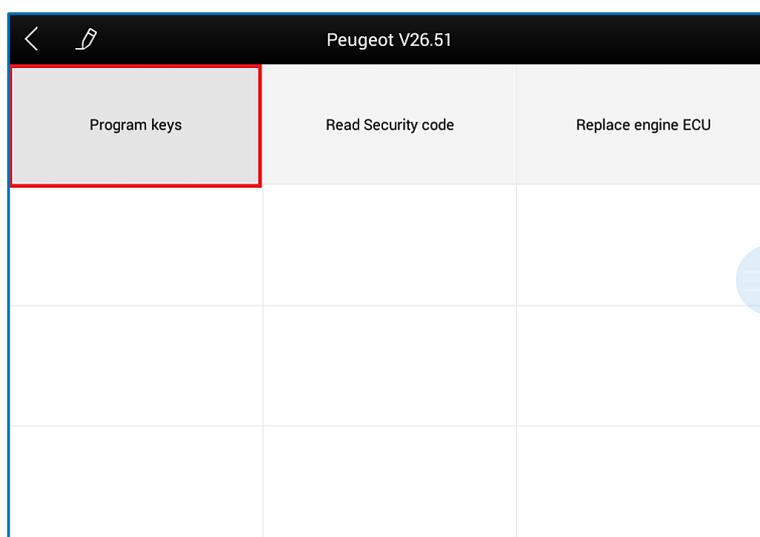
**Figura 3.62** Generar transponder  
(Captura obtenida del equipo MINI-VVDI)

Finalmente, una vez ubicado el chip virgen en la posición adecuada del equipo, en la aplicación se seleccionó generar transponder como se indica en la Figura 3.62, se espera algunos segundos hasta que se complete la clonación de un nuevo chip de manera exitosa. Con la obtención del código de transponder en un chip clonable, se procedió a la conexión del equipo X100 PAD 2 PRO al vehículo mediante el puerto OBD II para la selección de todas las características necesarias que el equipo ordene.



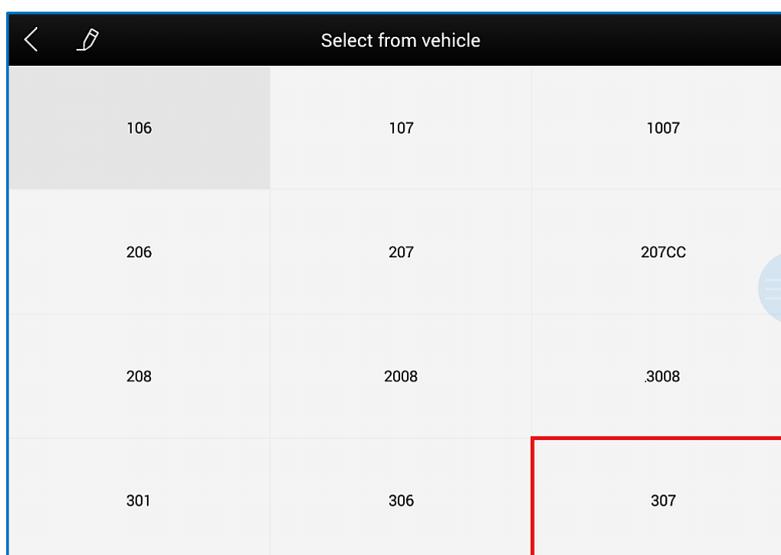
**Figura 3.63** Selección marca del vehículo – Menú inmovilizador  
(Captura obtenida del equipo X100 PAD 2 PRO)

Para el inicio de programación una nueva llave se ingresó al menú inmovilizador del equipo, donde se encontró diferentes marcas europeas de automóviles, en este menú se escogió la marca Peugeot como se observa en la Figura 3.63.



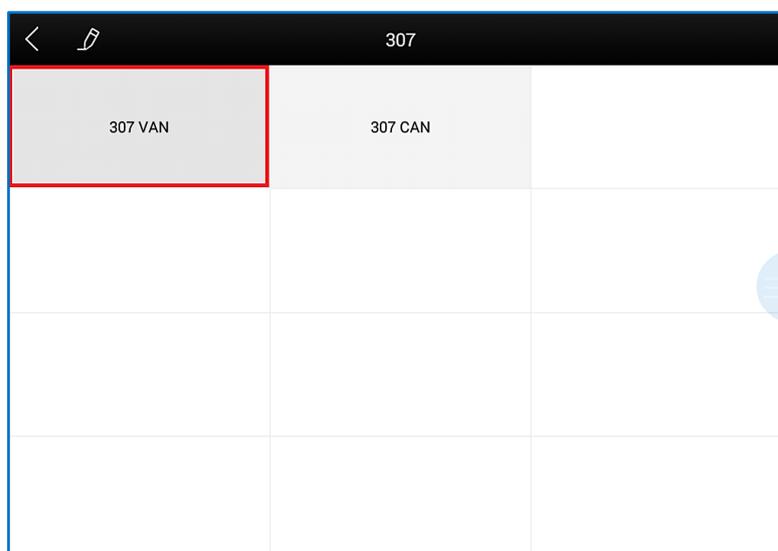
**Figura 3.64** Selección PROGRAM KEYS – Menú inmovilizador  
(Captura obtenida del equipo X100 PAD 2 PRO)

Seleccionado la marca del vehículo, el equipo programador presenta tres opciones; de las cuales se escogió la opción PROGRAM KEYS como se muestra en la Figura 3.64.



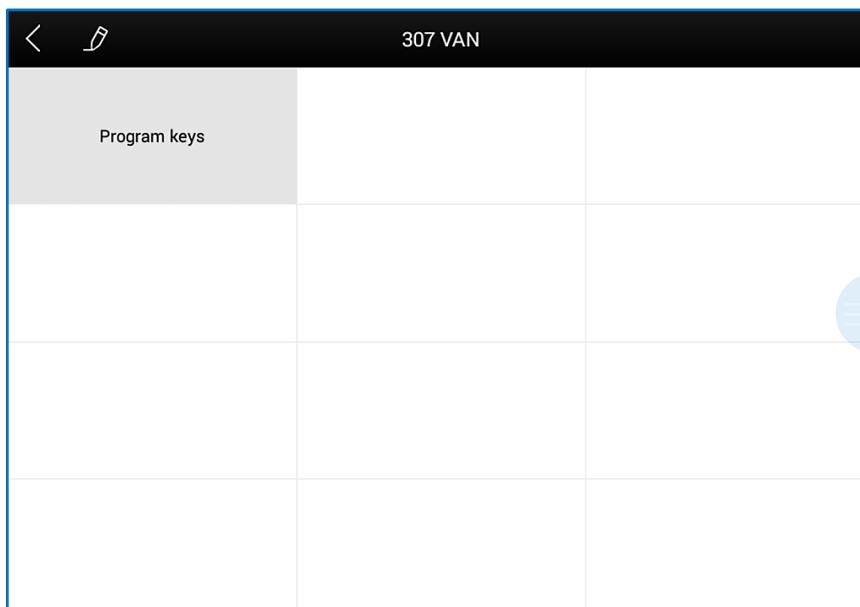
**Figura 3.65** Selección modelo del vehículo – Menú inmovilizador  
(Captura obtenida del equipo X100 PAD 2 PRO)

Después de escoger la opción de programar llaves, la pantalla del equipo muestra los diferentes modelos de vehículos en los que se puede realizar este proceso, para lo cual se seleccionó el modelo 307 como se muestra en la Figura 3.65.



**Figura 3.66** Selección modo de comunicación  
(Captura obtenida del equipo X100 PAD 2 PRO)

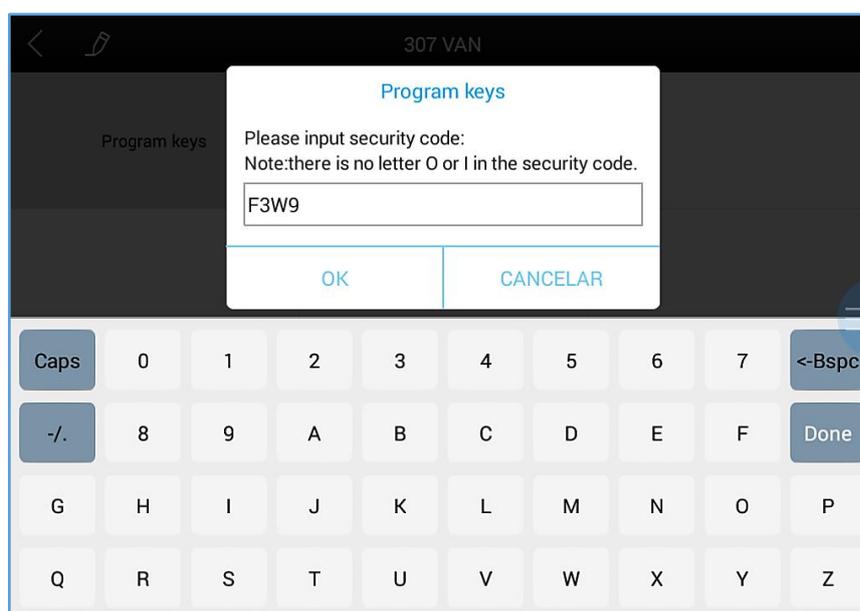
Posteriormente se despliega un menú con dos opciones 307 VAN y 307 CAN como se indica en la Figura 3.66. Se seleccionó 307 VAN debido a que el vehículo utilizado transfiere la información mediante esta red de comunicación.



**Figura 3.67** Opcion program keys  
(Captura obtenida del equipo X100 PAD 2 PRO)

Posteriormente de haber seleccionado el modo de comunicación del vehículo, la pantalla del equipo indica solo la opción programar llaves como se indica en la Figura 3.67, para proseguir con el proceso se escogió la única opción que se dispone.

Después de escoger la función de programar llaves, las copias de llaves existentes en el vehículo incluyendo la llave original serán borradas como método de seguridad.



**Figura 3.68** Opción ingresar PIN CODE  
(Captura obtenida del equipo X100 PAD 2 PRO)

Una vez borrado las llaves existentes, el equipo despliega una advertencia; si el código de seguridad o el PIN CODE ingresado es incorrecto la programación de las llaves no se podrá efectuar, una vez dada la afirmación de proseguir con el proceso de reprogramación se despliega una nueva ventana la cual pide ingresar el código de seguridad o PIN CODE, como se puede apreciar en la Figura 3.68, ingresado el código de seguridad se presionó en OK para seguir con el proceso.

A continuación, el equipo da una afirmación de validación del código de seguridad ingresado. Después, se despliega un menú para ingresar el número de llaves que se requiere programar, en este caso se programó dos llaves.

Para empezar con la programación, el equipo requiere de diferentes acciones manuales a realizar, en primera instancia se giró el switch de encendido a OFF y se retiró la llave, después aparece una advertencia que requiere que en un lapso de 15 segundos se debe de insertar la primera llave y girar el switch de encendido en ON, una vez realizado esta acción el equipo informa e informa la correcta programación de la primera llave, para programar la segunda llave el equipo requiere de las mismas acciones anteriormente descritas, una vez terminado el proceso de programar, el equipo brinda la información de los pasos a seguir para programar un control de bloque y desbloqueo de la puertas, pero para este caso no fue necesario realizar esta acción.

### **3.4 EVALUACIÓN DE ARCHIVOS DE VOLCADO**

En este apartada se dará a conocer la correcta lectura de los archivos de volcado de los módulos de control del vehículo como la ECU, BSI y panel de instrumentos mediante la verificación del PIN CODE y numero VIN del vehículo. Por otro lado, se analizará el tamaño de los archivos encontrados de los tipos de memorias y microcontroladores de la ECU. Además, se hará una comparación de archivos para verificar si existe algún cambio de bytes en el DUMP. Finalmente se analizará los archivos de volcado del módulo de la carrocería y panel de instrumentos para identificar la ubicación exacta de los códigos emitidos del kilometraje recorrido y expresar un algoritmo matemático de como interactúan estos códigos al momento de un aumento de kilómetros al movilizarse el vehículo.

### 3.4.1 ANÁLISIS ARCHIVO DE VOLCADO ECU

En este apartado se realiza un análisis de los archivos de volcado obtenidos de la ECU Bosch ME 7.4.4 del automóvil Peugeot 307 año 2003, los archivos de volcado obtenidos son de la memoria EEPROM ST95160, memoria Flash AM29F400BB Mapas, memoria Micro ST10C167. El análisis en este primer punto corresponde a verificar que la lectura es correcta y determinar el tamaño del archivo.

#### 3.4.1.1 Verificación de lectura archivo de volcado ECU

Para verificar una lectura correcta del archivo de volcado de la unidad electrónica del motor ECU, el método que se utilizó fue la identificación del PIN CODE en uno de los tres archivos de volcado del módulo de control, una vez obtenido dicho código, el mismo que se encuentra escrito en alfanumérico, por tal motivo el PIN CODE debe de ser transformado de alfanumérico a hexadecimales.

En la Tabla 3.11, se encuentra un sistema de transformación de números y letras a códigos hexadecimales, en esta tabla se encuentra caracteres del número 0 al 9 y de la letra A y la Z, este proceso de transformación es utilizado por el grupo PSA que corresponde a Citroën y Peugeot, con este sistema se puede transformar un código alfanúmero a hexadecimal, el sistema de transformación será utilizado para traducir el PIN CODE encontrado anteriormente.

**Tabla 3.11** Transformación de numero PIN a valor hexadecimal

PIN	Hexadecimal	PIN	Hexadecimal	PIN	Hexadecimal
0	CF	C	BC	O	B0
1	CE	D	BB	P	AF
2	CD	E	BA	Q	AE
3	CC	F	B9	R	AD
4	CB	G	B8	S	AC
5	CA	H	B7	T	AB
6	C9	I	B6	U	AA
7	C8	J	B5	V	A9
8	C7	K	B4	W	A8
9	C6	L	B3	X	A7
A	BE	M	B2	T	A6
B	BD	N	B1	Z	A5

El PIN CODE extraído con el equipo X100 PAD 2 PRO fue F3W9, para traducir este código a hexadecimales y posteriormente poder localizarlo en un archivo de volcado se utilizó la tabla descrita anteriormente.

En la Tabla 3.12, se detalla el método de transformación del PIN CODE, modificando sus valores alfanuméricos a valores hexadecimales, por la tanto el código de seguridad F3W9 en hexadecimales es B9, CC, A8, C6.

**Tabla 3.12** Transformación Pin Code de alfanumérico a hexadecimales

PIN CODE de alfanumérico a hexadecimales				
Pin Code	F	3	W	9
Pin Code en hexadecimales	B9	CC	A8	C6

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000540	00	00	00	00	00	00	00	00	00	00	00	00	00	00	AC	FF	.....~ý
00000550	00	00	00	00	00	00	00	00	00	00	00	00	00	00	AB	FF	.....<ý
00000560	00	08	00	00	00	00	00	00	00	00	00	00	00	00	A2	FF	.....çý
00000570	00	00	00	00	00	00	00	00	00	00	00	00	00	00	A9	FF	.....@ý
00000580	00	00	00	00	00	00	00	00	00	00	00	00	00	00	A8	FF	.....ý
00000590	58	01	46	33	57	39	B9	CC	A8	C6	01	FE	04	FB	54	F9	X.F3W9^I^Æ.p.úTù
000005A0	58	01	46	33	57	39	B9	CC	A8	C6	01	FE	04	FB	54	F9	X.F3W9^I^Æ.p.úTù
000005B0	19	76	19	68	00	00	00	00	00	00	00	00	00	00	95	FE	.v.h..... þ
000005C0	19	76	19	68	00	00	00	00	00	00	00	00	00	00	95	FE	.v.h..... þ
000005D0	AA	AA	AA	00	00	00	00	00	00	00	00	00	00	00	A5	FD	aaa.....#ý
000005E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	A2	FF	.....çý
000005F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	A2	FF	.....çý
00000600	00	00	00	00	00	00	00	00	00	00	00	00	00	00	A0	FF	.....ý
00000610	00	04	00	00	00	00	00	00	00	00	00	00	00	00	9B	FF	..... ý
00000620	00	00	00	00	00	00	00	00	00	00	00	00	00	00	9E	FF	..... ý
00000630	00	00	00	00	00	00	00	00	00	00	00	00	00	00	9D	FF	.....ý
00000640	00	00	00	00	00	00	00	00	00	00	00	00	00	00	9C	FF	..... ý
00000650	80	00	00	00	00	00	00	00	00	00	00	00	00	00	1B	FF	.....ý
00000660	00	00	00	00	00	00	00	00	00	00	00	00	00	00	9A	FF	..... ý
00000670	00	00	00	00	00	00	00	00	00	00	00	00	00	00	99	FF	..... ý

**Figura 3.69** Ubicación Pin Code en memoria EEPROM ST95160

En la Figura 3.69, se puede apreciar la ubicación del PIN CODE del vehículo, este se encuentra repetido en 2 grupos de 4 bytes, los hexadecimales correspondientes para este código son B8, CC, A8, C6, este se lo puedo encontrar en el archivo de volcado de la memoria EEPROM ST95160, en la dirección 590 y registros 06, 07, 08, 09, este código se repite en la dirección 5A0 y registros 06, 07, 08, 09.

### 3.4.1.2 Tamaño archivos de volcado ECU

- **Memoria EEPROM ST95160**

En este punto con la ayuda del programa FAIRDELL HEXCMP se procede a analizar el tamaño del archivo de volcado obtenido de la memoria EEPROM ST95160 de la ECU Bosch ME 7.4.4.

File size	Value
- HEX	0x800
- DECIMAL	2048
Offset	
- HEX	0x0
- DECIMAL	0
Value	
- Character	'Z'
- Byte (HEX)	0x5A
- Byte (DEC)	90
- Word (HEX)	0x5A
- Word (DEC)	90
- DWord (HEX)	0x5A
- DWord (DEC)	90

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	A	00	00	00	02	00	60	01	05	60	A5	4E	EC	7B	84	FC
00000010	0D	04	02	02	02	00	00	00	CF	AB	34	58	A2	03	3D	FD
00000020	0D	04	02	02	02	00	00	00	CF	AB	34	58	A2	03	3D	FD
00000030	01	83	69	74	3E	67	FF	F5								
00000040	01	83	69	74	3E	67	FF	F5								
00000050	53	0A	83	02	99	0D	A9	01	04	00	20	04	6F	0E	24	FD
00000060	53	0A	83	02	99	0D	A9	01	04	00	20	04	6F	0E	24	FD
00000070	1B	00	6E	0E	2F	12	00	00	00	00	00	00	00	00	21	FF
00000080	1B	00	6E	0E	2F	12	00	00	00	00	00	00	00	00	21	FF
00000090	FF	20	00	00	00	00	00	00	00	00	00	00	00	00	D8	FE
000000A0	FF	20	00	00	00	00	00	00	00	00	00	00	00	00	D8	FE
000000B0	00	00	00	00	00	00	00	00	0D	03	00	00	00	00	E5	FF
000000C0	00	00	00	00	00	00	00	00	0D	03	00	00	00	00	E5	FF
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	F3	FF
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	F3	FF
000000F0	0A	2E	FF	FF	00	40	27	00	17	01	C6	00	00	00	76	FC
00000100	AE	07	00	00	AE	07	00	00	00	00	00	00	0A	00	7C	FE
00000110	00	00	00	00	00	80	00	00	00	80	00	00	00	00	EF	FE
00000120	FD	FF	00	00	00	00	00	00	3D	00	F6	3E	5C	00	25	FC
00000130	01	00	D6	FF	00	00	C3	F6								
00000140	62	31	62	31	62	31	2B	4E	2B	4E	02	B5	10	0D	6D	FC
00000150	00	00	D6	FF	00	00	00	00	00	00	D6	FF	D6	FF	6C	FA
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	EA	FF
00000170	00	00	00	00	00	00	00	80	00	80	00	80	00	80	E9	FD
00000180	00	00	00	00	00	00	00	00	FD	FF	94	B0	00	00	A8	FC

**Figura 3.70** Tamaño archivo de volcado de la memoria EEPROM ST95160

El tamaño de un archivo se determina por el número de bytes que abarque, por lo tanto, el tamaño de la memoria EEPROM ST95160 es de 2048 bytes, como se puede observar en la Figura 3.70.

- **Memoria Flash AM29F400BB Mapas**

Para la verificación del tamaño de la memoria Flash AM29F400BB Mapas se lo realizó con el programa del punto anterior.

The screenshot shows the Fairdell HexCmp application interface. On the left, a 'File size' table is displayed with the following data:

File size	Value
- HEX	0x80000
- DECIMAL	524288
Offset	
- HEX	0x0
- DECIMAL	0
Value	
- Character	'I'
- Byte (HEX)	0x88
- Byte (DEC)	136
- Word (HEX)	0x9088
- Word (DEC)	-28536
- DWord (HEX)	0x80889088
- DWord (DEC)	-2138533752

The main window displays a hex dump for the file 'First File - C:\Users\fabry.LAPTOP-COPG002K\OneDrive\Escritorio\Pruebas Tesis\1'. The dump shows the following data:

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	88	90	88	80	E0	19	DA	80	8E	00	49	80	2D	2D	DA	80
00000010	F2	00	48	40	2D	15	E7	F8	5A	00	F7	F8	EF	F9	E7	FA
00000020	93	00	F7	FA	F0	F9	F7	8E	EE	F9	DE	E0	48	90	2D	04
00000030	E0	1C	DA	00	66	64	08	91	9A	E0	F8	80	DF	E0	0D	24
00000040	E7	F8	A5	00	F7	F8	EE	F9	F7	8E	EF	F9	F7	8E	F0	F9
00000050	DA	86	A4	F3	B5	4A	B5	B5	E0	08	08	81	46	F8	40	9C
00000060	8D	FC	DA	80	8A	18	0D	10	E7	F8	5A	00	F7	F8	EF	F9
00000070	F7	8E	EE	F9	F7	8E	F0	F9	E6	F4	5D	00	F6	F4	0A	AA
00000080	E6	F5	A2	FF	F6	F5	2E	AA	98	80	98	90	DB	00	88	70
00000090	88	60	E1	1E	E0	0C	E1	0C	C0	CE	5C	3E	D4	4E	4E	6A
000000A0	48	41	3D	17	D4	4E	52	6A	E6	FD	A8	F7	C0	C5	E0	F2
000000B0	20	25	5C	12	00	D2	A8	5D	66	F5	FF	03	40	45	ED	07
000000C0	D4	4E	54	6A	A8	5D	66	F5	FF	03	40	45	9D	05	E1	0E
000000D0	0D	03	F0	4C	09	81	F0	C4	09	C1	47	FC	10	00	8D	DC
000000E0	F0	4C	47	F8	10	00	8D	01	E1	0E	F1	8E	98	60	98	70
000000F0	DB	00	8A	E0	02	30	E0	14	DB	00	E0	04	DB	00	44	44
00000100	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000110	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000120	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000130	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000140	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000150	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000160	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000170	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
00000180	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44

**Figura 3.71** Tamaño archivo de volcado de la memoria Flash AM29F400BB Mapas

El tamaño del archivo de volcado obtenido de la lectura a la memoria Flash AM29F400BB Mapas es de 524 288 bytes, como se indica en la Figura 3.71.

- **Memoria Micro ST10C167**

En esta sección de igual manera que en los puntos anteriores se utilizó el programa FAIRDELL HEXCMP para el análisis del tamaño del archivo de volcado obtenido de la memoria Micro ST10C167 de la ECU Bosch ME 7.4.4.

The screenshot shows the Fairdell HexCmp application window. On the left, a 'File size' table is displayed with the following data:

File size	Value
- HEX	0x8000
- DECIMAL	32768
Offset	
- HEX	
- DECIMAL	
Value	
- Character	
- Byte (HEX)	
- Byte (DEC)	
- Word (HEX)	
- Word (DEC)	
- DWord (HEX)	
- DWord (DEC)	

The main area displays a hex dump for the file 'C:\Users\fabry.LAPTOP-COPG002K\OneDrive\Escritorio\Pruebas Tesis\1'. The dump shows a sequence of hexadecimal values starting from offset 00000000. The first few lines are:

```

00000000  FA 00 70 64 FA 82 04 00 FA 00 18 67 FA 82 0C 00
00000010  FA 00 30 67 FA 82 14 00 FA 00 48 67 FA 82 1C 00
00000020  FA 82 20 00 FA 82 24 00 FA 00 60 67 FA 82 2C 00
00000030  FA 82 30 00 FA 82 34 00 FA 82 38 00 FA 82 3C 00
00000040  FA 82 40 00 FA 82 44 00 FA 82 48 00 FA 82 4C 00
00000050  FA 82 50 00 FA 82 54 00 FA 82 58 00 FA 82 5C 00
00000060  FA 82 60 00 FA 82 64 00 FA 82 68 00 FA 82 6C 00
00000070  FA 82 70 00 FA 82 74 00 FA 82 78 00 FA 82 7C 00
00000080  FA 82 80 00 FA 82 84 00 FA 82 88 00 FA 82 8C 00
00000090  FA 00 2C 13 FA 00 C0 1E FA 00 6C 17 FA 00 96 13
000000A0  FA 82 A0 00 FA 82 A4 00 FA 82 A8 00 FA 00 CA 22
000000B0  FA 00 2A 22 FA 82 B4 00 FA 82 B8 00 FA 82 BC 00
000000C0  FA 82 C0 00 FA 82 C4 00 FA 82 C8 00 FA 82 CC 00
000000D0  FA 82 D0 00 FA 82 D4 00 FA 82 D8 00 FA 82 DC 00
000000E0  FA 00 7A 22 FA 82 E4 00 FA 82 E8 00 FA 82 EC 00
000000F0  FA 82 F0 00 FA 82 08 01 FA 82 F8 00 FA 82 FC 00
00000100  FA 82 00 01 FA 82 04 01 FA 82 08 01 FA 82 0C 01
00000110  FA 82 10 01 FA 82 14 01 FA 82 18 01 FA 82 1C 01
00000120  FA 82 20 01 FA 82 24 01 FA 82 28 01 FA 82 2C 01
00000130  FA 82 30 01 FA 82 34 01 FA 82 38 01 FA 82 3C 01
00000140  FA 82 40 01 FA 00 78 67 FA 00 A8 67 FA 82 4C 01
00000150  FA 82 50 01 FA 82 54 01 FA 82 58 01 FA 82 5C 01
00000160  FA 82 60 01 FA 82 64 01 FA 82 68 01 FA 82 6C 01
00000170  FA 82 70 01 FA 82 74 01 FA 82 78 01 FA 82 7C 01
00000180  FA 82 80 01 FA 82 84 01 FA 82 88 01 FA 82 8C 01

```

**Figura 3.72** Tamaño archivo de volcado de la memoria Micro ST10C167

El tamaño del archivo de volcado de la memoria Micro ST10C167 es de 32 768 byts, como se indica en la Figura 3.72.

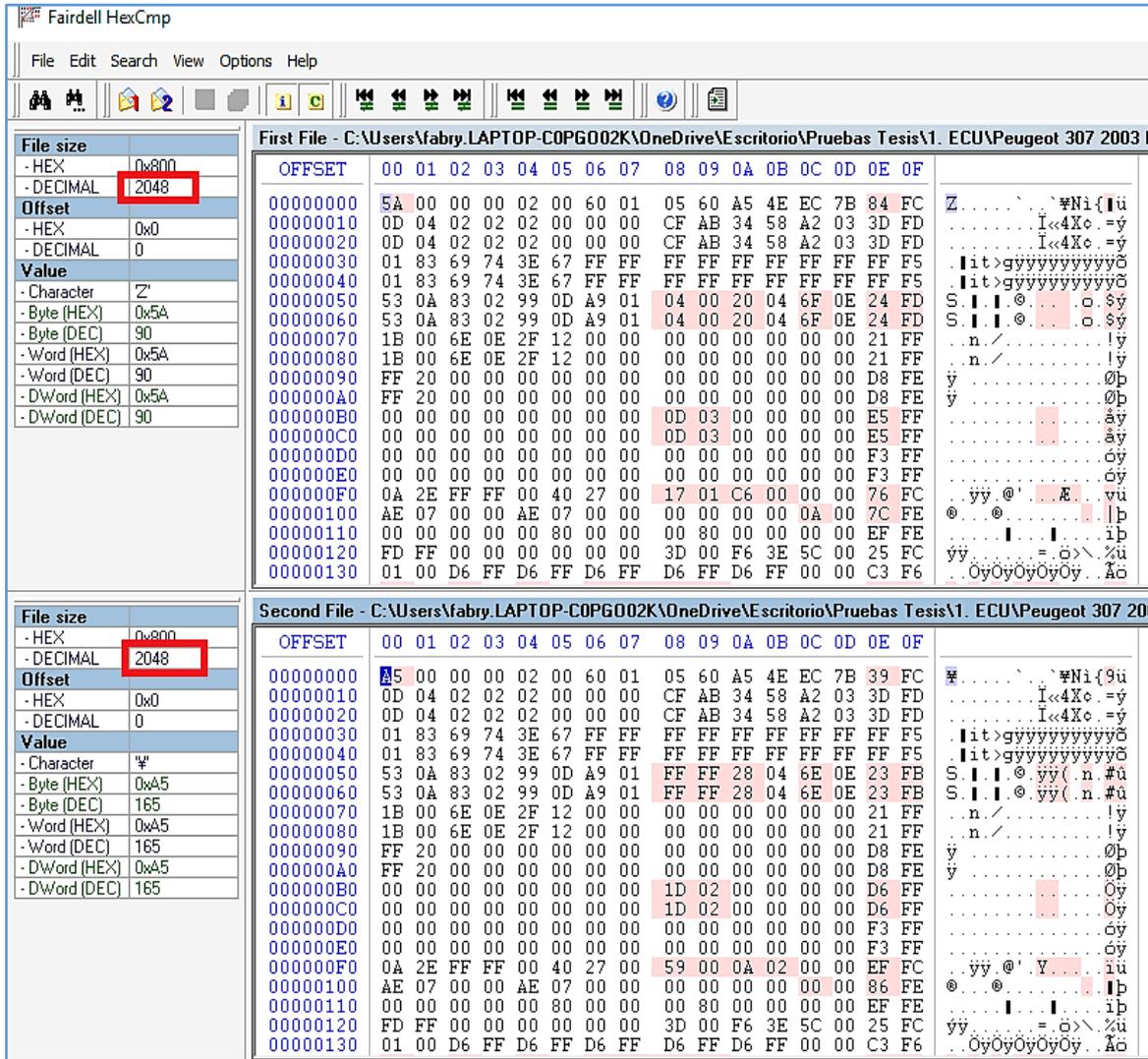
### 3.4.2 COMPARACIÓN DE ARCHIVOS DE VOLCADO DE LA ECU ANTES Y DESPUÉS DE UN RESET

En este punto se realizó una comparación entre los archivos de volcado obtenidos antes de realizar y después de realizar un proceso de reseteo a las funciones especiales encontradas en el vehículo, como se mencionó en capítulos anteriores la dos funciones que dispone el automóvil son: Reset luz de servicio y un ajuste del ángulo de dirección.

Por lo tanto, se procedió a comparar el tamaño entre archivos y el cambio de datos o bytes que existe después de haber realizado el Reset en la memoria EEPROM ST95160, memoria Flash AM29F400BB Mapas, memoria Micro ST10C167.

### 3.4.2.1 Análisis memoria EEPROM ST95160 después del Reset luz de servicio

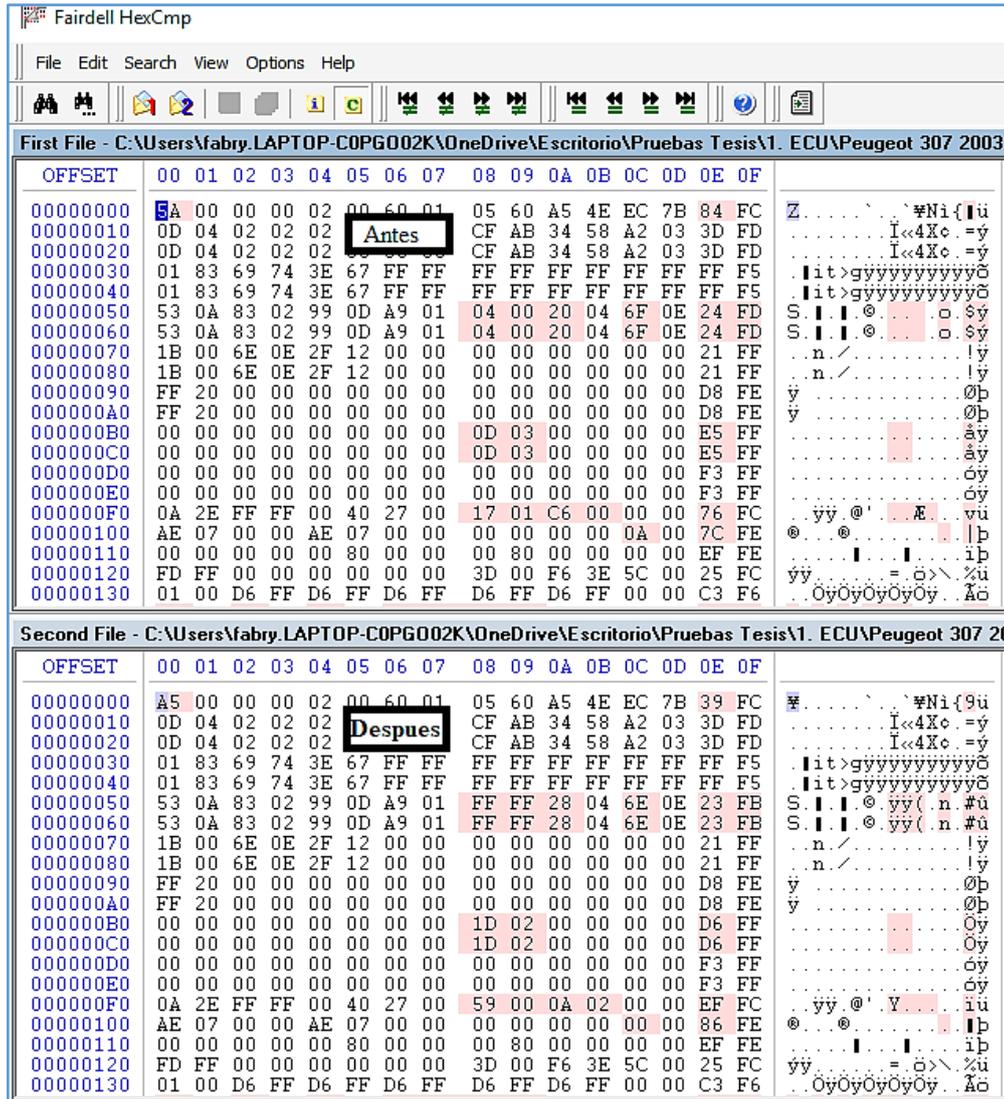
La comparación del tamaño del archivo de volcado de la memoria EEPROM ST95160 después de un proceso de reseteo de la luz de servicio con el archivo original determina que la relación de este parámetro no presenta ningún cambio.



**Figura 3.73** Tamaño del archivo Memoria EEPROM ST95160 antes y después del Reset luz de servicio

El tamaño del archivo de volcado obtenido después de haber realizado el Reset luz de servicio es de 2048 bytes al igual que el tamaño del archivo de volcado original como se indica en la Figura 3.73.

El siguiente paso fue analizar la cantidad y la escritura de datos o bytes que cambian después del Reset de luz de servicio en la memoria EEPROM ST95160.



**Figura 3.74** Comparación de datos o bytes de la memoria EEPROM ST95160 antes y después del Reset luz de servicio

EL software FAIRDELL HEXCMP subraya mediante un color predeterminado el cambio de datos o bytes hexadecimales que existe entre dos archivos de volcado como se puede ver en la Figura 3.74.

Después del Reset luz de servicio en la memoria EEPROM ST95160 se puede apreciar que existe diferentes cambios de datos o bytes, a continuación, se explica el cambio en cada dirección (fila) y registro (columna) que se ha producido.

En la Tabla 3.13, se indica los datos o bytes que cambiaron en la memoria EEPROM ST95160 después de haber realizado el Reset de luz de servicio, la escritura que se generó en el segundo archivo de volcado es en 22 direcciones, dando como un total de 102 bytes modificados.

**Tabla 3.13** Análisis bytes memoria EEPROM ST95160 antes y después Reset luz de servicio

EEPROM ST95160	BYTES ANTES DEL RESET LUZ DE SERVICIO																BYTES DESPUÉS DEL RESET LUZ DE SERVICIO															
	REGISTROS																															
DIRECCIÓN	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	5A														84		A5														39	
50								04	00	20		6F		24	FD										FF	FF	28		6E		23	FB
60								04	00	20		6F		24	FD										FF	FF	28		6E		23	FB
B0								0D	03					E5											1D	02					D6	
C0								0D	03					E5											1D	02					D6	
F0								17	01	C6	00			76											59	00	0A	02			EF	
100												0A		7C														00		86		
140	62		62		62		2B	4E	2B	4E				6D	FC	3B		3B		3B		ED	4D	ED	4D					60	FB	
1A0												2B	4E	C3													AE	41	02			
1B0			00		BB	03	3C						A6	B1	FD			01		06	00	5B						AE	41	FE		
270	01						FD	0C						09		C7						24	0D							BB		
2B0	15	19		04		01								A1		17	1B		08		00									9A		
2DA					00									54							20									34		
2E0		43									B4			24			3C										6C			73		
330								1E	1E					3B										14	17					4C		
340			9C	4E	86									69				9B	4D	64										8D		
350		01		15		01								9F			03		18		03									98		
4FD								03						6F										02						70		
510	00	00		00	00	00	00	00	00	00				AF	FF	03	44		1E	03	34	5B	02	03	28					8B	FE	
520		00	00	00	00	00	00	00	00	00	00			AF	FF		02	01	FF	0D	49	80	7F	FF	FF	2B	14			1A	FB	
610		00												9F			04													9B		
7F0													5A	26	FE														A5	DB	FD	

A continuación, se detalla de cada dirección y registro los bytes o datos modificados. El primer byte que cambió es en la dirección 00 y registro 00, el valor hexadecimal modificado es de 5A por A5, y en la misma dirección y registro 0E hay otro dato modificado de 84 a 39.

En la dirección 50 hay una modificación de 6 bytes en los registros 08, 09, 0A, 0C, 0E, 0F, los bytes modificados son 04, 00, 20, 6F, 24, FD a FF, 28, 6E, 23, FB, los datos o bytes modificados en la dirección 60 son en los mismos tanto en los registros como en bytes que la dirección 50.

En las direcciones B0 y C0 hay tres datos modificados en cada uno, los bytes modificados son en las direcciones 08, 09, 0E de 0D, 03, E5 a 1D, 02, D6 en cada dirección.

En la dirección F0 hay 5 datos modificados en los registros 08, 09, 0A, 0B, 0E de 17, 01, C6, 00, 76 a 59, 00, 0A, 02, EF.

En la dirección 100 y registros 0C, 0E los bytes modificados son de 0A, 7C a 00, 86.

En la dirección 140 y registros 00, 02, 04, 06, 07, 08, 09, 0E, 0F los bytes modificados son de 62, 62, 62, 2B, 4E, 2B, 4E, 6D, FC a 3B, 3B, 3B, ED, 4D, ED, 4D, 60, FB.

En la siguiente dirección 1A0 hay 3 bytes modificados en los registros 0C, 0D, 0F los datos modificados son de 2B, 4B, 3C a AE, 41, 02.

En la dirección 1B0 y registros 02, 04, 05, 06 0D, 0E, 0F los bytes modificados son de 00, BB, 03, 3C a 01, 06, 00, 5B, AE, 41, FE.

En la dirección 270 hay 4 datos modificados en los registros 01, 06, 07, 0E en los bytes modificados son de 01, FD, 0C, 09 a C7, 24, 0D, BB.

En la dirección 2B0 hay 5 bytes modificados en los registros 00, 01, 03, 05, 0E los datos modificados son de 15, 19, 04, 01, A1 a 17, 1B, 08, 00, 9A.

En la dirección 2DA hay 2 bytes modificados en los registros 04, 0E los datos que cambian son de 00, 54 a 20, 34.

En la dirección 2EA hay 3 bytes modificados en los registros 01, 0A, 24 los datos modificados son de 43, B4, 24 a 3C, 6C, 73.

En la dirección 330 hay 3 bytes modificados en los registros 08, 09, 0E los datos modificados son de 1E, 1E, 3B a 14, 17, 4C.

En la dirección 340 se han modificado bytes en los registros 02, 03, 04, 0E los datos modificados son de 9C, 4E, 86, 69 a 9B, 4D, 64, 8D.

En la dirección 350 y registros 01, 03, 05, 0E hay cambios en los bytes de 01, 15, 01, 9F a 03, 18, 03, 98. En la dirección 4FD y registros 07, 0E se hay modificaciones de 03, 6F a 02, 70.

En las direcciones 510 y 520 se han modificado 10 bytes en cada una, de los cuales 8 bytes en cada dirección no tienen registrado un comando o una orden.

En la dirección 510 y en los registros 00, 01, 03, 04, 05, 06, 07, 08, 09, 0E, 0F hay modificaciones en los bytes de 00, 00, 00, 00, 00, 00, 00, 00, AF, FF a 03, 44, 1E, 03, 34, 5B, 02, 03, 28, 8B, FE.

En la dirección 520 y registros 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0E, 0F los bytes modificados son de 00, 00, 00, 00, 00, 00, 00, 00, AF, FF a 02, 01, FF, 0D, 49, 80, 7F, FF, FF, 2B, 14, 1A, FB.

En la dirección 610 y registros 01, 0E los bytes que se han modificado son de 00, 9F a 04, 9B. Por último, en la dirección 7F0 en los registros 0D, 0E, 0F en los bytes se han modificado de 5A, 26, FE a 5A, DB, FD.

#### **3.4.2.2 Análisis memoria Flash AM29F400BB Mapas después del Reset luz de servicio**

En este ítem se procedió a verificar que el tamaño del archivo de volcado de la memoria Flash AM29F400BB Mapas obtenido después del reseteo de luz de servicio, sea igual al tamaño del archivo de volcado obtenido antes del reseteo.

The image displays two screenshots of the Fairdell HexCmp application. Each screenshot shows a file comparison window with a metadata panel on the left and a hex dump on the right. The metadata panel includes fields for File size (HEX: 0x80000, DECIMAL: 524288), Offset (HEX: 0x0, DECIMAL: 0), and Value (Character: T, Byte (HEX): 0x88, Byte (DEC): 136, Word (HEX): 0x9088, Word (DEC): -28536, DWord (HEX): 0x80889088, DWord (DEC): -2138533752). The hex dump shows a sequence of bytes from offset 00000000 to 00000130. The top window is labeled 'First File' and the bottom window is labeled 'Second File'. Both windows show identical hex dump data, with a red box highlighting the value '524288' in the 'DECIMAL' field of the 'Value' section in both.

**Figura 3.75** Tamaño del archivo Memoria Flash AM29F400BB Mapas antes y después del Reset luz de servicio

El tamaño del archivo de volcado de la memoria Flash AM29F400BB Mapas, correspondiente a la lectura después del Reset luz de servicio es de 524 288 bytes al igual que el tamaño del archivo de volcado original de la misma memoria, esto se puede observar de una manera más detallada en Figura 3.75.

La lectura de la memoria Flash AM29F400BB Mapas después de haber realizado el Reset luz de servicio no presenta ninguna modificación en los bytes.

### 3.4.2.3 Análisis memoria Micro ST10C167 después del Reset luz de servicio

En este punto se realizó el análisis del tamaño del archivo de volcado de la memoria Micro ST10C167 obtenido después del reseteo de luz de servicio para verificar si existen cambios en con el tamaño del archivo de volcado obtenido antes de este proceso.

The screenshot displays the Fairdell HexCmp application interface. It features a menu bar (File, Edit, Search, View, Options, Help) and a toolbar with various file operations. The main workspace is divided into two panes, each showing a memory dump comparison.

**First File - C:\Users\Fabry.LAPTOP-C0PG002K\OneDrive\Escritorio\Pruebas Tesis\1. ECU\Peugeot 307 2003 B**

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	FA	00	70	64	FA	82	04	18	67	FA	82	0C	00	00	00	00	ú.pdú ..ú. .gú ..
00000010	FA	00	30	67	FA	82	14	48	67	FA	82	1C	00	00	00	00	ú.0gú ..ú.Hgú ..
00000020	FA	82	20	00	FA	82	24	00	FA	00	60	67	FA	82	2C	00	ú ..ú \$.ú ..gú ..
00000030	FA	82	30	00	FA	82	34	00	FA	82	38	00	FA	82	3C	00	ú 0.ú 4.ú 8.ú <
00000040	FA	82	40	00	FA	82	44	00	FA	82	48	00	FA	82	4C	00	ú @.ú D.ú H.ú L.
00000050	FA	82	50	00	FA	82	54	00	FA	82	58	00	FA	82	5C	00	ú P.ú T.ú X.ú N.
00000060	FA	82	60	00	FA	82	64	00	FA	82	68	00	FA	82	6C	00	ú '.ú d.ú h.ú l.
00000070	FA	82	70	00	FA	82	74	00	FA	82	78	00	FA	82	7C	00	ú p.ú t.ú x.ú .
00000080	FA	82	80	00	FA	82	84	00	FA	82	88	00	FA	82	8C	00	ú l.ú l.ú l.ú l.
00000090	FA	00	2C	13	FA	00	C0	1E	FA	00	6C	17	FA	00	96	13	ú. .ú. Á.ú. l.ú.  .
000000A0	FA	82	A0	00	FA	82	A4	00	FA	82	A8	00	FA	00	CA	22	ú ..ú P.ú ..ú.É"
000000B0	FA	00	2A	22	FA	82	B4	00	FA	82	B8	00	FA	82	BC	00	ú.*ú ..ú ..ú K.
000000C0	FA	82	C0	00	FA	82	C4	00	FA	82	C8	00	FA	82	CC	00	ú Á.ú Á.ú É.ú I.
000000D0	FA	82	D0	00	FA	82	D4	00	FA	82	D8	00	FA	82	DC	00	ú Ð.ú Ó.ú Ø.ú Ü.
000000E0	FA	00	7A	22	FA	82	E4	00	FA	82	E8	00	FA	82	EC	00	ú.z.ú á.ú è.ú i.
000000F0	FA	82	F0	00	FA	82	08	01	FA	82	F8	00	FA	82	FC	00	ú Ë.ú ..ú è.ú ü.
00000100	FA	82	00	01	FA	82	04	01	FA	82	08	01	FA	82	0C	01	ú ..ú ..ú ..ú .
00000110	FA	82	10	01	FA	82	14	01	FA	82	18	01	FA	82	1C	01	ú ..ú ..ú ..ú .
00000120	FA	82	20	01	FA	82	24	01	FA	82	28	01	FA	82	2C	01	ú ..ú \$.ú ..ú .
00000130	FA	82	30	01	FA	82	34	01	FA	82	38	01	FA	82	3C	01	ú 0.ú 4.ú 8.ú <

**Second File - C:\Users\Fabry.LAPTOP-C0PG002K\OneDrive\Escritorio\Pruebas Tesis\1. ECU\Peugeot 307 200**

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	FA	00	70	64	FA	82	04	18	67	FA	82	0C	00	00	00	00	ú.pdú ..ú. .gú ..
00000010	FA	00	30	67	FA	82	14	48	67	FA	82	1C	00	00	00	00	ú.0gú ..ú.Hgú ..
00000020	FA	82	20	00	FA	82	24	00	FA	00	60	67	FA	82	2C	00	ú ..ú \$.ú ..gú ..
00000030	FA	82	30	00	FA	82	34	00	FA	82	38	00	FA	82	3C	00	ú 0.ú 4.ú 8.ú <
00000040	FA	82	40	00	FA	82	44	00	FA	82	48	00	FA	82	4C	00	ú @.ú D.ú H.ú L.
00000050	FA	82	50	00	FA	82	54	00	FA	82	58	00	FA	82	5C	00	ú P.ú T.ú X.ú N.
00000060	FA	82	60	00	FA	82	64	00	FA	82	68	00	FA	82	6C	00	ú '.ú d.ú h.ú l.
00000070	FA	82	70	00	FA	82	74	00	FA	82	78	00	FA	82	7C	00	ú p.ú t.ú x.ú .
00000080	FA	82	80	00	FA	82	84	00	FA	82	88	00	FA	82	8C	00	ú l.ú l.ú l.ú l.
00000090	FA	00	2C	13	FA	00	C0	1E	FA	00	6C	17	FA	00	96	13	ú. .ú. Á.ú. l.ú.  .
000000A0	FA	82	A0	00	FA	82	A4	00	FA	82	A8	00	FA	00	CA	22	ú ..ú P.ú ..ú.É"
000000B0	FA	00	2A	22	FA	82	B4	00	FA	82	B8	00	FA	82	BC	00	ú.*ú ..ú ..ú K.
000000C0	FA	82	C0	00	FA	82	C4	00	FA	82	C8	00	FA	82	CC	00	ú Á.ú Á.ú É.ú I.
000000D0	FA	82	D0	00	FA	82	D4	00	FA	82	D8	00	FA	82	DC	00	ú Ð.ú Ó.ú Ø.ú Ü.
000000E0	FA	00	7A	22	FA	82	E4	00	FA	82	E8	00	FA	82	EC	00	ú.z.ú á.ú è.ú i.
000000F0	FA	82	F0	00	FA	82	08	01	FA	82	F8	00	FA	82	FC	00	ú Ë.ú ..ú è.ú ü.
00000100	FA	82	00	01	FA	82	04	01	FA	82	08	01	FA	82	0C	01	ú ..ú ..ú ..ú .
00000110	FA	82	10	01	FA	82	14	01	FA	82	18	01	FA	82	1C	01	ú ..ú ..ú ..ú .
00000120	FA	82	20	01	FA	82	24	01	FA	82	28	01	FA	82	2C	01	ú ..ú \$.ú ..ú .
00000130	FA	82	30	01	FA	82	34	01	FA	82	38	01	FA	82	3C	01	ú 0.ú 4.ú 8.ú <

**Figura 3.76** Comparación archivo de volcado de la memoria Micro ST10C167 antes y después del Reset luz de servicio.

El tamaño del archivo de volcado de la memoria Micro ST10C167 correspondiente a la lectura después del Reset luz de servicio es de 32 768 bytes, igual al tamaño del archivo de volcado de la misma memoria antes de realizar el proceso de reseteo como se indica en la Figura 3.76.

Por otro lado, se determinó que la lectura de la memoria Micro ST10C167 después de haber realizado el Reset luz de servicio no presenta ninguna modificación en los bytes.

### 3.4.2.4 Análisis memoria EEPROM ST95160 después del ajuste del sensor de aceleración

En este punto se procedió a verificar que el tamaño del archivo de volcado de la memoria EEPROM ST95160 obtenido después del ajuste del sensor de aceleración, sea igual al tamaño del archivo de volcado obtenido antes del reseteo. Cabe mencionar que el archivo de volcado antes de realizar el ajuste del sensor de aceleración es el mismo archivo que se obtuvo después de haber realizado el reseteo de la luz de servicio.

The screenshot displays the Fairdell HexCmp application interface. It shows two files being compared, both with a size of 2048 bytes. The 'File size' section on the left for both files shows:
 

- HEX: 0x800
- DECIMAL: 2048

 The main window displays the hex dump for both files. The first file is 'C:\Users\fabry.LAPTOP-C0PG002K\OneDrive\Escritorio\Pruebas Tesis\1. ECU\Peugeot 307 2003 B' and the second is 'C:\Users\fabry.LAPTOP-C0PG002K\OneDrive\Escritorio\Pruebas Tesis\1. ECU\Peugeot 307 200'. Both files show identical hex data starting from offset 00000000 to 00000130.

**Figura 3.77** Tamaño del archivo Memoria EEPROM ST95160 antes y después del ajuste del sensor de aceleración

El tamaño del archivo de volcado de la memoria EEPROM ST95160 después de haber realizado el ajuste del sensor de aceleración es de 2048 bytes, mismo tamaño del archivo de volcado antes de haber realizado este proceso como se muestra en la Figura 3.77.

El siguiente paso fue analizar la cantidad y la escritura de datos o bytes que cambian después del ajuste del sensor de aceleración en la memoria EEPROM ST95160.

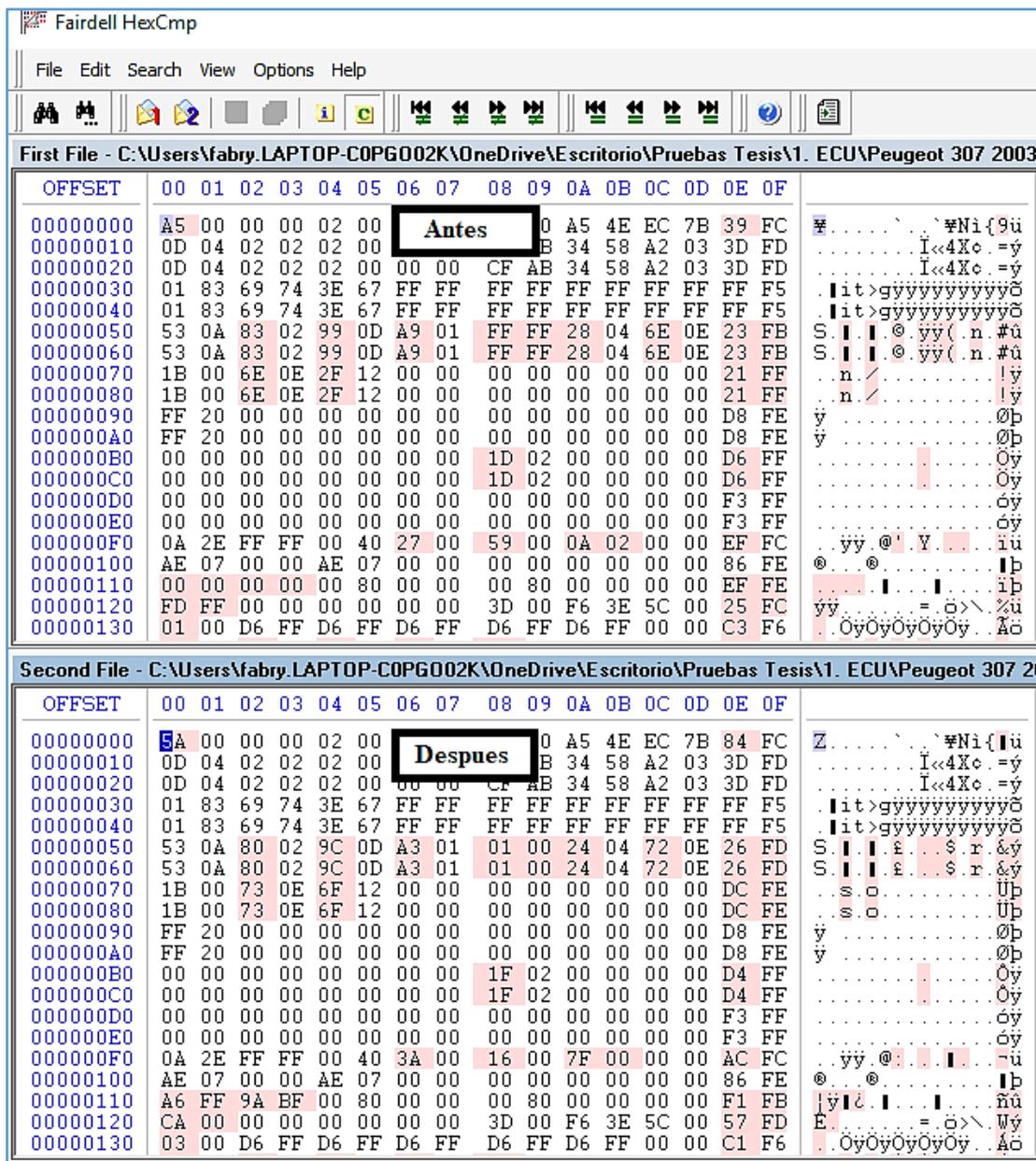


Figura 3.78 Comparación de datos o bytes de la memoria EEPROM ST95160 antes y después del ajuste del sensor de aceleración

Después del ajuste del sensor de aceleración en la memoria EEPROM ST95160 se puede apreciar que existe diferentes cambios de datos o bytes como se observa en la figura 3.78, a continuación, se explica el cambio en cada dirección (fila) y registro (columna) que se ha producido.

En la Tabla 3.14, se indica los datos o bytes que cambiaron en la memoria EEPROM ST95160 después de haber realizado el ajuste del sensor de aceleración, la escritura que se generó en el segundo archivo de volcado es en 29 direcciones, dando como un total de 126 bytes modificados.

**Tabla 3.14** Análisis bytes memoria EEPROM ST95160 antes y después del ajuste del sensor de aceleración

EEPROM ST95160	BYTES ANTES DEL RESET ÁNGULO DE DIRECCIÓN																BYTES DESPUÉS DEL RESET ÁNGULO DE DIRECCIÓN																				
	REGISTROS																																				
DIRECCIÓN	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F					
00	A5														39		5A															84					
50			83		99		A9		FF	FF	28		6E		23	FB			80		9C		A3		01	00	24		72		26	FD					
60			83		99		A9		FF	FF	28		6E		23	FB			80		9C		A3		01	00	24		72		26	FD					
70			6E		2F										21	FF			73		6F									DC	FE						
80			6E		2F										21	FF			73		6F									DC	FE						
B0									1D						D6											1F					D4						
C0									1D						D6											1F					D4						
F0							27		59		0A	02			EF								3A		16		7F	00			AC						
110	00	00	00	00											EF	FE	A6	FF	9A	BF											F1	FB					
120	FD	FF													25	FC	CA	00													57	FD					
130	01														C3		03														C1						
140	3B		3B		3B		ED		ED						60		27		27		27		CE		CE						DA						
160													00	00	EA	FF													F4	FF	F7	FD					
180									FD	FF	94	BD			A8	FC									CA	00	CC	AD			A5	FD					
1A0													ED		02													CE			21						
1BA							5B	00			4A	4A			AE	41							90	25			4B	4B		BA	60						
270	C7	09	1D				24	0D							BB		87	0B	22				E3	13							2F						
2B0	17	1B	01	08											9A		94	0F	0C	19												0D					
2D0					20										34								30									24					
2E0		3C													73			05														61					
330								00	14	17	AB	AB			4C										01	16	15	BF	BE			24					
340					64					00	00	60		30	8D							69										31	E5				
350	00	03	18	15	03										98	FF	05	01	73	63	01											EE	FE				
380	63	40													FD		46	5B															FF				
4FD										28					70																		71				
500						70					4E				F5	FA							43									23	FB				
510										28					8B																			8C			
520					0D	49					2B				1A							0B	43											27			
7F0														A5	DB	FD																			5A	26	FE

En la dirección 00 y en los registros 00, 0E los bytes modificados son de A5, 39 a 5A, 84.

En la dirección 50 y 60 se han modificado 9 bytes en cada una, se presenta un cambio en los registros 02, 04, 06, 08, 09, 0A, 0C, 0E, 0F de 83, 99, A9, FF, FF, 28, 6E, 23, FB a 80, 9C, A3, 01, 00, 24, 72, 26, FD; los bytes modificados son iguales en la dirección 50 y 60.

En las direcciones 70 y 80 se presentan las mismas modificaciones de bytes en los registros 02, 04, 0E, 0F, los datos modificados son de 6E, 2F, 21, FF a 73, 6F, DC, FF.

En la dirección B0 hay modificaciones de bytes en los registros 08, 0E de 1D, D6 a 1F, D4, de igual forma en la dirección C0 hay las mismas modificaciones de bytes que en la dirección B0 y en los mismos registros.

En la dirección F0 hay 5 modificaciones de bytes en los registros 06, 08, 0A, 0B, 0E los datos modificados son de 27, 59, 0A, 02, EF a 3A, 16, 7F, 00, AC.

En la dirección 110 y registros 00, 01, 02, 03, 0E, 0F se modifican los bytes de 00, 00, 00, 00, EF, EF a A6, FF, 9A, BF, F1, FB.

En la dirección 120 hay 4 modificaciones de bytes en las direcciones 00, 01, 0E, 0F de 00, FF, 25, FC a CA, 00, 57, FD.

En la dirección 130 y registros 00, 0E hay bytes modificados de 01, C3 a C3, C1.

En la dirección 140 hay 6 bytes modificados en los registros 00, 02, 04, 06, 08, 0E los datos modificados son de 3B, 3B, 3B, ED, ED, 60 a 27, 27, 27, CE, CE, DA.

En la dirección 160 y registros 0C, 0D, 0E, 0F los bytes modificados son de 00, 00, EA, FF a F4, FF, F7, FD.

En la dirección 180 y registros 08, 09, 0A, 0B, 0E, 0F los bytes modificados son de FD, FF, 94, BD, A8, FC a F4, FF, F7, FD.

En la dirección 1A0 y registros 0C, 0E los bytes modificados son de ED, 02 a CE, 21.

En la dirección 1BA hay 6 bytes modificados en los registros 06, 07, 0A, 0B, 0D, 0E los datos modificados son de B5, 00, 4A, 4A, AE, 41 a 90, 25, 4B, 4B, BA, 60.

En la dirección 270 y registros 00, 01, 02, 06, 07, 0E hay bytes modificados de C7, 09, 1D, 24, 0D, BB a 87, 0B, 22, E3, 13, 2F.

En la dirección 2B0 hay 5 bytes modificados en los registros 00, 01, 02, 03, 0E los datos modificados son de 17, 1B, 01, 08, 9A a 9A, 0F, 0C, 19, 0D.

En la dirección 2D0 y registros 05, 0E los bytes modificados son de 20, 34 a 30, 24.

En la dirección 2E0 y registros 01, 0A, 0E los bytes modificados son de 01, 0A, 0E a 05, B5, 61.

En la dirección 330 y en los registros 07, 08, 09, 0A, 0B, 0C, 0E los bytes modificados son de 00, 14, 17, AB, AB, 4C a 01, 16, 15, BF, BE, 24.

En la dirección 340 hay 6 bytes modificados en los registros 04, 09, 0A, 0B, 0D, 0E los datos modificados son de 64, 00, 00, 60, 30, 8D a 69, 01, 01, 00, 31, E5.

Los bytes modificados en la dirección 350 y registros 00, 01, 02, 03, 04, 0E, 0F son de 00, 03, 18, 15, 03, 98, FF a 05, 01, 73, 63, 01, EE, FE.

En la dirección 380 y registros 00, 01, 0E los bytes modificados son de 63, 40, FD a 45, 5B, FF.

En la dirección 4FD y registros 09, 0E los bytes modificados son 28, 70 a 27, 71.

En la dirección 500 hay 4 bytes modificados en los registros 05, 0A, 0E, 0F los datos modificados son de 70, 4E, F5, FA a 43, 4D, 23, FB.

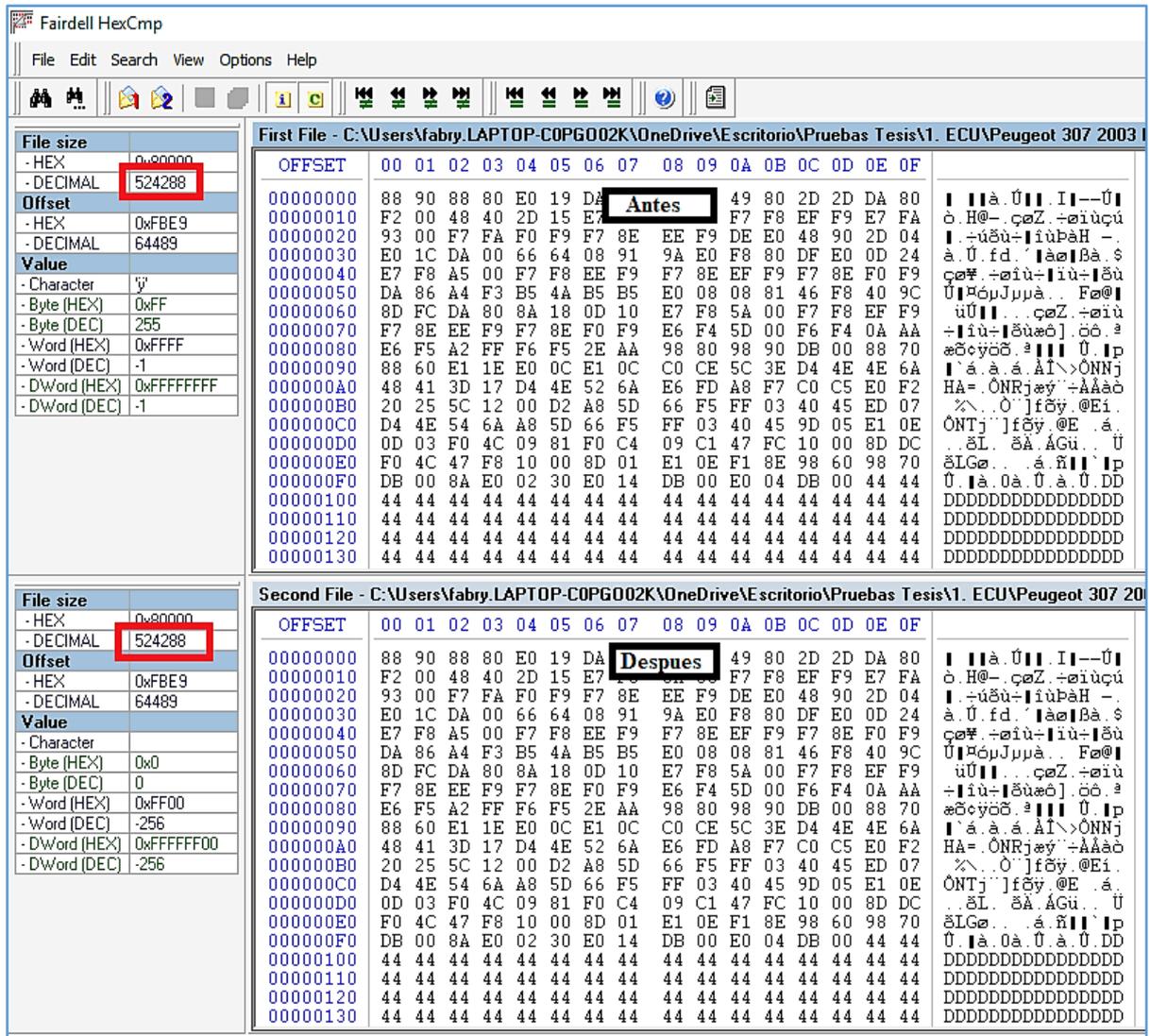
En la dirección 510 y registros 09, 0E los bytes modificados son de 28, 8B a 27, 8C.

En la dirección 520 hay 4 bytes modificados en los registros 04, 05, 0A, 0E los datos modificados son de 0D, 49, 2B, 1A, a 0B, 43, 27, 27.

Y en la dirección 7F0 y en los registros 0D, 0E, 0F los bytes que se han modificado son de A5, DB, FD a 5A, 26, FE.

### 3.4.2.5 Análisis memoria Flash AM29F400BB Mapas después del ajuste del sensor de aceleración

Para el análisis de la memoria Flash se procedió a verificar que el tamaño del archivo de volcado de la memoria obtenido después del ajuste del sensor de aceleración, sea igual al tamaño del archivo de volcado obtenido antes del reseteo.



**Figura 3.79** Comparación archivo de volcado de la memoria Flash AM29F400BB Mapas antes y después del ajuste del sensor de aceleración

El tamaño del archivo de volcado de la memoria Flash AM29F400BB Mapas correspondiente a la lectura después ajuste del sensor de aceleración es de 524 288 bytes similar al tamaño del archivo de volcado de la misma memoria antes de este proceso como se puede observar en la Figura 3.79.



El tamaño del archivo de volcado de la memoria Micro ST10C167 correspondiente a la lectura después del ajuste del sensor de aceleración es de 32768 bytes, con esto se determinó que el tamaño del archivo de volcado antes de realizar el reseteo de esta función tiene los mismos bytes como se indica en la Figura 3.80.

Con los resultados obtenidos, la lectura de la memoria Micro ST10C167 después de haber realizado el ajuste del sensor de aceleración no presenta ninguna modificación en los bytes.

### 3.4.2.7 Análisis de archivos entre Reset luz de servicio y ajuste del sensor de aceleración

Es necesario realizar una comparación entre los valores hexadecimales después de cada Reset de funciones especiales realizadas al vehículo Peugeot 307, con esto se determinó que se tiene una similitud en lo referente a los bytes modificados. Para el presente análisis se utilizará los datos de las tablas 3.13 y 3.14, los bytes a analizar se encuentran en las direcciones 00 y 7F0 y en los registros 00, 0D, 0E, 0F.

En la Tabla 3.15, se describe el análisis realizado entre los archivos de volcado antes mencionados.

**Tabla 3.15** Registros cíclicos

REGISTROS												
	Antes del Reset luz de servicio				Después del Reset luz de servicio/ Antes del ajuste del sensor de aceleración				Después del ajuste del sensor de aceleración			
Dirección	00	0D	0E	0F	00	0D	0E	0F	00	0D	0E	0F
<b>00</b>	5A		84		A5		39		5A		39	
<b>7F0</b>		5A	26	FE		A5	DB	FD		A5	DB	FD

Para la entrada de diagnóstico o reseteo de funciones especiales, entre el equipo de programación y la memoria de la ECU se tiene una comunicación de ciclos, antes de realizar el Reset de la luz de servicio se tiene en la dirección 00 y registros 00, 0E los siguientes bytes 5A, 84 y en la dirección 7F0 y registros 0D, 0E, 0F los siguientes bytes 5A, 26, FE y después de realizar el proceso de reseteo de esta función se tiene un cambio de datos, en la dirección 00 registros 00, 0E se tiene los datos 05, 39 y en la dirección 7F0 registros 0D, 0E, 0F se tiene los datos A5, DB, FD.

Después de realizar el ajuste del sensor del acelerador los bytes obtenidos, son los mismos que se obtuvo en el proceso del reseteo de la luz de servicio; en la dirección 00 y registros 00, 0E se tiene los bytes 5A, 84, por otro lado, en la dirección 7F0 registros 0D, 0E, 0F se tiene los bytes A5, DB, FD.

Los bytes que se muestran en la tabla 3.15 de antes y después de cada reseteo de funciones especiales nos indica la comunicación en hexadecimales que tiene el equipo de diagnóstico y la memoria EEPROM ST95160 de la ECU Bosch ME 7.4.4 del automóvil.

Los bytes de comunicación son cíclicos, es decir el equipo cuándo va a realizar un reseteo o ajuste de una función, para abrir la comunicación o indicar que va a realizar una modificación en el archivo de volcado escribe los bytes A5, 39 y para cerra la comunicación o indicar que el trabajo se ha realizado escribe los siguientes bytes A5, DB, F9.

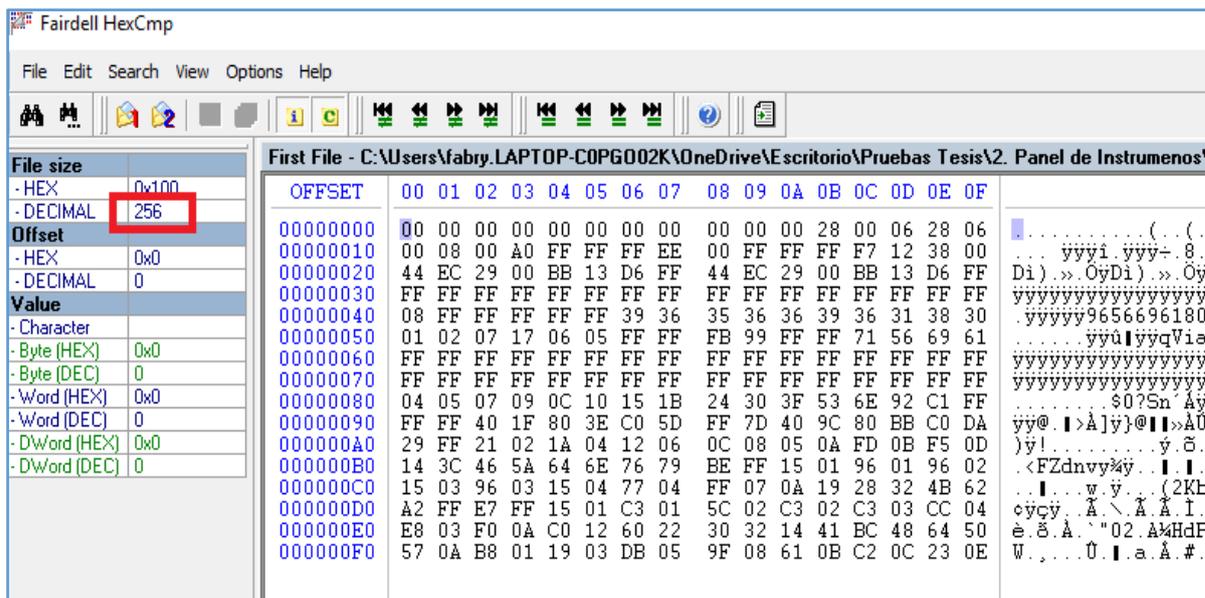
Después cuando el equipo hace otra modificación para abrir la comunicación o indicar que quiere realizar un reseteo escribe los bytes 5A, 39 y para cerrar la comunicación o indicar que el proceso está completado escribe los siguientes bytes A5, DB, FD.

### **3.4.3 ANÁLISIS ARCHIVO DE VOLCADO PANEL DE INSTRUMENTOS**

En este apartado se explica todo lo referente al análisis de los archivos extraídos de la memoria EEPROM 25C020 del panel de instrumentos. En el proceso de pruebas se realizó diferentes lecturas, la mismas que sirven para determinar la dirección y el registro de donde se encuentra escrito el kilometraje.

#### **3.4.3.1 Tamaño de la memoria EEPROM 25C020**

Para determinar el tamaño de cada archivo de volcado obtenido de la memoria EEPROM 25C020 se utilizó el programa FAIRDELL HEX CMP.



**Figura 3.81** Tamaño archivo de volcado panel de instrumentos

El tamaño del del archivo de volcado referente a la primera lectura realizada en el panel de instrumentos es de 256 bytes como se puede apreciar en la Figura 3.81.

### 3.4.3.2 Análisis de lecturas realizadas memoria EEPROM 25C020

Para poder entender de mejor manera el algoritmo en el cálculo del kilometraje, se realizó un total de 6 lecturas a la memoria del panel electrónico de control.

Nombre	Fecha de modificación	Tipo	Tamaño
Lectura 1 km 274746	23/06/2020 10:32 a. m.	Archivo	1 KB
Lectura 2 km 274 825	2/07/2020 11:41 a. m.	Archivo	1 KB
Lectura 3 km 274 880	7/07/2020 3:24 p. m.	Archivo	1 KB
Lectura 4 km 274 940	12/07/2020 9:30 a. m.	Archivo	1 KB
Lectura 5 km 275 103	22/07/2020 10:34 a. m.	Archivo	1 KB
Lectura 6 km 275 112	23/07/2020 5:36 p. m.	Archivo	1 KB

**Figura 3.82** lecturas realizadas memoria EEPROM 25C020

Cada lectura realizada fue en diferentes kilómetros recorridos por el automóvil como se indica en la Figura 3.82. Los archivos de volcado obtenidos de las lecturas efectuadas al panel de instrumentos se las realizo cuando el marcador del kilometraje estaba en 274 746 km, 274 825 km, 274 88 km, 274 940 km, 275 103 km, 275 112 km.

### 3.4.3.3 Análisis de datos o bytes

Entre la lectura 1 y la lectura 2 se realizó un análisis del tamaño de los bytes que cambian entre cada archivo, siendo estos parámetros de gran importancia para poder determinar que las lecturas están correctamente y poder localizar la ubicación del kilometraje dentro del archivo de volcado.

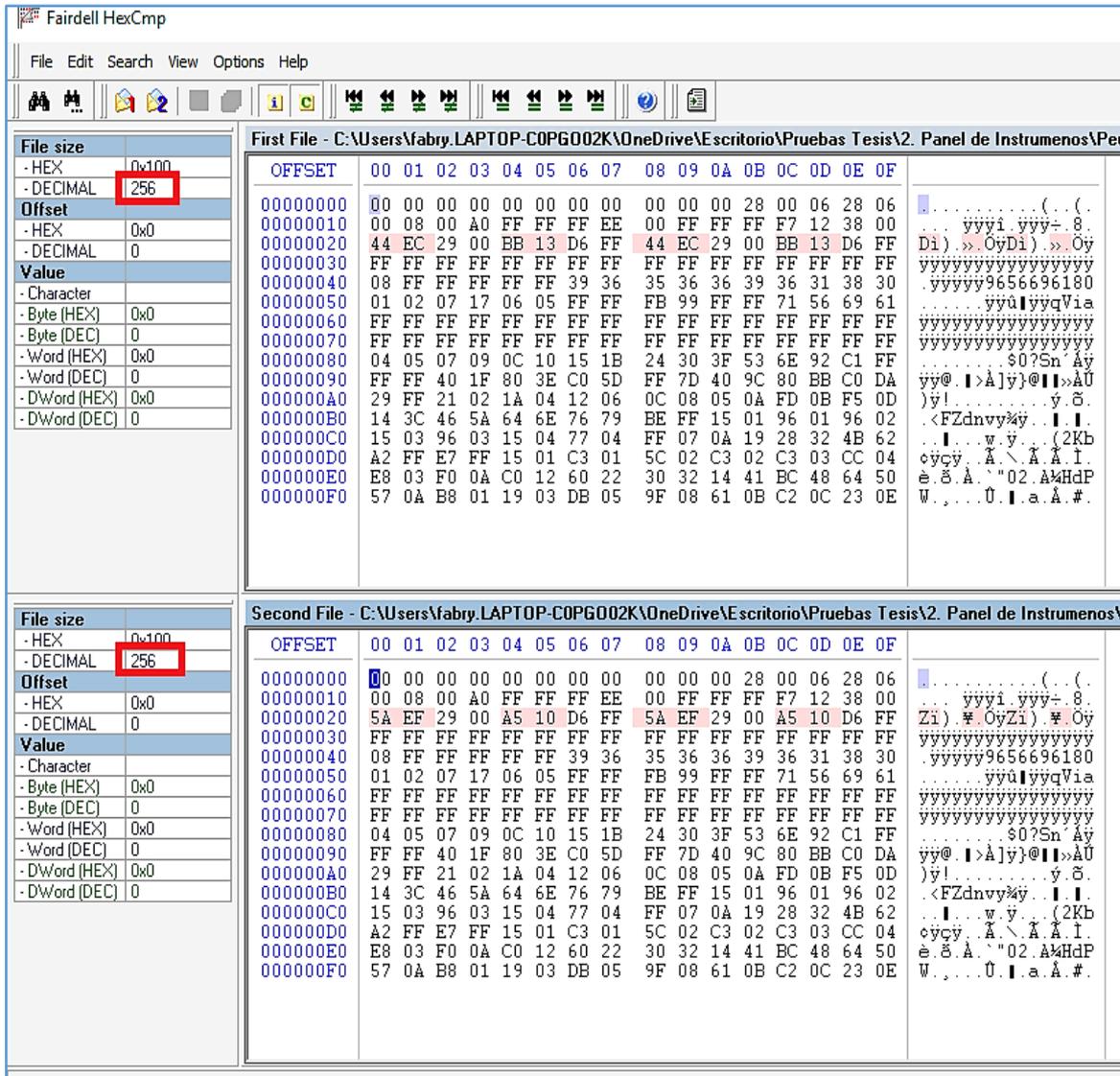


Figura 3.83 Lectura de archivos de 274 746 km y 274 825 km

En la Figura 3.81, se puede observar que el tamaño de los dos archivos es de 256 bytes, con este análisis verificamos que las lecturas realizadas son correctas, después de haber recorrido ciertas distancias el vehículo no existe una alteración en el tamaño de la memoria del panel de instrumentos.

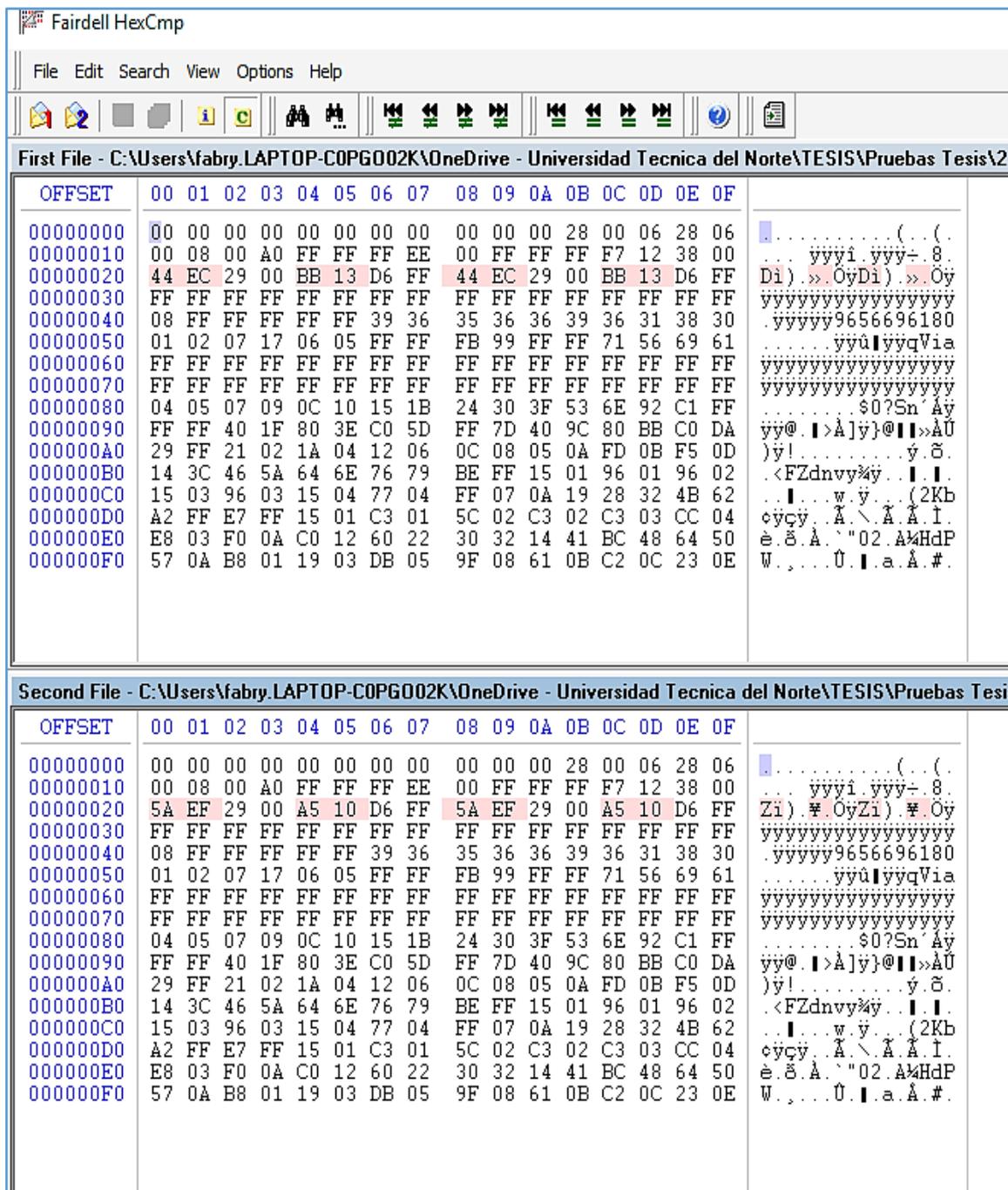
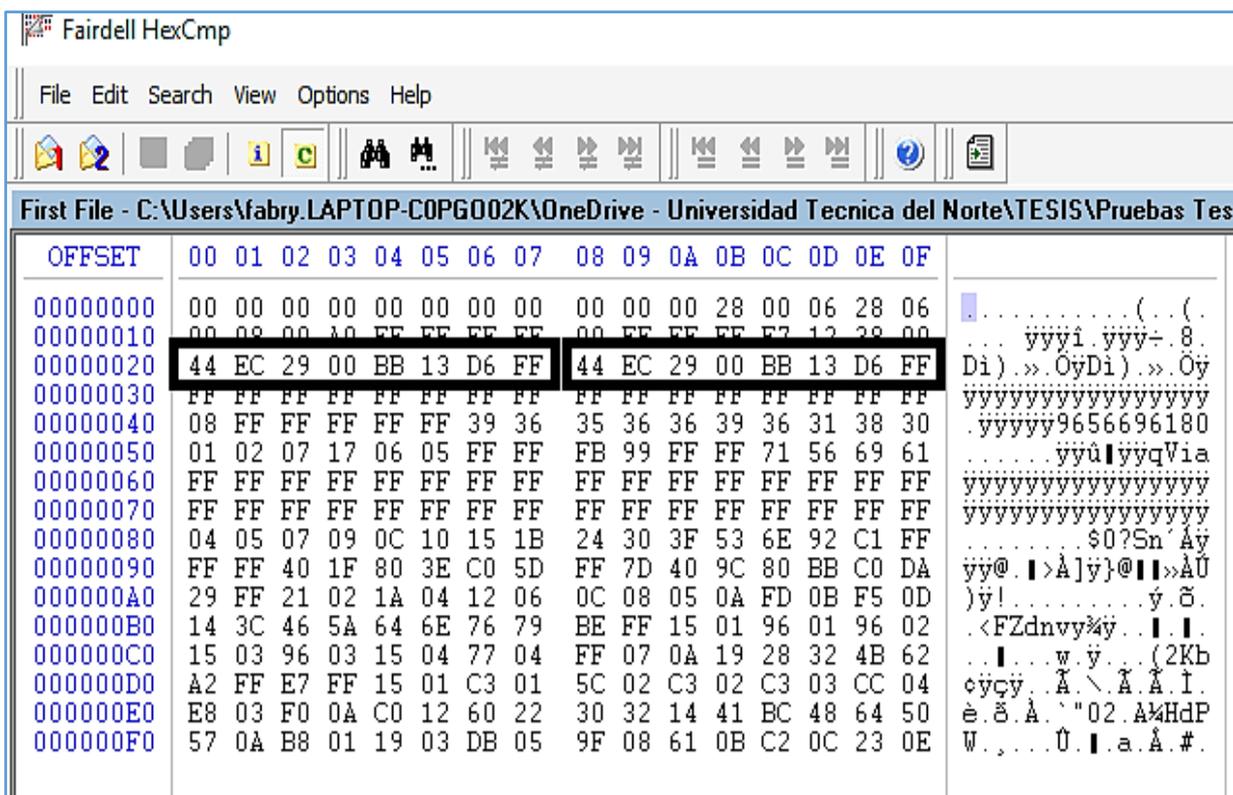


Figura 3.84 Comparación entre archivos

En las lecturas realizadas de 274 746 km y 274 825 km hay un cambio de bytes en la dirección (fila) 20 y registros (columnas) 00, 01, 04, 05, 08, 09, BB, 13 como se indica en la Figura 3.84. Se pudo deducir que el cambio en los datos de la dirección y registros son correspondientes al kilometraje recorrido que tiene el automóvil, cabe recalcar que entre cada lectura realizada a la memoria del panel de instrumentos solo hubo variación de datos y no tamaño.



**Figura 3.85** Representación del kilometraje

En la comparación los entre archivos obtenidos se pudo determinar la ubicación del kilometraje en valores hexadecimales. Además, se verifico que está representado por 2 grupos de 8 bytes. Los datos del primer grupo están ubicados en la dirección 20 y registros 00, 01, 02, 03, 04, 05, 06, 07 y los datos del segundo grupo se encuentran en la dirección 20 y registros 08, 09, 0A, 0B, 0C, 0D, OE, OF como se puede observar en la Figura 3.85.

#### 3.4.3.4 Análisis del algoritmo de cálculo de kilometraje

Con las 6 lecturas realizadas a la memoria del panel de instrumentos se obtuvo datos suficientes para deducir el algoritmo del kilometraje, en el análisis se utilizó el primer grupo de 8 bytes de los archivos obtenidos.

En la Tabla 3.16, están representados los datos en valores hexadecimales del primer grupo de 8 bytes de cada lectura de la memoria EEPROM 25C020. Los datos utilizados para el análisis del cálculo del algoritmo del kilometraje son de la dirección 20 y registros 00, 01, 02, 03, 04, 05, 06, 07.

**Tabla 3.16** Datos del kilometraje en valores hexadecimales

LECTURA	Km	00	01	02	03	04	05	06	07
1	274 746	44	EC	29	00	BB	13	D6	FF
2	274 825	5A	EF	29	00	A5	10	D6	FF
3	274 880	80	F1	29	00	7F	0E	D6	FF
4	274 940	D8	F3	29	00	27	0C	D6	FF
5	275 103	36	FA	29	00	C9	05	D6	FF
6	275 112	90	FA	29	00	6F	05	D6	FF

Con los resultados obtenidos, se deduce que para el kilómetro 0, el grupo de 8 bytes tendrá los siguientes datos en hexadecimales 00, 00, 00, 00, FF, FF, FF, FF y en decimales viene a ser 00, 00, 00, 00, 255, 255, 255, 255, en los registros que tiene como escritura los datos hexadecimales 00 y FF están en blanco o no están escritos.

En la Tabla 3.17, están representados los datos en decimales del primer grupo de 8 bytes de cada lectura obtenida. Se opta en transformar los datos en decimales por su fácil interpretación y familiaridad. Para la transformación de los datos de hexadecimal a decimal se utiliza una calculadora tipo programador la misma que viene en el computador.

**Tabla 3.17** Datos de kilometraje en valores decimales

LECTURA	Km	00	01	02	03	04	05	06	07
1	274 746	068	236	041	000	187	019	214	255
2	274 825	090	239	041	000	165	016	214	255
3	274 880	128	241	041	000	127	014	214	255
4	274 940	216	243	041	000	039	012	214	255
5	275 103	054	250	041	000	201	005	214	255
6	275 112	144	250	041	000	111	005	214	255

Con los datos obtenidos se analizó el algoritmo que utiliza la columna 00 y la columna 04, en la lectura 5 y lectura 6 se tiene 275 103 km y 275 112 km respectivamente, además para estos kilómetros en la columna 00 se tiene los datos 054, 144 en decimal y en la columna 04 se tiene los datos 201 y 111 en decimal.

A continuación, se obtiene la diferencia entre el kilómetro de la lectura 5 y 6:

$$(275\ 112 - 275\ 103)km = 9\ km$$

La diferencia entre el kilómetro 275 112 y 275 103 es de 9 kilómetros, ahora se obtiene la diferencia entre los datos de la lectura 5, 6 y columnas 00, 04 de la tabla 3.17.

$$\text{Diferencia lectura 5 y 6 columna 00} \quad 144 - 54 = 90$$

$$\text{Diferencia lectura 5 y 6 columna 04} \quad 201 - 111 = 90$$

La diferencia de los datos de la columna 00 y la columna 04 de la lectura 5 y 6 en valores decimales es de 90, a continuación, se realiza una división entre la diferencia del kilometraje y la diferencia de los decimales.

$$90 \div 9 = 10$$

Por cada kilómetro recorrido se suma o se resta 10 decimales dependiendo el caso.

Ejemplo:

El kilómetro 275 103 de la lectura 5 está representado del siguiente grupo de decimales:

**054-250-041-000-201-005-214-255**

Como se sabe que para cada kilómetro de recorrido se suma o se resta de 10 en 10 decimales, podemos calcular el kilómetro 275 103.

$$54 + 10 = 64 \text{ y } 201 - 10 = 191$$

Entonces el kilómetro 275 104 estará representado por el siguiente grupo de decimales:

**064-250-041-000-191-005-214-255**

Además, en la columna 00 y 04 de la tabla 2 tenemos los siguientes datos en decimales. En la columna 00 hay los siguientes datos 068, 090, 128, 216, 054, 144 y en la columna 04 hay los siguientes datos 187, 165, 127, 039, 201, 111.

Con estos datos se puede deducir que para la columna 00 se tiene los siguientes números, 00, 02, 04, 06, 08 y en la columna 04 serían los siguientes números 255, 253, 251, 249, 247, esto es dependiendo el caso.

Cómo se mencionó anteriormente el kilómetro 0 está representado por el siguiente grupo de números decimales 00 00 00 00 255 255 255 255 para entender de mejor manera se detalla una tabla a continuación.

En la Tabla 3.18, se tiene un ejemplo de cómo cambian los valores de la columna 00 y la columna 04, en la columna 00 por cada kilómetro que recorre el valor aumenta de 10 en 10 hasta aproximarse a 255 y en la columna 04 los valores disminuyen de 10 en 10 hasta aproximarse a cero.

**Tabla 3.18** Ejemplo de inicio de kilometraje

Km	DATOS							
	00	01	02	03	04	05	06	07
<b>0</b>	00	00	00	00	255	255	255	255
<b>1</b>	10	00	00	00	245	255	255	255
<b>2</b>	20	00	00	00	235	255	255	255

En la columna 01 y en la columna 05 los valores van aumentando o disminuyendo de uno en uno respectivamente, cuando ya ha recorrido un aproximado de 25 kilómetros los valores de la columna 00 se aproximen a 255 y los valores de la columna 04 se aproximen a cero, entonces en la columna 01 el dato 00 cambia a 01 y el dato de la columna 05 disminuye de 255 a 244.

Ejemplo:

Km 25 en decimal: 250 **000** 000 000 005 **255** 255 255

Km 26 en decimal 004 **001** 000 000 251 **244** 255 255

Como se puede observar en el ejemplo anterior después de recorrer un cierto número de kilómetros el valor de la columna 00 se aproxima a 255 y el valor de la columna 04 se aproxima a cero, entonces se cumple un ciclo y cambia el valor de la columna 01 y el valor de la columna 05.

En Tabla 3.19, está representado el algoritmo del kilometraje en decimales de 0 km a 255 km, todos los datos representados son cíclicos, por ende se puede calcular diferentes valores para kilometraje en decimal, en la fila de kilómetros están los ciclos de kilometraje de 0-25 de 26-51 entre otros, en la fila de km recorridos indica la cantidad de kilómetros que recorre en cada ciclo, por ejemplo de 0 a 25 se recorre 25 km, de 26 a 51 se recorre 25 km, en la fila de valores de la columna 00 y los valores de la columna 04 están los números que se utilizaran para cada ciclo de kilometraje, y en las filas los valores de la columna 01 y los valores de la columna 05 se tiene los valores correspondientes a la columna 01 y 05, ejemplo

si se quiere representar el kilómetro 77 en decimales seria, 002 003 000 000 253 253 255 255 y si se quiere representar el kilómetro 78 se debe de sumar 10 a 003 y se debe de restar 10 a 253 y como aún no se cumple el ciclo el valor 003 y el segundo valor de 253 se mantienen.

**Tabla 3.19** Algoritmo del cálculo de kilometraje

<b>KILÓMETROS</b>	<b>0-25</b>	<b>26-51</b>	<b>52-76</b>	<b>77-102</b>	<b>103-127</b>	<b>128-153</b>	<b>154-179</b>	<b>180-204</b>	<b>205-230</b>	<b>231-255</b>
km recorridos	25	25	24	25	24	25	25	24	25	24
Valores columna 00	000	004	008	002	006	000	004	008	002	006
Valores columna 04	255	251	247	253	249	255	251	247	253	249
Valores columna 01	000	001	002	003	004	005	006	007	008	009
Valores columna 05	255	254	253	252	251	250	249	248	247	246

Todos los valores utilizados para realizar el análisis del algoritmo del cálculo del kilometraje están representados en decimales, después de haber realizado el cálculo de cualquier kilometraje se debe pasar a hexadecimales debido a que en un archivo de volcado los datos se encuentra representados de esta manera.

### **3.4.4 ANÁLISIS ARCHIVO DE VOLCADO BSI**

En este apartado se procede a explicar lo referente al análisis de los archivos extraídos de la caja de servicios inteligentes o BSI. En el proceso de pruebas se realizó diferentes lecturas, las mismas que sirven para determinar las diferentes eventualidades que sucede dentro de dicho módulo.

#### **3.4.4.1 Verificación de lectura archivo de volcado BSI**

Para verificar una lectura correcta del archivo de volcado del módulo BSI, el método que se utilizo fue la identificación del PIN CODE y numero VIN del vehículo encontrado en la matricula del propietario. Estos valores se encuentran en valores alfanuméricos y deben ser transformados a valores hexadecimales.

Como se mencionó anteriormente, en la tabla 3.11 se encuentra un sistema de transformación de números y letras a códigos hexadecimales, datos necesarios para verificar que existe una lectura correcta del archivo.

Los valores hexadecimales del PIN CODE se encontró en apartados anteriores y se muestra en la tabla 3.12, por lo tanto, se prosiguió con el análisis en el DUMP de la unidad BSI.

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000550	04	1B	7A	07	8A	82	31	52	84	1B	8B	07	84	86	31	54
00000560	04	1B	8A	07	84	86	31	54	84	1B	05	00	84	FF	31	55
00000570	04	1B	05	00	72	FF	30	E6	84	1B	06	00	71	FF	31	55
00000580	04	1B	8B	07	69	36	31	55	84	1B	84	01	69	FF	31	55
00000590	04	1B	8A	27	69	4B	31	55	84	1B	06	00	69	FF	31	55
000005A0	04	1B	06	02	81	40	31	55	84	1B	8C	07	81	40	31	55
000005B0	04	1B	8B	27	82	55	31	55	84	1B	87	07	7F	7D	31	57
000005C0	04	1B	8A	07	7F	7D	31	57	84	1B	8A	07	7F	7B	31	57
000005D0	04	1B	8B	07	80	7B	31	57	84	1B	89	07	7F	7D	31	57
000005E0	04	1B	89	07	7F	7E	31	57	84	1B	05	00	86	FF	31	5D
000005F0	04	1B	7A	03	8E	82	31	5D	84	1B	05	00	84	FF	31	62
00000600	1F	05	05	01	2F	01	02	08	2C	02	26	01	00	B9	FF	FF
00000610	FF	3C	CD	ED	6B	66	0B	FF	FF							
00000620	FF	B9	CC													
00000630	A8	C6	B9	A0	B0	B4	EE	EE	EE	EE	00	00	00	00	00	00
00000640	00	00	00	00	00	00	FF	E1	F9	FF	FE	FE	F1	B4	E7	AF
00000650	87	69	FA	FF	F8	FF	FF	EB	FE	FA	97	FF	DE	DB	FF	A9
00000660	B9	CC	CC	BC	B1	C9	BE	AF	CC	A6	CF	CF	C8	CA	C8	C8
00000670	77	AF	DD	29	AF	FF	FF	FF	3C	DC	FF	D6	0E	7F	3C	66
00000680	FF	D6	0E	7F	3C	86	FF	D6	0E	7F	C3	FE	FF	D6	0E	7F
00000690	C3	DE	FF	D6	0E	7F	C3	FE	FF	D6	0E	7F	C3	FF	FF	D6
000006A0	0E	7F	C3	DF	FF	D6	0E	7F	C3	FF	FA	05	FD	C8	FA	05
000006B0	FD	C8	FF	E1	1B	42	FF	E1	1B	42	FF	45	59	1E	FF	45
000006C0	59	1E	FF	FF	F8	2F	FF	FD	5D	AB	FF	D6	13	B8	FF	FF
000006D0	FF	FF	FF	FD	85	3B	FF	FB	45							
000006E0	6A	1D	FF	2E	FD	18	FD	30	FF	FB	FE	34	FF	97	FF	A8
000006F0	FA	EB	FF	F8	FF	FD	FF	FF	DF							
00000700	DD	7B	C3	A6	FA	05	FE	DB	FA	05	FE	DB	FF	E1	1B	7F
00000710	FF	E1	1B	7F	FF	45	5A	55	FF	45	5A	55	FF	FF	F8	2F
00000720	FF	FD	5D	AB	FF	D6	13	B8	FF	FF	FF	FF	FF	FD	38	92

Figura 3.86 Ubicación Pin Code-archivo de volcado BSI

En la Figura 3.86, se indica la ubicación del grupo de 4 bytes pertenecientes al Pin Code, este grupo está en el archivo de volcado de la lectura al microcontrolador MC68HC912DG128 del módulo de control BSI, el Pin Code se encuentra en la dirección 620 registros 0E, 0F y dirección 630 registros 00, 01.

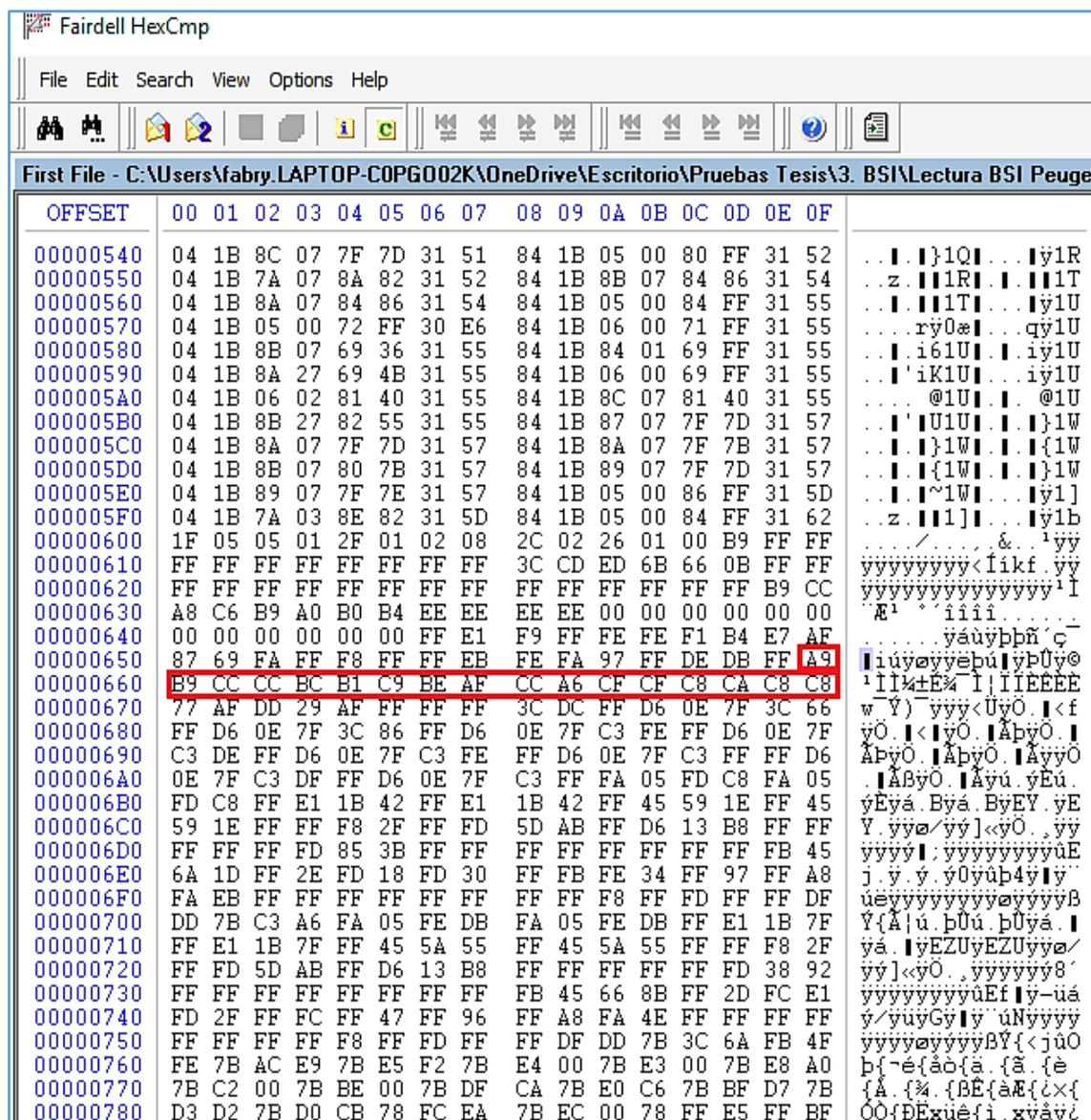
Para la identificación del número VIN del vehículo en el archivo de volcado de la BSI, en primer lugar, se transformó el código de 17 dígitos expresado en valores alfanuméricos a valores hexadecimales.

En la Tabla 3.20, se encuentra el número VIN transformado a valores hexadecimales, para este análisis de cambio de dígitos es necesario el uso de la tabla 3.11.

**Tabla 3.20** Número VIN de alfanumérico a hexadecimales

Número VIN de alfanumérico a hexadecimales																	
Número VIN	V	F	3	3	C	N	6	A	P	3	Y	0	0	7	5	7	7
Número VIN en hexadecimales	A9	B9	CC	CC	BC	B1	C9	BE	AF	CC	A6	CF	CF	C8	CA	C8	C8

Una vez obtenido el número VIN en códigos hexadecimales, se prosiguió a determinar su ubicación en el DUMP del archivo de volcado de la unidad BSI.



**Figura 3.87** Ubicación del número VIN en el archivo de volcado BSI

La ubicación del número VIN en el archivo de volcado extraído se encuentra en la dirección 650 registro 0F y en la dirección 660 registros 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F como se puede observar en la Figura 3.87.

### 3.4.4.2 Tamaño del archivo de volcado BSI

En este punto con la ayuda del programa FAIRDELL HEXCMP se procedió con el análisis del tamaño del archivo de volcado del microcontrolador MC68HC912DG128 de la unidad electrónica BSI.

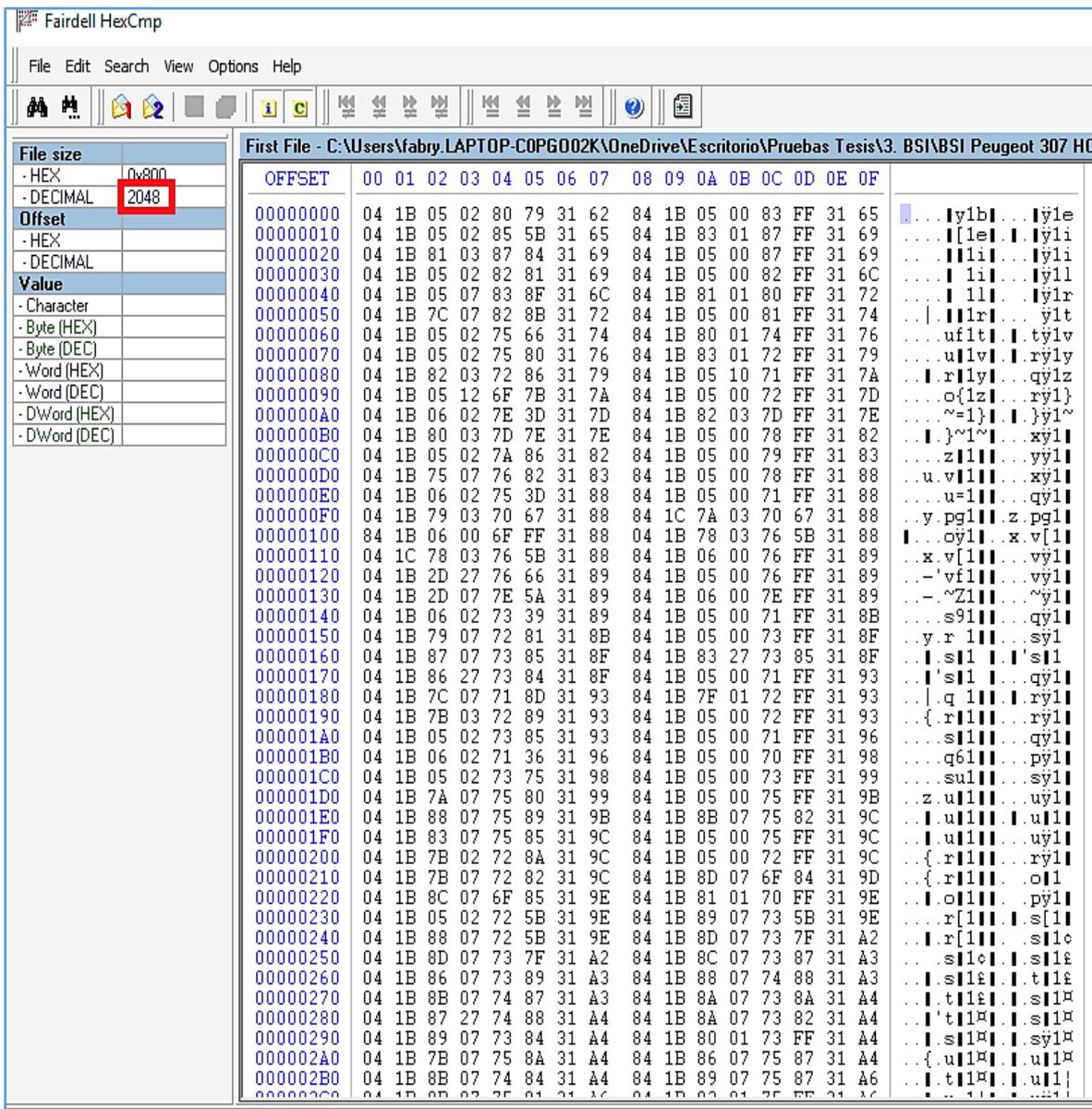


Figura 3.88 Tamaño del archivo de volcado BSI

Con los resultados obtenidos se puede determinar que el tamaño del archivo de volcado del microcontrolador del módulo de la carrocería es de 2048 Bytes como se aprecia en la Figura 3.88.

### 3.4.4 COMPARACIÓN DE ARCHIVOS DE VOLCADO BSI

Para la comparación de los archivos de volcado del módulo de control de la carrocería se analizó las lecturas realizadas en el método práctico.

Nombre	Fecha de modifica...	Tipo
 BSI Lectura 1 Km 274880	7/07/2020 11:12 a. ...	Archivo
 BSI Peugeot Antes de Proga Llaves	13/08/2020 10:22 a...	Archivo
 BSI Peugeot Despues de Prog Llaves	13/08/2020 12:04 ...	Archivo

**Figura 3.89** Lecturas realizadas módulo BSI

La primera lectura se la realizo cuando él cuenta kilómetros del vehículo marcaba 274 880 km, la segunda y tercera lectura se las realizo antes y después de realizar una programación de llaves como se puede ver en la Figura 3.89.

Cabe mencionar que, cuando se realizó las dos últimas lecturas él cuenta kilómetros marcaba 275 423 km.

#### 3.4.4.1 Análisis Archivo de kilometraje

Se realizo un análisis al primer archivo de volcado extraído del microcontrolador de la BSI, aquí se procede determinar la ubicación de la escritura del kilometraje en el archivo de volcado, esta lectura se la realizo cuando él cuenta kilómetros marco 274 880 para este dato el valor correspondiente en hexadecimales es 80 F1 29 00 7F 0E D6 FF el mismo que se lo obtuvo de la tabla 3.16.

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000550	04	1B	7A	07	8A	82	31	52	84	1B	8B	07	84	86	31	54	..z.!!!1R!..!!!1T
00000560	04	1B	8A	07	84	86	31	54	84	1B	05	00	84	FF	31	55	..!..!!!1T!...!y!U
00000570	04	1B	05	00	72	FF	30	E6	84	1B	06	00	71	FF	31	55	...ry0æ!...qy!U
00000580	04	1B	8B	07	69	36	31	55	84	1B	84	01	69	FF	31	55	...i6!U!..!iy!U
00000590	04	1B	8A	27	69	4B	31	55	84	1B	06	00	69	FF	31	55	...!iK!U!..!iy!U
000005A0	04	1B	06	02	81	40	31	55	84	1B	8C	07	81	40	31	55	... @!U!.. @!U
000005B0	04	1B	8B	27	82	55	31	55	84	1B	87	07	7F	7D	31	57	...! !U!U!..! !}!W
000005C0	04	1B	8A	07	7F	7D	31	57	84	1B	8A	07	7F	7B	31	57	...! !}!W!..! !}!W
000005D0	04	1B	8B	07	80	7B	31	57	84	1B	89	07	7F	7D	31	57	...! !}!W!..! !}!W
000005E0	04	1B	89	07	7F	7E	31	57	84	1B	05	00	86	FF	31	5D	...! !}!W!..! !}!W
000005F0	04	1B	7A	03	8E	82	31	5D	84	1B	05	00	84	FF	31	62	..z.!!!1}!...!y!b
00000600	1F	05	05	01	2F	01	02	08	2C	02	26	01	00	B9	FF	FF	.../...&...'yy
00000610	FF	3C	CD	ED	6B	66	0B	FF	FF	yyyyyyyy<likf.yy							
00000620	FF	B9	CC	yyyyyyyyyyyyyyyy!i													
00000630	A8	C6	B9	A0	B0	B4	EE	EE	EE	EE	00	00	00	00	00	00	"Æ! "iiii.....
00000640	00	00	00	00	00	00	FF	E1	F9	FF	FE	FE	F1	B4	E7	AF	.....yáuybbñ'ç-
00000650	87	69	FA	FF	F8	FF	FF	EB	FE	FA	97	FF	DE	DB	FF	A9	!iúyøyyébú!ypÛy@
00000660	B9	CC	CC	BC	B1	C9	BE	AF	CC	A6	CF	CF	C8	CA	C8	C8	'!i!±E% ! !IEEEE
00000670	77	AF	DD	29	AF	FF	FF	FF	3C	DC	FF	D6	0E	7F	3C	66	w'Y)-yyy<Ûy0.  <f
00000680	FF	D6	0E	7F	3C	86	FF	D6	0E	7F	C3	FE	FF	D6	0E	7F	y0.  < y0.  Ápy0.
00000690	C3	DE	FF	D6	0E	7F	C3	FE	FF	D6	0E	7F	C3	FF	FF	D6	Ápy0.  Ápy0.  Áyy0
000006A0	0E	7F	C3	DF	FF	D6	0E	7F	C3	FF	FA	05	FD	C8	FA	05	.. ÁBy0.  Áyú.yEú.
000006B0	FD	C8	FF	E1	1B	42	FF	E1	1B	42	FF	45	59	1E	FF	45	yEya. Byá. ByEY. yE
000006C0	59	1E	FF	FF	F8	2F	FF	FD	5D	AB	FF	D6	13	B8	FF	FF	Y.yyø/yý <y0. .yy
000006D0	FF	FF	FF	FD	85	3B	FF	FD	FF	FF	FF	FF	FF	FF	FB	45	yyyy! .yyyyyyyyúE
000006E0	6A	1D	FF	2E	FD	18	FD	30	FF	FB	FE	34	FF	97	FF	A8	j.y.y y0yðb4y y'
000006F0	FA	EB	FF	F8	FF	FD	FF	FF	DF	úeyyyyyyyyyyøyyyyB							
00000700	DD	7B	C3	A6	FA	05	FE	DB	FA	05	FE	DB	FF	E1	1B	7F	Y{A}ú. pÙú. pÙyá.
00000710	FF	E1	1B	7F	FF	45	5A	55	FF	45	5A	55	FF	FF	F8	2F	yá.  yEZUyEZUyyø/
00000720	FF	FD	5D	AB	FF	D6	13	B8	FF	FF	FF	FF	FF	FD	38	92	yý <y0. .yyyyy8'
00000730	FF	FB	45	66	8B	FF	2D	FC	E1	yyyyyyyyúE y-úa							
00000740	FD	2F	FF	FC	FF	47	FF	96	FF	A8	FA	4E	FF	FF	FF	FF	y/yúyGy y' úNyyyy
00000750	FF	FF	FF	FF	F8	FF	FD	FF	FF	DF	DD	7B	3C	6A	FB	4F	yyyyyøyyýBÛ{<jú0
00000760	FE	7B	AC	E9	7B	E5	F2	7B	E4	00	7B	E3	00	7B	E8	A0	p{-é{âò{ã. {ã. {è
00000770	7B	C2	00	7B	BE	00	7B	DF	CA	7B	E0	C6	7B	BF	D7	7B	{A. {%. {BÉ{âÆ{çx{
00000780	D3	D2	7B	D0	CB	78	FC	EA	7B	EC	00	78	FF	E5	FF	BF	00{DÉxúè{i.xyâyç
00000790	FF	F7	FF	FF	FF	FF	FF	F7	FF	y-yyyy-y-yyyyyy							
000007A0	FF	FB	C3	1F	FF	FF	FF	FF	yyyyyyyyyyúA.yyyy								
000007B0	FF	yyyyyyyyyyyyyyyy															
000007C0	2E	D3	02	39	01	BB	01	BD	04	06	00	00	24	30	30	39	.0.9.»%...\$009
000007D0	36	31	30	37	36	30	35	32	35	30	01	FF	FF	FF	FF	FF	6107605250.yyyyyy
000007E0	FF	yyyyyyyyyyyyyyyy															
000007F0	13	09	05	01	92	03	96	36	76	05	80	96	44	09	83	80	.....!6v.!!D.!!

Figura 3.90 Ubicación Kilometraje BSI lectura 1

El kilometraje en el archivo de volcado de la lectura número 1 se encuentra escrito en 8 grupos de 4 bytes como se indica en la Figura 3.90.

Los grupos de bytes que representan al kilometraje están ubicados en las siguientes direcciones y registros, el primer grupo está en la dirección 670 registros 0A, 0B, 0C, 0D, el segundo grupo se encuentra en la dirección 680 registros 00, 01, 02, 03, el tercer grupo se

encuentra en la dirección 680 registros 06, 07, 08, 09, el cuarto grupo se encuentra en la dirección 680 registros 0C, 0D, 0E, 0F, el quinto grupo se encuentra en la dirección 690 y registros 02, 03, 04, 05, el sexto grupos se encuentra en la dirección 690 y registros 08, 09, 0A, 0B, el séptimo grupo se encuentra en la dirección 690 registros 0E, 0F y sigue en la dirección 6A0 registros 00, 01 respectivamente y el octavo grupo se encuentra en la dirección 6A0 y registros 04, 05, 06, 07.

En la Tabla 3.21, se da a conocer cada grupo que representa el kilometraje, escrito en un grupo de 4 bytes de la siguiente forma FF D6 0E 7F. En el panel de instrumentos el kilometraje se encuentra representado en dos grupos de 8 bytes representado de la siguiente forma 80 F1 29 00 7F 0E D6 FF.

**Tabla 3.21** Representación del kilometraje en la BSI Valeo

DIRECCION	REGISTROS															
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
<b>670</b>											FF	D6	0E	7F		
<b>670</b>	FF	D6	0E	7F			FF	D6	0E	7F			FF	D6	0E	7F
<b>690</b>			FF	D6	0E	7F			FF	D6	0E	7F			FF	D6
<b>6A0</b>	0E	7F		FF	D6	0E	7F									

El kilometraje en el microcontrolador del módulo de la carrocería esta escrito con el mismo algoritmo que en el panel de instrumentos, con la diferencia que en la BSI solo está representado por los últimos 4 bytes y estos se encuentran invertidos, ejemplo los 4 últimos bytes del kilometraje de 274 880 en el panel son 7F 0E D6 FF y en la BSI son FF D6 0E 7F.

#### 3.4.4.2 Análisis de lecturas antes y después de la reprogramación de llaves

En este punto se analizó la diferencia entre los archivos de volcado obtenidos antes y después de haber realizado la programación de llaves, además se realizó una comparación del tamaño de dichos archivos.

The screenshot displays the Fairdell HexCmp application interface. On the left, a metadata panel shows the file size as 2048 bytes. The main area contains two hex dump windows. The top window, titled 'First File', shows a hex dump starting at offset 00000150. The bottom window, titled 'Second File', shows a hex dump starting at the same offset. In the 'Second File' dump, several bytes are highlighted in red, indicating differences from the 'First File'. These changes occur at offsets 00000170, 00000180, 00000190, 000001A0, 000001B0, 000001C0, 000001D0, 000001E0, 000001F0, and 00000200.

File size	Value
- HEX	0x800
- DECIMAL	2048
Offset	
- HEX	0x0
- DECIMAL	0
Value	
- Character	'I'
- Byte (HEX)	0x84
- Byte (DEC)	132
- Word (HEX)	0x1B84
- Word (DEC)	7044
- DWord (HEX)	0x78A1B84
- DWord (DEC)	126491524

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000150	84	1B	82	27	82	81	33	8F	04	1B	88	07	82	80	33	8F
00000160	84	1B	05	00	84	FF	33	8F	84	1C	7B	03	84	84	33	8F
00000170	04	1C	7B	03	84	81	33	8F	84	1B	8C	07	83	62	33	7E
00000180	04	1B	8C	07	83	63	33	7E	84	1B	8C	07	82	66	33	7F
00000190	04	1B	8C	07	82	66	33	7F	84	1B	8C	07	82	67	33	7F
000001A0	04	1B	8C	07	82	6A	33	7F	84	1B	8C	07	82	6C	33	7F
000001B0	04	1B	88	07	82	6D	33	7F	84	1B	88	07	82	6E	33	7F
000001C0	04	1B	8C	07	82	6F	33	7F	84	1B	8C	07	82	71	33	7F
000001D0	04	1B	88	07	81	78	33	80	84	1B	8A	07	81	79	33	80
000001E0	04	1B	8C	07	80	7D	33	80	84	1B	8D	07	80	7E	33	80
000001F0	04	1B	8C	07	80	81	33	81	84	1B	8B	07	81	81	33	81
00000200	04	1B	8C	07	80	82	33	81	84	1B	8A	07	81	81	33	81
00000210	04	1B	8C	07	81	80	33	81	84	1B	8B	07	80	7E	33	81
00000220	04	1B	8B	07	81	7D	33	81	84	1B	8C	07	81	7E	33	81
00000230	04	1B	8D	07	81	7E	33	81	84	1B	8B	07	81	7E	33	81
00000240	04	1B	8B	07	83	80	33	81	84	1B	8B	07	83	7F	33	81
00000250	04	1B	8C	07	82	7F	33	81	84	1B	8B	07	82	7F	33	81
00000260	04	1B	8A	07	81	7F	33	82	84	1B	8A	07	82	80	33	82
00000270	04	1B	8B	07	82	80	33	82	84	1B	8C	07	81	7F	33	82
00000280	04	1B	8A	07	80	80	33	82	84	1B	8C	07	80	80	33	82

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000150	84	1B	82	27	82	81	33	8F	04	1B	88	07	82	80	33	8F
00000160	84	1B	05	00	84	FF	33	8F	84	1C	7B	03	84	84	33	8F
00000170	04	1C	7B	03	84	81	33	8F	84	20	05	00	86	FF	33	8F
00000180	84	1F	05	00	86	FF	33	8F	04	20	05	00	86	FF	33	8F
00000190	04	1F	05	00	86	FF	33	8F	84	1C	77	03	86	6A	33	8F
000001A0	04	1C	05	00	86	FF	33	8F	84	17	77	01	86	FF	33	8F
000001B0	84	2C	76	01	86	FF	33	8F	84	2F	76	01	FF	FF	33	8F
000001C0	84	3D	76	01	FF	FF	33	8F	04	2C	77	01	FF	FF	33	8F
000001D0	04	2F	77	01	86	FF	33	8F	04	3D	77	01	86	FF	33	8F
000001E0	04	17	05	00	86	FF	33	8F	84	17	05	00	86	FF	33	8F
000001F0	84	2C	05	00	86	FF	33	8F	84	2F	05	00	FF	FF	33	8F
00000200	84	3D	05	00	FF	FF	33	8F	84	1B	8A	07	81	81	33	81
00000210	04	1B	8C	07	81	80	33	81	84	1B	8B	07	80	7E	33	81
00000220	04	1B	8B	07	81	7D	33	81	84	1B	8C	07	81	7E	33	81
00000230	04	1B	8D	07	81	7E	33	81	84	1B	8B	07	81	7E	33	81
00000240	04	1B	8B	07	83	80	33	81	84	1B	8B	07	83	7F	33	81
00000250	04	1B	8C	07	82	7F	33	81	84	1B	8B	07	82	7F	33	81
00000260	04	1B	8A	07	81	7F	33	82	84	1B	8A	07	82	80	33	82
00000270	04	1B	8B	07	82	80	33	82	84	1B	8C	07	81	7F	33	82
00000280	04	1B	8A	07	80	80	33	82	84	1B	8C	07	80	80	33	82

**Figura 3.91** Archivo de volcado de antes y después de la programación de llaves

El tamaño correspondiente entre las dos lecturas realizadas es de 2 048 bytes como se puede observar en la Figura 3.91, mismo valor que concuerda con el tamaño de la primera lectura referente al kilometraje de la BSI, además existen diversos cambios de bytes en la escritura del archivo obtenido después de la programación.

En la Tabla 3.22, se indica los datos o bytes que cambiaron después de haber realizado la reprogramación de llaves, en el archivo se generó un cambio de escritura en 16 direcciones y un total de 130 bytes modificados.



A continuación, se detalla de cada dirección y registro los bytes o datos modificados, los primeros bytes modificados son en la dirección 170 y registros 09, 0A, 0B, 0C, 0D, 0F los bytes modifica son 1B, 8C, 07, 83, 62, 7E a 20, 05, 00, 86, FF, 8F.

En la dirección 180 y registros 00, 01, 02, 03, 04, 05, 07, 08, 09, 0A, 0B, 0C, 0D, 0F los bytes que cambian son 04, 1B, 8C, 07, 83, 63, 7E, 84, 1B, 8C, 07, 82, 66, 7F a 84, 1F, 05, 00, 86, FF, 8F, 04, 20, 05, 00, 86, FF, 8F.

En la dirección 190 y registros 01, 02, 03, 04, 05, 07, 09, 0A, 0B, 0C, 0D, 0F los bytes que cambian son 1B, 8C, 07, 82, 66, 7F, 1B, 8C, 07, 82, 67, 7F a 1F, 05, 00, 86, FF, 8F, 1C, 77, 03, 86, 6A, 8F.

En la dirección 1A0 y registros 01, 02, 03, 04, 05, 07, 09, 0A, 0B, 0C, 0D, 0F los bytes que cambian son 1B, 8C, 07, 82, 6A, 7F, 1B, 8C, 07, 82, 6C, 7F a 1C, 05, 00, 86, FF, 8F, 17, 77, 01, 86, FF, 8F.

En la dirección 1B0 y registros 00, 01, 02, 03, 04, 05, 07, 09, 0A, 0B, 0C, 0D, 0F los bytes que cambian son 04, 1B, 88, 07, 82, 6D, 7F, 1B, 88, 07, 82, 6E, 7F a 84, 2C, 76, 01, 86, FF, 8F, 2F, 76, 01, FF, FF, 8F.

En la dirección 1C0 y registros 00, 01, 02, 03, 04, 05, 07, 08, 09, 0A, 0B, 0C, 0D, 0F los bytes que cambian son 04, 1B, 8C, 07, 82, 6F, 7F, 84, 1B, 8C, 07, 82, 71, 7F a 84, 3D, 76, 01, FF, FF, 8F, 04, 2C, 77, 01, FF, 8F.

En la dirección 1D0 y registros 02, 03, 04, 05, 07, 08, 09, 0A, 0B, 0C, 0D, 0F los bytes que cambian son 1B, 88, 07, 81, 78, 80, 84, 1B, 8A, 07, 81, 79, 80 a 2F, 77, 01, 86, FF, 8F, 04, 3D, 77, 01, 86, FF, 8F.

En la dirección 1E0 y registros 02, 03, 04, 05, 07, 09, 0A, 0B, 0C, 0D, 0F los bytes que se modifican son 1B, 8C, 07, 80, 7D, 80, 1B, 8D, 07, 80, 7E, 80 a 13, 05, 00, 86, FF, 8F, 17, 05, 00, 86, FF, 8F.

En la dirección 1F0 y registros 00, 01, 02, 03, 04, 05, 07, 09, 0A, 0B, 0C, 0D, 0F los bytes que cambian son 04, 1B, 8C, 07, 80, 81, 81, 1B, 8B, 07, 81, 81, 81 a 84, 2C, 05, 00, 86, FF, 8F, 2F, 05, 00, FF, FF, 8F.

En la dirección 200 y registros 00, 01, 02, 03, 04, 05, 07 los bytes que cambian son 04, 1B, 8C, 07, 80, 82, 81 a 84, 3D, 05, 00, FF, FF, 8F.

En la dirección 610 y registros 0E, 0F los bytes que se modifican son FF. FF a E1, 5D. en la dirección 620 y registros 00, 01 los bytes que se modifican son FF, FF a F2, EA.

En la dirección 630 y registros 06, 07, 08, 09 los bytes que se modifican son EE, EE, EE, EE por AA, 5C, 8E, 44.

En la siguiente dirección 640 y registros 0A, 0B los bytes que se modifican son FE, FE a FD, FD.

En la dirección 670 y registro 09 el byte que se modificó es DC por A0. Y por último en la dirección 6A0 y registros 02, 03 los bytes que cambian son C3, 28 a 3C, B3.

### **3.5 INFILTRACIÓN EN LA RED CAN**

En este apartado se da a conocer en primer lugar, los pasos para realizar una conexión entre los equipos y el puerto DCL del automóvil, necesarios para poder infiltrarse en la red can del automóvil, también se detalla el procedimiento de reprogramación del ángulo de acelerador utilizando el equipo de diagnóstico X-100 PAD2 PRO, proceso necesario para poder obtener los códigos de reprogramación emitidos durante el ajuste del ángulo del sensor del acelerador, además se realiza un análisis del protocolo de comunicación entre el equipo diagnóstico y la ECU.

#### **3.5.1 OBTENCIÓN CÓDIGOS RED CAN**

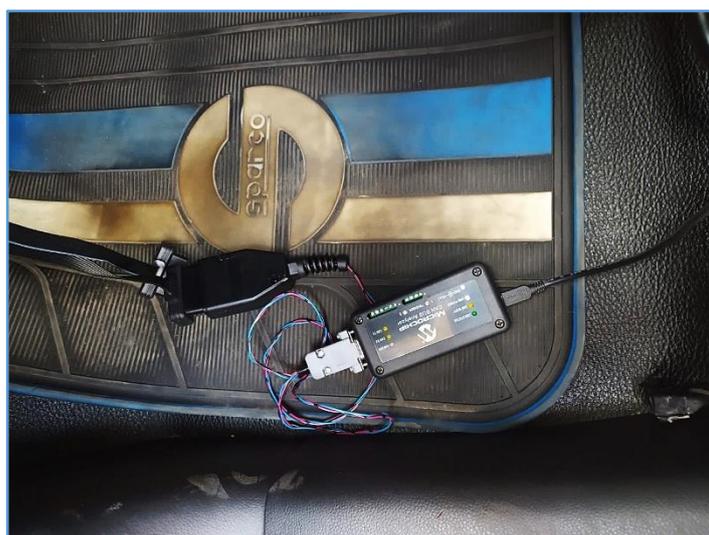
Para la obtención de códigos mediante la infiltración en la Red CAN se utilizó el del vehículo Kia Sportage R año 2011, la infiltración en la red can se la realizo para indagar en los códigos hexadecimales de programación que emite un equipo de diagnóstico. A este vehículo se le realizo un ajuste al ángulo del acelerador.



**Figura 3.92** Vehículo Kia Sportage R

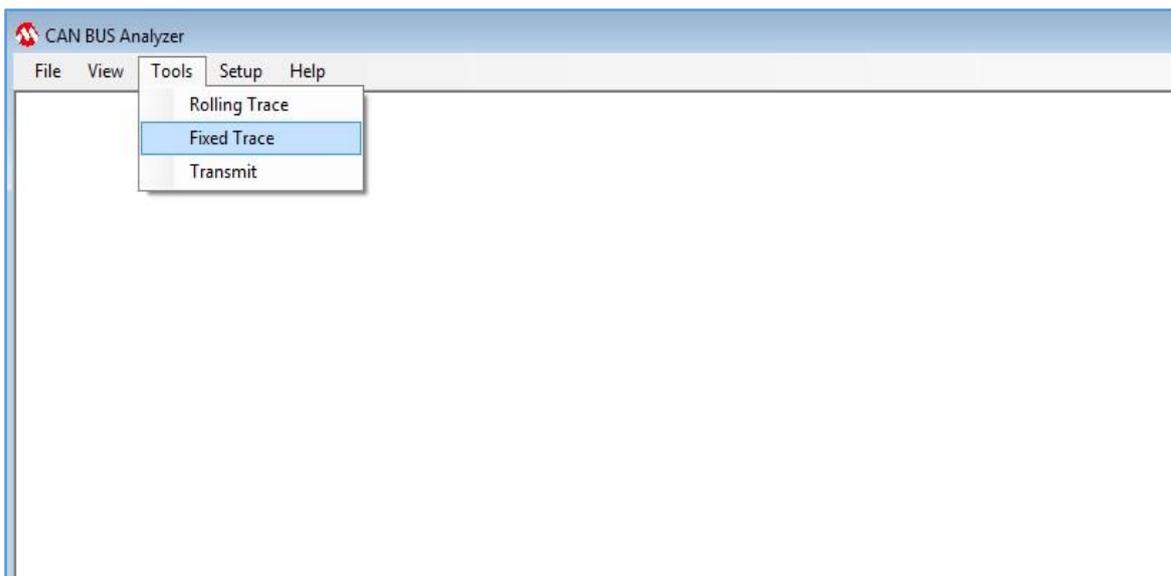
En la Figura 3.92, se puede observar el vehículo de la marca Kia modelo Sportage R, en el cual se realizó todo el procedimiento referente a la infiltración en la red CAN.

Para infiltrarse en la red can, se identificó el puerto DLC del vehículo, el mismo que se encuentra ubicado encima del pedal de aceleración y a la derecha de la caja de fusibles.



**Figura 3.93** Conexión CAN BUS ANALYZER Y X100 PAD 2 PRO

En la Figura 3.93, se puede apreciar la conexión que se realizó entre los equipos y el puerto DLC, las conexiones fueron realizadas mediante el cable doble conector OBD II y SPLITTER de 16 pines macho a hembra, del extremo conector macho se conectó al puerto DLC, de los dos extremos del conector hembra se conectó los equipos X 100 PAD2 PRO y Can Bus Analyzer, finalmente con un cable DB9 a USB se realizó las conexiones a una computadora portátil.



**Figura 3.94** Software CAN BUS Analyzer

Después de haber realizado las conexiones pertinentes entre los equipos y el puerto DLC se abre la interfaz del CAN BUS Analyzer, previamente instalado en una laptop, donde se va a la opción Tools y se seleccionó FIXED TRACE como se indica en la Figura 3.94.

 The image shows the 'CAN BUS Analyzer' software window with the 'Fixed Trace' window open. The window displays a table of communication data. The table has the following columns: TRACE, ID, DLC, DATA 0, DATA 1, DATA 2, DATA 3, DATA 4, DATA 5, DATA 6, DATA 7, TIME STAMP (sec), TIME DELTA (sec), and COUNTER. The data is as follows:
 

TRACE	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	TIME STAMP (sec)	TIME DELTA (sec)	COUNTER
RX	0x580	8	0x00	324,9502	0,021	10455							
RX	0x4B1	8	0x00	324,9502	0,020	10453							
RX	0x316	8	0x01	0x00	0x00	0x00	0x00	0x0C	0x00	0x80	324,9582	0,010	21987
RX	0x18F	8	0xFE	0x00	0x00	0x00	0x00	0x45	0x00	0x00	324,9593	0,010	21983
RX	0x260	8	0x00	0x00	0x0C	0x00	0x09	0x00	0x5F	0x12	324,9592	0,010	21961
RX	0x2A0	8	0x28	0x00	0x7F	0x00	0xDA	0x7B	0x54	0x09	324,9592	0,010	21957
RX	0x329	8	0xDC	0xAA	0x61	0x12	0x11	0x27	0x00	0x14	324,9592	0,010	21946
RX	0x545	8	0x5E	0x00	0x00	0x79	0x00	0x00	0x00	0x00	324,9502	0,010	21981
RX	0x350	8	0x00	0x10	0x63	0x7D	0x89	0x00	0x00	0x87	324,9653	0,010	21951
RX	0x5A0	8	0x00	324,4362	0,994	221							
RX	0x5A1	8	0x4E	0x53	0x5A	0x31	0x39	0x41	0x69	0x47	324,5252	0,994	219
RX	0x5A2	4	0x09	0x00	0x00	0x00					324,4362	0,994	220
RX	0xA0	8	0x00	0x77	0x00	0x00	0x00	0x24	0x01	0x00	324,8942	0,100	2192
RX	0xA1	8	0x80	0x80	0x00	0x00	0x4D	0x00	0x00	0x00	324,8942	0,099	2196

 At the bottom of the window, there is a status bar with the following information: Tool Connected, 500 Kbps, Normal Mode (highlighted in green), Error Normal, TX ERR: 0, RX ERR: 0, Termination: OFF, Trace Active, Logging Inactive, ID in HEX, DATA in HEX.

**Figura 3.95** Ventana FIXED TRACE

En la Figura 3.95, se puede observar la ventana FIXED TRACE la cual indica los mensajes de comunicación a través de la red CAN de las diversas actividades que está realizando el vehículo, por ejemplo, podemos detectar si se abre o se cierra una puerta, o si se activa el limpia parabrisas.

Estos mensajes son previos a la comunicación con el equipo de diagnóstico, el siguiente paso a seguir es seleccionar los parámetros necesarios en el X-100 PAD 2 PRO para poder realizar el reseteo del ángulo del acelerador.

Fixed Trace													
TRACE	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	TIME STAMP (sec)	TIME DELTA (sec)	COUNTER
RX	0x316	8	0x01	0x17	0x64	0x08	0x17	0x17	0x00	0x5B	502,7762	0,010	39831
RX	0x18F	8	0xFE	0x00	0x00	0x00	0x00	0x48	0x00	0x00	502,7762	0,010	39827
RX	0x260	8	0x10	0x17	0x17	0x30	0x0B	0x6E	0x6C	0x21	502,7762	0,010	39785
RX	0x2A0	8	0x48	0x00	0x80	0x18	0xDB	0x7B	0x54	0x09	502,7763	0,010	39776
RX	0x329	8	0x0C	0xA5	0x61	0x10	0x11	0x20	0x00	0x14	502,7772	0,010	39764
RX	0x545	8	0x5C	0x00	0x00	0x8A	0x00	0x00	0x00	0x00	502,7772	0,010	39812
RX	0x350	8	0x00	0x10	0x83	0x7D	0x8A	0x00	0x00	0x64	502,7812	0,010	39769
RX	0x5A0	8	0x00	502,3802	0,994	398							
RX	0x5A1	8	0x4E	0x53	0x5A	0x31	0x39	0x41	0x69	0x47	502,3602	0,994	396
RX	0x5A2	4	0x09	0x00	0x00	0x00					502,3802	0,994	397
RX	0xA0	8	0x5A	0x73	0x60	0x0B	0x00	0x20	0x02	0x00	502,7022	0,099	3972
RX	0xA1	8	0x8B	0x80	0x00	0x00	0x1A	0x00	0x00	0x00	502,7022	0,099	3978
RX	0x7DF	8	0x02	0x01	0x0C	0x00	0x00	0x00	0x00	0x00	502,7152	0,117	566
RX	0x7E8	8	0x04	0x41	0x0C	0x0B	0x68	0x00	0x00	0x00	502,7182	0,119	495

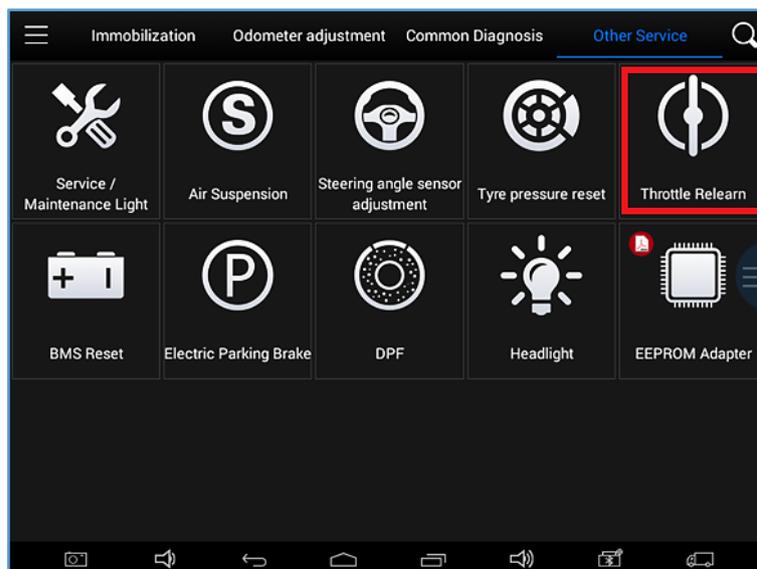
**Figura 3.96** Comunicación con valores hexadecimales

Para que se genere una comunicación entre el X-100 PAD 2 PRO y la computadora el equipo de diagnóstico envía un mensaje, la computadora recibe el mensaje y le da una respuesta, los mensajes enviados y de recepción se realiza mediante valores hexadecimales como se muestra en la Figura 3.96.

El X-100 PAD 2 PRO, siendo una unidad externa debe de iniciar la comunicación con la computadora automotriz, para esta acción el equipo envía el código 7DF, y la computadora como respuesta a este saludo envía el código 7E8. Las dos últimas filas corresponden a la comunicación entre el equipo y la ECU.

### 3.5.2 REPROGRAMACIÓN DEL ÁNGULO DEL ACELERADOR - KIA SPORTAGE R

En este punto se detalla el proceso que se realizó para poder realizar un ajuste al ángulo del acelerador utilizando el equipo de diagnóstico X-100 PAD 2 PRO, los pasos a seguir son sumamente fáciles, debido a que consiste en seleccionar las características generales del automóvil, como origen, marca, modelo y año de fabricación.



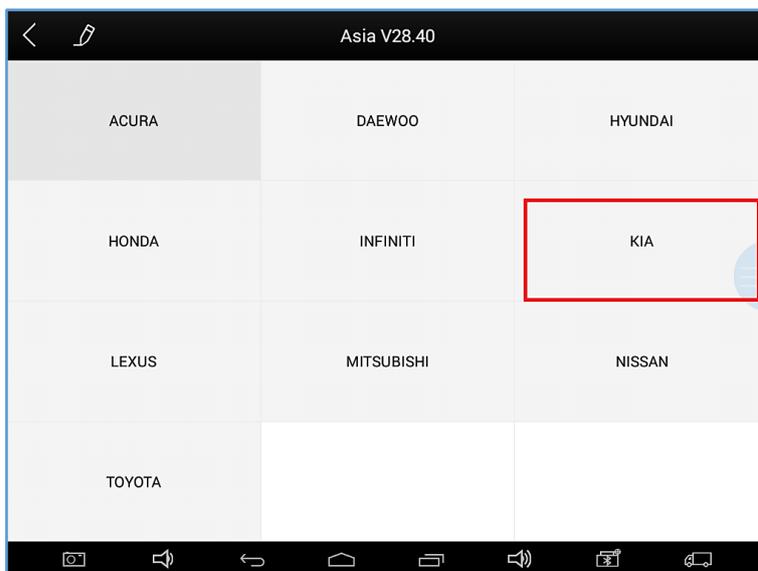
**Figura 3.97** Menú funciones especiales – Reset ajuste del ángulo del acelerador  
(Captura obtenida del equipo X100 PAD 2 PRO)

El primer paso para realizar el ajuste del ángulo del acelerador, en el menú que arroja el equipo dirigió a funciones especiales y se escogió la función ajuste del sensor de aceleración como se indica en la Figura 3.97.



**Figura 3.98** Menú selección origen del vehículo  
(Captura obtenida del equipo X100 PAD 2 PRO)

Una vez que se escogió la función de ajuste del sensor de aceleración se abrió una nueva ventana en donde se seleccionó el origen del vehículo como se puede apreciar en la Figura 3.98. En este caso se escogió la opción ASIÁTICA debido a que el vehículo del caso práctico es de marca de automóviles surcoreano.



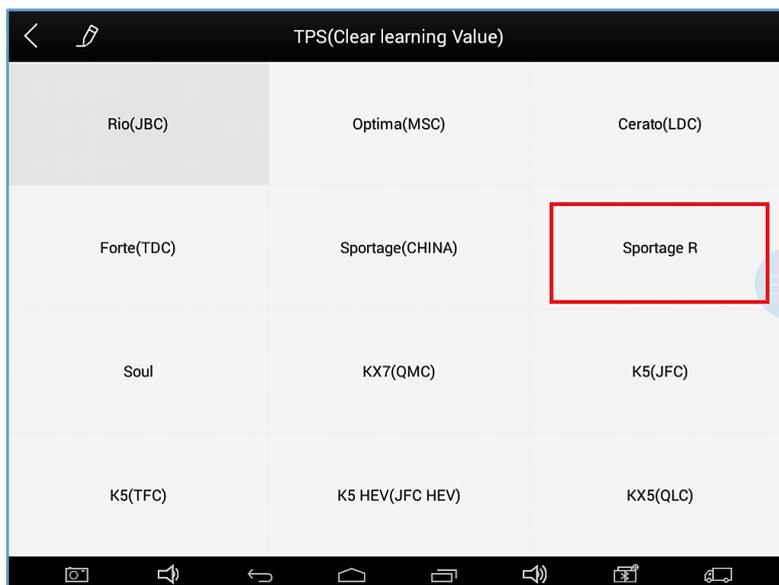
**Figura 3.99** Menú selección marca del vehículo  
(Captura obtenida del equipo X100 PAD 2 PRO)

Posteriormente, se abre una ventana con el menú de las marcas procedentes de Asia como se indica en la Figura 3.99, en el caso práctico la marca del vehículo es KIA por tal motivo se escogió esta opción.



**Figura 3.100** Menú selección TPS (CLEAR LEARNING VALUE)  
(Captura obtenida del equipo X100 PAD 2 PRO)

Una vez seleccionado la procedencia y la marca del vehículo, se despliega otra ventana con una sola opción a elegir como se muestra en la Figura 3.100, esta opción hace relación a TPS (CLEAR LEARNING VALUE) o borrado de valor de aprendizaje del sensor de posición del acelerador, se seleccionó dicha opción para proseguir con el procedimiento.



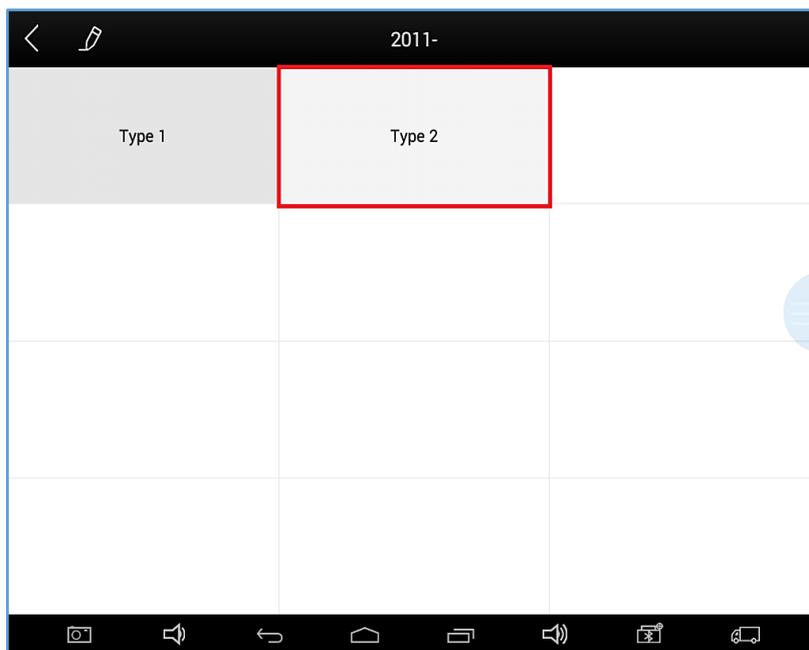
**Figura 3.101** Menú selección modelo vehículo  
(Captura obtenida del equipo X100 PAD 2 PRO)

En la Figura 3.101, se puede observar el siguiente menú que arroja el equipo, en este se puede apreciar los diferentes modelos para la marca KIA, aquí se seleccionó Sportage R debido a que este es el modelo del vehículo utilizado en la práctica



**Figura 3.102** Menú selección año del vehículo  
(Captura obtenida del equipo X100 PAD 2 PRO)

Una vez seleccionado las características generales del vehículo como procedencia, marca y modelo. Se procede a la elección del año del vehículo, en este caso se escogió el año 2011 debido a que es el año de fabricación del carro.



**Figura 3.103** Menú selección tipo de vehículo  
(Captura obtenida del equipo X100 PAD 2 PRO)

Posteriormente se abrió una ventana con dos opciones como se indica en la Figura 3.103, en esta pantalla se observa las opciones tipo 1 y tipo 2, en esta parte del proceso no se supo a ciencia cierta qué opción escoger, entonces se procedió a seleccionar la opción tipo 1 y dio como resultado comunicación fallida, debido a que no se tuvo una comunicación exitosa se seleccionó la opción tipo 2 y la comunicación fue exitosa.

Para finalizar este proceso se despliega una nueva ventana para confirmar el ajuste de posición del ángulo del acelerador, antes de realizar este paso se debe de verificar que el Switch de encendido debe de estar en posición ON y no se debe de arrancar el motor mientras dure el proceso.

### 3.5.3 ANÁLISIS DE CÓDIGOS DE PROGRAMACIÓN

La conexión realizada entre el puerto DLC y la herramienta CAN BUS Analyzer se la realizo con uno de los dos conectores macho del cable doble conector OBD II y SPLITTER de 16 pin macho a hembra que sirve como puente de infiltración, esta herramienta va conectada a una computadora en la cual ya está instalado el software de la herramienta, del otro conector macho va conectado el equipo X-100 PAD 2 PRO.

De esta forma se puede observar todos los códigos en hexadecimales que son enviados desde el equipo de programación a la computadora durante una reprogramación, además se puede observar los códigos de respuesta que emite la computadora al equipo que se ha ejecutado un trabajo solicitado por este.

El software del CAN BUS Analyzer es muy fácil de interpretar, debido a que este dispone de un panel principal semejante a una hoja de cálculo de Microsoft Excel, en este panel principal se puede observar en tiempo real los códigos que están siendo emitidos por el equipo de diagnóstico a la ECU y de igual forma se puede apreciar la información de respuesta.

Fixed Trace													
TRACE	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	TIME STAMP (sec)	TIME DELTA (sec)	COUNTER
RX	0x316	8	0x01	0x00	0x00	0x00	0x00	0x0B	0x00	0x58	1168.8332	0,009	106740
RX	0x18F	8	0xFE	0x00	0x00	0x00	0x00	0x41	0x00	0x00	1168.8332	0,010	106725
RX	0x260	8	0x00	0x00	0x0B	0x00	0x02	0x62	0x66	0x3A	1168.8333	0,009	106668
RX	0x2A0	8	0x68	0x00	0x87	0x00	0x0B	0x7B	0x54	0x09	1168.8342	0,010	106593
RX	0x329	8	0x4F	0x85	0x61	0x12	0x11	0x26	0x00	0x14	1168.8342	0,010	106580
RX	0x545	8	0x5E	0x00	0x00	0x7B	0x00	0x00	0x00	0x00	1168.8342	0,009	106696
RX	0x350	8	0x00	0x10	0x03	0x7D	0x95	0x00	0x00	0x2B	1168.8302	0,010	106479
RX	0x5A0	8	0x00	1168.5052	0,994	1062							
RX	0x5A1	8	0x4E	0x53	0x5A	0x31	0x39	0x41	0x69	0x47	1168.1362	0,994	1056
RX	0x5A2	4	0x09	0x00	0x00	0x00					1168.5052	0,994	1066
RX	0xA0	8	0x00	0x7F	0x00	0x00	0x00	0x24	0x01	0x00	1168.7692	0,100	10657
RX	0xA1	8	0x80	0x80	0x00	0x00	0x4D	0x00	0x00	0x00	1168.7692	0,100	10669
RX	0x7DF	8	0x01	0x00	1001.1152	0,850	4722						
RX	0x7E8	8	0x03	0x7F	0x11	0x12	0x00	0x00	0x00	0x00	1168.3772	0,030	4557
RX	0x7E0	8	0x02	0x11	0xFA	0x00	0x00	0x00	0x00	0x00	1168.3733	0,029	3

**Figura 3.104** Códigos hexadecimales de programación.

En la Figura 3.104, se puede apreciar los códigos en hexadecimales de reprogramación emitidos por el X-100 PAD 2 PRO hacia la ECU y los códigos de respuesta de la computadora hacia el equipo, durante el proceso de ajuste del sensor del acelerador del vehículo.

El protocolo de comunicación durante el proceso de reprogramación es el siguiente, El X-100 PAD 2 PRO envía las ordenes de trabajo a la ECU del vehículo por la ID 0x7E0, la ECU responde a las órdenes del equipo de diagnóstico por la ID 0x7E8, así se cumple el protocolo de comunicación J2534, este protocolo es el que define a las interfaces que se encuentran estandarizadas en reprogramación de la ECU.



Los códigos en hexadecimales emitidos por el X-100 PAD 2 PRO son de suma importancia ya que con estos datos se puede realizar una reprogramación del TPS sin necesidad de tener un equipo de costo elevado, para poder definir el ajuste del sensor de aceleración el equipo envió 7 órdenes de trabajo.

En la tabla anterior se puede apreciar que el escáner para realizar la primera orden de trabajo mediante la ID 0X7E0 envía códigos en hexadecimales a la ECU en el Data 0, Data 1, Data 2, los códigos emitidos son 0X01, 0X11 y 0XF8A respectivamente. En segunda instancia el equipo envía ordenes de trabajo en el Data 0, Data 1, Data 2, Data 4 con los códigos hexadecimales 0X03, 0X30, 0XB4, 0X04.

Posteriormente el equipo envía una tercera orden de trabajo en las mismas columnas de trabajo, con los siguientes códigos hexadecimales 0X03, 0X30, 0XBC, 0X04.

Las tres siguientes ordenes de trabajo se realizan en el Data 0, Data 1, Data 2, Data 4, para la cuarta orden los códigos hexadecimales emitidos son 0X03, 0X30, 0XC3, 0X04, para la quinta orden los códigos hexadecimales emitidos son 0X03, 0X30, 0XCA, 0X04, para la sexta orden los códigos hexadecimales emitidos son 0X03, 0X30, 0XD1, 0X04 y la última orden de trabajo se realiza en el Data 1, Data 2 los códigos hexadecimales emitidos son 0X01, 0X02.

### **3.5.4 VALIDACIÓN CÓDIGOS DE REPROGRAMACIÓN**

Una vez obtenido todos los códigos hexadecimales emitidos por el equipo de programación a la ECU, durante el proceso de ajuste del ángulo del sensor del acelerador, se puede realizar una reprogramación sin necesidad de ocupar el equipo de diagnóstico, para este proceso se puede utilizar una herramienta como el equipo CAN BUS Analyzer.

Para realizar una reprogramación en la interface del equipo en el menú Tools se debe de seleccionar la opción transmitir, esta ventana permite interactuar con otros nodos del CAN BUS transmitiendo mensajes, se puede ingresar cualquier ID, DLC o combinación de bytes en este caso datos en valores hexadecimales, además permite transmitir un máximo de nueve mensajes separados, los mensajes se los puede transmitir con un modo de repetición limitado a la velocidad periódica.

FORMAT	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	PERIOD (msec)	REPEAT	TRANSMIT
HEX	7E0	8	01	11	FA	00	00	00	00	00	100	50	Send
HEX	7E0	8	03	30	B4	04	00	00	00	00	100	50	Send
HEX	7E0	8	03	30	BC	04	00	00	00	00	100	50	Send
HEX	7E0	8	03	30	C3	04	00	00	00	00	100	50	Send
HEX	7E0	8	03	30	CA	04	00	00	00	00	100	50	Send
HEX	7E0	8	03	30	D1	04	00	00	00	00	100	50	Send
HEX	7E0	8	01	20	00	00	00	00	00	00	100	50	Send
HEX											0	0	Send
HEX											0	0	Send

**Figura 3.105** Códigos de reprogramación

En la Figura 3.105, se puede observar los códigos de reprogramación obtenidos en el proceso de ajuste del ángulo del sensor del acelerador, para ingresar los códigos en la columna FORMAT se selecciona hexadecimales (HEX), en la columna ID se ingresa 7E0 debido a que este mensaje fue emitido por el equipo de diagnóstico para enviar la orden de trabajo, en la columna DLC se escribió 8 ya que este fue el dato que se obtuvo en el proceso, y en las columnas DATA 0 y hasta la DATA 9 se puso todos los datos hexadecimales que se detallan en la tabla 3.23.

En la columna PERIOD (msec) se ingresa el número 100 debido a que es la mínima velocidad a la que el equipo puede mandar estos mensajes, y finalmente en la columna REPEAT, se escribe 50 ya que muchas veces los mensajes son perdidos, esto se debe a la prioridad de la computadora en el momento de recibir un mensaje, todos los datos se deben de ir ingresado uno por uno y finalmente antes de seleccionar SEND o enviar se debe de verificar que todos los datos ingresados sean correctos.

Con los datos obtenidos se empleó un nuevo método de programación manual para reprogramar de manera directa en la RED CAN, estos datos pueden ser utilizados por cualquier usuario en futuros trabajos de investigación.

## CAPITULO IV

### 4. CONCLUSIONES Y RECOMENDACIONES

#### 4.1 CONCLUSIONES

- En vehículos con sistemas electrónicos las funciones especiales que posee cada uno son dependientes del año de fabricación o de las tecnologías que vienen incorporadas, los procesos de reprogramación mediante un equipo de diagnóstico profesional son diferentes para cada función y para cada modelo de vehículo, por lo tanto, con la implementación del equipo X100 PAD 2 PRO se logró identificar funciones como: Reset luz de servicio, ajuste de sensor del ángulo de dirección, Reset presión de neumático, ajuste sensor de acelerador, Reset de la batería, Reset del freno eléctrico de parking y Reset del filtro de partículas diésel DPF.
- En el vehículo Peugeot 307 al inspeccionar los datos internos se encontró que este dispone de las funciones Reset de luz de servicio y ajuste de sensor de acelerador, en el caso de la primera función encontrada esta indica que el vehículo ha alcanzado un límite de kilómetros recorridos por lo que es necesario un mantenimiento preventivo. En el caso de la función ajuste del sensor de aceleración es para corregir el ángulo de apertura de la mariposa, a través de este ajuste se corrige fallas relacionadas de ralentí inestable o problemas directos con la aceleración.
- En la obtención de archivos de volcado, se comprobó que la ECU BOSCH 7.4.4 PSA, cuenta con un microcontrolador ST10C167, una memoria flash AM29F400BB, y una memoria EEPROM ST95160. Por otro lado, se identificó que la unidad BSI VALEO, cuenta con un microcontrolador MC68HC912DG128, siendo un microcontrolador muy diminuto, y finalmente, en el módulo del panel de instrumentos se logró determinar que esta unidad cuenta con una memoria EEPROM 25C020 tipo SOIC eléctricamente borrrable con 8 pines de montaje superficial.
- Mediante la obtención de archivos de volcado de los módulos de control, se determinó que el código de seguridad o PIN CODE y el número VIN del vehículo se encuentran encriptados en la memoria EEPROM ST95160 de la ECU y microcontrolador

MC68HC912DG128 de la BSI respectivamente, estos datos fueron de gran relevancia debido a que sirven a manera de ejercicio para cerciorarse de que las lecturas realizadas de los archivos estén de forma correcta. Además, en el vehículo Peugeot 307 se utilizó el PIN CODE en el proceso de programación de llaves donde se pudo concluir que, la información de las llaves programadas no se guarda directamente en la BSI se encripta, esto no pasa en otros vehículos que si lo hace.

- Después de la lectura de los diferentes archivos de volcado, mediante el proceso de evaluación, se logró identificar los códigos hexadecimales referentes al Reset luz de servicio y ajuste del sensor de aceleración, de este análisis se pudo deducir que, la memoria EEPROM ST95160 de la ECU guarda información referente a las funciones especiales, cuando el vehículo necesite resetear o ajustar alguna función, la memoria de la ECU es la encargada de enviar esta información al microcontrolador MC68HC912DG128 de la BSI y posteriormente ser visualizada en el panel de instrumentos. En la memoria del módulo de la BSI se encuentra encriptado el PIN CODE, número VIN, información del transponder de llaves y kilometraje y la memoria EEPROM 25C020 del panel de instrumentos guarda información del kilometraje.
- En la red CAN del vehículo Kia Sportage R año 2011 y con la ayuda del equipo microchip can bus Analyzer se pudo observar en tiempo real los códigos que están siendo emitidos por una herramienta de programación profesional a la ECU automotriz, en el proceso de reprogramación el equipo profesional utiliza el código 7DF para establecer comunicación con la ECU, cuando el equipo envía una orden de trabajo a la ECU utiliza el código 7E0, esta efectúa dicho trabajo y le da una respuesta de la orden de trabajo con el código 7E8, cumpliendo el protocolo de comunicación J2534.
- En el reseteo y ajuste de funciones especiales se obtuvieron tres diferentes métodos; el primero es con un equipo de programación profesional, el segundo método es mediante la obtención de códigos hexadecimales, escribirlos en el archivo de volcado de la memoria y reprogramar, y un tercer método que consiste en detectar los códigos de reprogramación emitidos en la red CAN, escribir estos códigos y enviar los códigos con una herramienta desarrolladora y depuradora de red CAN.

## 4.2 RECOMENDACIONES

- Los ajustes o reprogramaciones de funciones especiales son sencillas de efectuar si se sigue los pasos que le indica el equipo, por lo tanto, se recomienda antes de efectuar un reseteo o un ajuste se debe de tener una lectura previa de la unidad electrónica del vehículo para así poder comparar con la lectura que se obtenga después de efectuar dicha acción, debido a que estos procesos se presentan después de un ciclo de trabajo del vehículo.
- Para la inspección de los datos que contiene el vehículo, es recomendable antes de retirar las unidades de control electrónico, ECU, BCM y módulo del panel de instrumentos, se debe girar el switch de encendido a la posición off, abrir el capó del vehículo, bloquear todas las puertas y esperar entre 5 a 15 minutos para que la ECU deje de comunicarse con los diferentes módulos, una vez esperado cierto tiempo se procede a desconectar el borne negativo de la batería y posteriormente se puede retirar la computadora del vehículo.
- Cuando se va a trabajar en el circuito impreso para las lecturas de los archivos de volcado de los diferentes módulos del automóvil, se debe efectuar dicha acción en una superficie limpia, libre de líquidos y sustancias que puedan causar daños en su estructura y su funcionamiento, además es importante utilizar equipo de protección para evitar descargas electrostáticas a los circuitos integrados.
- En el proceso de infiltración en la red CAN, en el momento de identificar los códigos hexadecimales de comunicación de la ECU y un equipo externo, descifrar correctamente cual pertenece a la orden de trabajo y cual pertenece a la respuesta de la computadora de la orden efectuada, debido a que estos datos en el software de una herramienta desarrolladora y depuradora de red CAN, se presentan en forma desconcertada es decir primero se puede presentar el trabajo efectuado por la ECU y posteriormente la orden de trabajo del equipo externo hacia la ECU o viceversa.
- Se debe de tomar en cuenta que los códigos de reprogramación de funciones especiales presentados en este trabajo de investigación no pueden ser utilizados en cualquier vehículo, debido a que, los datos obtenidos mediante el método de obtención de códigos

hexadecimales a través de la lectura de archivos de volcado de memorias y microcontroladores son válidos únicamente para vehículos Peugeot 307 año 2003, y los datos obtenidos mediante el método de detectar los códigos de reprogramación emitidos en la red CAN son válidos para vehículos Kia Sportage R año 2011.

- Los procesos de reprogramación relacionados a funciones de kilometraje deben realizarse de una manera ética y profesional por ingenieros y técnicos automotrices, por lo que se recomienda a la Ley Orgánica de Transporte Terrestre, Tránsito y seguridad vial que establezca un reglamento que sancione a personas que modifiquen el kilometraje recorrido de un vehículo con el fin de evitar daños y perjuicios a futuros propietarios.
- Tomar en consideración este trabajo como una base para futuras investigaciones y profundizar este estudio en vehículos actuales que posean un campo más amplio de funciones especiales como, ajuste de sensor de acelerador, reseteo de servicio de aceite, reseteo de batería, calibración del ángulo de dirección, reseteo EPB, reseteo DPF entre otras.

## BIBLIOGRAFÍA

1. Alientech. (s.f.). *Programador a banco para centralitas de motor y transmisión automática* [Mensaje de un blog]. Recuperado el 28 de Diciembre de 2019, de <https://www.alientech-tools.com/es/k-tag/>
2. Alves Da Silva, G. (2017). *Projeto e Simulação de Funções Embarcadas Automotivas: Estudo de Caso para Carrocerias*. (tesis de grado). Uniersidade Tecnológica Federal Do Paraná, Curitiba, Brasil.
3. Andino , F., & Corrales, L. (2017). *Diseño y construcción de un banco de pruebas para la emulación de las señales generadas por los sensores empleados en el sistema de inyección electrónica automotriz*. (tesis de grado) Escuela Politécnica Nacional, Quito, Ecuador. Obtenido de <http://bibdigital.epn.edu.ec/handle/15000/19042>
4. Angulo, J., y Angulo, I. (2003). *Microcontroladores PIC Diseño práctico de aplicaciones Primera parte. El PIC16F84 Lenguajes PBASIC y Ensamblador*. (3ra ed.). Madrid, España: Concepción Fernández Madrid. Concepción Fernández Madrid
5. Arteaga, L. R. (22 de diciembre de 2007). ELECTRONICA DESARROLLO- Lenguajes de Programación para PICs [Mensaje de un blog]. Recuperado el 2 de Enero de 2019, de <http://electronicadesarrollo.blogspot.com/2007/12/programacin-de-pics.html>
6. Augeri , F. (21 de octubre de 2011). Reparacion de computadoras (Nivel 1). Recuperado de <http://www.cise.com/portal/capacitacion/noticias-de-los-cursos/item/461-curso-de-reparaci%C3%B3n-de-computadoras-automotrices-nivel-i.html>
7. Auto Avance . (24 de Agosto de 2013). Tipos de Memorias y Programación de Computadoras Automotrices Ecus [Mensaje de un blog]. Recuperado el 16 de Diciembre de 2019, de <https://www.autoavance.co/blog-tecnico-automotriz/167-procesamiento-de-datos-controladores-y-memorias-computadoras-automotrices/>
8. Auto Avance. (25 de Enero de 2016). Metodos para Flasheo de Memorias y Reprogramación de ECUS [Mensaje de un blog]. Recuperado el 30 de Diciembre de 2019, de <https://www.autoavance.co/blog-tecnico-automotriz/202-metodos-para-flasheo-de-memorias-y-reprogramacion-de-ecus/>

9. Auto Avance. (9 de Agosto de 2019). Chip Tuning o Reprogramación de Ecu [Mensaje de un blog]. Recuperado el 10 de Enero de 2020, de <https://www.autoavance.co/blog-tecnico-automotriz/chip-tuning-o-reprogramacion-de-ecu/>
10. AutoExacto. (Enero de 2019). Tecnología de punta para la industria automotriz [Mensaje de un blog]. Recuperado el 30 de Diciembre de 2019, de <https://www.diagnostico-automotriz.com/programador-de-llaves-x100-pad2-full.html>
11. Báez , L., & Cabrera , F. (2010). *Diseño e implementación de un sistema movil anti-robo comandado por voz a un sistema de seguridad electrónica para vehículo.*(trabajo de grado). Universidad Internacional del Ecuador, Quito, Ecuador. Obtenido de <https://repositorio.uide.edu.ec/handle/37000/182>
12. Barra, O., & Barra, F. (2015). *Microcontroladores PIC con programación PBP.* (1ra ed.). Madrid, España: RA-MA, S.A.
13. Binary Electronic y Sistem. (21 de Marzo de 2015). Funciones de la ECU [Mensaje de un blog]. Recuperado el 17 de Octubre de 2019 , de <https://www.binaryelectronic.com/blog/electronica-automovil/funciones-de-la-ecu/>
14. Buitrago , J. (2017). *Estudio del Protocolo de Comunicación serial BUS CAN y la aplicación en la industria de vehículos de transporte BUSSCAR DE COLOMBIA.* (trabajo de grado). Universidad Tecnologica de Pereira, Pereira, Colombia. Obtenido de <http://repositorio.utp.edu.co/dspace/handle/11059/9251>
15. Caldera, H. (2016). *Infraestructura de software para el desarrollo de aplicaciones de servicio automotriz.* (trabajo de grado). Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, México. Obtenido de <http://hdl.handle.net/20.500.11961/2878>
16. Cano, A. (2012). *Mantenimiento de redes multiplexadas* (1 ed.). Malga, España: IC Editorial. Recuperado de <http://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=3211651>
17. Carpio, C. (2013). *Manual de procedimientos para interactuar entre protocolos de comunicación automotriz.* (trabajo de grado). Universidad del Azuay, Azuay, Ecuador. Obtenido de <http://dspace.uazuay.edu.ec/handle/datos/2210>

18. Cevallos , A., & Reina , S. (2013). *Determinación de los niveles de iluminación de diferentes tipos de faros e implementación de un sistema automatizado en el automóvil.* (trabajo de grado) Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/7100>
19. Cevallos , C. (2016). *Análisis eléctrico de circuitos eléctricos del sistema de inyección de combustible del motor S4A del vehículo Chevrolet Sail.* (trabajo de grado). Universidad Internacional del Ecuador, Guayaquil, Ecuador. Obtenido de <http://repositorio.uide.edu.ec/handle/37000/1156>
20. Chinga, A. (2013). *Elaboración de un banco de pruebas para inmovilizador de Chevrolet Corsa evolution mediante elementos electrónicos instalados en un panel para la implementación de material didáctico en el taller automotriz de la facultad de ciencias de la ingeniería de la UTE.* (trabajo de grado). Universidad Tecnológica Equinoccial, Quito, Ecuador. Obtenido de <http://repositorio.ute.edu.ec/handle/123456789/4798>
21. Crespo, F., & León, D. (2017). *Construcción de un banco digital comprobador de computadoras automotrices para vehículos con sistemas de inyección electrónica a gasolina para el Laboratorio de Autotrónica de la Carrera de Ingeniería Automotriz de la ESPOCH.* (trabajo de grado). Escuela Superior Politécnica de Chimborazo, Chimborazo, Ecuador. Obtenido de <http://dspace.espoch.edu.ec/handle/123456789/7008>
22. Díaz, M., & Sopa, G. (2013). *Reprogramación electrónica del vehículo Corsa Evolution 1.4, para elevar los parámetros característicos del motor de combustión interna.* (trabajo de grado). Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/xmlui/handle/21000/7169>
23. Encyclopædia Britannica. (29 de mayo de 2020). *Technology* [Mensaje de un blog]. Recuperado el 12 de Mayo de 2019, de <https://www.britannica.com/technology/technology>
24. Flóres, H. (2010). *Diseño lógico: fundamentos de electrónica digital.* (1ra ed.). Bogotá, Colombia: Ediciones de la U .

25. García, D., & Rubio, M. (2017). *Introducción a la Informática Básica*. (1ra ed.). Madrid, España: UNED.
26. Grijalva, P. (2012). *Reprogramación de la memoria Eprom del vehículo Chevrolet Corsa*. (trabajo de grado). Universidad del Azuay, Cuenca, Ecuador. obtenido de <http://dspace.uazuay.edu.ec/handle/datos/1439>
27. Ibáñez Galindo, M. (2015). *Integración de un sistema para la obtención de datos de vehículos automotores basados en los protocolos CAN bus y OBD-II*. (trabajo de grado) Universidad Nacional Autónoma de México, México D.F, México. Obtenido de [https://ru.dgb.unam.mx/handle/DGB\\_UNAM/TES01000730818](https://ru.dgb.unam.mx/handle/DGB_UNAM/TES01000730818)
28. Infotaller. (11 julio de 2016). Sistema de antirrobo: inmovilizador y alarma. Cada vez más difícil para los ladrones [Mensaje de un blog]. Obtenido el 24 de febrero de 2020, de [https://www.infotaller.tv/blogs/indave/Sistema-antirrobo-inmovilizador-dificil-ladrones\\_7\\_1016668324.html](https://www.infotaller.tv/blogs/indave/Sistema-antirrobo-inmovilizador-dificil-ladrones_7_1016668324.html)
29. Lara Rivero, A. Á. (2014). De sistema mecánico a sistema tecnológico complejo. El caso de los automóviles. *Universidad Autónoma Metropolitana–Xochimilco*, 2, 11-39. doi:10.1016/S0186-1042(14)71253-7
30. Lopera, J. D., Ramírez, C. A., Zuluaga, M. U., & Ortiz, J. (2010). El método analítico como método natural. *Revista Crítica de Ciencias Sociales y Jurídicas*, 25(1), (2020, pag 17). Obtenido de <http://hdl.handle.net/10495/5501>
31. Mamani, C. (09 de Mayo de 2015). *Diseño e implementación de un simulador de parámetros de unidades de control electrónico automotrices (ECU) de vehículos a gasolina en la Región Puno para su diagnóstico modular*. (trabajo de grado). Universidad Nestor Cazares Velasquez, Puno, Perú. Obtenido de <http://repositorio.uancv.edu.pe/handle/UANCV/446>
32. Martínez, A. (2017). *Introducción a CAN bus: Descripción, ejemplos y aplicaciones de tiempo real*. (tesis de maestría). Universidad Politécnica de Madrid, Madrid, España. Obtenido de <http://oa.upm.es/48061/>
33. Mejía, A. (2005). *Guía práctica para manejar y reparar el computador* (11va ed.). Medellín, Colombia: Ditel Ltda.

34. Molina , L. (2014). *Implementación de un equipo de verificación para programación de llaves y Sistemas Inmovilizadores de Vehículos*. (tesis de grado). Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/7563>
35. Molina, K. R. (2015). Comunicación electrónica del automóvil: Sistema CAN-BUS. *Escuela Especializada en Ingeniería ITCA-FEPADE*, 6(6), 39-42. Recuperado de :<http://www.redicces.org.sv/jspui/bitstream/10972/1865/1/7.pdf>
36. Niza, V. (2009). *Análisis, estudio e implementación de un prototipo de comunicación con redes de área de control (CAN) en automóviles electrónicos*. (tesis de grado) Universidad Politécnica Salesiana, Quito, Ecuador. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/17943>
37. Noguera, F., & Terrén, D. (2010). *Programación* (1ra ed.). Barcelona, España: UOC. Obtenido de <https://ebookcentral.proquest.com/lib/utnortesp/reader.action?docID=3219238&pg=3>
38. Noroña, M., & Gómez, M. (2019). Desarrollo e innovación de los sistemas mecatrónicos en un automóvil: una revisión. *Enfoque UTE*, 10(1), 117-127. doi:<https://doi.org/10.29019/enfoqueute.v10n1.350>
39. OOCITIES.ORG. (31 de octubre de 2009). *Unidad de control electrónica* [Mensaje de un blog]. Recuperado el 01 de Noviembre de 2019, de <http://www.oocities.org/mecanicoweb/1.htm>
40. Paredes, F. (2011). *Implementación de un Banco Simulador de Fallas de un Sistema de Inyección Electrónica de Combustible (gasolina) en un Motor Corsa de Cuatro Cilindros 1.6L MPFI*. (tesis de grado). Universidad Internacional del Ecuador, Quito, Ecuador. Obtenido de <http://repositorio.uide.edu.ec/handle/37000/954>
41. Pérez, D. (2018). Evolución de los dispositivos electrónicos en un automóvil. *INNOVA Research Journal*, 3(2), 1-7. Recuperado de :<https://doi.org/10.33890/innova.v3.n2.2018.616>
42. Pumarica, D., & Tonato, M. (2018). *Investigación para elevar los parámetros de desempeño del motor de combustión interna T18SED L4 DOHC 1.8L a través de la*

- reprogramación electrónica mediante la conexión en paralelo de una ECU programable.* (tesis de grado). Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <https://repositorio.espe.edu.ec/handle/21000/14026>
43. Rodríguez, J. (2019). Análisis electrónico de una unidad de control de motor (ECU) Ford-WV tipo EEC-IV. *Ingeniería: Ciencia, Tecnología e Innovación*, 6(1), 45-56. doi:10.33890/innova.v2.n9.2017.482.
44. Romo , L., Cajisaca , J., & Pinos , S. (2016). *Diseño y construcción de un banco comprobador de ECUS de sistemas de inyección de gasolina usando una tarjeta adquisición Arduino Mega.*(tesis de grado). Universidad del Azuay, Cuenca, Ecuador. Obtenido de <http://dspace.uazuay.edu.ec/handle/datos/5741>
45. Sánchez, J., & Taípe, R. (2018). *Investigación de los parámetros de funcionamiento del sistema de inyección de combustible de los vehículos Hyundai para desarrollar la construcción de un banco de pruebas de verificación de ECU'S.* (tesis de grado) Universidad de Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/14152>
46. Semante, D. A. (2016). *Reprogramación de la unidad de control electrónica del vehículo Honda civic si, para mejorar los parámetros característicos del motor de combustión interna serie D16A9.* (tesis de grado) Universidad de Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/12408>
47. Senplades. (2017). *Plan Nacional de Desarrollo Todo una Vida.* Obtenido de Secretaría Nacional de Planificación y Desarrollo: [planificacion.gob.ec/wp-content/uploads/downloads/2017/10/PNBV-26-OCT-FINAL\\_0K.compressed1.pdf](http://planificacion.gob.ec/wp-content/uploads/downloads/2017/10/PNBV-26-OCT-FINAL_0K.compressed1.pdf)
48. Serravalle, J. (11 de febrero de 2011). Electrónica Bloques de trabajo en una ECU automotriz [Mensaje de un blog]. Recuperado el 23 de Enero de 2020, de <http://www.cise.com/portal/notas-tecnicas/item/327-bloques-de-trabajo-en-una-ecu-automotriz.html?tmpl=component&print=1>
49. Sharma, P., & Möller, D. (2018). Protecting ECUs and Vehicles Internal Networks. *IEEE International Conference on Electro/Information Technology (EIT)*, 1(1), 0465-0470. doi:10.1109/EIT.2018.8500295

50. Tancara, C. (1993). La Investigación Documental. *Temas Sociales no.17* , 91-106. Obtenido de <http://www.revistasbolivianas.org.bo/pdf/rts/n17/n17a08.pdf>
51. Tapia , S. (2007). *Diseño de circuitos inmovilizadores en vehículos con inyección electrónica de gasolina*. (tesis de grado). Universidad de Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/4157>
52. Tinazaray, D. G. (2015). *Reprogramación electrónica del control de inyección y encendido del motor CBR 600 F4i, para el vehículo Formula Student*. (tesis de grado) Universidad de Fuerzas Armadas ESPE, Latacunga, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/xmlui/handle/21000/12102>
53. Tocci , R., Widmer , N., & Moss, G. (2007). *Sistemas digitales - Principios y aplicaciones* (10ma ed.). Ciudad de México, México: Pearson Educación.
54. Torres , E. (2015). *Sistema Electrónico por comando de voz para la seguridad física, iluminación y accionamiento automático de los accesorios primarios de un automóvil*. (tesis de grado). Universidad Técnica de Ambato, Ambato, Ecuador. Obtenido de <https://repositorio.uta.edu.ec/jspui/handle/123456789/19381>
55. UNAM. (s.f.). Unidad de Apoyo para el aprendizaje - Lenguajes de programación [Mensaje de un blog]. Recuperado el 28 de Enero de 2020, de [https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod\\_resource/content/1/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html)
56. Universidad Internacional de Valencia. (21 de marzo de 2018). Conociendo el language de máquina [Mensaje de un blog]. Recuperado el 18 de Febrero de 2020, de <https://www.universidadviu.com/conociendo-lenguaje-maquina/>
57. Verle, M. (2009). *PIC microcontrollers Programming in C*. (1er ed.) Mikroelektronika. Obtenido de <https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-c-con-ejemplos/introduccion-al-mundo-de-los-microcontroladores>
58. Vlacic, L., Parent, M., & Harashima, F. (Eds.). (2001). *Intelligent vehicle technologies*. (1er ed.). Gran Bretania, Inglaterra. Butterworth-Heinemann: Obtenido de [http://ceb.ac.in/knowledge-center/E-BOOKS/Intelligent\\_Vehicle\\_Technologies\\_-\\_Ljubov\\_Vlacic.pdf](http://ceb.ac.in/knowledge-center/E-BOOKS/Intelligent_Vehicle_Technologies_-_Ljubov_Vlacic.pdf)

59. XTOOL. (n.d.). Experto en funciones especiales X100 PAD2 Pro [Mensaje de un blog]. Recuperado el 18 de Febrero de 2020, de <http://www.xtooltech.com/EN/ProductView/X100PAD2pro.html>

## **ANEXOS**

## ANEXO I

### FICHAS TÉCNICAS EQUIPOS DE PROGRAMACIÓN

	
FICHA TÉCNICA X100 PAD2 PRO	
<b>DESCRIPCIÓN</b>	
<p>X100 PAD2 Pro es un maravilloso programador de teclas para tabletas! No solo proporciona a los clientes una programación de teclas profesional, sino también las funciones especiales más necesarias para el taller. X-100 PAD2 es una tableta con las últimas tecnologías para realizar programación clave, ajuste de kilometraje, reinicio de la luz de servicio de aceite, reinicio de la luz de servicio de la correa de distribución, reinicio de la luz de advertencia de presión de los neumáticos, reinicio del EPB, reinicio del cuerpo del acelerador, reinicio del DPF, reinicio de la batería, dirección Calibración de ángulo y diagnóstico de motor OBD-II, y se agrega especialmente con VW 4th &amp; 5th IMMO.</p>	
CARACTERÍSTICAS	
Sistema operativo	<ul style="list-style-type: none"> <li>• Android 4.4.4</li> </ul>
Conectores	<ul style="list-style-type: none"> <li>• LÍNEA PRINCIPAL DE PRUEBA</li> <li>• OBD II-16</li> <li>• KIA-20</li> <li>• KIA HYNDAY-10</li> <li>• HONDA-3</li> <li>• HAMAI-17</li> </ul>
Contenido del kit de herramientas del EQUIPO	<ul style="list-style-type: none"> <li>• Power conversation line</li> <li>• Power cable</li> <li>• Lector de memorias EEPROM</li> <li>• KC-100</li> </ul>
Capacidad	<ul style="list-style-type: none"> <li>• Asia: para Toyota / para Leus / tor Honde / para Aoura / tor Mazde / para Nesen / para infinit / tor Mitsubishi / para Suberu / para Suma / tor Kie / para Hyunde / para Deawoo / para SSANGYONG / para Proton, etc</li> <li>• Europa: para Audi / para VW / para Skode / para Seet / para Porsche / para Opel / para Citroen / para Puget / para Jeguar / para Land Rover / para Lincoln / para Reneult, etc</li> <li>• América: para GM / para Cacilec / para Buick / para Chryolor / tor Ford, etc.Europa: para Greet Wall / para Chery / para Brillenco-auto / para Geely / para LFAN / tor HAFEI / para CHANGAN / para LANDWND / para BYD / tor JAC, etc</li> <li>• Australia: Auford Honde</li> </ul>
Eléctricas	<ul style="list-style-type: none"> <li>• Voltaje de energía 12V</li> <li>• Capacidad de la batería 4000 mAh, 7.4 V</li> </ul>
Funciones EEPROM	<ul style="list-style-type: none"> <li>• Lectura de PIN code</li> <li>• Inicialización ECU</li> <li>• Programación de Chip</li> </ul>

**Figura AI.1.** Ficha técnica X100 PAD 2 PRO

	
<b>FICHA TÉCNICA CAN BUS ANALYZER TOOL</b>	
DESCRIPCIÓN	REFERENCIA
<p>La herramienta de análisis CAN BUS es un monitor de bus CAN de bajo costo y fácil de usar que puede utilizarse para desarrollar y depurar una red CAN de alta velocidad. La herramienta es compatible con CAN 2.0b e ISO11898-2 y con una amplia gama de funciones que permiten su uso en varios segmentos del mercado, como el automotriz, el industrial, el médico y el marino. El kit de herramientas viene con todo el hardware y el software necesarios para conectar una red CAN a un PC. La interfaz gráfica de usuario facilita la observación e interpretación rápida del tráfico de autobuses.</p>	
<b>CARACTERÍSTICAS</b>	
Requisitos del sistema	<ul style="list-style-type: none"> <li>• Windows® XP</li> <li>• NET framework versión 3.5</li> <li>• Puerto serie USB</li> </ul>
Requisitos de los cables	<ul style="list-style-type: none"> <li>• Cable mini USB - para comunicarse con el software de la PC</li> <li>• La herramienta CAN Analyzer se puede conectar a una red CAN utilizando lo siguiente:               <ul style="list-style-type: none"> <li>• A través del conector DB9</li> <li>• A través de terminales de rosca</li> </ul> </li> </ul>
Contenido del kit de herramientas del analizador	<ul style="list-style-type: none"> <li>• Herramienta del analizador CAN BUS</li> <li>• Cable USB mini para la conexión al PC</li> <li>• Guía de usuario/CD de software</li> </ul>
Capacidad	<ul style="list-style-type: none"> <li>• Soporta CAN 2.0b e ISO11898-2</li> <li>• Interfaz de usuario intuitiva para PC para funciones como configuración, rastreo, transmisión, filtro, registro, etc.</li> <li>• Funciones mejoradas en la GUI del PC para el soporte del microcontrolador PIC del microchip, como la vista de registro ECAN en la GUI</li> <li>• Acceso directo a las señales CAN H y CAN L, CAN TX y CAN RX para una depuración robusta</li> <li>• Opciones flexibles de interfaz de bus CAN, es decir, conector estándar DB9 o terminales de tornillo</li> <li>• Control por software de la resistencia de terminación y pantalla LED para el estado, el tráfico, el error del BUS</li> </ul>
Eléctricas	<ul style="list-style-type: none"> <li>• Se necesita una fuente de alimentación (9 a 24 voltios) cuando se opera sin el PC y cuando actualizando el firmware en el USB PIC MCU</li> <li>• El analizador CAN BUS también puede ser alimentado a través del puerto USB</li> </ul>

**Figura AI.2.** Ficha técnica CAN BUS Analyzer

### FICHA TÉCNICA UPA-USB DEVICE PROGRAMMER

DESCRIPCIÓN	REFERENCIA
<p>El dispositivo principal se puede conectar directamente a Elrasoft.com UUSP (UPA-USB programador de serie) o por un cable de extensión DB9 macho/hembra 1:1 opcional. 2 está equipado con un zócalo ZIF de 40 pines para todos los dispositivos DIP / DIL estrechos y anchos, así como con un zócalo SOIC de 16 pines ZIF (150mil, estrecho) para cubrir los dispositivos más utilizados. Nuevo PROGRAMADOR USB UPA con adaptadores completos</p>	 <p>The image shows a red printed circuit board (PCB) for the UPA-USB V1.3 device programmer. It features a central white label with a QR code, a barcode, and the text 'UPA - USB V1.3', 'Torn invalid', and 'QC: z1002'. The board is populated with various electronic components, including a large black ZIF socket on the left side and a USB connector on the right. The board is mounted on a larger, partially visible PCB.</p>
CARACTERÍSTICAS	
Requisitos del sistema	<ul style="list-style-type: none"> <li>Windows 7</li> </ul>
Requisitos de los cables	<ul style="list-style-type: none"> <li>Cable mini USB - para comunicarse con el software de la PC:</li> <li>A través del conector DB9</li> </ul>
Capacidad	<ul style="list-style-type: none"> <li>STMicroelectronics ST62 *:ST6240... ST6245... ST6249 NSC CR16, Motorola HC05, Motorola HC08, Motorola HC11, Motorola HC12, Motorola HCS12, Freescale (Motorola) HCS12X, Freescale (Motorola) HCS12XE *: Atmel 8051 arquitectura, Atmel AVR de 8 bits de riesgo: Microchip PIC12, Microchip PIC16: EEPROM I2C, EEPROM, Microwire, EEPROM, SPI, EEPROM de productos:</li> </ul>
Eléctricas	<ul style="list-style-type: none"> <li>Voltaje: 12V</li> </ul>

**Figura AI.3.** Ficha técnica UPA-USB

## FICHA TÉCNICA K-TAG MASTER

DESCRIPCIÓN	REFERENCIA
<p>La puesta a punto de los parámetros de gestión del motor almacenados en un ECU. Puede ser llevado a cabo en la comunicación en serie o en el modo de arranque, y sustancialmente consiste en tres operaciones:</p> <p>Lectura del archivo original almacenado en el ECU.            Modificar el archivo original usando un software de ajuste.            Escribir el archivo modificado en el ECU.</p>	

CARACTERÍSTICAS	
Requisitos del sistema	<ul style="list-style-type: none"> <li>Windows 7, Windows 8, Windows 10</li> </ul>
Requisitos de los cables	<ul style="list-style-type: none"> <li>Cable mini USB - para comunicarse con el software de la PC</li> <li>A través del conector DB9</li> <li>A través de terminales de rosca</li> </ul>
Capacidad	<ul style="list-style-type: none"> <li>Para la mayoría de los vehículos, KESS V5.017 puede leer y escribir ECU a través del Cable OBD2 directamente.</li> <li>KTAG V7.020 lea y escriba ECU a través de TRICORE, BDM u otro modo con cable especial</li> </ul>
Eléctricas	<ul style="list-style-type: none"> <li>12 a 24 Voltios DC</li> </ul>

**Figura AI.4.** Ficha técnica K-TAG Master



## FICHA TÉCNICA XHORSE MINI VVDI KEY

DESCRIPCIÓN	REFERENCIA
<p>Máquina de clonación y programación, Xhorse MINI VVDI Key Tool es compatible con los tipos de TELEMANDOS más comunes en el mercado; es fiable y fácil de usar. La herramienta VVDI puede realizar estas funciones:</p> <ul style="list-style-type: none"> <li>• Clonar transpondedor</li> <li>• Editar transponder especial</li> <li>• Preparar/Generar remoto</li> <li>• Renovar</li> <li>• Probar frecuencia</li> <li>• Desbloquear tarjeta inteligente</li> </ul>	

CARACTERÍSTICAS	
Requisitos del sistema	<ul style="list-style-type: none"> <li>• Android, IOS</li> </ul>
Capacidad	<ul style="list-style-type: none"> <li>• support identification all immobile Transponder in current market clone &amp; edit partial common Transponder</li> <li>• Support editing transponder types 36/46Hitag2). 470Hitag3). 48. 4D(40/80). 7935</li> <li>• Support cloning transponder types Support 36/46 original or aftermarket transponder offline cloning</li> <li>• Support 40-40 bit 60 63 83 67 69, et offline doning Support 11/12/13/4c/30/33/42 offline cloning</li> <li>• Support Toyota 72-G transponder online cloning</li> <li>• Support Hyundai/KIA 70140-80) bit online doning</li> <li>• Support Ford 83-80 bit online cloning Support all 48 online cloning</li> <li>• Support unlocking transponder type Support unlockirg Xhorse special 4D. 46 transponder</li> <li>• Support unlocking all 7935 locked transponder</li> </ul>
Eléctricas	<ul style="list-style-type: none"> <li>• Voltaje de la batería 3.6-4.2 V</li> <li>• Capacidad de la batería 2500 mAh</li> </ul>

**Figura AI.5.** Ficha técnica XHORSE MINI VVDI KEY

## ANEXO II

### ARCHIVOS DE VOLCADO ECU

File size	Value
- HEX	0x800
- DECIMAL	2048
<b>Offset</b>	
- HEX	0x0
- DECIMAL	0
<b>Value</b>	
- Character	Z'
- Byte (HEX)	0x5A
- Byte (DEC)	90
- Word (HEX)	0x5A
- Word (DEC)	90
- DWord (HEX)	0x5A
- DWord (DEC)	90

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	5A	00	00	00	02	00	60	01	05	60	A5	4E	EC	7B	84	FC	
00000010	0D	04	02	02	02	00	00	00	CF	AB	34	58	A2	03	3D	FD	
00000020	0D	04	02	02	02	00	00	00	CF	AB	34	58	A2	03	3D	FD	
00000030	01	83	69	74	3E	67	FF	F5									
00000040	01	83	69	74	3E	67	FF	F5									
00000050	53	0A	83	02	99	0D	A9	01	04	00	20	04	6F	0E	24	FD	
00000060	53	0A	83	02	99	0D	A9	01	04	00	20	04	6F	0E	24	FD	
00000070	1B	00	6E	0E	2F	12	00	00	00	00	00	00	00	00	21	FF	
00000080	1B	00	6E	0E	2F	12	00	00	00	00	00	00	00	00	21	FF	
00000090	FF	20	00	00	00	00	00	00	00	00	00	00	00	00	00	D8	FE
000000A0	FF	20	00	00	00	00	00	00	00	00	00	00	00	00	00	D8	FE
000000B0	00	00	00	00	00	00	00	00	0D	03	00	00	00	00	00	E5	FF
000000C0	00	00	00	00	00	00	00	00	0D	03	00	00	00	00	00	E5	FF
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	F3	FF
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	F3	FF
000000F0	0A	2E	FF	FF	00	40	27	00	17	01	C6	00	00	00	76	FC	
00000100	AE	07	00	00	AE	07	00	00	00	00	00	00	0A	00	7C	FE	
00000110	00	00	00	00	00	80	00	00	00	80	00	00	00	00	00	EF	FE
00000120	FD	FF	00	00	00	00	00	00	3D	00	F6	3E	5C	00	25	FC	
00000130	01	00	D6	FF	00	00	C3	F6									
00000140	62	31	62	31	62	31	2B	4E	2B	4E	02	B5	10	0D	6D	FC	
00000150	00	00	D6	FF	00	00	00	00	00	00	D6	FF	D6	FF	6C	FA	
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	EA	FF	
00000170	00	00	00	00	00	00	00	80	00	80	00	80	00	80	E9	FD	
00000180	00	00	00	00	00	00	00	00	FD	FF	94	B0	00	00	A8	FC	
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	E7	FF	
000001A0	00	00	00	00	D6	FF	D6	FF	00	00	00	00	2B	4E	C3	FB	
000001B0	00	00	00	00	BB	03	3C	00	00	00	4A	4A	00	A6	B1	FD	
000001C0	D6	FF	11	F3													
000001D0	00	80	00	80	00	80	00	80	00	80	00	80	00	00	E3	FC	
000001E0	00	80	00	80	00	80	00	80	00	80	00	80	00	80	62	FC	
000001F0	01	FF	00	E2	F8												
00000200	FF	00	E7	F8													
00000210	FF	00	E6	F8													
00000220	FF	00	FF	00	FF	00	00	80	00	80	00	80	00	80	E1	FA	
00000230	01	FF	DD	F8													
00000240	01	FF	00	80	00	00	5C	FA									
00000250	00	00	00	00	00	40	00	40	00	40	00	40	00	00	DB	FE	
00000260	A4	01	D6	FF	D6	FF	00	00	00	00	D6	FF	00	00	B6	F9	
00000270	A1	09	1D	00	00	00	FD	0C	00	00	00	00	00	00	09	FE	
00000280	00	00	00	00	00	00	D6	FF	D6	FF	D6	FF	D6	FF	84	F8	
00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	D7	FF	
000002A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	D6	FF	
000002B0	15	19	01	04	00	01	00	00	00	00	00	00	00	00	A1	FF	
000002C0	00	00	00	FF	1A	FF	00	00	00	00	00	00	00	00	DC	FD	

Figura AII.1. Archivo de volcado Memoria EEPROM ST95160

Fairdell HexCmp		First File - C:\Users\USUARIO\UNIVERSIDAD TECNICA DEL NORTE\TESIS\PRUEB																
File size		OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
- HEX	0x80000	00000000	88	90	88	80	E0	19	DA	80	8E	00	49	80	2D	2D	DA	80
- DECIMAL	524288	00000010	F2	00	48	40	2D	15	E7	F8	5A	00	F7	F8	EF	F9	E7	FA
Offset		00000020	93	00	F7	FA	F0	F9	F7	8E	EE	F9	DE	E0	48	90	2D	04
- HEX	0x0	00000030	E0	1C	DA	00	66	64	08	91	9A	E0	F8	80	DF	E0	0D	24
- DECIMAL	0	00000040	E7	F8	A5	00	F7	F8	EE	F9	F7	8E	EF	F9	F7	8E	F0	F9
Value		00000050	DA	86	A4	F3	B5	4A	B5	B5	E0	08	08	81	46	F8	40	9C
- Character	'I'	00000060	8D	FC	DA	80	8A	18	0D	10	E7	F8	5A	00	F7	F8	EF	F9
- Byte (HEX)	0x88	00000070	F7	8E	EE	F9	F7	8E	F0	F9	E6	F4	5D	00	F6	F4	0A	AA
- Byte (DEC)	136	00000080	E6	F5	A2	FF	F6	F5	2E	AA	98	80	98	90	DB	00	88	70
- Word (HEX)	0x9088	00000090	88	60	E1	1E	E0	0C	E1	0C	C0	CE	5C	3E	D4	4E	4E	6A
- Word (DEC)	-28536	000000A0	48	41	3D	17	D4	4E	52	6A	E6	FD	A8	F7	C0	C5	E0	F2
- DWord (HEX)	0x80889088	000000B0	20	25	5C	12	00	D2	A8	5D	66	F5	FF	03	40	45	9D	07
- DWord (DEC)	-2138533752	000000C0	D4	4E	54	6A	A8	5D	66	F5	FF	03	40	45	9D	05	E1	0E
		000000D0	0D	03	F0	4C	09	81	F0	C4	09	C1	47	FC	10	00	8D	DC
		000000E0	F0	4C	47	F8	10	00	8D	01	E1	0E	F1	8E	98	60	98	70
		000000F0	DB	00	8A	E0	02	30	E0	14	DB	00	E0	04	DB	00	44	44
		00000100	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000110	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000120	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000130	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000140	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000150	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000160	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000170	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000180	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000190	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		000001A0	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		000001B0	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		000001C0	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		000001D0	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		000001E0	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		000001F0	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
		00000200	DA	80	0C	02	DB	00	DA	80	0C	02	DB	00	C2	F4	1E	F8
		00000210	D7	50	06	02	F4	A4	88	22	CC	00	69	A1	3A	88	88	36
		00000220	4A	88	03	66	F3	F8	1A	00	69	81	3A	88	88	36	4A	88
		00000230	01	61	F3	F8	1A	00	69	82	3A	88	88	36	4A	88	01	62
		00000240	F3	F8	1A	00	69	84	3A	88	88	36	4A	88	01	63	C2	F4
		00000250	28	F8	F0	54	5C	25	20	54	C2	F4	1E	F8	00	45	E6	F1
		00000260	E0	2A	00	14	A9	81	69	81	3A	88	88	36	4A	88	00	6D
		00000270	A9	81	69	82	3A	88	88	36	4A	88	00	6B	A9	81	69	84
		00000280	3A	88	88	36	4A	88	00	6C	A9	81	67	F8	08	00	3A	88
		00000290	88	36	4A	88	00	6A	E6	F1	E8	2A	C2	F4	1E	F8	00	14
		000002A0	A9	81	C0	84	68	41	3A	88	88	36	4A	88	00	68	A9	81
		000002B0	C0	84	68	42	3A	88	88	36	4A	88	01	69	A9	81	C0	84
		000002C0	C0	84	68	42	3A	88	88	36	4A	88	01	69	A9	81	C0	84

Figura AII.2. Archivo de volcado Memoria FLASH AM29F400BB

Fairdell HexCmp

File Edit Search View Options Help

First File - C:\Users\USUARIO\UNIVERSIDAD TECNICA DEL NORTE\TESIS\PRUE

File size	
- HEX	0x8000
- DECIMAL	32768
Offset	
- HEX	0x0
- DECIMAL	0
Value	
- Character	'I'
- Byte (HEX)	0x88
- Byte (DEC)	136
- Word (HEX)	0x9088
- Word (DEC)	-28536
- DWord (HEX)	0x80889088
- DWord (DEC)	-2138533752

OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	FA	00	70	64	FA	82	04	00	FA	00	18	67	FA	82	0C	00
00000010	FA	00	30	67	FA	82	14	00	FA	00	48	67	FA	82	1C	00
00000020	FA	82	20	00	FA	82	24	00	FA	00	60	67	FA	82	2C	00
00000030	FA	82	30	00	FA	82	34	00	FA	82	38	00	FA	82	3C	00
00000040	FA	82	40	00	FA	82	44	00	FA	82	48	00	FA	82	4C	00
00000050	FA	82	50	00	FA	82	54	00	FA	82	58	00	FA	82	5C	00
00000060	FA	82	60	00	FA	82	64	00	FA	82	68	00	FA	82	6C	00
00000070	FA	82	70	00	FA	82	74	00	FA	82	78	00	FA	82	7C	00
00000080	FA	82	80	00	FA	82	84	00	FA	82	88	00	FA	82	8C	00
00000090	FA	00	2C	13	FA	00	C0	1E	FA	00	6C	17	FA	00	96	13
000000A0	FA	82	A0	00	FA	82	A4	00	FA	82	A8	00	FA	00	CA	22
000000B0	FA	00	2A	22	FA	82	B4	00	FA	82	B8	00	FA	82	BC	00
000000C0	FA	82	C0	00	FA	82	C4	00	FA	82	C8	00	FA	82	CC	00
000000D0	FA	82	D0	00	FA	82	D4	00	FA	82	D8	00	FA	82	DC	00
000000E0	FA	00	7A	22	FA	82	E4	00	FA	82	E8	00	FA	82	EC	00
000000F0	FA	82	F0	00	FA	82	08	01	FA	82	F8	00	FA	82	FC	00
00000100	FA	82	00	01	FA	82	04	01	FA	82	08	01	FA	82	0C	01
00000110	FA	82	10	01	FA	82	14	01	FA	82	18	01	FA	82	1C	01
00000120	FA	82	20	01	FA	82	24	01	FA	82	28	01	FA	82	2C	01
00000130	FA	82	30	01	FA	82	34	01	FA	82	38	01	FA	82	3C	01
00000140	FA	82	40	01	FA	00	78	67	FA	00	A8	67	FA	82	4C	01
00000150	FA	82	50	01	FA	82	54	01	FA	82	58	01	FA	82	5C	01
00000160	FA	82	60	01	FA	82	64	01	FA	82	68	01	FA	82	6C	01
00000170	FA	82	70	01	FA	82	74	01	FA	82	78	01	FA	82	7C	01
00000180	FA	82	80	01	FA	82	84	01	FA	82	88	01	FA	82	8C	01
00000190	FA	82	90	01	FA	82	94	01	FA	82	98	01	FA	82	9C	01
000001A0	FA	82	A0	01	FA	82	A4	01	FA	82	A8	01	FA	82	AC	01
000001B0	FA	82	B0	01	FA	82	B4	01	FA	82	B8	01	FA	82	BC	01
000001C0	FA	82	C0	01	FA	82	C4	01	FA	82	C8	01	FA	82	CC	01
000001D0	FA	82	D0	01	FA	82	D4	01	FA	82	D8	01	FA	82	DC	01
000001E0	00	00	00	00	FF	3F	00	00	CF	F5	A0	0F	30	0A	5F	F0
000001F0	00	40	00	00	FF	7F	00	00	B3	16	47	0F	4C	E9	B8	F0
00000200	D8	7C	00	00	4E	00	E6	7C	00	00	4E	00	D8	7C	00	00
00000210	9C	00	D8	7C	00	00	9C	00	D8	7C	00	00	9C	00	D8	7C
00000220	00	00	9C	00	26	17	00	00	00	00	D8	7C	00	00	5E	00
00000230	E6	7C	00	00	5E	00	D8	7C	00	00	BC	00	D8	7C	00	00
00000240	BC	00	D8	7C	00	00	BC	00	D8	7C	00	00	BC	00	26	17
00000250	00	00	00	00	00	00	00	02	00	02	00	00	00	00	2A	02
00000260	2A	02	00	00	62	20	00	00	24	4F	00	00	CA	7C	00	00
00000270	4C	0D	00	00	F4	7C	00	00	3A	10	00	00	74	02	00	00
00000280	01	00	C8	E2	4E	7D	00	00	3A	10	00	00	84	02	00	00
00000290	01	00	CA	E2	F8	20	00	00	D6	21	00	00	3A	10	00	00
000002A0	94	02	02	00	01	00	CC	E2	EE	21	00	00	F8	2F	00	00
000002B0	58	7D	00	00	3A	10	00	00	A8	02	01	00	01	00	CE	E2
000002C0	24	2F	00	00	4E	00	26	17	00	00	00	00	24	2F	00	00

Figura AII.3. Archivo de volcado Memoria MICRO ST10C167

## ANEXO III

### ARCHIVO DE VOLCADO BSI

Fairdell HexCmp		First File - C:\Users\USUARIO\UNIVERSIDAD TECNICA DEL NORTE\TESIS\PRUEB																
File size		OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
- HEX	0x800	00000000	04	1B	05	02	80	79	31	62	84	1B	05	00	83	FF	31	65
- DECIMAL	2048	00000010	04	1B	05	02	85	5B	31	65	84	1B	83	01	87	FF	31	69
Offset		00000020	04	1B	81	03	87	84	31	69	84	1B	05	00	87	FF	31	69
- HEX	0x0	00000030	04	1B	05	02	82	81	31	69	84	1B	05	00	82	FF	31	6C
- DECIMAL	0	00000040	04	1B	05	07	83	8F	31	6C	84	1B	81	01	80	FF	31	72
Value		00000050	04	1B	7C	07	82	8B	31	72	84	1B	05	00	81	FF	31	74
- Character		00000060	04	1B	05	02	75	66	31	74	84	1B	80	01	74	FF	31	76
- Byte (HEX)	0x4	00000070	04	1B	05	02	75	80	31	76	84	1B	83	01	72	FF	31	79
- Byte (DEC)	4	00000080	04	1B	82	03	72	86	31	79	84	1B	05	10	71	FF	31	7A
- Word (HEX)	0x1B04	00000090	04	1B	05	12	6F	7B	31	7A	84	1B	05	00	72	FF	31	7D
- Word (DEC)	6916	000000A0	04	1B	06	02	7E	3D	31	7D	84	1B	82	03	7D	FF	31	7E
- DWord (HEX)	0x2051B04	000000B0	04	1B	80	03	7D	7E	31	7E	84	1B	05	00	78	FF	31	82
- DWord (DEC)	33889028	000000C0	04	1B	05	02	7A	86	31	82	84	1B	05	00	79	FF	31	83
		000000D0	04	1B	75	07	76	82	31	83	84	1B	05	00	78	FF	31	88
		000000E0	04	1B	06	02	75	3D	31	88	84	1B	05	00	71	FF	31	88
		000000F0	04	1B	79	03	70	67	31	88	84	1C	7A	03	70	67	31	88
		00000100	84	1B	06	00	6F	FF	31	88	04	1B	78	03	76	5B	31	88
		00000110	04	1C	78	03	76	5B	31	88	84	1B	06	00	76	FF	31	89
		00000120	04	1B	2D	27	76	66	31	89	84	1B	05	00	76	FF	31	89
		00000130	04	1B	2D	07	7E	5A	31	89	84	1B	06	00	7E	FF	31	89
		00000140	04	1B	06	02	73	39	31	89	84	1B	05	00	71	FF	31	8B
		00000150	04	1B	79	07	72	81	31	8B	84	1B	05	00	73	FF	31	8F
		00000160	04	1B	87	07	73	85	31	8F	84	1B	83	27	73	85	31	8F
		00000170	04	1B	86	27	73	84	31	8F	84	1B	05	00	71	FF	31	93
		00000180	04	1B	7C	07	71	8D	31	93	84	1B	7F	01	72	FF	31	93
		00000190	04	1B	7B	03	72	89	31	93	84	1B	05	00	72	FF	31	93
		000001A0	04	1B	05	02	73	85	31	93	84	1B	05	00	71	FF	31	96
		000001B0	04	1B	06	02	71	36	31	96	84	1B	05	00	70	FF	31	98
		000001C0	04	1B	05	02	73	75	31	98	84	1B	05	00	73	FF	31	99
		000001D0	04	1B	7A	07	75	80	31	99	84	1B	05	00	75	FF	31	9B
		000001E0	04	1B	88	07	75	89	31	9B	84	1B	8B	07	75	82	31	9C
		000001F0	04	1B	83	07	75	85	31	9C	84	1B	05	00	75	FF	31	9C
		00000200	04	1B	7B	02	72	8A	31	9C	84	1B	05	00	72	FF	31	9C
		00000210	04	1B	7B	07	72	82	31	9C	84	1B	8D	07	6F	84	31	9D
		00000220	04	1B	8C	07	6F	85	31	9E	84	1B	81	01	70	FF	31	9E
		00000230	04	1B	05	02	72	5B	31	9E	84	1B	89	07	73	5B	31	9E
		00000240	04	1B	88	07	72	5B	31	9E	84	1B	8D	07	73	7F	31	A2
		00000250	04	1B	8D	07	73	7F	31	A2	84	1B	8C	07	73	87	31	A3
		00000260	04	1B	86	07	73	89	31	A3	84	1B	88	07	74	88	31	A3
		00000270	04	1B	8B	07	74	87	31	A3	84	1B	8A	07	73	8A	31	A4
		00000280	04	1B	87	27	74	88	31	A4	84	1B	8A	07	73	82	31	A4
		00000290	04	1B	89	07	73	84	31	A4	84	1B	80	01	73	FF	31	A4
		000002A0	04	1B	7B	07	75	8A	31	A4	84	1B	86	07	75	87	31	A4
		000002B0	04	1B	8B	07	74	84	31	A4	84	1B	89	07	75	87	31	A6
		000002C0	04	1B	8D	07	75	81	31	A6	84	1B	82	01	75	FF	31	A6

Figura AIII.1. Archivo de volcado Microcontrolador MC68HC912DG128 BSI Valeo

## ANEXO IV

### ARCHIVO DE VOLCADO PANEL DE INSTRUMENTOS

Fairdell HexCmp		First File - C:\Users\USUARIO\UNIVERSIDAD TECNICA DEL NORTE\TESIS\PRUE																	
File size		OFFSET	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
- HEX	0x100	00000000	00	00	00	00	00	00	00	00	00	00	28	00	06	28	06		
- DECIMAL	256	00000010	00	08	00	A0	FF	FF	FF	EE	00	FF	FF	FF	F7	12	38	00	
Offset		00000020	44	EC	29	00	BB	13	D6	FF	44	EC	29	00	BB	13	D6	FF	
- HEX	0x0	00000030	FF																
- DECIMAL	0	00000040	08	FF	FF	FF	FF	FF	FF	39	36	35	36	36	39	36	31	38	30
Value		00000050	01	02	07	17	06	05	FF	FF	FB	99	FF	FF	71	56	69	61	
- Character		00000060	FF																
- Byte (HEX)	0x4	00000070	FF																
- Byte (DEC)	4	00000080	04	05	07	09	0C	10	15	1B	24	30	3F	53	6E	92	C1	FF	
- Word (HEX)	0x1B04	00000090	FF	FF	40	1F	80	3E	C0	5D	FF	7D	40	9C	80	BB	C0	DA	
- Word (DEC)	6916	000000A0	29	FF	21	02	1A	04	12	06	0C	08	05	0A	FD	0B	F5	0D	
- DWord (HEX)	0x2051B04	000000B0	14	3C	46	5A	64	6E	76	79	BE	FF	15	01	96	01	96	02	
- DWord (DEC)	33889028	000000C0	15	03	96	03	15	04	77	04	FF	07	0A	19	28	32	4B	62	
		000000D0	A2	FF	E7	FF	15	01	C3	01	5C	02	C3	02	C3	03	CC	04	
		000000E0	E8	03	F0	0A	C0	12	60	22	30	32	14	41	BC	48	64	50	
		000000F0	57	0A	B8	01	19	03	DB	05	9F	08	61	0B	C2	0C	23	0E	

**Figura AIV.1.** Archivo de volcado Memoria EEPROM 25C020

**ANEXO V**  
**CÓDIGOS EMITIDOS EN LA RED CAN**

TRACE	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7
RX	0x7E8	8	0x03	0x7F	0x11	0x12	0x00	0x00	0x00	0x00
RX	0x7E0	8	0x01	0x11	0xFA	0x00	0x00	0x00	0x00	0x00
RX	0x7E8	8	0x03	0x70	0xB4	0x04	0x00	0x00	0x00	0x00
RX	0x7E0	8	0x03	0x30	0xB4	0x04	0x00	0x00	0x00	0x00
RX	0x7E8	8	0x03	0x70	0xBC	0x04	0x00	0x00	0x00	0x00
RX	0x7E0	8	0x03	0x30	0xBC	0x04	0x00	0x00	0x00	0x00
RX	0x7E8	8	0x03	0x70	0xC3	0x04	0x00	0x00	0x00	0x00
RX	0x7E0	8	0x03	0x30	0xC3	0x04	0x00	0x00	0x00	0x00
RX	0x7E8	8	0x03	0x7F	0x30	0x12	0x00	0x00	0x00	0x00
RX	0x7E0	8	0x03	0x30	0xCA	0x04	0x00	0x00	0x00	0x00
RX	0x7E8	8	0x03	0x70	0xD1	0x04	0x00	0x00	0x00	0x00
RX	0x7E0	8	0x03	0x30	0xD1	0x04	0x00	0x00	0x00	0x00
RX	0x7E8	8	0x01	0x60	0x00	0x00	0x00	0x00	0x00	0x00
RX	0x7E0	8	0x01	0x20	0x00	0x00	0x00	0x00	0x00	0x00

**Figura AV.1.** Códigos emitidos en la RED CAN