

Manual modificación kilómetros (leer/escribir) cuadro de Vw T4.

Se puede aplicar a más cuadros del grupo vag e incluso otras marcas y modelos.

Bien, no sé si definir esto de manual, o no, en todo casos sería un "manual" un procedimiento o un ajuste en todo caso, hay varios por internet, además de mucha información sobre cuadros etc... muy útil si se quiere acontecer a este tipo de modificaciones. Tan solo unifico aquí mi experiencia y resultados, para el foro furgovw.org, por petición de uno o varios miembros.

Primero de todo, la modificación de kilómetros en un cuadro con el fin de vender un vehículo mejor, es un delito. Además en la ITV, se anotan los kilómetros cada revisión, y darán cuenta de ello.

Con esto solo quiero decir, que este manual es para aquellos que quieran cambiar los kilómetros o deban cambiar los kilómetros, porque hayan cambiado o modificado el cuadro original y deban modificar los kilómetros para tener los que realmente tiene el vehículo.

En segundo lugar, no me hago responsable de las posibles averías, errores, roturas, o problemas derivados de la utilización de los procesos de este manual.

En tercer lugar **no** solo se utiliza este proceso para cambiar los kilómetros del marcador, se pueden realizar otras modificaciones, como variar la iluminación de los instrumentos, activar en test de agujas al inicio y otros más, como he dicho hay mucha información en varias webs sobre estos temas.

Vamos allá...

Voy a empezar nombrando que es lo que se va a hacer y luego veremos el cómo.

Parece una pérdida de tiempo, pero tiene sentido, para situarnos.

Lo que se va a hacer es leer un cuadro y escribir modificando algo (pueden ser los kilómetros, u otras modificaciones como he dicho antes). Tenemos que tener en cuenta que un cuadro es una unidad de control electrónica. (por lo menos todos los modelos, que llevan línea de diagnóstico, lo son)

Básicamente, es como reprogramar una unidad de control de motor u otra cualquiera. Quiero decir en cuanto al proceso, los datos modificables son diferentes.

Detallemos esto un poco más, es decir, dentro de los cuadros hay una memoria, esta almacena varias cosas como:

Kilómetros totales.

Número de bastidor.

Datos del inmovilizador.

Mapas de control de los relojes del cuadro.

Opciones de configuración (switchs) de las diferentes opciones del cuadro.

Etc, etc, etc.... (no sé cuantas más)

Todo esto suele ir almacenado en un chip , normalmente un EEPROM. Esos datos pasados a formato hexadecimal o decimal, con un programa de edición de estos documentos, se puede llegar a entender o extraer lo que queremos de ellos. Sin estos programas, difícilmente entenderemos algo (quizá Neo en matrix XD).

Esta es la pinta que tiene un chip EEPROM 93c56, bastante común:



Bien no soy experto, ni técnico, ni entendido, ni si quiera puedo llegar a nivel de aficionado, en estos temas de edición y demás en estos formatos binario, hexadecimal, decimal etc... ni si quiera a la electrónica en general, pero como he dicho antes hay mucha información en internet.

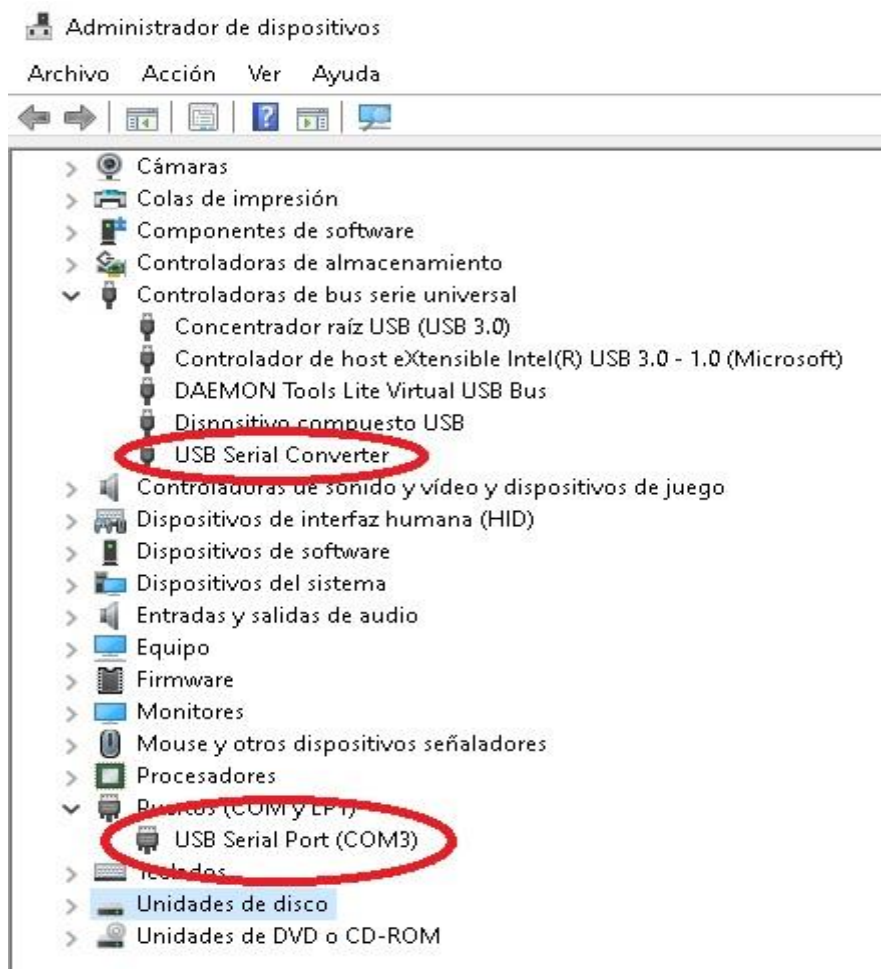
Si hay alguien que de verdad se dedique a esto, y quiera arrojar un poco más de información o corregir lo que aquí explico adelante. Que lo comunique a través del foro, y lo modificamos hasta tener un proceso claro.

Hasta donde yo sé, hay tres formas de leer y escribir este chip.

- 1- Primera y **más fácil**, utilizando un cable o interface por el conector OBD, y a través de la línea K de diagnóstico del cuadro. Hay varios cables/interfaces de este tipo.

No sirve el vag-com, por lo menos no cualquiera, hay cables vag-com, antiguos (409 creo), que si se pueden usar. O incluso otros cambiando el driver del cable por uno que deje una interface con varios dispositivos separados.

Pues para funcionar los diferentes programas necesitan una interfaz con un dispositivo y su driver de tipo:



Como se ve, la interface “USB Serial Converter”, crea la conexión: USB Serial Port (COM3), que es la que utilizaran los diferentes programas para a través de la línea de diagnóstico acceder a la información del chip.

Bien, como digo hay varias interfaces/cables , las más conocidas Vag k+can commander, vag tacho o vag EEPROM programmer por poner algunos ejemplos.

Respecto a si se necesita una interfaz, con línea CAN de comunicación, en principio no, lo dejo aquí claro, para los cuadros, por lo menos para los que he probado (para los más actuales no lo sé y supongo que sí) , pero lo que he visto es, que aunque se utilice un vag k+can commander, comunica al cuadro y lee/escrbe, única y exclusivamente a través de la línea K de diagnóstico, que para los que no lo sepan, es un único cable, una única conexión/pin del conector del cuadro.

Ni si quiera necesita la línea L, que sería otro cable, otra conexión/pin. Y que no la tienen todas las unidades de control/cuadros.

Bien, vamos a los programas.

También hay varios, y no voy a entrar a decir, si hace falta uno u otro para cada tipo de cable/interfaz, porque algunos son compatibles otros no.....bla bla bla, (luego explicaré la compatibilidad que encontré y que me funciona), y la mayoría solo necesitan el puerto antes nombrado, configurado correctamente para funcionar.

Bien, hay varios programas para hacer los mismo, unos con más opciones que otros, pero entre ellos están, los mismo que he nombrado en las interfaces. Las principales y comunes:

-Vag k+can commander, hay varias versiones, muchas públicas por internet, algunas con soporte CAN otras no (como he dicho antes en principio da igual).

Tiene otras funciones, como diagnóstico de unidades, configuración de algunos parámetros en algunas unidades de control, y también lee/escribe algunas unidades de control de motor y quizá más cosas ...

La que nos interesa, lee/escribe eeprom de cuadro, y tiene un editor hexadecimal integrado, que nos permite ver los datos y modificarlos directamente, además extrae de estos datos el kilometraje y número vin y datos inmovilizador.

-Vag EEPROM programmer normalmente versión 1.19, también pública por internet, y hace como el anterior, quizá un poco más reducido en cuanto a diagnóstico y configuraciones, *pero creo que también puede leer resetear unidades de airbag, inmo box y algo más.*

Como el anterior, lee/escribe eeprom de cuadro, y tiene un editor hexadecimal integrado, que nos permite ver los datos y modificarlos directamente, además extrae de estos datos el kilometraje y número vin y datos inmovilizador.

Mucho lio.

O no.

Ahora vamos al proceso:

Como hoy en día todo el mundo usa Windows 10, diré que hay bastantes problemas de configuración de estos programas para su correcto funcionamiento, y describiré mi experiencia con ellos.

En primer lugar , uso Windows 10, y tengo cable/interface “vag k+can commander 1.4” , y lo he utilizado con el programa “vag eeprom programmer 1.19”. El cable es como este:



Y ahora viene la explicación, no conseguí hacer funcionar el cable “BIEN” con su programa, ni con máquinas virtuales, ni probando mil configuraciones distintas en Windows etc etc etc. Cuando digo “bien” es que a veces hacia como que leía y después nada de nada.

La única forma fue utilizar el otro programa (vag eeprom programmer 1.19) descargado de internet, y puedo asegurar que es compatible con el cable.

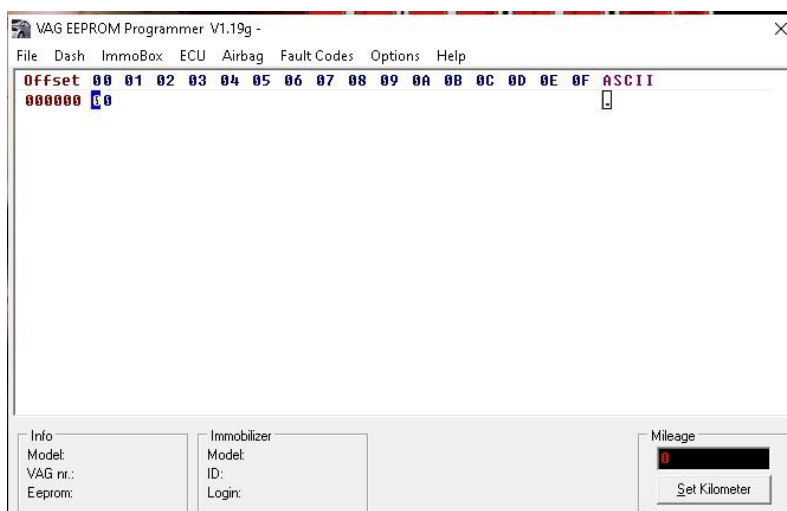
Hay que configurar el puerto “USB Serial Port COM x”, correctamente, creo que al final fue algo así:

Panel de control> Sistema> Hardware> Administrador de dispositivos> Puertos (COM y LPT), buscar el "USB Serial Port COM x ", ir a la pestaña "Configuración del puerto", luego haga clic en "Avanzado" y configure el Temporizador de latencia (mseg) en 1 , y poner un puerto COM, bajo 1,2,3,(comprobar los libres) después aceptar y aceptar.

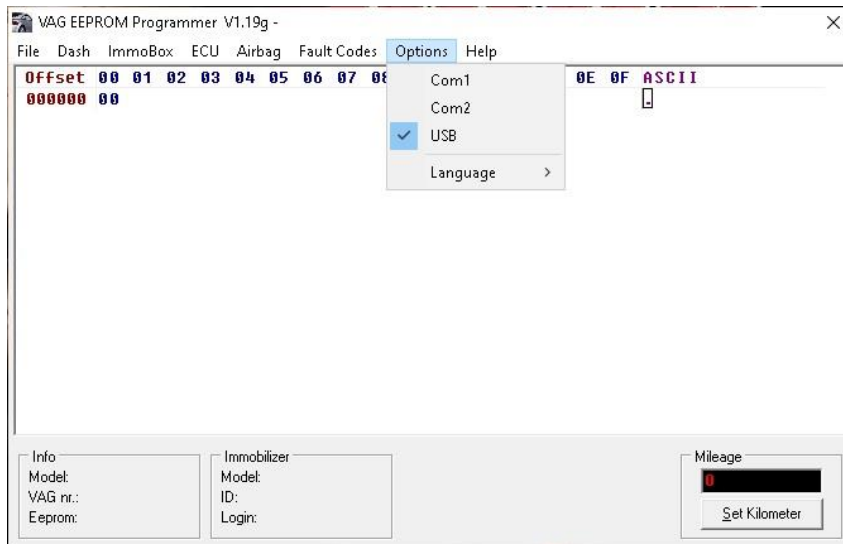
Además de esto, recomiendo cerrar todo tipo de aplicaciones abiertas mientras es hace la lectura/escritura de la eeprom, por supuesto acceder a la aplicación con permiso de administrador, desactivar el antivirus completamente, y en mi caso tuve que añadir una excepción en el firewall para el programa funcionará correctamente.

Conectamos el cable USB, y al vehículo en el OBD (si el cuadro esta montado, en mi caso lo hacia sobre la mesa con adaptadores), abrimos el programa.

Si esta todo bien, tendremos algo así:



Comprobamos que está configurado para USB. Options, USB.

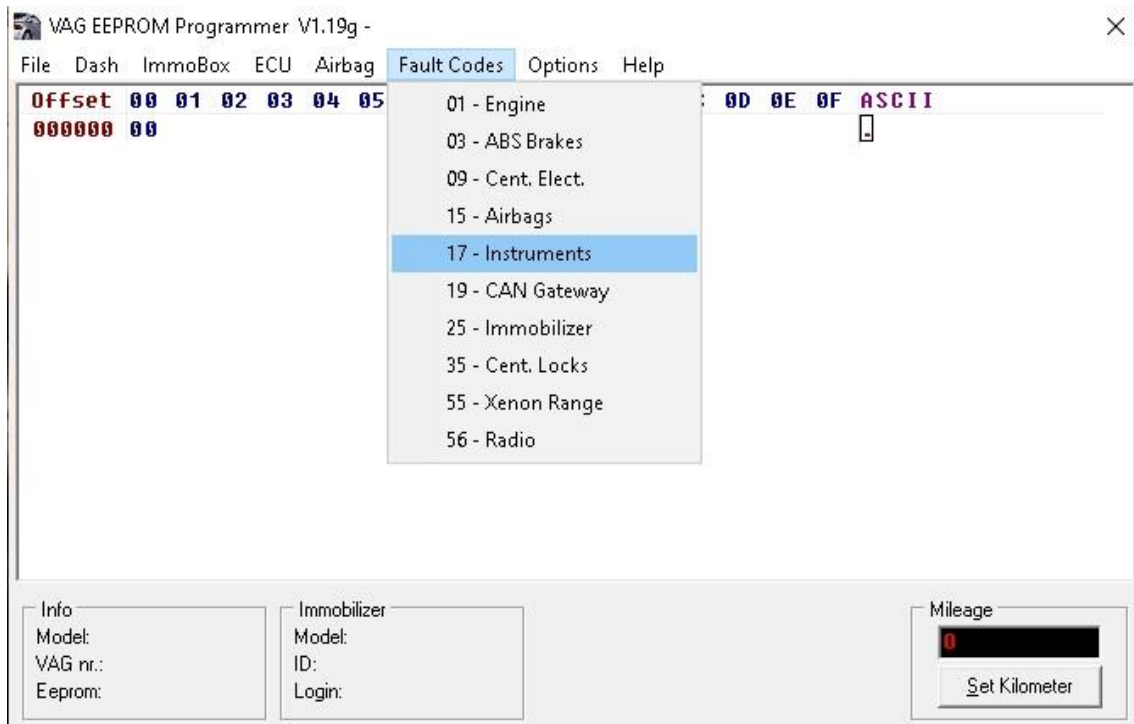


Bien, ponemos el contacto. Y primero probamos que la comunicación del cable/interfaz con el programa y con el vehículo sea correcta.

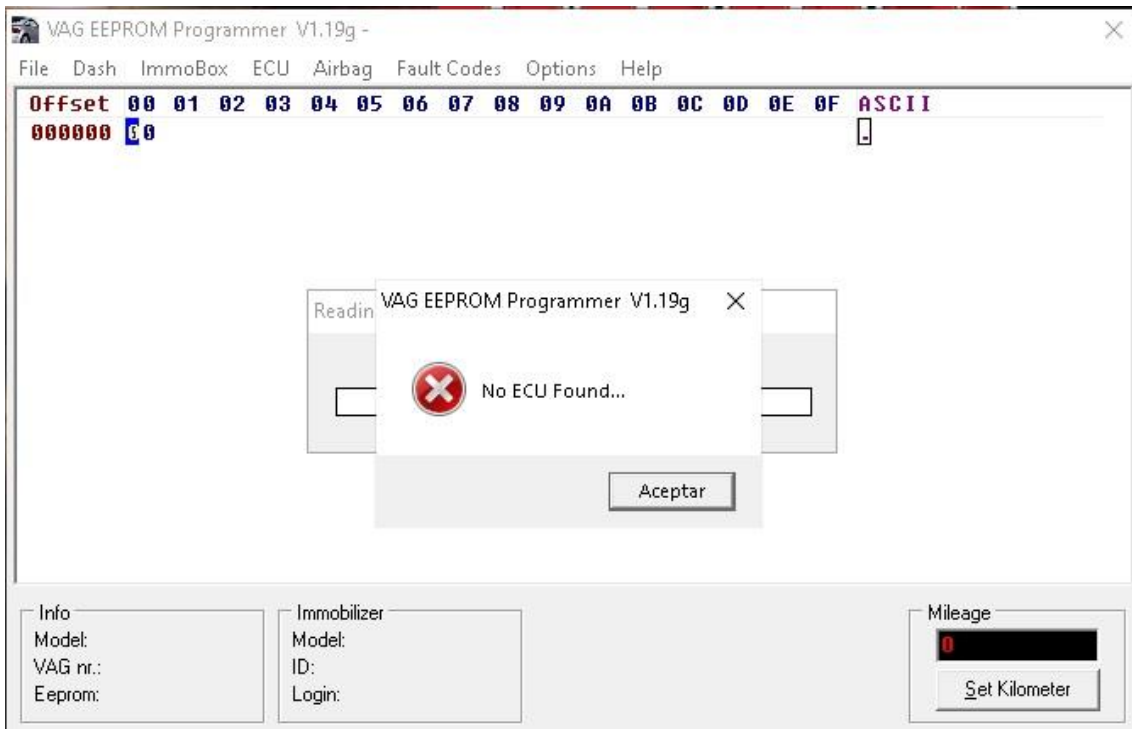
Apunte: Los cables/interfaces necesitan una tensión, de al menos 12,5V o más. Se que es difícil, pero normalmente todas las herramientas y sus aplicaciones de lectura/escritura de unidades de control, ya sean de cuadros de instrumentos, de motor, caja de cambio etcetcetc necesitan una fuente de alimentación fiable. Y muchas monitorizan el voltaje mientras trabajan.

Yo suelo poner un cargador, cuando leo unidades de motor. Porque para hacer las lecturas el motor tiene que estar parado.

Diagnosticamos el cuadro, Fault codes, 17-Instruments.



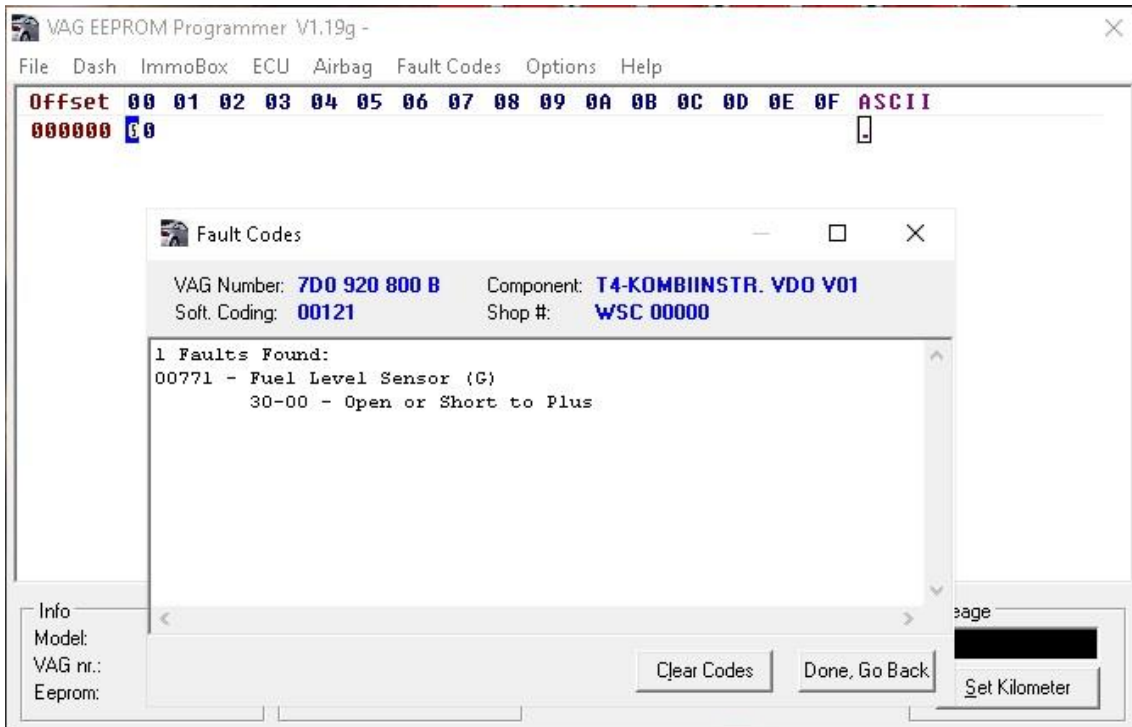
Podemos tener dos resultados:



En este caso, algo no esta bien configurado o conectado. Revisar las cosas que he nombrado antes, o googlear como locos. A mi a veces me tocaba poner en vez de USB, el puerto Com al que estaba conectado, (véase que solo puede ser 1 o 2), o cambiando entre estas opciones, funcionaba o no.

La de veces que he visto ese letrero....

O la siguiente:



Donde ha hecho un diagnóstico, y nombra en este caso (desconectado de todo, solo negativo, positivo y línea K, sobre la mesa, (ver fotos siguientes) da fallo de conexión del sensor de nivel de combustible.

Además vemos, que nos da el código de configuración de software, (software coding) en este caso 00121. La referencia del cuadro 7D0 920 800 B. Nos dice que el componente es un cuadro de instrumentos de T4, marca VDO Versión 01. Y el código de taller o tienda en este caso WSC00000.

Un pequeño stop, para ver el detalle de cómo está mi cuadro montado:



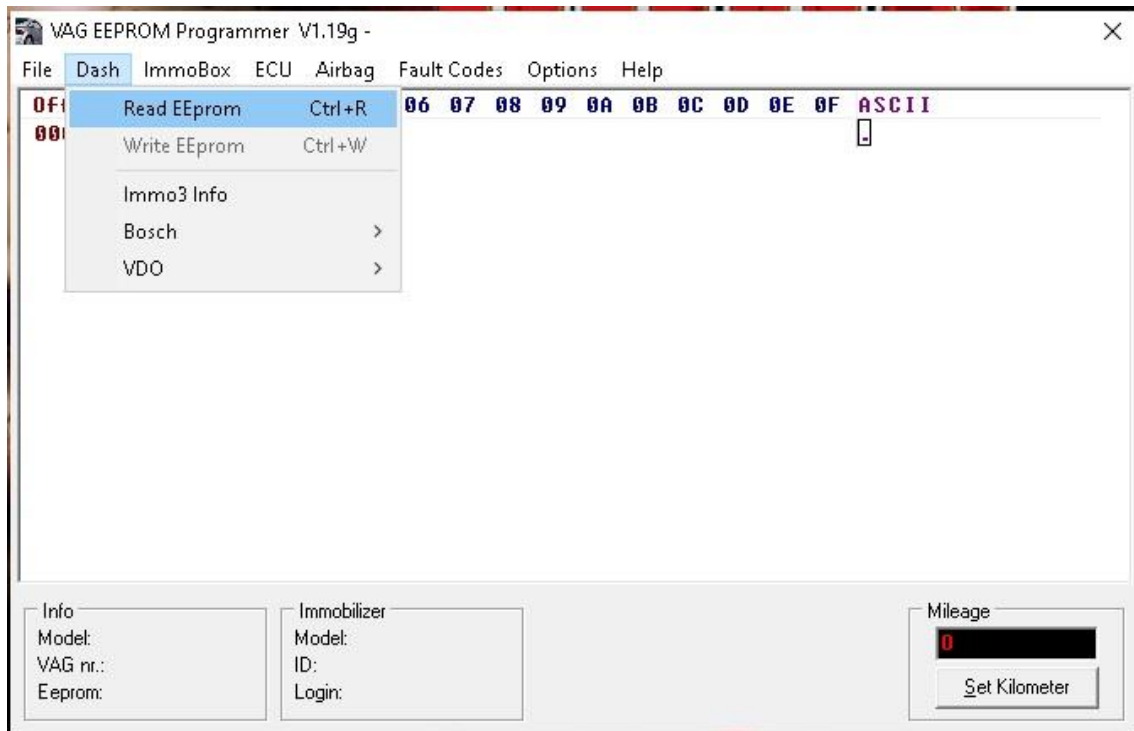
Y los cables que hay conectados, solo 4:

(+30,+15,31,K)



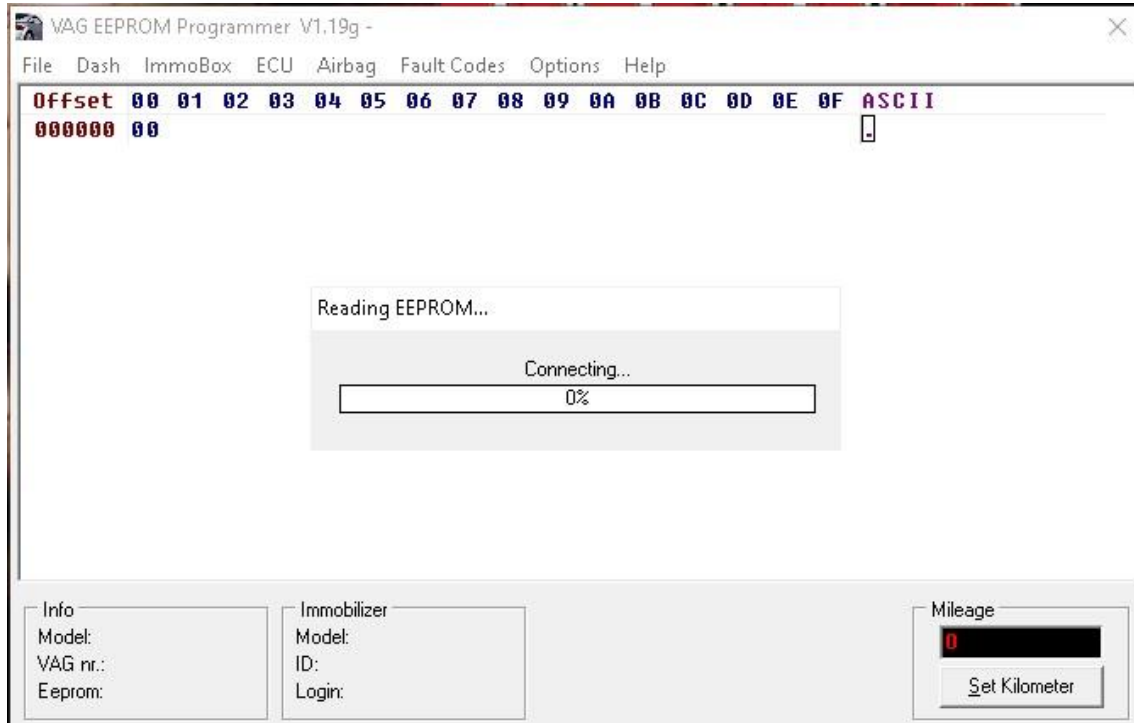
Bien, sabemos que hay comunicación entre el programa la interface y la unidad de control, maravilloso.

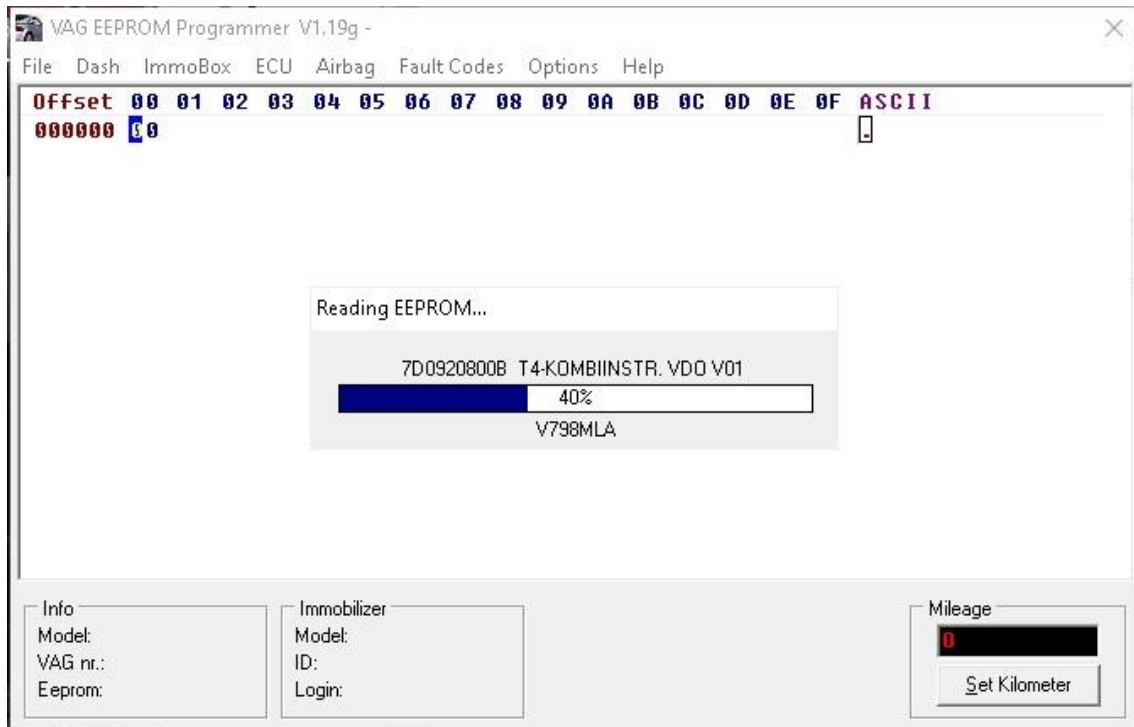
Siguiente paso leer la eeprom: Dash, Read eeprom.



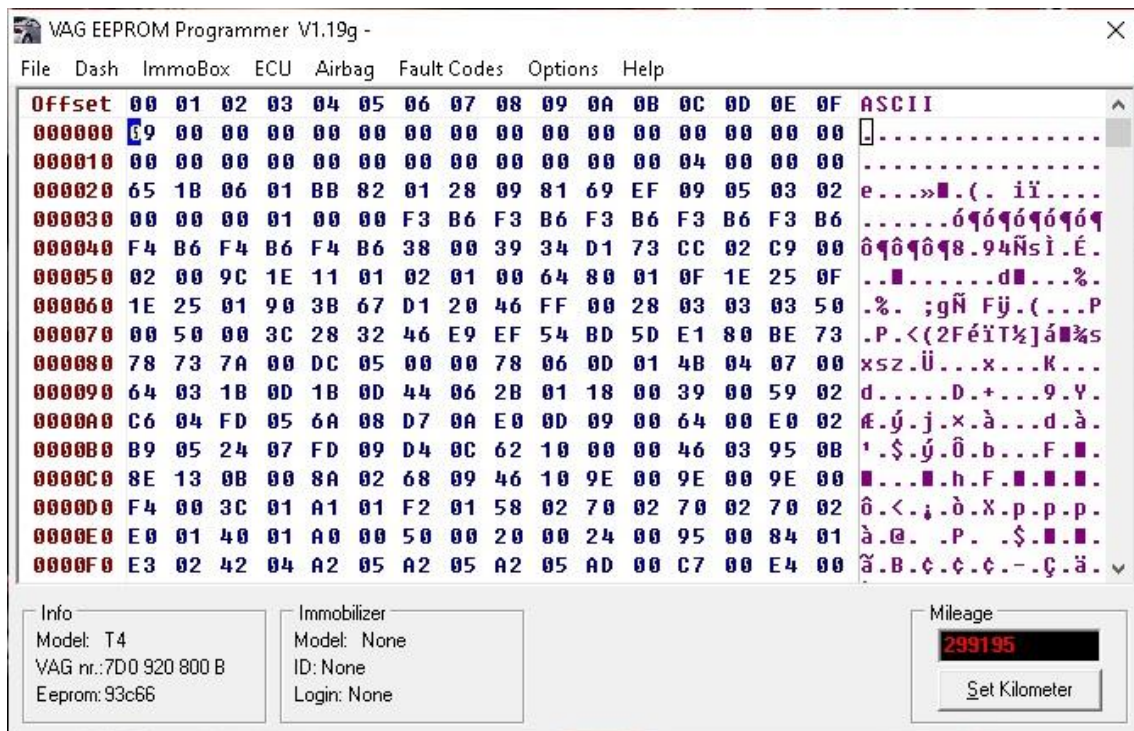
Y puede salir el dichoso leterito de antes , de que no hay comunicación.....

O normalmente puede empezar a leer:





Y terminar de leer y acabar tal que así:



Donde vemos, como había comentado antes, **un editor hexadecimal** (ventana grande llena de números y letras) con el volcado de la memoria leída, ahora podemos guardar esto como un documento, File, save as... y lo guardamos donde queramos (extensión de archivo binario tipo .BIN) y lo tendremos para posibles consultas modificaciones etc...

Y el propio programa de ese volcado nos extrae los kilómetros y algunos datos de información, como Modelo, Referencia del cuadro y el tipo de eeprom (En

este caso lleva una 93c66). Además de los datos de inmovilizador, que en este caso el cuadro no lleva y por eso pone NONE.

Estos archivos .BIN , se pueden abrir con otros editores hexadecimales , hay muchos gratuitos por internet, con más o menos opciones. Pero no hacen la función de sacar del archivo los kilómetros y los otros datos automáticamente, y habría que buscar donde están esparcidos entre todos los números y letras (normalmente repetidos dos veces) manualmente.

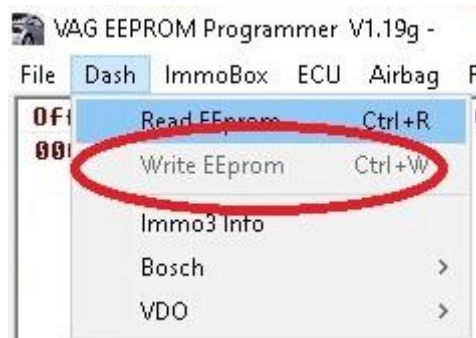
Mientras tanto este programa, como el vag k+can commader , te ponen a disposición directamente los kilómetros en la ventana que se ven abajo a la derecha, Mileage, (en vag k+can commander es similar), para cambiarlos, pones el cursor y con el teclado los cambias.

Tras hacerlo, el volcado de la eeprom que hay en la ventana del editor hexadecimal cambia, se puede ver mientras se escribe; y después se puede volver a guardar el documento, si se quiere, no es necesario, ya con los kilómetros cambiados.

Bien, si solo queremos cambiar los kilómetros, ya está hecho.

Volvemos a la pestaña Dash, pero esta vez Write eeprom.

(no tengo foto y por eso sale en gris pero es esa opción)



Tanto la lectura como la escritura, son breves, no suelen llegar a un minuto.

Cuando ha acabado de escribir, puedes quitar el contacto y volverlo a poner y comprobar los kilómetros, sino han cambiado directamente.

En mi caso, diré que todas las veces que he hecho esta modificación, me suele poner un kilómetro más, la verdad es que no sé porque.

Dado por finalizado este primer proceso, como he dicho conozco dos formas , más de hacer esto mismo. Que voy a nombrar , a continuación pero más rápidamente, pues son más complicadas, por lo menos para mí.

- 2- La segunda opción (y la tercera en parte) para leer la eeprom del cuadro, requiere desmontar el cuadro **completamente**. Y utilizar un lector externo de chips, con sus drivers y programa específicos, y una pinza adaptador específica para el tipo de chip.

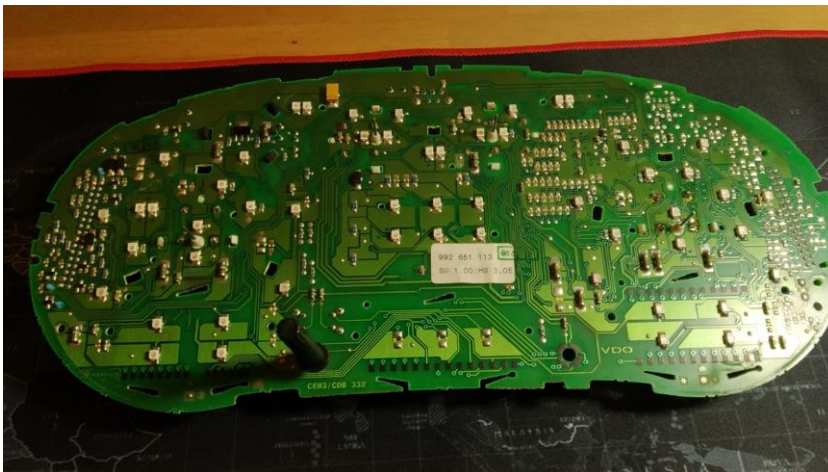
Como para esto ya hay bastante información no me detengo en dar detalles de desmontaje/montaje del cuadro de instrumentos. Si quiero anotar, que tengo una guía de como reajustar la posición de las agujas , que es extracto de una de internet con traducción y aportaciones mías, y que compartiré también en el foro.

Hay que llegar a tener solo la placa, tal que así:

Por detrás.



Y por delante.



Y en la placa, buscar el chip eeprom, donde este ubicado y soldado, hay varios chips parecidos, así tipo “garrapata” con 8 patas, os digo mi forma de saber cual es, pongo en Google, los números de identificación del chip (los que están escritos encima, ver la imagen de chip que he puesto antes), y tras eso pongo datasheet.

La búsqueda te suele llevar a encontrar el datasheet con todos los datos del chip en cuestión, el que nosotros buscamos debe de poder almacenar datos, es decir , es una memoria, por lo tanto la hoja debe de tener una cifra de Kb.

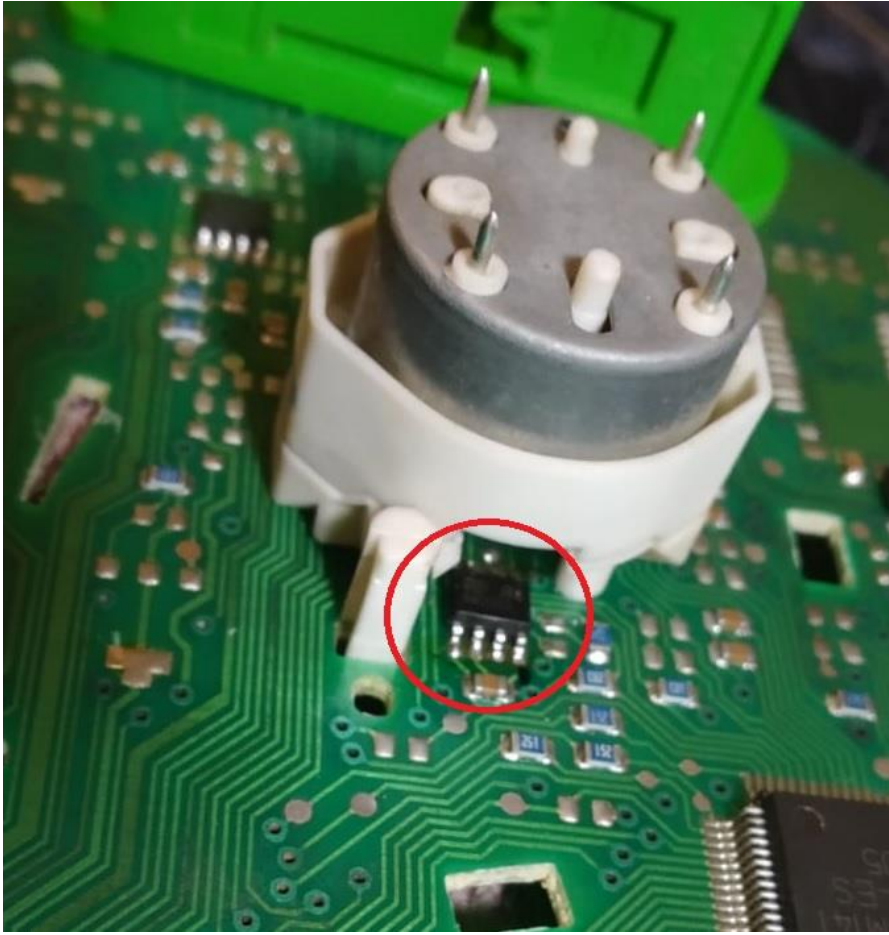
Sino tiene , seguir buscando.

En el caso de este cuadro, el chip estaba debajo de uno de los motores de movimiento de una de las agujas.

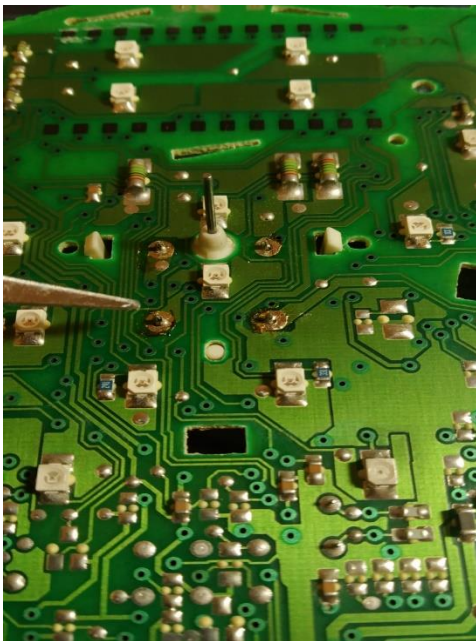


No es el de la derecha.

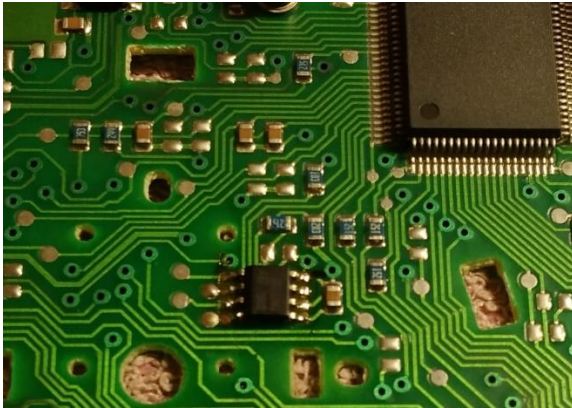
Es este: (Podría ser más fácil, pero no XD)



Pues sin entrar en detalles, el motor lleva unas pestañas de plástico y creo recordar 4 pines, y se desueldan los pines y después apretando las pestañas se quita.



Y queda el chip a vistas:



Y una vez llegados aquí, la primera forma es usar una pinza, adaptador para un lector de chips externo, para leer el chip sin sacarlo de la placa.

Utilizo un lector externo XGECU TL866 II plus, que usa un programa específico llamado Xgpro. Y una pinza soip8. Pero hay mil diferentes.

Lo muestro en la siguiente foto:



Hay que alinear bien los pines del chip, y la pinza, tal y como el programa del lector te marque, buscando el pin número 1; información que también está en el datasheet anteriormente consultado.

*Hay otras mil marcas, modelos, más completos o menos, de lectores de chips de este tipo, con más o menos posibilidades (en cuanto a los tipos de chips que leen), cada uno utiliza su software, tanto a drivers como a **programa específico de lectura (del lector externo)**. Y es completamente diferente a los programas nombrados en la primera forma, o el primer proceso. Lo repito, no nos liemos, ni son los drivers de los cables/interfaces de antes, ni los mismos programas de antes.*

Tras leer con el lector externo, **(XGpro)** “OJO, repito, no es como los de los programas anteriormente nombrados, no saca los datos que nos interesan (kilometros, vin, inmovilizador) aunque si incluye editor hexadecimal es universal” <<<<No voy a entrar a detallar cada programa o uno en concreto, se parecen todos >>>> para lo que nos interesa, solo guarda el archivo en el ordenador.

Y después de esto, **Si se puede abrir el archivo con el vag eeprom programmer**, File, open... para que automáticamente “este programa” detecte los datos que nos interesan , y poder modificar los kilómetros desde su interface. Así nos evitamos lidiar con editores hexadecimales generales/estándar que nos obligarían a buscar entre los datos manualmente.

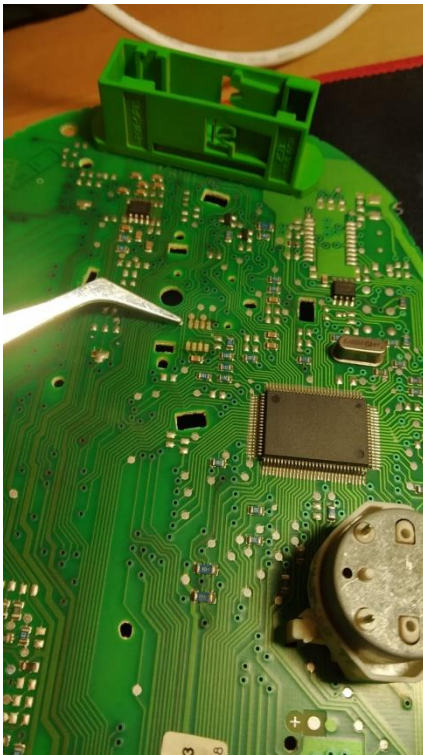
Después de modificarlos, al igual que en el primer proceso, se puede guardar, lo hacemos, lo abrimos con el programa específico (recordamos en mi caso XGpro) del lector externo y lo escribimos.

Resumen del proceso: Leer (programa del lector XGpro), guardar, abrir con el otro programa (vag eeprom programmer), modificar, guardar, abrir con el programa del lector (Xgpro) y volver a escribir.

3- La tercera forma, parte de **hacer todo como en la anterior**, cuadro **completamente** desmontado, quitar el motor, y utilizar un lector externo de chips.

Pero en esta ocasión, sin usar la pinza adaptador. Es decir, desoldando el chip de la placa, las ocho patas.

Placa sin el chip:



Y ponerlo en el zócalo del lector externo.

Tal que así:



Para realizar el mismo proceso que antes, con el programa específico del lector externo(XGpro), leer, guardar, abrir con el otro programa (vag eeprom programmer), modificar, volver a guardar, y volver a escribir otra vez con el programa específico del lector (XGpro).

Vale, finalizamos y en cualquiera de los dos últimos procesos hay de volver a soldar chip+motor, o solo motor, más montar todo el cuadro pieza por pieza.

Voy a hacer unos apuntes finales:

En cualquiera de los últimos procesos, hay que ajustar las agujas, y como he nombrado antes, tengo una guía que publicaré de cómo hacerlo correctamente.

Además, mi experiencia, es que la pinza soip8, difícilmente lee bien el chip, normalmente en mi caso daba errores por falta de tensión. Es probable que con un lector de mejor calidad si se pueda hacer bien.

(Lo he nombrado antes)Para hacer el primer proceso, sobre mesa, o en el vehículo, hace falta una fuente de tensión, de al menos 12,5 a 13v. Si es menos no funciona el cable/interface.

No se puede leer/escribir una unidad (sea la que sea) con el motor en marcha, utilizar un cargador de baterías o estabilizador de tensión para trabajar sobre unidades de control.

Alejar teléfonos móviles, y otras fuentes de posibles interferencias de los aparatos, sus cables y los equipos con los que se va a leer /escribir unidades de control.

No desconectar los cables ni del OBD, ni del USB, ni del cuadro, u otras unidades de control (sea la que sea), mientras se realiza cualquiera de estos procesos de lectura /escritura de unidades de control.

No quitar el contacto, mientras se lee/escribe una unidad de control, sea la que sea.

No quitar fusibles, ni relés, mientras se lee/escribe una unidad de control, sea la que sea.

Sino se toman estas medidas de precaución, se puede bloquear una unidad de control, y puede quedar inútil.

Es importante, una vez leída bien una unidad de control, guardar el archivo .BIN original, y hacer otra copia para modificar, por si hace falta volver al original.

Se puede conectar como se ve en las fotos, para los cuadros de instrumentos, solo positivo 30, positivo15, negativo 31, y línea K.

Como he nombrado en el inicio, hay otras opciones y o elementos que se pueden modificar en este volcado de memoria de la unidad de control. Pongo aquí algunos links:

<https://www.uk-mkivs.net/threads/code-for-needle-sweep-on-some-mk4-cluster.610937/>

<https://www.uk-mkivs.net/threads/needle-sweep-and-dashdumpedit.653680/>

https://theresias.com/tutorials/1h_q4t.htm

Manual creado por Fermín Albert Gozalbo.

Compartido en furgovw.org

Fotos, y contenido propios, salvo un par de fotos buscadas en Google.